

巨集處理器 (一)

- 巨集指令 (Macro Instruction) — 將一群常用的指令予以適當的組合，並定義一條指令用以替代該群指令，此一替代的指令稱之為巨集指令。

- 如 **ADD5** 即是程式設計者所定義之巨集指令，用以替代 (1)、(2) 及 (3) 三條指令。

MACRO

ADD5 &VAR1,&VAR2

L 1,&VAR1 (1)

A 1,=F'5' (2)

ST 1,&VAR2 (3)

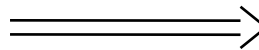
MEND

- 
- 巨集展開 (Macro Expansion) —
程式設計師於程式中直接使用自行定義的巨集指令。當所設計之程式經由巨集處理器 (Macro Processor) 處理時，巨集處理器會將程式中的巨集指令轉換為原來的指令群，此動作稱之為巨集展開。

■ 範例：

```
TEST    START
        USING    *,15
        ADD5    ONE,RESULT
ONE      DC      F '1'
RESULT  DS      F
        END
```

巨集處理器



```
TEST    START
        USING    *,15
        L      1,ONE
        A      1,=F '5'
        ST     1,RESULT
ONE      DC      F '1'
RESULT  DS      F
        END
```



- 使用巨集指令優點 —

- (1) 減輕程式設計者之負擔，縮短程式設計的時間。

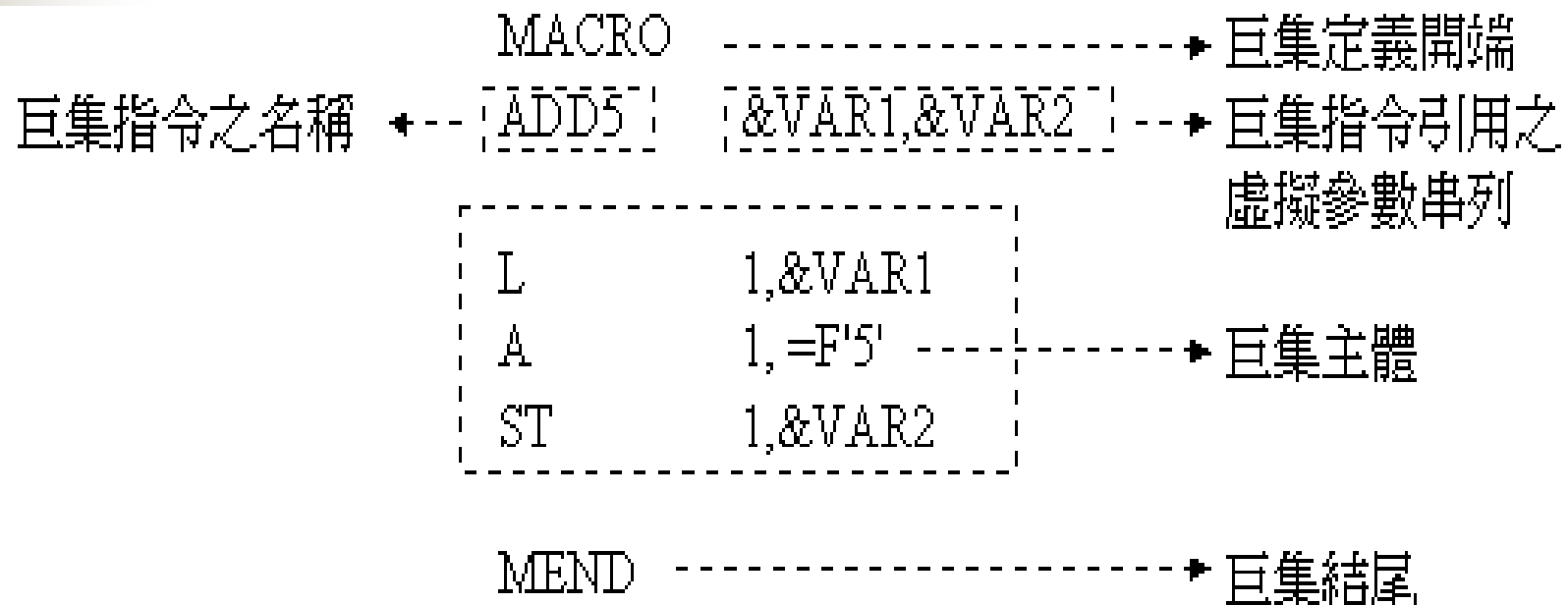
- (2) 巨集指令可視為高階程式語言指令，具有如同使用高階程式言的優點，卻又不失去組合語言之精神。

- (3) 巨集指令能促進程式的標準化、減輕程式除錯、修改與維護之負擔。

- (4) 巨集指令可以讓計算機內的系統軟體更具有彈性。

巨集撰寫之格式

■ 巨集指令之格式 —





- 巨集指令參數傳遞 —

巨集指令的參數傳遞方式可分成兩種：

- (1) 位置參數法：憑藉虛擬參數於參數串列中的位置，使實際參數與虛擬參數逐一對應。將要引用之虛擬參數與實際參數的位置相互對應，不想引用之虛擬參數則於實際參數串列中以空白取代
- (2) 關鍵字參數法：直接將要引用之虛擬參數與所對應的實際參數例出，不想引用者，則不列出。

■ 範例：位置參數法

```
MACRO
ADD5      &VAR1,&VAR2 -----▶ 虛擬參數串列
L        1, &VAR1
A        1, =F'5'
ST       1,&VAR2
MEND
TEST     START
        USING *,15
        ADD5 ONE,RESULT -----▶ 實際參數串列
ONE      DC      F'1'
RESULT   DS      F
        END
```

在此範例中，於巨集展開時，將會以實際參數 ONE 取代些擬參數 &VAR1 及實際參數 RESULT 取代虛擬參數 &VAR2。

■ 範例：關鍵字參數法

```

MACRO
ADD5    &VAR1,&VAR2 -----> 虛擬參數串列
L      1, &VAR1
A      1, =F'5'
ST     1,&VAR2
MEND
TEST   START
USING  *,15 -----> 實際參數
ADD5   &VAR=ONE,&VAR2=RESULT
ONE    DC    F'1' -----> 虛擬參數
RESULT DS    F
      END

```



- 巨集指令之址標引用 —

在巨集指令之本體中可能會有址標的出現，若該巨集指令經多次呼叫展開後，則會使該址標於程式中多次的出現，產生了符號多重定義的錯誤。解決此種問題的方法便是巨集處理器利用一個計數器來記錄巨集指令呼叫展開之次數，並將此次數值附加於址標之後。

■ 範例：

MACRO			PROG	START
REPEAT	&DATA1	&DATA2		USING *15
LOOP	L	1,&DATA1	<u>LOOP1</u>	L 1,C
	A	1,&DATA1		A 1,C
	ST	1,&DATA2		ST 1,B
	CLI	&DATA2,4		CLI B,4
	BNE	LOOP		BNE <u>LOOP1</u>
MEND			<u>巨集處理</u> <u>LOOP2</u>	L 1,X
PROG	START			A 1,X
	USING	*15		ST 1,Y
	REPEAT	C,B		CLI B,4
	REPEAT	X,Y		BNE <u>LOOP2</u>
C	DC	F'1'	C	DC F'1'
X	DC	F'2'	X	DC F'2'
B	DS	F	B	DS F
Y	DS	F	Y	DS F
	END			END



- 巢狀式巨集 —

所謂狀式巨集 (Nested Macro) 即是巨集定義內又含有巨集呼叫。巨集定義內含有巨集呼叫，則於巨集展開時會去呼叫另一個巨集。於是在展開一個巨集的當中，它又會去展開另一個巨集。如果巨集定義內之巨集呼叫為呼叫巨集本身，則會形成遞迴呼叫。

■ 範例：

```
MACRO
ADD4    &VAR
L       1,&VAR
A       1,=F'4'
ST      1,&VAR
MEND
MACRO
ADD5    &VAR1,&VAR2,&VAR3
ADD4    &VAR1
ADD4    &VAR2
ADD4    &VAR3
MEND
```

呼叫 ADD5 DATA1, DATA2
,DATA3 後，ADD5 中
每一個 ADD4 均展開為
三條指令。

```
{ L 1,DATA1
  A 1,=F'4"
  ST 1,DATA1
{ L 1,DATA2
  A 1,=F'4"
  ST 1,DATA2
{ L 1,DATA3
  A 1,=F'4"
  ST 1,DATA3
```



- 巨集定義內含巨集定義 —

在巨集定義內含有巨集定義的情況中，巨集定義之名稱亦可視為是參數。於內層的巨集定義必須被外層的巨集呼叫才會有效，不可於程式中直接單獨呼叫內層的巨集定義。

■ 範例:

此範例中，MAIN 為外層巨集定義，而 ROUTINE 為內層巨集定義，ROUTINE 僅可被 MAIN 呼叫，不能在主程式中直接被呼叫。

		MACRO		
		MAIN	&ROUTINE	巨集名稱：MAIN；
		.		
		.		參數：ROUTINE (為內層之巨集義)
		.		
MAIN 巨集定義	ROUTINE 巨集定義	MACRO		虛擬巨集名稱
		&ROUTINE	&X	內層巨集
		CNOP	0,4	調整界限
		BAL	1,*+4	設定暫存器 1 的值為計數器值加 4
		DC	A(&X)	定義 &X 的位址值
		L	15,=V(&ROUTINE)	把副程式位址載入暫存器 15 內
		BALR	14,15	轉移控制權到副程式內
		MEND		
		.		
		.		
		MEND		



- 條件式巨集展開 —

於巨集處理中，AIF 與 AGO 是兩個非常重要之虛擬指令，能依其條件滿足與否，做適當的巨集展開。但是展開後，AIF 與 AGO 並不會出現在展開的程式片斷中。提供條件式巨集展開將使得程式更具有彈性與效率。



- (1) AIF 指令：

AIF 指令有如高階語言中的 IF 指令，為一條有條件跳躍之虛擬指令。其指令架構如下：

AIF (測試條件) 巨集址標

此敘述表示如果測試條件滿足，則跳躍至址標處執行。



- (2) AGO 指令：

AGO 指令有如高階語言中的 GOTO 指令，為一條無條件跳躍之虛擬指令。其指令架構如下：

AGO 巨集址標

此敘述表示遇到 AGO 指令時，則無條件跳躍至址標處執行。

<註>：條件式巨集指令中之址標於展開後，並不會出現出在主程式中。

■ 範例：

```
MACRO
TEST &COND,&ARG1,&ARG2,&ARG3
A 1,&ARG1
AIF (&COND EQ1) ENDING (測試 COND 是否為 1)
A 1,&ARG2
AIF (&COND EQ2) ENDING (測試 COND 是否為 2)
A 1,&ARG3      ENDING
MEND

.
.
.
LOOP1  TEST 3,DATA1,DATA2,DATA3
.
.
.
LOOP2  TEST 2,DATA3,DATA2
.
.
.
LOOP3  TEST 1,DATA1
.
DATA1  DC  F'4'
DATA2  DC  F'9'
DATA3  DC  F'10'
```

展開

```

.
.
.
LOOP1  A 1,DATA1
        A 1,DATA2
        A 1,DATA3
.
.
.
LOOP2  A 1,DATA3
        A 1,DATA2
.
.
.
LOOP3  A 1,DATA1
.
.
.
```