電脳技術叢書

微電脳組合程式

工業技術研究院 電子工業研究所 編印

微電腦組合程式
著作人：柯俊暉
發行人：方賢齊
發行所：工業技術研究院
出版者：工業技術出版社
新竹縣竹東鎮中興路 4 段 195 之 4 號 (02)5428212 (036)952041
行政院新聞局登記證：局版臺業字第 1529 號
印刷所：華泰印刷廠 (035)225941
初　版：七十年六月
再　版：七十年十二月
定　價：新台幣 620 元

序　言

電腦技術發展中心一般系統部所出版的技術叢書系列，目的在將本中心內研製計劃有關之技術資料公諸社會，其一方面可供各級學校利用，另一方面也可供工商研究機關參考。電腦技術發展中心叢書之出版，盼望能給目前電腦工業發展帶來直接助益。

本書爲叢書之一，目的在介紹微電腦組合程式結構，書中所採實例，則以目前本中心現有之微電腦組合程式爲主。

本書在編寫期間，承蒙凌淑明小姐協助整理，在此致謝！

柯　俊　暉
周　進　興
識於電子工業研究所電腦技術發展中心一般系統部
中華民國六十九年十月一日

目　錄

第一章　前　言

我們將微電腦系統的組合程式（ Assembler ）設計實例，加以分析討論，藉以瞭解該組合程式的架構與其設計之要領，然後將所得之心得及整個組合程式的架構加以整理，而成此書。本書的目的，在於提供給修習計算機系統軟體（ Software ）的人員，一個具體的設計實例，使能體會出組合程式是如何設計出來的；同時也提供給即將設計組合程式的人員，做一參考，希望對其設計工作有所助益。

由於本書所述的組合程式，爲一設計實例，必然與其所實施的機器，有很大的關連（ Dependent ），故我們希望讀者在研讀本書時，能隨時參閱微電腦系統使用手册，如此更能加深對組合程式的瞭解。

微電腦系統組合程式的設計，有下列幾個特性：

1. 富於結構性設計（ Structure Programming）；因此本書的說明，將以一個程式單元爲單位，分別說明其功能與特性，同時也希望讀者能注意微電腦系統的組合程式是如何劃分各個程式單元，其中有因功能不同而劃分，有因處理的資料性質不同而劃分，讀者可拿本書，作爲設計結構性程式的參考。
2. 符號表的大小不受限制；此因使用了符號表流溢檔（ Symbol Table Overflow File ）的設計方法。
3. 原始程式檔的大小，不受限制；一般在設計組合程式時，常因符號表的大小受到限制，因而使得原始程式的大小也受到限制。但微電腦系統的組合程式，因採用符號表流溢檔的技巧，故原始程式的大小，也因而不受限制。
4. 組合程式分成三個過程（ PASS ），採用記憶體重疊（ Overlay ）的技巧完成。
5. 採用輸入輸出暫存區（ Buffer ），和中間過程暫存檔的技巧完成。並設定動態的（ Dynamically ）暫存區位址及大小，亦即在各個過程中，暫存區的位址及大小是可變動的，以增加程式執行的效率。

第二章　概　論

第一節　組合程式的功能簡介

組合程式是一個系統程式，處理以組合語言（Assembly Language）所寫成的原始程式（Source Program），處理過後，產生目標程式（Object Program），此目標程式除了含有與原始程式相對應的機器語言（Machine Language）外，還必須含有其他的資料，以便利餵入程式（Loader）或連結程式（Linker）使用，我們可用下圖表示之：

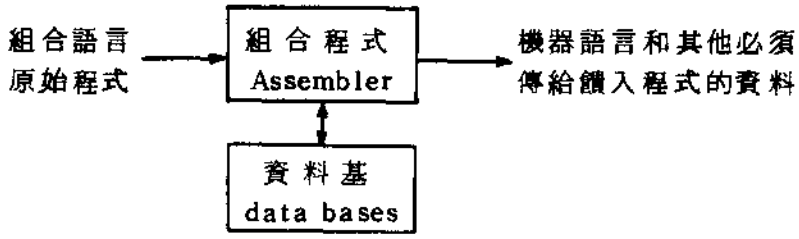

圖　1－1　組合程式的功能圖

微電腦系統的組合程式，所產生的目標程式含有四種資料：

1. 標頭（Header）
2. 外部符號表（External Symbol Table）
3. 機器碼（Machine Code）
4. 內部符號表（Internal Symbol Table）

其格式（Format）及用途，將於第三章中詳細敘述。

本書所要討論的是分析一個組合程式的設計實例，探討其中的架構及資料結構，使讀者能體會出理論與實例是如何配合的。

微電腦系統的組合程式是指令巨組合程式（Macro Assembler），它分成三個過程來完成，並採取記憶體重疊（Memory Overlay）的技巧，以避免佔用太多的記憶體空間。所謂記憶體重疊，就是將一個程式分成幾個過程處理，而這幾個過程分別在不同時間餵入（Load in）記憶體，已處理完畢的過程，其所佔用的記憶體可以讓給即將進入的過程，以節省記憶體空間。

在探討組合程式的架構之前，在此先將微電腦系統的組合語言，做一概略的介紹。

第二節　微電腦系統組合語言陳述的規格

組合語言程式的陳述包括四個欄，即址標欄（LABEL），操作指令欄（OPCODE），運算元欄（OPERAND），及註解欄（COMMENT）。

LABEL：址標；由一個以上的文數字（Alphanumeric）及"？"，"—"所組成，第一個字必須爲文字，且前六個字在整個程式中必須是唯一的。

OPCODE：操作碼，亦即指令欄。

OPERAND：爲運算元欄；可有一或兩個運算元，若爲兩個運算元則以逗號"，"分開。

COMMENT：註解欄；組合程式不做任何處理，註解前需加一分號"；"。

例如：

址標欄	操作指令欄	運算元欄	註解欄
VALUE:	DEFW	1000H	
	LD	HL,VALUE	;GET VALUE

例中VALUE 定義爲一個位址，此位址存放的資料爲1000H；LD爲一饋入的指令，HL，VALUE 爲兩個運算元，此指令之目的在於將VALUE 所代表的值饋入HL記錄器（Register），註解即說明此指令之意義。

運算元可爲一運算表示式，如LD A，X＋Y　X＋Y即爲一個表示式"＋"爲運算子（Operator）。組合程式，有下述的運算子：

運算子	功能	優先順序（Precedence Order）
+	UNARY PLUS	1
-	UNARY MINUS	1
.NOT.或＼	LOGICAL NOT	1
.RES.	RESULT	1
**	EXPONENTIATION	2
*	MULTIPLICATION	3
/	DIVISION	3
.MOD.	MODULO	3
.SHR.	LOGICAL SHIFT RIGHT	3
.SHL.	LOGICAL SHIFT LEFT	3
+	ADDITION	4
-	SUBTRACTION	4
.AND.或 &	LOGICAL AND	5
.OR.或 ↑	LOGICAL OR	6
.XOR.	LOGICAL XOR	6
.EQ.或 =	EQUALS	7
.GT.或 >	GREATER THAN	7
.LT.或 <	LESS THAN	7
.UGT.	UNSIGNED GREATER THAN	7
.ULT.	UNSIGNED LESS THAN	7

虛指令（Pseudo Instruction）

虛指令經組合程式處理過後，並不產生機器碼，其目的在於通知組合程式去完成特定工作。此類指令包括有：

ORG	nn	設定位址計數器的起始值爲 nn
EQU	nn	設定旗標的值爲 nn ，同樣的旗標只能設定一次
DEFL	nn	設定旗標的值爲 nn ，同樣的旗標可重覆設定
END		指示組合程式，原始程式已完
DEFB	n	定義位址計數器所指的位址，其內容爲 n
DEFB	"S"	定義位址計數器所指的位址，其內容爲字母 S
DEFW	nn	定義位址計數器所指的位址和下一位址，其內容爲 nn
DEFS	nn	自位址計數器所指的位址開始，保留 nn 個位元組（Bytes）
DEFM	"S"	定義字串所在的起始位址
DEFT	"S"	自位址計數器所指的位址開始，存字串的長度，下一位址存字串的起始位址
MACRO	#P1 #P2	定義指令羣
ENDM		指令羣定義結束記號
COND	nn	定義條件 nn ，若條件 nn 爲眞（nn 值≠ 0），則陳述 COND 和 ENDC 間的陳述被包含入，否則不被包含入
ENDC		條件結束記號

數字基（Number Bases）

微電腦系統的組合程式，可處理的數字基有：二進位、八進位、十進位、十六進位，在數字式後，分別加 B 、O 、D 、H 。例 1011B 、134O 、100D 、80H

組合程式命令（Command）

* Eject	跳頁
* Heading S	使字串 S 印在每一頁的開頭
* Listing OFF	暫停列表（List）
* Listing ON	繼續列表
* Maclist OFF	暫停指令羣擴展列表
* Maclist ON	繼續指令羣擴展列表
* Include filename	使原始程式 "filename" ，包含入此程式內
* Page line	告訴組合程式，一頁要印幾行

指令罩定義形式如下：

```
<name>  MACRO  <#P0>  <#P1>  <#P2>……<#Pn>
          ⋮
        ENDM
```

例：

```
FILLBL      MACRO       #BUFFER        #COUNT
            LD          B，#COUNT
            LD          HL，#BUFFER
FL#$YM      LD          (HL)，
            INC         HL
            DJNZ        FL#$YM
            ENDM
```

擴展如下：

```
            FILLBL      INBUF    50
            LD          B，50
            LD          HL，INBUF
FL0000      LD          (HL)，
            INC         HL
            DJNZ        FL0000
            ENDM
```

以上爲微電腦系統之組合語言，讀者若想更進一步的了解，請參閱微電腦系統組合語言程式手册和組合程式使用手册。

第三節　微電腦系統組合程式的命令使用法

組合程式命令（Command）的一般格式如下：

ASM　　file-name* 〔(Options)〕

ASM爲組合程式第一過程（PASS1）的執行名稱，第一過程執行完畢後，再把第二過程（ASM2）餵入記憶體執行，然後執行第三過程。"file-name"爲原始程式的檔案名稱，它是由使用者利用編校程式（EDITOR），先行建立在磁碟檔上；"file-name"右上角的星號"*"，表示組合程式同時可以處理一個以上的原始程式，將它們組成一個目標程式（Object Program）。

[(Options)]，中括號〔 〕表示其中的選擇性參數（Options）是可以取捨的，若使用者選用某一個或一個以上的參數時，則必須以小括號將它們含括起來。這些參數，便利使用者控制組合程式，於執行過程中有那些檔案或資料要輸出，及特殊處理規定。

以下將這些參數列出，並說明其用途：

Macro	使能處理指令群定義，若無此參數，則指令群定義和指令群的呼叫（Call）會被視爲錯誤。
Symbol	告訴組合程式要執行第三過程（PASS 3），以產生ISD (Internal Symbols) 塊（Block）附加在目標程式檔的後面。符號偵錯（Symbolic Debug）時，會使用到這些資料。
Xref	告訴組合程式要執行第三過程，以產生縱橫參考表（Cross Reference Table），附加在列表檔（List File）的後面。
Absolute	表示整個程式要被定址在絕對位址上（Absolute Address），若無此參數表爲可重新定址（Relocatable），因此目標程式將來在連結（Link）時可以重新定址。
NOList	告訴組合程式，不必產生列表檔（List File）
NOMaclist	告訴組合程式，指令群擴展陳述不包含在列表檔內。
NOOject	告訴組合程式，不必建立目標程式檔。
NOWarning	告訴組合程式，不必將警告訊息顯示出來。
Print	告訴組合程式，將原始程式的列表（Listing）輸出在SYSLST上，若無此參數，則只建立列表檔。 SYSLST是一個邏輯單元（Logical Unit）請參閱微電腦系統之微電腦作業系統使用手册。
I=file-name	定義中間檔（Intermediate File）的檔案名稱。
L=file-name	定義列表檔（Listing File）的檔案名稱。
M=file-name	定義指令群檔（Macro File）的檔案名稱。
O=file-name	定義目標程式檔（Object File）的檔案名稱。
T=file-name	定義符號表流溢檔（Symbol Table Overflow File）的檔案名稱。
X=file-name	定義縱橫參考檔的檔案名稱。

以上爲微電腦系統組合程式的命令使用法概述，讀者若想更進一步的瞭解其使用法，請參閱微電腦系統組合程式使用手册（筆者由於篇幅限制，無法在此詳述，特在此深致歉意）。

第三章　資料結構

第一節　資料之流程

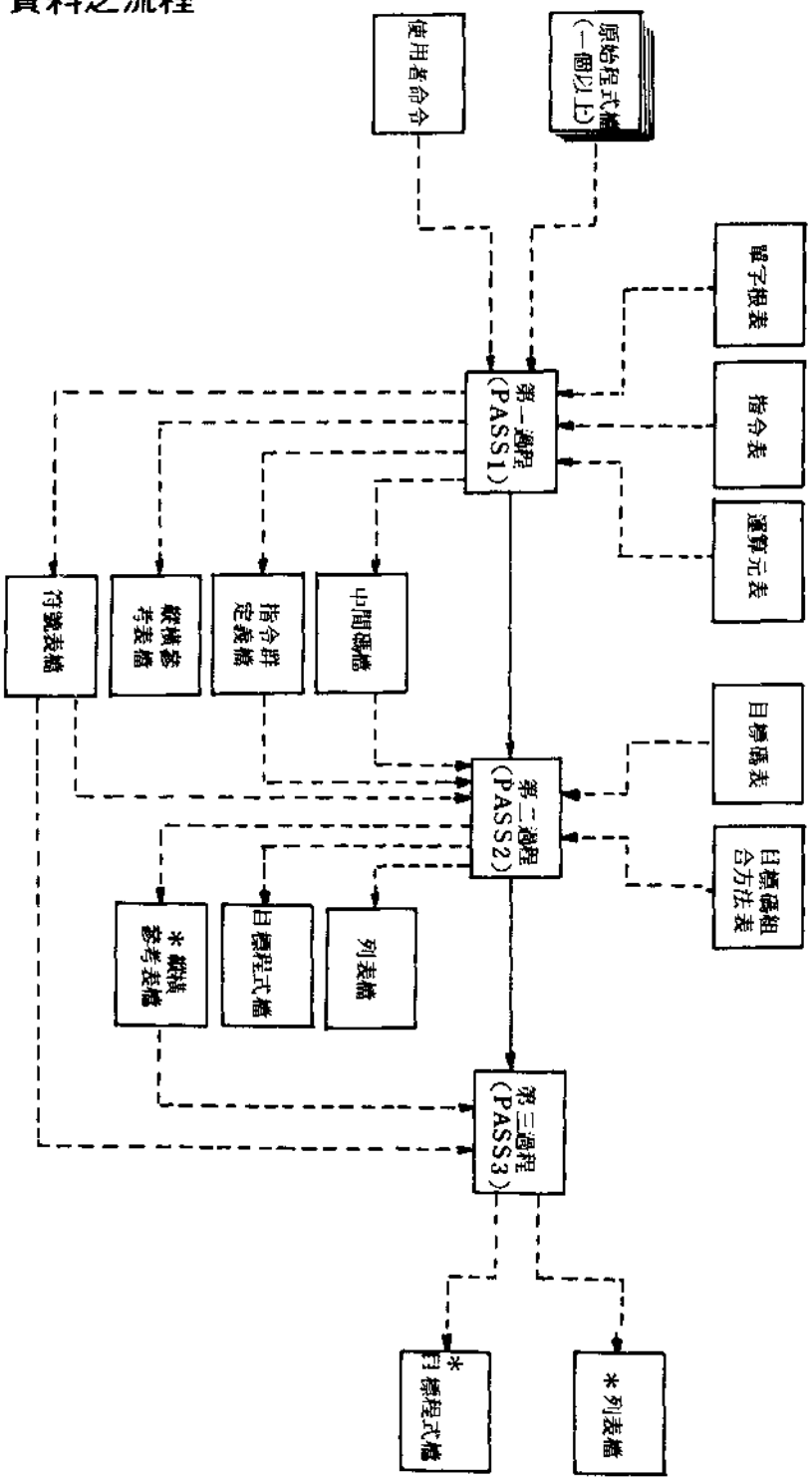

圖 3－1

第二節　檔案描述 (File Description)

組合程式根據用者在組合程式命令行中所指定的選擇性參數，來決定要建立那些輸出檔。

輸出檔可分爲下列幾項：

中間檔（Intermediate File）、列表檔（List File）、指令群定義檔（Macro Definition File）、目標程式檔（Object File）、符號表流溢檔（Symbol Table Overflow File）、縱橫參考檔（Cross Reference File）。其中中間檔、符號表流溢檔、縱橫參考檔、指令群檔都是暫時性檔案（Scratch File），若用者沒有指定一個完整的名稱予檔案，則組合處理完畢後，會自動消失。

例如：I＝5　使中間檔分配給主裝置驅動器5上的一個暫時性檔案，而I＝5／ I.FILE 則使中間檔 I.FILE 於組合處理完後，仍然保存在主裝置的驅動器5上。

以下對各個檔案的格式、作用，配合例3－2的範例，做一詳細的敍述。

一原始檔（Source File）：

原始檔爲執行組合程式時的輸入資料，其格式與用者在建立該檔時的格式完全相同。

```
LINE NO.                    STATEMENT
  001                GLOBAL       MAIN
  002                EXTERNAL     BUF LGTH
  003      MCR1      MACRO        #P1 #P2 #P3
  004                LD           A, #P1
  005                ADD          A, #P2
  006                LD           HL, #P3
  007                LD           (HL), A
  008                ENDM
  009      MCR2      MACRO        #P1 #P2
  010                LD           B, #P1
  011                LD           HL, #P2
  012      FL#$YM    LD           (HL), ' '
  013                INC          HL
  014                DJNZ         FL#$YM
  015                ENDM
  016      *M OFF
  017      SYM1      EQU          '0'
  018      SYM2      EQU          20H
  019      SUM       DEFS         2
```

```
020     MAIN    MCR1            SYM1 SYM2 SUM
                  :
                  :     MACRO EXPANSION
                  :
021             MCR2            SYM2 LGTH
                  :
                  :     MACRO EXPANSION
                  :
022             MCR2            SYM2 BUF
                  :
                  :     MACRO EXPANSION
                  :
023             LD              B, (HL)
024             LD              (HL), SYM2
025             SBC             A, (IX+SYM2)
026             LD              (IX+SYM2), SYM2
027             RET
028             END
```

例　3－2　程 式 範 例

例 3－2 的程式是用者在編校狀態（EDITOR MODE）時所建立的格式，而其眞正儲存在磁碟上的格式如表 3－3 所示。

```
09 47 4C 4F  42 41 4C 09  4D 41 49 4E  20 0D 09 45    *. GLOBAL. MAIN . . E*
58 54 45 52  4E 41 4C 09  42 55 46 20  4C 47 54 48    *XTERNAL. BUF LGTH*
0D 4D 43 52  31 09 4D 41  43 52 4F 20  09 23 50 31    *. MCR1. MACRO . #P1*
20 23 50 32  20 23 50 33  0D 09 4C 44  20 09 41 2C    * #P2 #P3. . LD . A, *
23 50 31 0D  09 41 44 44  09 41 2C 23  50 32 0D 09    *#P1. . ADD. A, #P2. . *
4C 44 09 48  4C 2C 23 50  33 0D 09 4C  44 09 28 48    *LD. HL, #P3. . LD. (H*
4C 29 2C 41  0D 09 45 4E  44 4D 0D 3B  0D 4D 43 52    *L), A. . ENDM. ; . MCR*
32 09 4D 41  43 52 4F 09  23 50 31 20  23 50 32 0D    *2. MACRO. #P1 #P2. *

09 4C 44 09  42 2C 23 50  31 0D 09 4C  44 09 48 4C    *. LD. B, #P1. . LD. HL*
2C 23 50 32  0D 46 4C 23  24 59 4D 09  4C 44 09 28    *, #P2. FL#*YM. LD. (*
48 4C 29 2C  27 20 27 0D  09 49 4E 43  09 48 4C 0D    *HL), ' '. . INC. HL. *
09 44 4A 4E  5A 09 46 4C  23 24 59 4D  0D 09 45 4E    *. DJNZ. FL#*YM. . EN*
44 4D 0D 3B  0D 2A 4D 20  4F 46 46 0D  53 59 4D 31    *DM. ; . *M OFF. SYM1*
09 45 51 55  09 27 30 27  0D 53 59 4D  32 09 45 51    *. EQU. '0'. SYM2. EQ*
55 09 32 30  48 0D 53 55  4D 09 44 45  46 53 09 32    *U. 20H. SUM. DEFS. 2*
0D 3B 0D 4D  41 49 4E 09  4D 43 52 31  09 53 59 4D    *. ; . MAIN. MCR1. SYM*

31 20 53 59  4D 32 20 53  55 4D 0D 09  4D 43 52 32    *1 SYM2 SUM. . MCR2*
09 53 59 4D  32 20 4C 47  54 48 0D 09  4D 43 52 32    *. SYM2 LGTH. . MCR2*
09 53 59 4D  32 20 42 55  46 0D 09 4C  44 09 42 2C    *. SYM2 BUF. . LD. B, *
28 48 4C 29  0D 09 4C 44  09 28 48 4C  29 2C 53 59    *(HL). . LD. (HL), SY*
4D 32 0D 09  53 42 43 09  41 2C 28 49  58 2B 53 59    *M2. . SBC. A, (IX+SY*
4D 32 29 0D  09 4C 44 09  28 49 58 2B  53 59 4D 32    *M2). . LD. (IX+SYM2*
29 2C 53 59  4D 32 0D 09  52 45 54 0D  09 45 4E 44    *), SYM2. . RET. . END*
0D FF AA AA  AA AA AA AA  AA AA AA AA  AA AA AA AA    *. . . . . *
```

表　3－3　原 始 檔

二中間檔

中間檔是介於原始檔與目標模組檔（Object Module File）間的一個暫時性檔案，其作用在於將原始檔內不同長度、型態的陳述，轉換成一致的格式，以便利組合程式處理。每一原始陳述（Source Statement）產生一對應的中間項（Intermediate Item）每一中間項包括有8個位元組，其存放內容如下所述：

位元組0、1——存放目前陳述的位址計數

位元組2 ——位元0～6存放目標碼的長度。
位元7標示陳述之址標欄中是否含有符號；若位元7爲1表含有符號址標，若爲0表不含符號址標。

位元組3 ——存放助憶指令（Mnemonic Instruction）亦即操作指令(Opcode)所轉譯得的中間碼。

位元組4 ——存放陳述開始位址至操作指令結束位址間之長度。

位元組5、6 ——存放運算元轉換所得的中間碼。
位元組5存放第一個運算元，位元組6存放第二個運算元。

位元組7 ——存放陳述被檢查到的第一個錯誤碼，若無錯誤發生則此位元組存放FEH。

LINE NO.		STATEMENT		INTERMEDIATE ITEM BYTE NO. 0	1	2	3	4	5	6	7
001		GLOBAL	MAIN	00	00	00	EC	07	FE	FE	FE
002		EXTERNAL	BUF LGTH	00	00	00	EB	09	FE	FE	FE
003	MCR1	MACRO	#P1 #P2 #P3	00	00	80	E7	0A	FE	FE	FE
004		LD	A, #P1	00	00	00	FE	03	FE	FE	FE
005		ADD	A, #P2	00	00	00	FE	04	FE	FE	FE
006		LD	HL, #P3	00	00	00	FE	03	FE	FE	FE
007		LD	(HL), A	00	00	00	FE	03	FE	FE	FE
008		ENDM		00	00	00	EA	05	FE	FE	FE
009	MCR2	MACRO	#P1 #P2	00	00	80	E7	0A	FE	FE	FE
010		LD	B, #P1	00	00	00	FE	03	FE	FE	FE
011		LD	HL, #P2	00	00	00	FE	03	FE	FE	FE
012	FL#$YM	LD	(HL),	00	00	00	FE	09	FE	FE	FE
013		INC	HL	00	00	00	FE	04	FE	FE	FE
014		DJNZ	FL#$YM	00	00	00	FE	05	FE	FE	FE
015		ENDM		00	00	00	EA	05	FE	FE	FE
016	*M OFF			00	00	00	FE	FE	FE	FE	FE
017	SYM1	EQU	'O'	00	00	80	E3	08	FE	FE	FE
018	SYM2	EQU	2OH	00	00	80	E3	08	FE	FE	FE
019	SUM	DEFS	2	00	00	80	E1	08	19	FE	FE
020	MAIN	MCR1	SYM1 SYM2 SUM	02	00	80	EE	06	FE	FE	FE
		:		02	00	02	4C	03	07	19	FE
		:	MACRO EXPANSION	04	00	02	58	04	07	19	FE
		:		06	00	03	AA	03	10	19	FE
		:		09	00	01	03	03	74	07	FE
021		MCR2	SYM2	0A	00	00	EE	02	FE	FE	FE
		:		oA	00	02	4C	03	00	19	FE
		:		0C	00	03	AA	03	10	19	FE
		:	MACRO EXPANSION	0F	00	82	4D	09	74	19	FE
		:		11	00	01	2D	04	10	EE	FE
		:		12	00	02	93	05	19	EE	FE
022		MCR2	SYM2 BUF	14	00	00	EE	02	EE	FE	FE
		:		14	00	02	4C	03	00	19	FE
		:		16	00	03	AA	03	10	19	FE
		:	MACRO EXPANSION	19	00	82	4D	09	74	19	FE
		:		1B	00	01	2D	04	10	FE	FE
023		LD	B, (HL)	1C	00	02	93	05	19	FE	FE
024		LD	(HL),SYM2	1E	00	01	02	03	00	74	FE
025		SBC	A, (IX+SYM2)	1F	00	02	4D	03	74	19	FE
026		LD	(IX+SYM20), SYM2	21	00	03	B2	04	07	81	FE
027		RET		24	00	04	CB	03	81	19	FE
028		END		28	00	01	3F	04	FE	EE	FE
				29	00	00	E5	FE	FE	FE	FE

表　3—4　中 間 檔

三指令群定義檔（Macro Definition File）

指令群定義檔內存放使用者自己定義的指令群，以便將來做指令群擴展(Macro Expansion）時使用。指令群檔所儲存的指令群是以記錄（Record)爲單位，每一指令群都有屬於自己的記錄，一指令群可以佔用一個以上的記錄，但一個記錄內不能存有二個以上的指令群。

指令群檔內記錄的格式，如下表所示：

```
4D 43 02 31  20 20 FE 09  4C 44 20 09  41 2C 23 31    *MC 1   . LD  A, #1*
0D 09 41 44  44 09 41 2C  23 32 0D 09  4C 44 09 48    *. . ADD. A, #2. . LD. H*
4C 2C 23 33  0D 09 4C 44  09 28 48 4C  29 2C 41 0D    *L, #3. LD. (HL), A. *
FF FE FE FE  FE FE FE FE  FE FE FE FE  FE FE FE FE    *. . . . . . . . . . . . . . . . *
FE FE FE FE  FE FE FE FE  FE FE FE FE  FE FE FE FE    *. . . . . . . . . . . . . . . . *
FE FE FE FE  FE FE FE FE  FE FE FE FE  FE FE FE FE    *. . . . . . . . . . . . . . . . *
FE FE FE FE  FE FE FE FE  FE FE FE FE  FE FE FE FE    *. . . . . . . . . . . . . . . . *
FE FE FE FE  FE FE FE FE  FE FE FE FE  FE FE FE FE    *. . . . . . . . . . . . . . . . *

4D 43 02 32  20 20 FE 09  4C 44 09 42  2C 23 31 0D    *MC 2   . . LD. B, #1. *
09 4C 44 09  48 4C 2C 23  32 0D 46 4C  23 30 09 4C    * LD. HL, #2. FL#0. L*
44 09 28 48  4C 29 2C 27  20 27 0D 09  49 4E 43 09    *D. (HL), ' '. . INC. *
48 4C 0D 09  44 4A 4E 5A  09 46 4C 23  30 0D FF FE    *HL   DJNZ FL#0. .  *
FE FE FE FE  FE FE FE FE  FE FE FE FE  FE FE FE FE    *. . . . . . . . . . . . . . . . *
FE FE FE FE  FE FE FE FE  FE FE FE FE  FE FE FE FE    *.    . . . . . . . . . . . . *
FE FE FE FE  FE FE FE FE  FE FE FE FE  FE FE FE FE    *. . . . . . . . . . . . . . . . *
FE FE FE FE  FE FE FE FE  FE FE FE FE  FE FE FE FE    *. . . . . . . . . . . . . . . . *
```

表 3—6 指 令 群 檔

在指令群記錄之開始保留7個位元組，其中6個位元組存放指令群的定義名稱，第7個位元組則表示此指令群名稱的結束，緊跟著下面的位元組即存放指令群的定義陳述（註一）。

存放在指令群檔中的指令群定義陳述與用者在建立原始檔時，唯一不同點即將指令群檔上所有參數符號轉換成在參數串（Parameter String）上的順序計數。如例3—2的原始程式範例中，有二個指令群定義MCR1及MCR2，在建立指令群檔時就有2個記錄MC.1及MC.2，如表3—6所述，讀者可比較原來在原始程式中的參數MCR1的#P1，#P2，#P3，#P4已轉換成#1，#2，#3，#4，MCR2中的#P1，#P2已轉換成#1，#2。

四符號表流溢檔（Symbol Table Overflow File）

符號表流溢檔是於記憶體中的符號表暫存區填滿時，自動在指定的主裝置邏輯單元8上所產生的檔案。符號表流溢檔存放所有在原始檔陳述中所定義的符號名稱，每一個符號名稱都是以佔8個位元組的符號項目(Symbol Item)來表示，其格式如下：