

定址方式

定址方式	格式	有效位址
立即資料定址	#data8	
暫存器定址	Rn (R0 ~ R7)	
直接定址	addr8	addr8
絕對定址	addr11	addr11
長程(絕對)定址	addr16	addr16
(暫存器)間接定址	@Ri (@R0 、 @R1)或@DPTR	R0 、 R1 、或 DPTR
(暫存器)相對定址	disp8	PC+符號擴展之 disp8
(基底)指標定址	@A+PC 或 @A+DPTR	A+PC 或 A+DPTR

定址方式

```
1 ;ex4.1-1.asm
2 ;Some examples of the notations of
3 ;addressing modes
----
4 DSEG AT 30H
0040 5 ORG 40H
0040 6 OPR1: DS 1
0041 7 OPR2: DS 1
----
8 CSEG AT 0000H
0000 9 ORG 0000H
0000 7825 10 TEST0: MOV R0,#25H ;immediate
0002 E8 11 MOV A,R0 ;register
0003 E540 12 MOV A,LOW OPR1 ;direct
0005 E6 13 MOV A,@R0 ;indirect
0006 7612 14 TEST1: MOV @R0,#12H;
0008 80F6 15 SJMP TEST0 ;relative
000A 0106 16 AJMP TEST1 ;absolute
000C 020000 17 LJMP TEST0 ;long
000F 83 18 MOVC A,@A+PC ;index
0010 93 19 MOVC A,@A+DPTR ;index
20 END
```

MCS-51資料轉移指令

指令	動作	CY	AC	OV	P
內部資料記憶體及暫存器					
MOV A,src-byte	$A \leftarrow \text{src-byte}$ $\text{src-byte} = \text{Rn, direct, @Ri, \#data8}$	-	-	-	*
MOV Rn,src-byte	$\text{Rn} \leftarrow \text{src-byte}$ $\text{src-byte} = \text{A, direct, \#data8}$	-	-	-	-
MOV direct,src-byte	$(\text{direct}) \leftarrow \text{src-byte}$ $\text{src-byte} = \text{A, Rn, direct, @Ri, \#data8}$	-	-	-	-
MOV @Ri,src-byte	$(\text{Ri}) \leftarrow \text{src-byte}$ $\text{src-byte} = \text{A, direct, \#data8}$	-	-	-	-
CLR A	$A \leftarrow 0$	-	-	-	*
XCH A,src-byte	$A \leftrightarrow \text{src-byte}$ $\text{src-byte} = \text{Rn, direct, @Ri}$	-	-	-	*
XCHD A,@Ri	$A[3:0] \leftrightarrow (\text{Ri})[3:0]$	-	-	-	*
SWAP A	$A[3:0] \leftrightarrow A[7:4]$	-	-	-	-

MCS-51資料轉移指令

外部資料記憶體					
MOV DPTR,\#data16	$\text{DPTR} \leftarrow \#data16$	-	-	-	-
MOVBX A,src-byte	$A \leftarrow \text{src-byte}$ $\text{src-byte} = @Ri, @DPTR$	-	-	-	*
MOVBX dest-byte,A	$\text{dest-byte} \leftarrow A$ $\text{dest-byte} = @Ri, @DPTR$	-	-	-	-
程式記憶體(中之表格)					
MOVC A,@A+PC	$A \leftarrow (A+PC)$	-	-	-	*
MOVC A,@A+DPTR	$A \leftarrow (A+DPTR)$	-	-	-	*

MCS-51資料轉移指令

```

1  ;ex4.2-2.a51
----
2      DSEG AT 30H
0030   3  OPR1:  DS  1
0031   4  OPR2:  DS  1
5  ;Exchange two words in memory
6  ;using DIRECT addressing mode
----
7      CSEG AT 0000H
0000 A830   8  SWAPBYTE: MOV  R0,LOW OPR1 ;get opr1
0002 A931   9      MOV  R1,LOW OPR2 ;get opr2
0004 8831  10      MOV  LOW OPR2,R0 ;save opr1
0006 8930  11      MOV  LOW OPR1,R1 ;save opr2
0008 22   12      RET
13      END

```

MCS-51資料轉移指令

```

1  ;ex4.2-3.a51
----
2      DSEG AT 30H
0030   3  OPR1:  DS  1
0031   4  OPR2:  DS  1
5  ;Exchange two words in memory
6  ;using DIRECT addressing mode
----
7      CSEG AT 0000H
0000 7830   8  SWAPBYTE: MOV  R0,#LOW OPR1;pointer for opr1
0002 7931   9      MOV  R1,#LOW OPR2;pointer for opr2
0004 E6  10      MOV  A,@R0 ;get opr1
0005 87F0  11      MOV  B,@R1 ;get opr2
0007 F7  12      MOV  @R1,A ;save opr1
0008 A6F0  13      MOV  @R0,B ;save opr2
000A 22  14      RET
15      END

```

MCS-51資料轉移指令

```

1 ;ex4.2-4.a51
---- 2      DSEG AT 30H
0030 3  OPR1: DS 1
0031 4  OPR2: DS 1
5 ;Exchange two words in memory
6 ;using DIRECT addressing mode
---- 7      CSEG AT 0000H
0000 E530 8  SWAPBYTE: MOV A,LOW OPR1 ;get opr1
0002 C531 9      XCH A,LOW OPR2 ;swap
0004 F530 10     MOV LOW OPR1,A ;save opr2
0006 22 11     RET
12     END
    
```

加法運算與旗號位元的關係

運算	SF ZF OV CY	帶號數	未帶號數
01000000 + 00101110 ----- 01101110	0 0 0 0	+64 + +46 ----- + 110	64 + 46 ----- 110
01000110 + 01010000 ----- 10010110	1 0 1 0	+70 + +80 ----- - 106(溢位)	70 + 80 ----- 150
01001110 + 10110010 ----- 10000000	0 1 0 1	+78 + -78 ----- + 0	78 + 178 ----- 256(C=1)
10110010 + 10100000 ----- 101010010	0 0 1 1	-78 + -96 ----- + 82(溢位)	178 + 160 ----- 82(C=1)

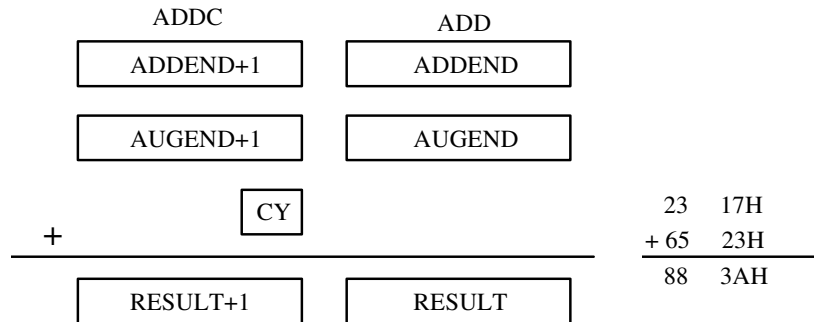
MCS-51二進制加法與減法指令

指令	動作	CY	AC	OV	P
ADD A,src-byte	$A \leftarrow A + \text{src-byte}$ src-byte = Rn, direct, @Ri, #data8	*	*	*	*
ADDC A,src-byte	$A \leftarrow A + \text{src-byte} + C$ src-byte = Rn, direct, @Ri, #data8	*	*	*	*
SUBB A,src-byte	$A \leftarrow A - \text{src-byte} - C$ src-byte = Rn, direct, @Ri, #data8	*	*	*	*

單精確制加法

```
1 ;ex4.3-1.a51
---- 2 DSEG AT 30H
0030 3 ADDEND: DS 1 ;addend
0031 4 AUGEND: DS 1 ;augend
0032 5 RESULT: DS 1 ;result
6 ;single-precision addition
---- 7 CSEG AT 0000H
0000 E530 8 ADD8: MOV A,LOW ADDEND ;get addend
0002 2531 9 ADD A,LOW AUGEND ;add them
0004 F532 10 MOV LOW RESULT,A ;store result
0006 22 11 RET
12 END
```

雙精確制加法



雙精確制加法

```

1 ;ex4.3-2.a51
2 DSEG AT 30H
0030 3 ADDEND: DS 2 ;addend
0032 4 AUGEND: DS 2 ;augend
0034 5 RESULT: DS 2 ;result
6 ;double-precision addition
---- 7 CSEG AT 0000H
0000 E530 8 ADD16: MOV A,LOW ADDEND ;perform
0002 2532 9 ADD A,LOW AUGEND ;low-byte
0004 F534 10 MOV LOW RESULT,A ;addition
0006 E531 11 MOV A,LOW ADDEND+1 ;perform
0008 3533 12 ADDC A,LOW AUGEND+1 ;high-byte
000A F535 13 MOV LOW RESULT+1,A ;addition
000C 22 14 RET
15 END
    
```

減法運算與旗號位元的關係

運算	SF ZF OV CY	帶號數	未帶號數
01000000 - 00101110 100010010	0 0 0 0	+64(大) - +46(小) + 18	64(大) - 46(小) 18
01000110 - 01010000 011110110	1 0 0 1	+70(小) - +80(大) - 10	70(小) - 80(大) 246(B=1)
01001110 - 10110010 010011100	1 0 1 1	+78 - -78 - 100(溢位)	78 - 178 156
10110010 - 10100000 100010010	0 0 0 0	-78(大) - -96(小) + 18	178(大) - 160(小) 18
10110010 - 11000100 011101110	1 0 0 1	-78(小) - -60(大) - 18	178(小) - 196(大) 238(B=1)
01100000 - 10110010 010101110	1 0 1 1	+96(大) - -78(小) - 82(溢位)	96(小) - 178(大) 238(B=1)
10110010 - 01111111 100110011	0 0 1 0	-78(小) - +127(大) + 51(溢位)	178(大) - 127(小) 51

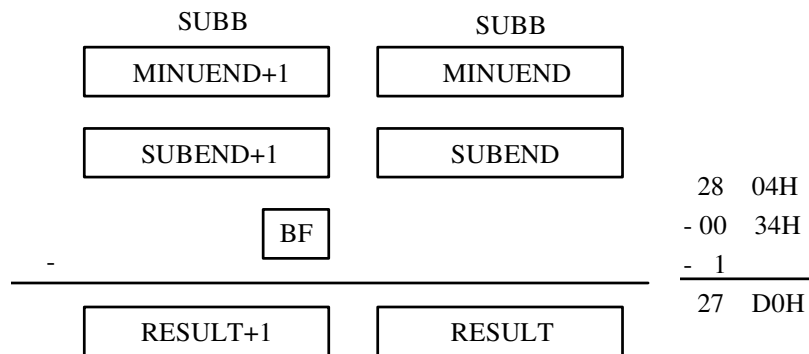
單精確制減法

```

1 ;ex4.3-3.a51
---- 2 DSEG AT 30H
0030 3 MINUEND: DS 1 ;minuend
0031 4 SUBEND: DS 1 ;subend
0032 5 RESULT: DS 1 ;result
6 ;single-precision subtraction
---- 7 CSEG AT 0000H
0000 E530 8 SUB8: MOV A,LOW MINUEND ;get minuend
0002 C3 9 CLR C ;clear carry
0003 9531 10 SUBB A,LOW SUBEND ;subtract them
0005 F532 11 MOV LOW RESULT,A ;store result
0007 22 12 RET
13 END

```

雙精確制減法



雙精確制減法

```

1  ;ex4.3-4,a51
---- 2      DSEG AT 30H
0030 3  MINUEND: DS 2  ;minuend
0032 4  SUBEND: DS 2  ;subtrahend
0034 5  RESULT: DS 2  ;result
6  ;double-precision subtraction (16 bits)
---- 7      CSEG AT 0000H
0000 C3 8  SUB16: CLR C ;clear carry flag
0001 E530 9      MOV A,LOW MINUEND;get minuend(lo)
0003 9532 10     SUBB A,LOW SUBEND ;subtract them
0005 F534 11     MOV LOW RESULT,A ;store result(lo)
0007 E531 12     MOV A,LOW MINUEND+1;get minuend(hi)
0009 9533 13     SUBB A,LOW SUBEND+1 ;subtract them
000B F535 14     MOV LOW RESULT+1,A;save result(hi)
000D 22 15     RET
16     END
    
```

MCS-51單運算元指令

指令	動作	CY	AC	OV	P
INC A	$A \leftarrow A + 1$	-	-	-	*
INC dst-byte	$(dst\text{-}byte) \leftarrow (dst\text{-}byte) + 1$ $dst\text{-}byte = Rn, \text{direct}, @Ri$	-	-	-	-
INC DPTR	$DPTR \leftarrow DPTR + 1$	-	-	-	-
DEC A	$A \leftarrow A - 1$	-	-	-	*
DEC dst-byte	$(dst\text{-}byte) \leftarrow (dst\text{-}byte) - 1$ $dst\text{-}byte = Rn, \text{direct}, @Ri$	-	-	-	-
CPL A	$A \leftarrow \overline{A}$	-	-	-	*

MCS-51乘法與除法運算指令

指令	動作	CY	AC	OV	P
MUL AB	$B:A \leftarrow A \times B$ $B \leftarrow$ 乘積高序位元組； $A \leftarrow$ 乘積低序位元組	0	-	*	*
DIV AB	$A \leftarrow A/B$ 的商數 $B \leftarrow A/B$ 的餘數	0	-	*	*

8位元乘法

```
1 ;ex4.3-5.a51
---- 2 DSEG AT 30H
0030 3 MULTR: DS 1 ;multiplier
0031 4 MULTAND: DS 1 ;multiplicand
0032 5 RESULT: DS 2 ;result
6 ;single-precision multiplication
---- 7 CSEG AT 0000H
0000 E530 8 MUL8: MOV A,LOW MULTR ;get multiplier
0002 8531F0 9 MOV B,LOW MULTAND ;get multiplicand
0005 A4 10 MUL AB ;multiply them
0006 85F033 11 MOV LOW RESULT+1,B;save high byte
0009 F532 12 MOV LOW RESULT,A ;save low byte
000B 22 13 RET
14 END
```

8位元除法

```
1 ;ex4.3-6.a51
---- 2 DSEG AT 30H
0030 3 DIVIDEND: DS 1 ;dividend
0031 4 DIVISOR: DS 1 ;divisor
0032 5 QUOTIENT: DS 1 ;quotient
0033 6 REMAINDER: DS 1 ;remainder
7 ;single-precision division
---- 8 CSEG AT 0000H
0000 E530 9 DIV8: MOV A,LOW DIVIDEND ;get dividend
0002 8531F0 10 MOV B,LOW DIVISOR ;get divisor
0005 84 11 DIV AB ;divide
0006 F532 12 MOV LOW QUOTIENT,A ;save quotient
0008 85F033 13 MOV LOW REMAINDER,B;save remainder
000B 22 14 RET
15 END
```

BCD加法的調整程序

AC = 1

49	0 1 0 0	1 0 0 1	
+ 58	+ 0 1 0 1	1 0 0 0	
107	1 0 1 0	0 0 0 1	
	+ 0 1 1 0	0 1 1 0	
	1 0 0 0 0	0 1 1 1	
	1 0	7	

因1010大於1001所以加0110

進位

← 因AC = 1所以加0110

← 總和(有進位)

BCD減法的調整程序

AC = 0

51	0 1 0 1	0 0 0 1	
- 69	+ 1 0 0 1	0 1 1 1	
82	1 1 1 0	1 0 0 0	
	+ 1 0 1 0	1 0 1 0	
	1 1 0 0 0	0 0 1 0	
	1 0	2	

因1110大於1001所以減0110

借位 = 0

← (69取2補數)

← 因AC = 0所以減0110

← 差(有借位)

MCS-51 BCD調整指令

指令	動作	CY	AC	OV	P
DA A	A ← 調整累積器 A 的內容為成立的 BCD 數字	*	-	-	*

BCD加法

```
1 ;ex4.3-9.a51
---- 2 DSEG AT 30H
0030 3 ADDEND: DS 1 ;addend
0031 4 AUGEND: DS 1 ;augend
0032 5 RESULT: DS 2 ;result
6 ;single-precision BCD addition
---- 7 CSEG AT 0000H
0000 E530 8 BCDADD: MOV A,LOW ADDEND ;get addend
0002 2531 9 ADD A,LOW AUGEND ;add them
0004 D4 10 DA A ;decimal adjust
0005 F532 11 MOV LOW RESULT,A ;save result(lo)
0007 7400 12 MOV A,#00 ;clear A
0009 3400 13 ADDC A,#00H ;get carry
000B F533 14 MOV LOW RESULT+1,A;save result(hi)
000D 22 15 RET
16 END
```

MCS-51條件性分歧指令

指令	動作	CY	AC	OV	P
JZ disp8	PC ← PC + 2; If A = 0 then PC ← PC + disp8	-	-	-	-
JNZ disp8	PC ← PC + 2; If A != 0 then PC ← PC + disp8	-	-	-	-
JC disp8	PC ← PC + 2; If C = 1 then PC ← PC + disp8	-	-	-	-
JNC disp8	PC ← PC + 2; If C = 0 then PC ← PC + disp8	-	-	-	-
CJNE A,direct,disp8	PC ← PC + 3; If A != (direct) then PC ← PC + disp8 If A < (direct) then C ← 1 else C ← 0	*	-	-	-
CJNE dest-byte, #data8, disp8	PC ← PC + 3; If dest-byte != #data8 then PC ← PC + disp8 If dest-byte < #data8 then C ← 1 else C ← 0	*	-	-	-

陣列資料搬移

```

1  ;ex4.4-1.a51
----
2      DSEG AT 30H
0030  3  SRCA: DS 8 ;source array
0038  4  DSTA: DS 8 ;destination array
5      ;block data move using one-byte-by-one-byte
6      ;transfer
----
7      CSEG AT 0000H
0000 E530  8  BLKMOV: MOV A,LOW SRCA ;transfer 1st byte
0002 F538  9      MOV LOW DSTA,A
0004 E531 10      MOV A,LOW SRCA+1;transfer 2nd byte
0006 F539 11      MOV LOW DSTA+1,A
          .....
          .....
001C E537 22      MOV A,LOW SRCA+7;transfer 8th byte
001E F53F 23      MOV LOW DSTA+7,A
0020 22    24      RET
          25      END

```

利用迴路的陣列資料搬移

```

1 ;ex4.4-2.a51
---- 2 DSEG AT 30H
0008 3 LENTH EQU 08H ;bytes of array
0030 4 SRCA: DS 8 ;source array
0038 5 DSTA: DS 8 ;destination array
6 ;block data move with counter
---- 7 CSEG AT 0000H
0000 7A08 8 BLKMOV: MOV R2,#LENTH;set count
0002 7830 9 MOV R0,#LOW SRCA ;set source pointer
0004 7938 10 MOV R1,#LOW DSTA ;set dest. pointer
0006 E6 11 MLOOP: MOV A,@R0 ;transfer them
0007 F7 12 MOV @R1,A
0008 08 13 INC R0 ;point to next
0009 09 14 INC R1 ;entry
000A 1A 15 DEC R2
000B BA00F8 16 CJNE R2,#0,MLOOP;repeat count times
000E 22 17 RET
18 END

```

多精確制BCD加法

```

1 ;ex4.4-3.a51
---- 2 DSEG AT 30H
0004 3 COUNT EQU 04H ;repeat times
0030 4 ADDEND: DS 4 ;addend
0034 5 AUGEND: DS 4 ;augend/result
6 ;multi-precision BCD addition
---- 7 CSEG AT 0000H
0000 7A04 8 MBCDADD: MOV R2,#COUNT ;put count in CX
0002 7830 9 MOV R0,#LOW ADDEND ;point to addend
0004 7934 10 MOV R1,#LOW AUGEND ;point to augend
0006 C3 11 CLR C ;clear carry flag
0007 E6 12 BEGIN: MOV A,@R0 ;add them
0008 37 13 ADDC A,@R1 ;
0009 D4 14 DA A ;decimal adjust
000A F7 15 MOV @R1,A ;save the result
000B 08 16 INC R0 ;point to the next item
000C 09 17 INC R1
000D 1A 18 DEC R2 ;repeat count times
000E BA00F6 19 CJNE R2,#00,BEGIN
0011 22 20 RET
21 END

```

旗號迴路

```

1 ;ex4.4-4.a51
2      DSEG AT 30H
0030 3  SRCA: DS 8 ;source array
0038 4  DSTA: DS 8 ;destination array
5 ;block data move with general exit
6      CSEG AT 0000H
0000 7830 7  BLKMOV: MOV R0,#LOW SRCA ;set source pointer
0002 7938 8      MOV R1,#LOW DSTA ;set dest. pointer
0004 E6 9  MLOOP: MOV A,@R0 ;transfer them
0005 B4FF02 10     CJNE A,#0FFH,SAVE ;end ?
0008 010F 11     AJMP DONE ;yes, done
000A F7 12  SAVE: MOV @R1,A ;save into destination
000B 08 13     INC R0 ;point to next
000C 09 14     INC R1 ;entry
000D 0104 15     AJMP MLOOP ;continue
000F 22 16  DONE: RET
17      END
    
```

MCS-51無條件分歧與跳躍指令

指令	動作	CY	AC	OV	P
SJMP disp8	$PC \leftarrow PC + 2; PC \leftarrow PC + \text{disp8}$	-	-	-	-
AJMP addr11	$PC[10:0] \leftarrow \text{addr11}$	-	-	-	-
LJMP addr16	$PC \leftarrow \text{addr16}$	-	-	-	-
JMP @A+DPTR	$PC \leftarrow A + \text{DPTR}$	-	-	-	-

整數平方根

25		35	
- 1	1	- 1	1
24		34	
- 3	2	- 3	2
21		31	
- 5	3	- 5	3
16		26	
- 7	4	- 7	4
9		19	
- 9	5 ← 結果	- 9	5 ← 結果
0		10	
		- 11	
		- 1	

整數平方根

```

1 ;ex4.4-5.a51
----
2 DSEG AT 30H
0030 3 TESTNUM: DS 1 ;test number
0031 4 SQRT: DS 1 ;square root value
5 ;program to find the approximate square
6 ;root of a given number by successive
7 ;subtraction.
8 CSEG AT 0000H
0000 E530 9 SQRT_FD: MOV A,LOW TESTNUM;get test number
0002 7801 10 MOV R0,#01H ;start value
0004 7900 11 MOV R1,#00H ;clear count
0006 C3 12 AGAIN: CLR C ;clear carry flag
0007 98 13 SUBB A,R0 ;we have done it
0008 4005 14 JC DONE ;when A < R0
000A 09 15 INC R1 ;increase count
000B 08 16 INC R0 ;get next odd number
000C 08 17 INC R0
000D 80F7 18 JMP AGAIN ;continue
000F 8931 19 DONE: MOV LOW SQRT,R1;save result
0011 22 20 RET
21 END

```

MCS-51迴路指令

指令	動作	CY	AC	OV	P
DJNZ Rn,disp8	PC ← PC + 2; Rn ← Rn - 1; If Rn != 0 then PC ← PC + disp8	-	-	-	-
DJNZ direct,disp8	PC ← PC + 2; (direct) ← (direct) - 1; If (direct) != 0 then PC ← PC + disp8	-	-	-	-

MCS-51迴路指令

```

1 ;ex4.4-6.a51
---- 2 DSEG AT 30H
0008 3 LENTH EQU 08H ;bytes of array
0030 4 SRCA: DS 8 ;source array
0038 5 DSTA: DS 8 ;destination array
6 ;block data move with counter
---- 7 CSEG AT 0000H
0000 7A08 8 BLKMOV: MOV R2,LENTH;set count
0002 7830 9 MOV R0,LOW SRCA ;set source pointer
0004 7938 10 MOV R1,LOW DSTA ;set dest. pointer
0006 E6 11 MLOOP: MOV A,@R0 ;transfer them
0007 F7 12 MOV @R1,A
0008 08 13 INC R0 ;point to next
0009 09 14 INC R1 ;entry
000A DAFA 15 DJNZ R2,MLOOP;repeat count times
000C 22 16 RET
17 END
    
```

MCS-51迴路指令

```

1 ;ex4.4-7.a51
---- 2      DSEG AT 30H
0008 3  LENTH EQU 08H ;bytes of array
0000 4  REG0 SET 00H ;R0 = 00H
0030 5  KEY: DS 1 ;test data
0031 6  INDEX: DS 1 ;data index
7 ;linear search using LOOPNE instruction
---- 8      CSEG AT 0000H
0000 7908 9      MOV R1,#LENTH;set count
0002 90001F 10     MOV DPTR,#TABLE-1;set table pointer
0005 75F0FF 11     MOV B,#-1 ;assume not find
0008 A830 12     MOV R0,LOW KEY ;get test data
000A A3 13  MLOOP: INC DPTR ;point to next entry
000B 7400 14     MOV A,#00H ;zero accumulator
000D 93 15     MOVC A,@A+DPTR

```

MCS-51迴路指令

```

000E B50009 16     CJNE A,REG0,NEXTLP;search the table
0011 7407 17  FOUND: MOV A,#LENTH-1;if found then
0013 C3 18     CLR C ;evalute the index
0014 99 19     SUBB A,R1 ;and return
0015 F5F0 20     MOV B,A ;else set index=-1
0017 02001C 21     JMP RETURN ;found
001A D9EE 22  NEXTLP: DJNZ R1,MLOOP ;not found yet
001C 85F031 23  RETURN: MOV LOW INDEX,B;save index
001F 22 24     RET
0020 12232467 25  TABLE: DB 12H,23H,24H,67H ;data table
0024 76982345 26     DB 76H,98H,23H,45H
27     END

```