

# 前言



微處理機問世以來，應用之廣與人類生存息息相關，短短幾年震撼了整個世界，無人不以學習「微電腦」列為首要，尤其「工專」「工職」學校，適應時代潮流與國家實際需要，開「微電腦」課程，為國家培育微電腦人才，實感欣慰。但大部分學校都以開BASIC程式課程為主，而忽略了微電腦基本理論。這對電子科、電工科學生來講，只學BASIC程式是不夠的，應該重視微電腦基本結構，以及微電腦組合語言、程式設計，以達培育微電腦技術人才之目的。或許由於BASIC較好教、較易學習，但為整體技術之發展，微電腦技術人才之培養，無論再難，還是應注重微電腦基本理論結構之教育。由於基本理論較難教，且較枯燥；而實習教材又較少，有鑑於此，特編此書，專以實習教材為主，配合ABC-80微電腦學習機，以實驗方式來證明指令功能，提供給各學校作為教學參考，各人自習用，期能略盡微薄之力。

微電腦之學習，必須由認識微處理機之內部結構，了解其動作原理開始，及熟悉每個基本指令作用，進而學習程式設計，再進一步了解輸入／輸出之技巧、輸入／輸出界面晶片之構造功能與程式規劃，最後將微處理器ROM、RAM界面晶片，以及週邊設備等諸系統組件連接在一起，加上必要之軟體構成一部完整之微電腦系統。

本書內容涵蓋上述範圍，第一篇介紹Z-80微處理器之內部結構與動作原理、各腳解說、Z-80指令集、8255技術資料。第二篇實習部分，共有30個實習，60多個例題，分四大部分——基本指令練習、副程式應用練習、實用程式練習、擴充應用程式練習。全部實習從最基本指令練習直至擴充應用，無論對微電腦

初學者或有基礎者都很實用。以實際實驗代替只讀原理則較能領悟微電腦之奧妙，且對於程式結構，及程式設計技巧也較容易了解體會，而達到學以致用的效果。

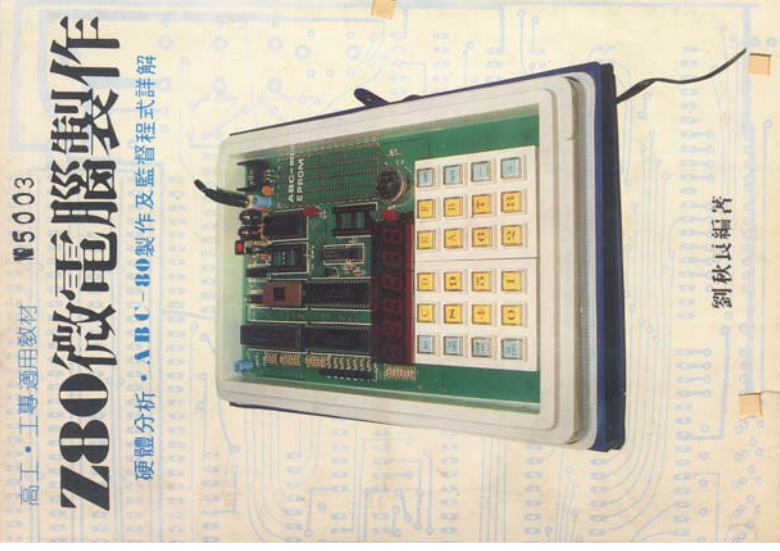
第三篇微電腦系統，以硬體結構最簡便，線路最容易了解且易操作之「ABC-80微電腦」為例，介紹其硬體、軟體規格，及操作、監督程式、副程式，以及EPROM燒錄器。讀者如能瞭解ABC-80之系統結構，就等於瞭解所有微電腦之基本結構了。

本書從基本原理、應用程式，微電腦系統依一貫性介紹，配合人手一機，時時練習，日久定能靈活應用，設計程式，人人成為微電腦高手。讓我們共同來為我國工業技術升級而努力。

本人才疏學淺，錯誤之處在所難免，誠盼專家先進不吝指正，不勝感激！

劉 秋 良

1982年6月



# 第一章

## Z80微處理器的結構

### 與各腳之功能

#### 1-1 Z-80 微處理器之結構

Z80為一具有40支接腳之大型積體電路。此40支接腳中，有16支是Z80微處理器輸出至外部記憶器之位址線（如圖之右側上方所示），此16條位址線構成了Z80之位址巴士（address bus）。由於具有十六位元之位址，因此，Z80微處理器最高能選取 $65536 (2^{16})$ 個外部（記憶或輸入／輸出）位置。

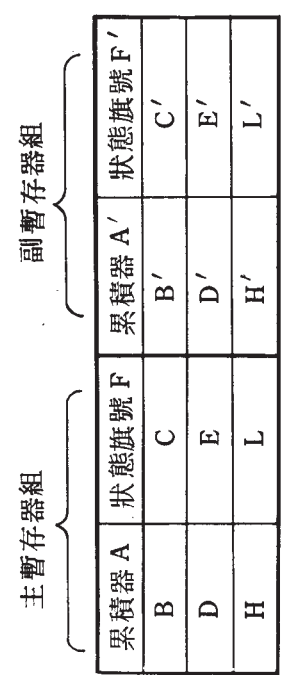
1-1圖之右側下方所示則為Z80微處理器與外部設備（如記憶器或輸入／輸出界面電路）互遞資訊的孔道。此一孔道包括8條資料能雙向（bidirectional）傳遞之電線，構成所謂的資料巴士（data bus）。

1-1圖之左側所示則為Z80微處理器之電源線與控制巴士（control bus）。控制巴士包括13種控制信號。這些控制信號可分為三類：系統控制、CPU控制、與CPU巴士控制。

#### CPU暫存器

Z80微處理器內部包含了一個208數元的讀寫記憶體，分別組合成18個8數元之暫存器與4個16數元的暫存器，其中8數元暫存器包括有兩套一般用途的暫存器，每組各有6個，他們可單獨的當8數元暫存器用，或者可以成對地組成16數元的暫存器。一般用途暫存器內還包括兩個累積器及兩個旗號暫存器（Flag register）。其他還有4個16數元的暫存器及8數元的

中斷向量暫存器 ( interrupt vector ) 及記憶翻新暫存器 ( Memory Refresh ) 亦各有其特殊的用途。



中斷向量高位址 暫存器 I	動態記憶體 翻新暫存器 R
指標暫存器	IX
指標暫存器	IY
堆疊指示器	SP
程式計數器	PC

圖 1 - 2 Z80 中處機內暫存器分類圖

### 1 - 1 - 1 一般用途暫存器

如圖 1 - 3 所示, Z80 微處理器包括十四個八位元之一般用途暫存器。此些暫存器分別稱作 A, B, C, D, E, H, L, 以及 A',

A		A'	
B	C	B'	C'
D	E	D'	E'
H	L	H'	L'

圖 1 - 3 Z80 之一般用途暫存器

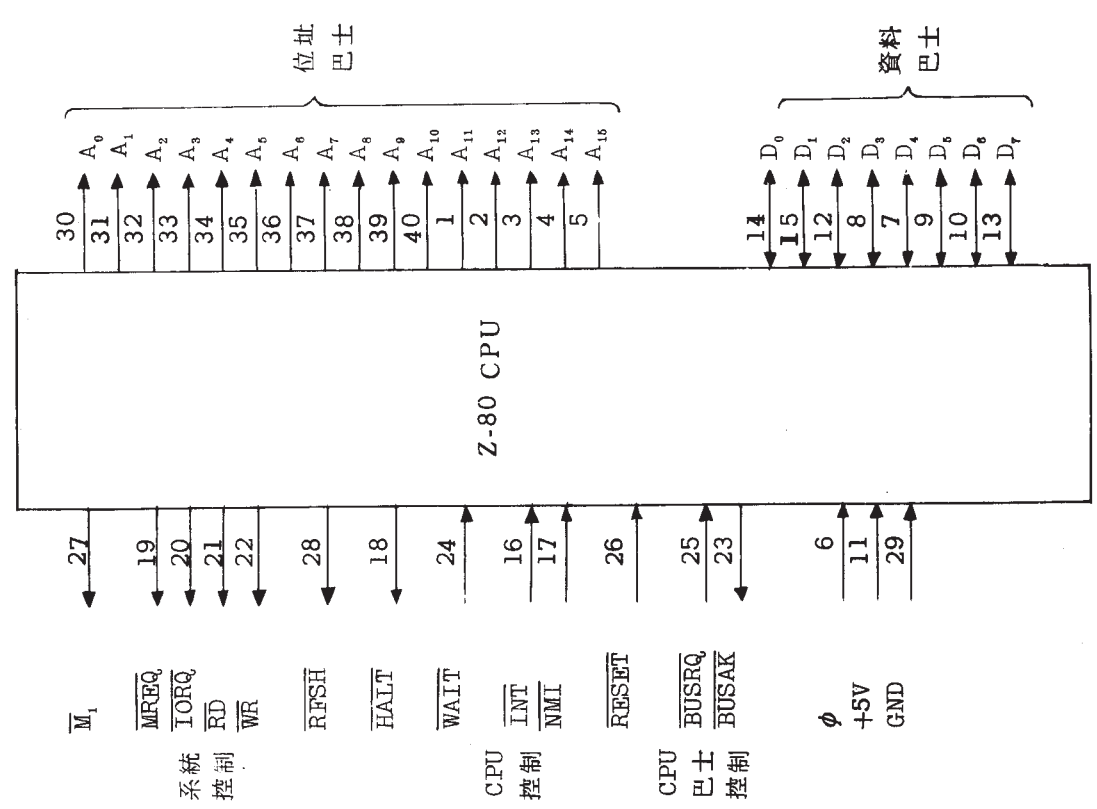


圖 1 - 1 Z-80 CPU 之接腳外觀圖

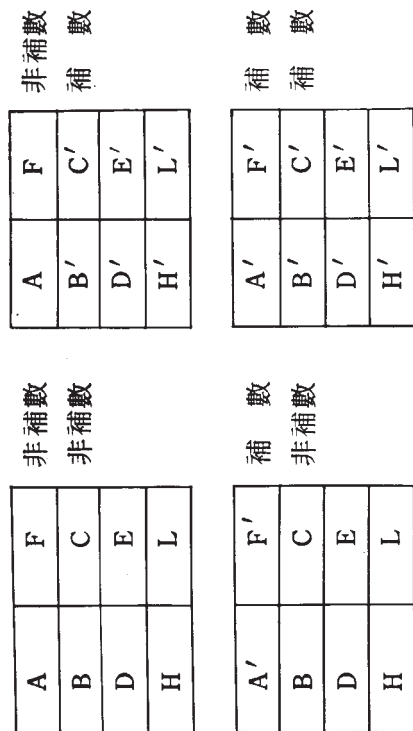


圖 1 - 4 Z80 一般用途暫存器之四種組合

B', C', D', H', 與 L'。在任一時刻，只有其中一組七個暫存器以及其對應之旗號暫存器 ( F 或 F' ) 能動作。有一特殊之 Z80 指令可用以從 A 與 F 或 A' 與 F' 兩組中選取一組動作，而另一指令則可用以於 B, C D E H L 或 B', C', D', E', H', L' 兩組中選取一組。AF 與其餘六個一般用途暫存器之四種組合情形，如圖 1 - 4。

具有兩組一般用途暫存器之優點為，程式設計者能迅速地由其中一組轉換至另一組。就最簡單之情況而言，這在微處理器內提供了更多之暫存器記憶設施。由於程式存取 CPU 暫存器之資料的速度，遠勝於存取記憶器者，因此，就資料處理速度而言，暫存器記憶優於記憶器記憶。此即所謂的記憶層次 ( memory hirarchy )；各種不同之記憶設備，具有不同之資料存取速度，宛如有“階級”一般。於較複雜之應用上，兩組暫存器之優點為，不用的一組可用以保存微處理器在接收到插斷以後的環境 ( environment ) ( 環境即為各種資料現況 )。

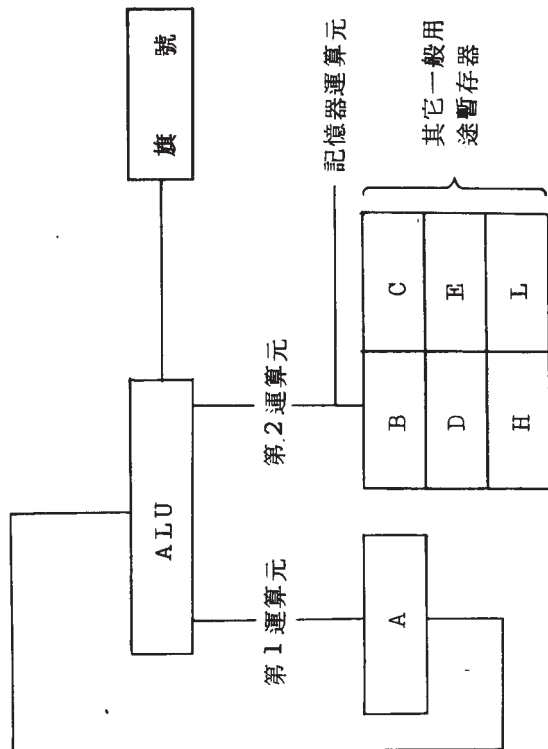


圖 1 - 5 算術與邏輯運算

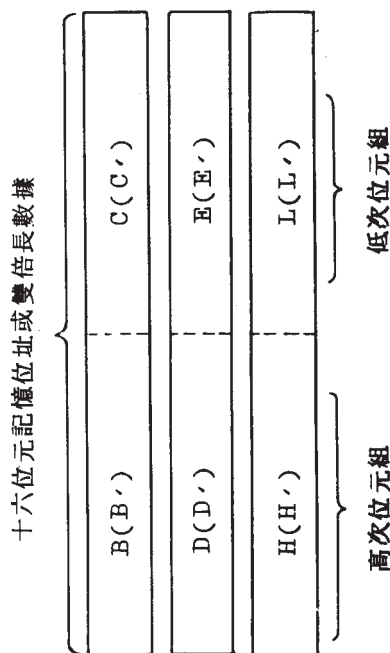


圖 1 - 6 暫存器對

正如 8080 一般，Z80 之一般用途暫存器之功能有點特殊。八位元資料可來回搬運於記憶器與任一（七者中）暫存器之間，或來回於任兩暫存器之間。不過，諸如兩運算之加或 XOR 等算術與邏輯運算，則僅能針對 A（或 A'）暫存器與另一暫存器或記憶位置。運算後之結果則恒存於 A 暫存器。換言之，A 暫存器即為一般所謂之累加器（accumulator）。一般而言，目前所選定之 A 暫存器即為 CPU 作算術與邏輯運算之主暫存器，此一情形如圖 1-5 所示。

A 暫存器除此六個一般用途暫存器，可兩兩組成三個暫存器對（register pair）：BC, DE, 與 HL。在 8008, 8080, 與 Z80 之許多運算上，此些暫存器對所儲存的即為一項位址。譬如，如圖 1-6 所示的，H 暫存器即儲存某項記憶位址之高（High）次八位元，而 L 暫存器則儲存該項記憶位址之低（Low）次八位元。BC 與 DE 兩暫存器對亦可儲存一記憶位址，因此，總共有三個暫存器對能儲存記憶位址指示器（pointer），指及記憶器中之資料。一般而言，這三個暫存器對將如圖 1-6 般地儲存記憶位址，但是，它們亦可作第二種用途——用於雙倍長算術（double-precision arithmetic），以儲存十六位元之數據。

雙倍長算術涉及十六位元數值之相加、相減、加一、或減一。Z80 之大多數算術與邏輯均以八位元為主，但卻容許三個暫存器對、堆疊指示器、與兩索引暫存器間進行有限之雙倍長算術。此一觀點主要在於便利記憶位址之運算。因為，若無十六位元算術，則長十六位元之記憶位址的運作，就必須分兩次進行。圖 1-7 所示即為以暫存器對作雙倍長算術之情形。運算所需之兩十六位元運算元皆來自暫存器對或十六位元暫存器，結果亦存回此些暫存器（對）。運算結果影響旗號。

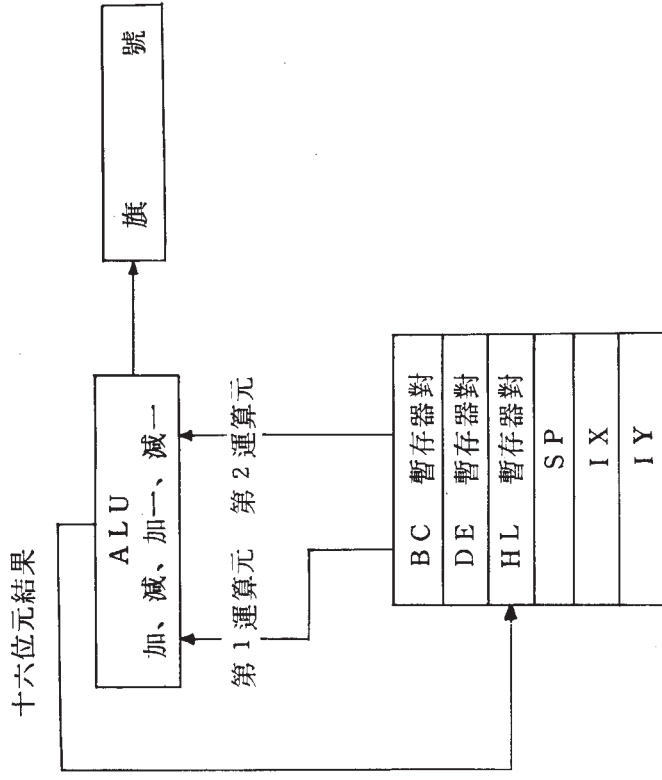


圖 1-7 暫存器對雙倍長運算

## 1-1-2 旗號暫存器

Z80 微處理器有兩個旗號暫存器 F 與 F'，但每一時刻僅能有一者被使用。F 或 F' 之選擇與 A 或 A' 之選擇一致。任一時刻，若非 A 與 F，則即為 A' 與 F' 被選取。如圖 1-8 所示，Z80 微處理器之旗號暫存器包括六種不同之旗號（flag）（旗號暫存器有兩個位元沒用）。此些旗號指明了，在算術、邏輯、或其他運算後，微處理器之各種狀態。緊接，我們將此六種旗號，分別介紹於下：

Z80 旗號暫存器之六種旗號中，有四個是可以指令加以測試的。它們是：



1. 進位旗號 ( C )。該旗號值即等於累加器最高次位元所產生之進位值。加法運算時，若累加器之最高次位元產生進位，則進位旗號值置定為 1；否則，其值為 0。減法運算時，若累加器之最高次位元產生借位，則該旗號值亦置定為 1。除此之外，

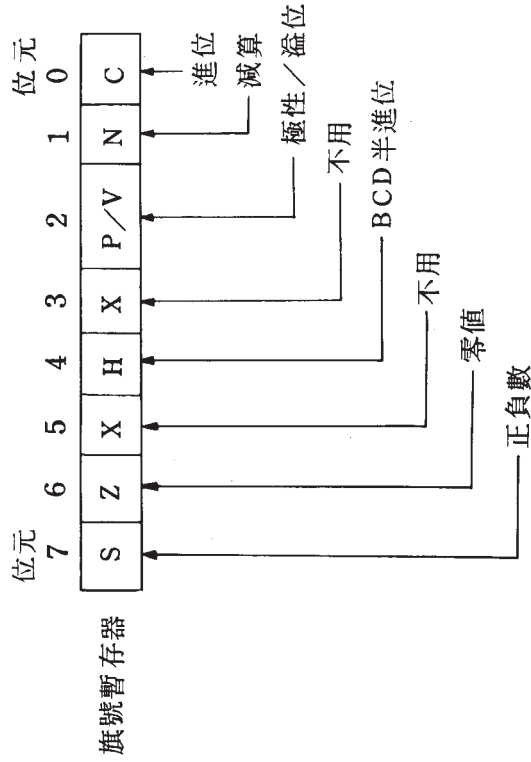


圖 1 - 8 Z80 微處理器之旗號暫存器

移位與旋轉指令亦會影響進位旗號之值。

2. 零值旗號 ( Z )。若運算結果使累加器之內含為零，則該旗號之值置定為 1；否則，其值為 0。

3. 正負數旗號 ( S )。該旗號主要用於有號數之運算。若運算所得之結果為負數，則該旗號之值置定為 1；否則，旗號之值為 0。由於八位元二進數之第 7 位元 ( MSB ) 即代表該數之正負數，並且以 1 代表負數，0 表正數。因此，事實上，正負旗號 ( S ) 之值即等於累加器之第 7 ( 最高 ) 位元值

4. 極性/溢位旗號 ( P/V )。該旗號為一雙用途旗號。邏輯運算 ( 如 AND A, B ) 時，其顯示累加器所含結果之極性 ( parity )。而有號 2 補數算術時，其代表溢位。

若算術運算所得之結果，超過八位元 2 補數形式所能表示之範圍 ( - 128 + 127 )，則溢位旗號之值為 1。

在邏輯運算 ( AND, OR, XOR ) 時，若所得結果為偶極性，則 P/V 旗號之值為 1，否則，若所得結果為奇極性，則該旗號之值為 0。

Z80 旗號暫存器之旗號中，有兩者是不可測試的，該兩旗號均用於 BCD 算術。

1. 半進位旗號 ( H )。此位元代表 BCD 算術時，低次四位元所產生之進位或借位。Z80 微處理器在執行 DAA ( 十進制調整 ) 指令時，該旗號值用以校正先前兩濃縮 ( packed )

BCD 數之加 ( 減 ) 算結果。

2. 減算旗號 ( N )。由於加算與減算後之 BCD 調整方式不同，因此，該旗號用以記錄 Z80 微處理器剛剛所執行的是加法運算或減法運算，以作為 DAA 運算校正之根據。

### 1 - 1 - 3 特殊用途暫存器

所謂特殊用途暫存器，即為在設置時即已指定其專作某一種用途之暫存器。Z80 微處理器具有六個特殊用途暫存器；茲將此六個暫存器分別描述於後：

#### 程式計數器 ( PC )

程式計數器即為儲存微處理器正在拿取之指令，所在位置之記憶位址的十六位元暫存器。每當程式計數器之內含被送出至位址巴士時，其內含值就自動加一。若程式跳越 ( program jump ) 發生，則新的內含值會自動存入程式計數器，蓋過原先之內含值。

#### 堆疊指示器 ( SP )

當程式計數器之內含指至外部記憶體中，次一欲執行指令之位置時，堆疊指示器則指至外部記憶體堆疊器之堆疊頂端位置。記憶體堆疊器之觀念雖非微處理器所僅有，但實際上幾乎每一微處理器均具有堆疊器之設施。所謂記憶堆疊器 ( memory stack )，事實上即為一段特別騰出，以作 CPU 暫存器，旗號暫存器，與程式計數器內含之暫時儲存的記憶區域。Z80 之記憶堆疊器，可座落於外部讀寫記憶體 ( RAM ) 之任意位置。由於用以儲存“堆疊頂端”之記憶位址，因之，Z80 之堆疊指示器的長度為十六位元。

記憶堆疊器具有後進先出 ( LIFO ) 之特性。Z80 可執行一推入 ( PUSH ) 指令，將某一 CPU 暫存器之內含推入堆疊器，同時亦可執行一拉取 ( POP 或 PULL ) 指令，將資料自堆疊器拉回至某一特定之 CPU 暫存器。自堆疊器拉取之資料，恒為最後被推入者，此一特性即如自助餐廳之堆盤一樣，最後被置於盤堆之盤子 ( 資料 )，一定位於盤堆之最頂端 ( 堆疊頂端 )，因此，亦為第一個被取出者。

由於 Z80 之記憶堆疊器是往位址漸小之方向延伸，因此，每一項資料被推入堆疊器，堆疊指示器之內含即自動減一。反之，每有一項資料自堆疊器取出，堆疊指示器之內含值即自動加 1，以指至新的堆疊頂端。圖 1 - 9 所示即為堆疊指示器之作業情形。

除了平常程式設計者可用之作為資料之暫時寄存所外，堆疊器之設置主要在於便利插斷 ( interrupt ) 之處置與副程式巢串 ( subroutine nesting )。

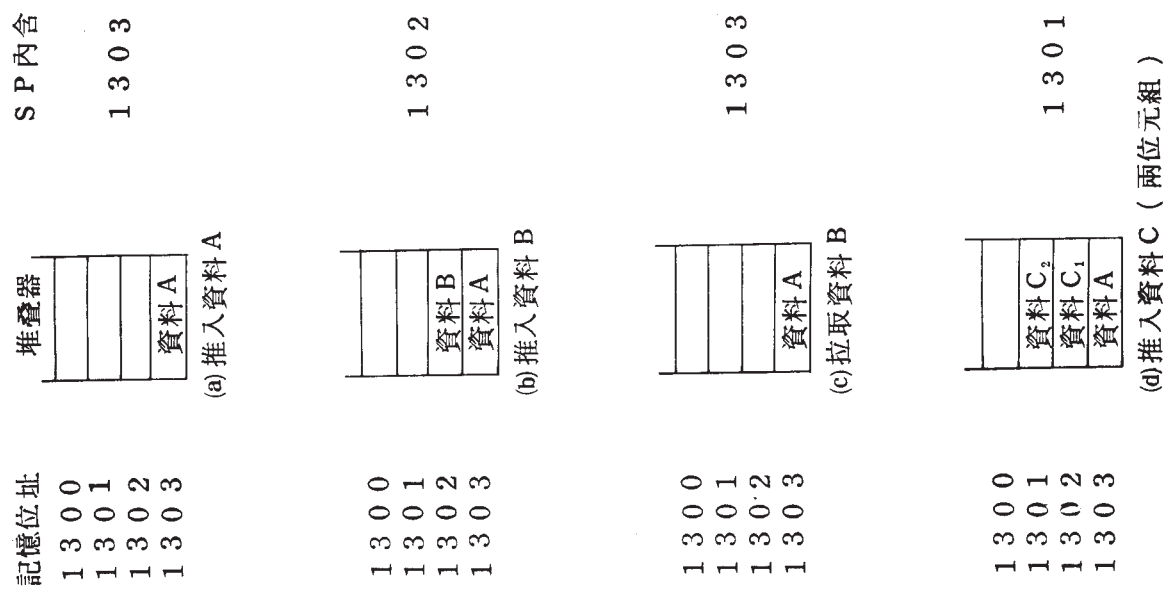


圖 1 - 9 堆疊指示器 ( SP ) 之作業

## 兩個指標暫存器 (IX, IY)

此兩個 16 數元暫存器在指標址法 (indexed Addressing mode) 中擔任一基準位址，另由指令提供一個數元組作為指標的移動量，在處理數據表 (Table) 時此種搜址法相當有用。指標的移動量是以 2 進位補數的形態直接存於指令，可由基準點前進 127 及後退 128 個數元組。

**中斷向量高位址暫存器 (I)**

8 數元暫存器，配合中斷發生時，由產生中斷之裝置送來 8 數元之低位址，形成一 16 數元之位址，中處機自動執行一間接呼叫 (indirect call)，由此位址取出中斷服務程式的真正開始位址，這樣只需改變此位址的內含即可任意地安排中斷服務程式的起始位置。中斷裝置送來之低位址的 LSB (least significant bit) 通常為零，而使中斷服務程式的起始位址永遠存於一偶數位址上。

## 記憶體翻新暫存器 (R)

Z80 微處理器在每次操作碼取還時都利用操作碼解碼時，位址排上空著的時間送出此 7 數元的暫存器到位址排之低位址上，配合微處理器上 RFSH 之控制信號，供動態記憶體不斷翻新 (refresh) 用，每一新操作碼取還後此暫存器便自動加一，RFSH 還用以控制記憶體之 CE (chipenable) 這樣不但使設計者能像處理靜態記憶體一般地容易使用動態記憶體，而且絲毫不減慢微處理器的運算速度，設計者可以指令檢查 R 暫存器的內含，但不具任何意義，故程式設計通常不用及此暫存器。

## 1-2 微處理器之各腳解說

Z80 微處理器為一 40 腳 DIP (Dual in-line) 形態之微處理器，其各出入腳如圖 1-1 所示。

**A<sub>0</sub>-A<sub>15</sub>**：三態輸出，高態動作 (Active high)，此 16 數元位址排可供 64 K 之記憶體交換數據時用，輸入輸出埠則利用 A<sub>0</sub>-A<sub>7</sub> 選擇 256 個輸入埠與 256 個輸出埠，當 RFSH 時 A<sub>0</sub>-A<sub>6</sub> 上則含有正確的翻新位址。A<sub>0</sub> 是 LSB (Least significant bit)，A<sub>15</sub> 是 MSB (most significant bit)。

**D<sub>0</sub>-D<sub>7</sub>**：雙向三態輸入/輸出，高態動作，此雙向數據排負責微處理器與記憶體及輸入輸出埠間之數據交換。

**M1**：輸出，低態動作 (Active low)，指令執行中每一操作碼 (op code) 取還週期時產生，即 M<sub>1</sub> 指示目前的機器週期是操作碼讀取週期，Z80 也使用兩個數元的操作碼，其開頭的數元組一定是 CBH, EDH 或 FDH (H 表示十六進位)，此時每一指令就有兩個機器週期完成操作碼的讀取，也就是有兩次 M1 產生。另外 M1 還與 IORQ 同時產生以表示一中斷接受 (interrupt Acknowledge) 週期。

**MREQ**：三態輸出，低態動作，在每一記憶體的讀或寫週期時產生，以指示目前的位址排上之位址應送到記憶體上去作讀或寫的工作。通常在與記憶體相關指令執行中產生。

**IORQ**：三態輸出，低態動作，IORQ 指示出，目前的位址排上之位址應用以選擇一輸出或輸入埠作一 I/O 的讀或寫，通常在輸入輸出指令執行時產生。IORQ 在輸入輸出指令執行時產生，但不與此指令讀取時之 M1 重疊，即 IORQ 與 M1 不會在 I/O 指令中同時出現，故 M1 與 IORQ 同時發生便可用作中斷接受的指示，此時中斷的最低位址部份經數據排進到微處理器，經一間接呼叫再進入中斷服務程式。



RD : 三態輸出, 低態動作, RD 指示微處理器要自記憶體或輸入輸出埠上經數據排讀入一數據。

WR : 三態輸出, 低態動作, WR 指示微處理器已準備好一穩定的數據在數據排上, 要寫入記憶體或輸入輸出埠內。

RFSH : 輸出, 低態動作, RFSH 指示位址排上最低的七個數元上的位址可供動態記憶體作記憶翻新 ( refresh ) 用, 此信號應與 MREQ 配合在適當的時間對所有的動態記憶作一遍翻新。

HALT : 輸出, 低態動作, HALT 指示微處理器正執行一暫停指令 ( HALT ), 此與 NOP 指令作用相同, 但程式計數器值並不改變。指令繼續執行時才能維持動態記憶體的翻新。只有中斷要求, 包括不可隱中斷及開放態下的可隱中斷, 才能使 HALT 繼續新位址的指令執行, HALT 中如有 BUSRQ 信號進入, 可使微處理器離開 HALT, 但 BUSRQ 消失後仍舊回到此 HALT 狀態。

WAIT : 輸入, 低態動作, WAIT 通知微處理器, 位址排所選到的記憶位置或輸入輸出埠上還沒準備好, 暫時不能作數據交換, 此時微處理器自動進入一連串的等待狀態, 並不斷檢視此 WAIT 信號, 到其成高態時才進行數據交換, 這樣微處理器便可與任何速度的記憶體或輸入輸出埠同步工作而無需降低微處理器的整個時序的頻率。

INT : 輸入, 低態動作, 微處理器在每一指令週期之最後一時序檢視此輸入, 如發現其為低態時, 在新指令週期一開始便送出 M<sup>1</sup> 及 IORQ, 當作中斷接受的指示, 微處理器此時有三種進入中斷程式的方式, 這也是微

處理器的一控制指令。微處理器要能認可 INT 信號前之先決條件為, 沒有 BUSRQ 信號進入且內部的中斷控制正反器 ( IFF ) 處於開放狀態 ( enable state ), 此 IFF 亦可經軟體指令 ( DI, EI ) 隨意地控制其開放與否。

NMI : 輸入, 負緣驅動 ( negative edge triggered ), NMI 較 INT 的優先次序 ( priority ) 為高, 故不論 IFF 是否處於開放態, 在 NMI 發生時的指令週期一結束, 微處理器便自動將程式計數器之值存入堆中, 而將 66 H 放入程式計數器中, 由該處繼續指令的執行。能夠阻止 NMI 進行的方法只有在 NMI 發生的指令週期內用 WAIT 使其永遠處於 WAIT 態或使用較 NMI 優先次序更高的 BUSRQ 來將 NMI 掩蓋掉 ( override )。

RESET : 輸入, 低態動作, RESET 強迫程式計數器歸零, 同時完成 4 件準備工作。

(1) 封閉 ( Disable ) IFF, 禁止所有 INT 的發生。

(2) 令 I 暫存器值為零。

(3) 令 R 暫存器值為零。

(4) 令微處理器以方式零來處理中斷情況。

在 RESET 過程中, 所有位址排與數據排進入高阻抗態, 所有控制信號輸出也都不動作 ( inactive state )。

BUSRQ : 輸入, 低態動作, BUSRQ 將強迫微處理器之位址排, 數據排的輸出進入高阻抗態, 而使其他的裝置取代微處理器來控制位址排與數據排, BUSRQ 是在每一機器週期中便作一次檢視, 不同於 INT 是在每一指令週期結束前才作一次檢視, 當 BUSRQ 回到高態時

## 第二章

# Z80指令集

CPU的硬體結構，前章我們已介紹過，本章旨在介紹Z80微處理器的指令集以便讀者能儘快寫程式。

首先概略的解說計算機的指令。

計算機之指令一般可分為五類：

1. 資料傳送指令→主要將資料傳遞於暫存器間或暫存器與記憶器之間及暫存器與輸入／輸出之間。
2. 資料處理指令→分為算術運算、邏輯運算，加一與減一指令、位元運算、移位與旋轉指令。
3. 測試與控制轉移指令→測試指令測試某一特定暫存器之某些位元值，看其為1或為0，最低程度其必須能測試旗號暫存器。控制轉移指令又分為跳越、相對跳越（條件跳越與無條件跳越兩種），副程式叫用等。
4. 輸入／輸出指令→這是管理輸入／輸出設備之特殊指令。
5. 控制指令→控制指令供應同步信號，並且延緩或插斷某一程式之執行。它們亦可作為中斷或模擬插斷。

Z80微處理器主要設計以取代Intel 8080微處理器，並提供額外之能力。因此，Z80之指令集包含了Intel 8080之所有指令，並且還增加一些新的指令。看八位元運算碼之有限位元數，您可能極想知道Z80之設計者究竟如何達成上述目的。其實，這乃用盡8080所餘之空白運算碼，以及索引指令時再增加一位元組之運算碼的結果。後者使有些Z80的指令長達五個位元組。

任何程式都有好幾種不同之寫法，徹底地認識與了解整個指

，微處理器再接著由該機器週期繼續工作。

**BUSAK**：輸出，低態動作，BUSAK是微處理器在交出位址排與數據排之控制權時，同時送出的通知外接裝置的信號。微處理器對BUSRQ取樣，然後同步地送出

BUSAK，所以BUSRQ與BUSAK波形相似，只是一為非同步輸入，一為同步的輸出。（同步是指系統時序而言）

**$\Phi$** ：單相且TTL位準的時序脈波，只需使用一330 $\Omega$ 之提升電阻到+5V上，便可開始工作。

令集，乃是寫出有效率之程式所不可或缺。初學程式設計者並不一定須寫最精簡、有效率之程式。因此，初讀本章時，詳細記得各個指令並不重要。重要的是要記得指令之種類並且研習典型之例子。然後，於寫程式時，再參考 Z80 指令集之說明，選擇最適合自己所需之指令。本章，我們以簡化的方式，歸納介紹 Z80 之各種指令型態。

## 2-1 資料傳送指令

於 Z80，資料傳送指令可分為四類：八位元傳送，十六位元傳送，堆疊作業，與區段傳送。茲將其分別介紹於後。

### 2-1-1 八位元傳送

於 Z80，所有八位元之資訊傳送皆以取入 (load) 指令達成。取入的動作是將資料自某一來源位置傳至目的位置，而不改變來源位置之原有內含。取入指令之格式為

LD dst, src

其中，dst 表資料之目的地，其可為某一 CPU 暫存器或某一記憶位置。src 為資料來源，其亦可為某一 CPU 暫存器或某一記憶位置。

將資料取入累加器 A 可使用暫存器，暫存器間接，索引，擴展，與立即等多種定址法。其指令格式分別為

LD A, r  
 (HL), (BC), (DE) (暫存器間接)  
 (IX+d), (IY+d) (索引)  
 (nn) (擴展)  
 n (立即)

將資料取入 B, C, D, E, H, 或 L 等暫存器，則可使用暫存器，暫存器間接，索引，與立即等定址法。

LD B, r

(HL)  
 (IX+d), (IY+d)

n

其中，r 代表 A, B, C, D, E, H, 或 L 等任一暫存器，而 n 為一八位元立即值。為了簡明起見，目的暫存器欄僅以 B 代表，其亦可為 C, D, E, H, 或 L。

將 CPU 暫存器內含“存出”至記憶器之指令，則只需將上述指令之目的欄與來源欄對調即可。當然，此時目的欄是不能放立即運算元的，其必須恒為一記憶位址。

附帶一提的，特殊暫存器 I 與 R 之內含皆可取入累加器 (利用隱含定址)。同時，反之，累加器之內亦可傳至 I 與 R。

下面我們舉幾個實際例子。

1. LD B, A

為一暫存器對暫存器傳輸指令。該指令將累加器 A 之內含抄入 B 暫存器。指令執行完後，累加器 A 之內含變成同於 B 暫存器之內含。於 Z80，任何兩八位元之一般用途暫存器間，資料皆可直接傳送。

2. LD A, (1234H)

該指令將位址 1234 (十六進制) 之記憶位置的內含，取入累加器 A。注意該擴展定址指令之記憶位址的寫法。“1234H”表“十六進數 1234”。(1234H) 表位址 1234 之記憶位置的“內含”。小括弧即表內含。該指令譯成機器碼儲存在記憶器中之情形為

位址	A	3A	運算碼
	A + 1	34	低次位址
	A + 2	12	高次位址

表 2-1 Z-80 八位元指令群 表 2-2

來源 目的	隱含		暫存器								暫存器間接		索引	擴展	立即
	I	R	A	B	C	D	E	H	L	(HL)	(BC)	(DE)			
A	ED 57	ED 5F	7F	78	79	7A	7B	7C	7D	7E	0A	1A	DD 7E d	FD 3A n	3E n
B			47	40	41	42	43	44	45	46			DD 46 d	FD 46 d	06 n
C			4F	48	49	4A	4B	4C	4D	4E			DD 4E d	FD 4E d	0E n
D			57	50	51	52	53	54	55	56			DD 56 d	FD 56 d	16 n
E			5F	58	59	5A	5B	5C	5D	5E			DD 5E d	FD 5E d	1E n
H			67	60	61	62	63	64	65	66			DD 66 d	FD 66 d	26 n
L			6F	68	69	6A	6B	6C	6D	6E			DD 6E d	FD 6E d	2E n
(HL)			77	70	71	72	73	74	75						36 n
(BC)			02												
(DE)			12												
(IX +d)			DD 77 d	DD 70 d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d					DD 36 d	3E n
(IY +d)			FD 77 d	FD 70 d	FD 71 d	FD 72 d	FD 73 d	FD 74 d	FD 75 d					FD 36 d	3E n
(m)			32 n												
I			ED 47												
R			ED 4F												

特別留意到，任何記憶位置之內含欲取入 A 以外之 CPU 暫存器，都必須利用 HL 之暫存器間接定址，而不能如取入累加器 A 般地使用擴展定址。例如，若欲將 25CD 存入 HL 暫存器對以十六位元取入指令），然後再執行

LD B, (HL)

指令。

3. LD C, 15H

該指令將立即數據一一十六進數 15 ( = 21<sub>10</sub> ) 取入 C 暫存器。

## 2-1-2 十六位元傳送

基本上，我們可將一十六位元之立即數據，某兩連續記憶位置之內含（擴展定址），或堆疊頂端之資料（即堆疊指示器所指之位置內含），取入任一十六位元之暫存器對，BC, DE, HL, SP, IX 或 IY。相反地，這些暫存器對之內含，亦可以相同方式存至某兩連續記憶位置或堆疊頂端。除此之外，HL, IX, 與 IY 等暫存器之內含，亦可取入 SP，此一能力簡便了多重堆疊器之製作。最後，AF 暫存器對之內含，亦可推入堆疊器，或將堆疊頂端資料取出存入 AF 暫存器。

表 2-2 所示即為 Z80 之所有十六位元取入指令。堆疊作業之推入（PUSH）與拉取（POP）指令，亦算在此一群指令內。留意，Z80 並無存取某一八位元暫存器之推入或拉取指令。所有堆疊作業指令均將資料來回傳遞於某一暫存器對與堆疊頂端之間。例如：

PUSH AF

指令，即為一將 AF 暫存器對之內含，存入堆疊器之單位元組指令，其運算碼為 F5H。該指令執行時所發生之作業系列為：

SP 值減 1

LD (SP), A

記得，於 Z80，堆疊指示器永遠指至堆疊頂端（即堆疊最頂之一項資料）位置，因此，在推入之前，堆疊指示器之值必須先減 1，以指至次一可供利用之位置。

拉取指令正巧與推入指令完全相反。例如，Z80 執行

```
POP AF
LD F, (SP)
SP 值加 1
LD A, (SP)
SP 值再加 1 (指至新堆疊頂端)
```

注意，每次取一個位元組後，堆疊指示器的內含值即自動加一，以指至新的堆疊頂端。

於堆疊作業時，運算元之十六位元中，高次之位元組恒先推入，後拉取。亦即：

```
PUSH DE 為 D 先推入，然後 E。
PUSH HL 為 H 先推入，然後 L。
POP HL 為先拉取至 L，後 H。
```

最後，我們分別舉一擴展定址與擴展立即定址之十六位元取入指令為例。

擴展立即定址指令：

```
LD IX, 0379H
```

將十六位元之立即數據 0379 取入索引暫存器 IX。指令執行完後，IX 索引暫存器將含 0379 (十六進)。

擴展定址指令

```
LD DE, (024AH)
```

將位址 024AH 之記憶位置的內含取入 E 暫存器，並且位址 024BH 之記憶位置的內含取入 D 暫存器。(圖 2-1)

特別注意，於 Z80，擴展定址指令儲存於記憶體時，運算元

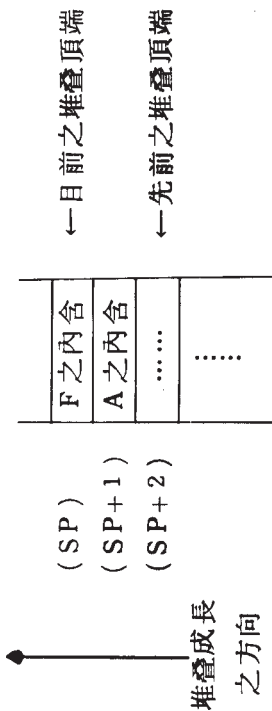
SP 值再減 1  
LD (SP), F

表 2-2 十六位元取入指令組與「推入」及「拉取」指令

目的 來源	暫存器					擴展 立即	暫存 擴展	暫存 器間接
	AF	BC	DE	HL	SP	IX	IY	(nn) (SP)
AF								F1
BC						01 n n		ED 4B C1 n n
DE						11 n n		ED 5B D1 n n
HL						21 n n		2A n n E1
SP				F9		31 n n	FD F9	ED 7B n n
IX						DD 21 n n		DD 2A E1 n n
IY						FD 21 n n		FD 2A E1 n n
擴展 定址	(nn)	ED 43 n n	ED 53 n n	22 n n	ED 73 n n	DD 22 n n	FD 22 n n	
暫存 器間接	(SP)	F5	D5	E5		DD E5	FD E5	

注意：每次執行後，推入與拉取指令均改變 SP 值。

指令執行過後，堆疊器之情形為





位址之低次八位元永遠在前——位址較小之記憶位置，而高次八位元恒在後。此外，所有十六位元資料傳送指令之作業亦同。任何暫存器對（或十六位元暫存器）之內含只要儲存至記憶器，低次之八位元一定儲存在前（位址較小之記憶位置），而高次八位元一定在後。

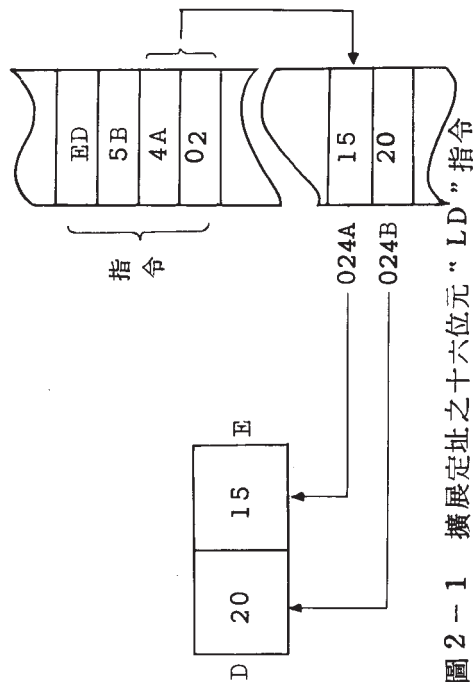


圖 2-1 擴展定址之十六位元“LD”指令

### 2-1-3 交換指令

暫存器間	隱含定址			
	AF	BC', DE', & HL'	HL	IX IX
AF	08			IY
BC DE & HL		D9		
DE			EB	
(SP)			E3	DD
			E3	FD
				E3

表 2-3 交換指令 EX 與 EXX

表 2-3 所示即為 Z80 特有之交換指令。交換指令將兩特定位置之內含互換，因此，為一雙重資料傳輸。EX 指令能將堆疊頂端之資料（連續兩個位元組），與 HL, IX, 或 IY 暫存器之內含互換。此外，其亦可將 DE 與 HL，或 AF 與 AF' 兩暫存器對之內含互換。

其次，EXX 指令將兩組六個一般用途暫存器之內含互換。

### 2-1-4 區段（整批）傳輸指令

區段傳輸指令導致整個記憶區段之資料（亦即，整批資料）的傳送，而非僅單位元組或雙位元組之資料。對微處理器之製造商而言，區段傳輸指令較難製作；但對程式設計而言，其却甚為好用。此種指令增加程式設計之便利，並能提高程式之性能，尤以當輸入／輸出作業時為然。

Z80 有四個區段傳輸指令：

LDI (Load and Increment)：取入，加一。

LDI (Load, Increment and Repeat)：取入，加一，並重複。

LDD (Load and Decrement)：取入，減一。

DDR (Load, Decrement and Repeat)：取入，減一，並重複。

所有區段傳輸指令皆使用三個暫存器對：

HL 指至資料之來源位置

DE 指至資料之目的位置

BC 為位元組計數器

程式設計者一旦將此些暫存器對佈置妥（置定其起始值），四個區段傳輸指令即可開始使用。LDI 指令將 HL 所指之記憶位置所含的位元組，移（抄）至 DE 所指之記憶位置。然後，HL 與 DE 兩暫存器對之內含均自動加一，指至次一位元組。同時，位元組計數器（BC）之值亦自動減一。當有整批資料必須移動，而且

表 2 - 4 區段傳輸指令

來源 暫存器 間接 (HL)	目的 暫存器 間接 (DE)
ED AO	LDI-((DE))-((HL)) HL及DE各加一，BC減一
ED B0	LDIR-((DE))-((HL)) HL及DE各加一，BC減一， 重複至BC = 0
ED A8	LDD-((DE))-((HL)) HL及DE各減一，BC減一
ED B8	LDDR-((DE))-((HL))，HL，DE， 及BC各減一，重複至BC = 0

HL 指至來源位置  
DE 指至目的位置  
BC 為位元組計數器

每移動一者後必須作其他型式之處理時，此一指令甚為好用。  
LDIR 指令則為 LDI 指令之延伸。資料移動，指示器值加一，計數器值減一之作業，一直重複，直至位元組計數器等於零為止。因此，僅此一指令即可將整批資料，自記憶器之某一區域搬至另一區域。圖 2 - 2 所示即為 LDIR 指令之執行情形。

留意，由於位元組計數器為一十六位元暫存器，因此，區段傳輸指令最高能搬運 64K (1K = 1024) 個位元組之資料。同時，資料可由任一記憶位置移至任一其他記憶位置。再者，由於三個暫存器對之內含無任何限制，因之，資料之來源區段與目的區段可彼此重疊。

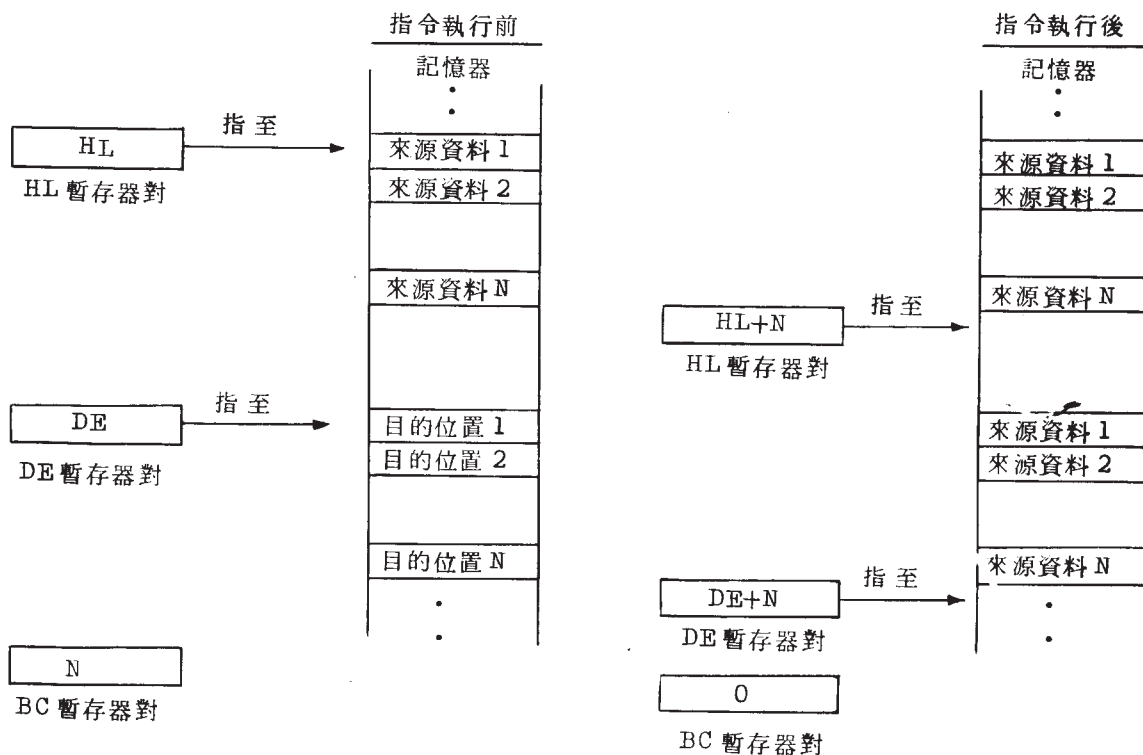


圖 2 - 2 LDIR 指令之效應

表 2 - 5 區段搜尋指令

LDD 與 LDDR 指令則分別類似於 LDI 與 LDIR 指令。唯一不同的是，每次移動一個位元組後，HL 與 DE 兩暫存器對之內含自動減一（而非加一），致使資料的搬運能自位址最高之位址開始，然後再位址漸小之位址。此一設施主要用於資料之來源區與目的區互相重疊時。於此種情況下，為免於來源區之資料於移走前被破壞，資料搬運工作必須自記憶位置最高之位址開始，逐漸往位址漸小的方向進行。

## 2 - 1 - 5 區段搜尋指令

區段搜尋指令在此順便介紹。Z80 有四個區段搜尋指令：  
CPI (Compare and increment) : 比較，並且加一。  
CPIR (Compare, increment and repeat) : 比較，加一，然後重複。  
CPD (Compare and decrement) : 比較，並且減一。  
CPDR (Compare, decrement and repeat) : 比較，減一，然後再重複。

CPI 指令將累加器之內含，與 HL 暫存器所指之記憶位置之內含相比，比較結果存於其中一旗號位元。然後，HL 暫存器對之內含加一，位元組計數器 (BC) 之值減一。

CPIR 則是 CPI 指令之延伸。該指令即等於 CPI 指令之一直反覆，直至找到欲取值或位元組計數值 (BC 之內含) 為零時為止。因此，僅此一指令即可搜尋整個記憶器。

CPD 與 CPDR 指令則分別類似於 CPI 與 CPIR 指令。唯一的差別是，每次搜尋後，HL 暫存器之內含減一（而非加一），致使搜尋能反向進行（由記憶位址最高之位址開始）。

有一點要指出的是，區段傳輸與搜尋指令於文字串 (string) 處理應用上甚為有用。

區段搜尋指令如表 2 - 5

搜尋位位置

暫存器間接 (HL)	
ED A1	'CPI' - 比較 HL 加一，BC 減一
ED B1	'CPIR' - 比較，HL 加一，BC 減一， 重複至 BC = 0 或找到為止
ED A9	'CPD' - 比較，HL 及 BC 各減一
ED B9	'CPDR' - 比較，HL 及 BC 各減一， 重複至 BC = 0 或找到為止

HL 指至欲與累加器內含相比之記憶位置。  
BC 為位元組計數器。

## 2 - 2 資料處理指令

Z80 之資料處理指令可分為三類來說：算術與邏輯指令，移位與旋轉指令，以及位元運作指令。

### 2 - 2 - 1 算術與邏輯指令

Z80 之算術與邏輯指令可區分為三組：八位元算術與邏輯指令，十六位元算術與邏輯指令，與一般用途之 AF 作業指令。

八位元算術與邏輯指令用以加、減、AND、OR、exclusive OR、或比較兩個八位元之運算元，其中一運算元恒來自累加器。另一運算元則可為立即數據，亦可來自某一 CPU 暫存器，或來自 HL 暫存器間接定址或索引定址所選取之記憶位置。兩運算

元經運算後，最後之結果則存於累加器，因此，運算後，累加器之原有內容已遭破壞，但另一運算元則保持不變。

十六位元之算術指令履行兩十六位元暫存器間之算術（加或減）運算。加一與減一指令亦能對某一暫存器對運算。

另外，有五個一般用途之算術指令能對累加器或旗號暫存器運算。

### 算術指令

Z80提供了兩種主要之算術運算指令：加與減。加法指令有一般加法（ADD）指令與進位加法（ADC）指令兩種。八位元之一般加法指令。

ADD A, s

將累加器A之內含，加上第二運算元（以s代表），結果存回累加器。進位加法指令

ADC A, s

將累加器A之內含，加上第二運算元以及現有進位旗號值，並將結果存回累加器。

減法指令則類似於加法指令。八位元一般減法指令。

SUB A, s

將累加器內含減去第二運算法及現有進位旗號值，結果存回累加器A。八位元之算術運算指令如表4-6所示。此些指令可使用暫存器、暫存器間接、索引、與立即等定址法。

例如，若累加器之原有內容為11000011（十六進數C3），則立即定址之加法指令

ADD A, 6

累加器A

11000011

(a) ADD A, 6 執行前

累加器A

11001001

(b) ADD A, 6 執行後

圖 2 - 3 加法運算表

執行完後，累加器內含將變為11001001（十六進數C9）。又例如，若累加器內含原為00100011（十六進數23），B暫存器之內含為00110101（十六進數35），且進位旗號為1，則暫存器定址之加法指令

ADC A, B

執行完後，累加器內含將變為01011001（十六進數59），B暫存器內含不變。

再舉一減法指令之例子。若累加器A之內含為11001010（十六進數CA），C暫存器之內含為00100010（十六進數22），則暫存器定址之

SUB A, C

指令執行完後，累加器內含將變為10101000（十六進數A8），而C暫存器之內含維持不變。

A 11001010

A

10101000

C 00100010

C

00100010

(a) SUB A, C 執行前

(b) SUB A, C 執行後

圖 2 - 4 減法運算

Z80有兩個指令，INC（加一）與DEC（減一），能將運算元之值加一或減一。八位元之INC與DEC指令能將任一八位元CPU暫存器，HL暫存器間接定址，或索引定址所選取之記憶位置的內含加一或減一。十六位元之INC與DEC指令則能對任一十六位元暫存器作業。

表2-7中含有三個特殊之算術運算指令：DAA, CPL, 與NEG。如圖2-5所示，CPL(ones-complement)指令將累

表 2 - 6 八位元算術與邏輯運算指令

來 源

	暫存器定址								暫存器 間接	索 引	立 即
	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	n
ADD	87	80	81	82	83	84	85	86	DD 86 d	FD 86 d	C6 n
進位加法 ,ADC	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n
減算 ,SUB	97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n
進位減算 ,SBC	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n
AND	A7	A0	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n
XOR	AF	A8	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n
OR	B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	F6 n
比較 ,CP	BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n
加一 ,INC	3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d	
減一 ,DEC	3D	05	0D	15	1D	25	2D	35	DD 35 d	FD 35 d	

表 2 - 7 一般用途之 AF 作業

十進調整 ,DAA	27
1補數 ,CPL	2F
2補數 ,NEG	ED 44
取進位旗號補數 ,CCF	3F
置定進位旗號 ,SCF	37

表 2 - 8 十六位元之算術運算指令

來 源	目 的 位置	HL	IX	IY	SP	HL	DE	BC	IX	IY
ADD		HL	09	19	29	39				
		IX	DD 09	DD 19	DD 29	DD 39				
		IY	FD 09	FD 19	FD 29	FD 39				
進位加法且置定 旗號 ,ADC		HL	ED 4A	ED 5A	ED 6A	ED 7A				
		HL	ED 42	ED 52	ED 62	ED 72				
進位減法且置定 旗號 ,SBC		HL	03	13	23	33				
		HL	0B	1B	2B	3B				
加一 ,INC		HL	03	13	23	33				
減一 ,DEC		HL	0B	1B	2B	3B				



加器A之內含取1補數——所有的0變1，1變0。NEG(NEGate)指令將累加器A之內含取1補數，然後再加1(亦即取2補數)。其實際效應即將累加器所含之數目變號——正變負，負變正。十進制調整DAA(Decimal Adjust Accumulator)指令，主要則用於BCD加減算術。該指令通常緊接於ADD, ADC, INC, SUB, DEC, 或NEG指令後，將運算所得之二進

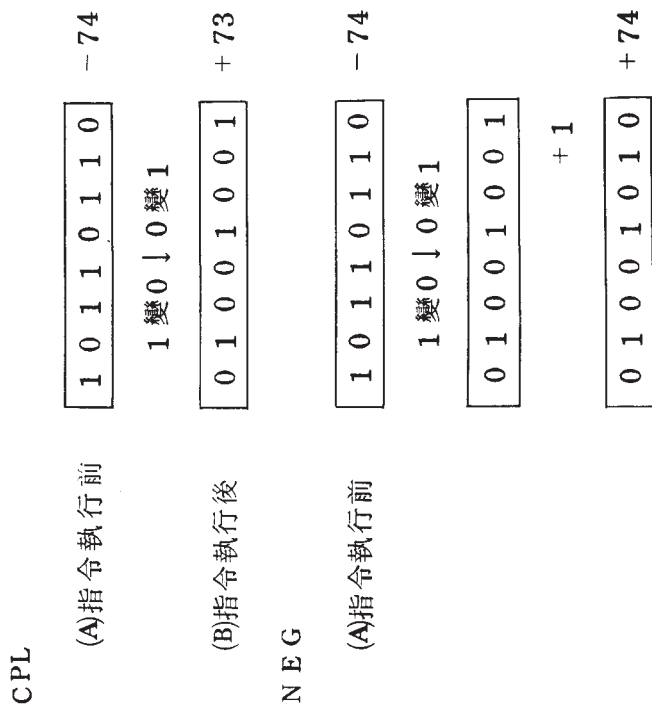


圖2-5 CPL與NEG指令之例子

結果，調整成符合十進制(BCD)算術之結果。

### 邏輯指令

如表2-6所示，Z80提供了三個邏輯運算指令，AND, OR, XOR(exclusive OR)，以及一比較指令C P。這些指

令皆對八位元資料運算，並且能使用暫存器，暫存器間接，索引，與立即等定址法。茲分別將其介紹如下：

### AND

每一邏輯運算皆以一真值表(truth table)表示。真值表顯示了在每一種可能之輸入狀況下，邏輯運算結果之值。AND運算之真值表為：

0AND 0 = 0	AND	0	1
0AND 1 = 0	或	0	0
1AND 0 = 0		1	0
1AND 1 = 1			1

AND運算之特性為：當兩輸入均為1時，輸出1，否則，輸出為0。換言之，若其中有一位元為0，則兩位元之AND運算結果必為0。此一特性常用以將資料之某一(或某些)位元變為0。此一技巧即稱為罩蓋(masking)或遮蓋。

AND指令之最大用途即以清除，或“遮蓋掉”(mask out)某項資料之某些位元。譬如，若我們欲將WORD位置之內含的低次四位元清除為零，則下列三個指令即為我們所需。

```
LD  A, (WORD) ; 資料字組取入A。
AND 11110000B ; "11110000"為面罩。
LD  (WORD), A ; 結果存回原位。
```

倘若位址WORD之記憶位址原來含10101010，則上述指令執行後，該位置之內含將變為10100000，最低次四位被清除為零。注意，“11110000B”即等於“二進數11110000”，B代表二進資料。

### OR

邏輯OR運算之真值表為

0 OR 0 = 0  
 0 OR 1 = 1  
 1 OR 0 = 1  
 1 OR 1 = 1

OR		0	1
0	0	0	1
1	1	1	1

或

顯然，邏輯 OR 運算之特性為：若兩者中有任一者為 1，則運算結果必為 1。否則，若兩輸入皆為 0，則運算結果為 0。

OR 指令最常用以將某一或某些位元置定為 1。例如，

```
LD A, (WORD)
OR A, 00001111 B
LD (WORD), A
```

三個指令即將 WORD 記憶位置之內含的低次四位元置定為 1。  
 XOR

XOR 代表 "exclusive OR"。此一邏輯運算之真值為

0 XOR 0 = 0  
 0 XOR 1 = 1  
 1 XOR 0 = 1  
 1 XOR 1 = 0

XOR		0	1
0	0	0	1
1	1	1	0

或

顯然，XOR 運算之特性為：若兩輸入有一者，而且僅有一者為 1，則結果為 1。否則，若兩輸入同為 0 或同為 1，則運算結果為 0。

XOR 指令主要用於比較兩項資料是否完全相同。若兩項資料中有一元不同，則兩則 XOR 之結果必不為 0。換言之，若兩字組 XOR 之結果為零，則兩字組必然完全相等。除此之外，於 Z80，XOR 指令亦可用以求某一資料字組之補數。利用 XOR 指令求補數之方法，是將其每一位元與 "1" 作 XOR 運算。例如，

```
LD A, (WORD)
XOR 11111111 B
```

若 WORD 位置原先含 10101010，則上述指令執行後，累加器 A 之內含將變為 01010101。是以，XOR 指令可用以製作一 "位元正反器" (bit toggle)，以作計時或為數 2 之計數器。下面之指令系列恰巧使由 LOOP 開始之迴路被執行兩次。

```
LD A, 0 ; 計數為零。
LOOP :
      (處理)
      :
      XOR 1 ; 2 數計數器。
      JP NZ, LOOP
      DONE :
```

## 2-2-2 移位與旋轉指令

Z80 的主要功能之一，是能將累加器、任一一般用途暫存器、或任一記憶位置 (由暫存器間接定址或索引定址選取) 之內含移位或旋轉。這些指令之運算碼如表 2-9 所示。表右邊之圖形解釋了各指令之功能。

Z80 有七個移位與旋轉指令，以及兩個特殊之 BCD 數字旋轉指令。SLA (Shift Left Arithmetic, 算術左移) 與 SRL (Shift Right Logical, 邏輯右移) 指令，分別將運算元向左與向右移動一個位元位置。於此兩種運算，被移出字組之位元恒進入進位位元，而尾端空出之位元位置則補 0。

SRA (Shift Right Arithmetic, 算術右移) 則為一特殊之右移指令。於 2 補數形式之負數運算，最左 (高次) 位元為符號位元。負數時，該位元值為 1；正數時，該位元值為 0。當以右移方式將某負數除 2 時，符號位元必須保持 1，數目方能繼

續保持於負數。SRL 指令顯然無法達成此一任務，因此必須採用 SRA 指令。於算術右移運算，進入最左邊位元者恒為符號位元。換言之，於連續右移之過程中，符號位元一直向右複製。例如，若累加器 A 原先之內含為

10000100 - 124

則

SRA A

執行後，累加器內含變為

11000010 - 62

注意，符號位元已向右複製。若再一次算術右移，則累加器內含將變為

11100001 - 31

圖 2 - 6 所，即為上述介紹之三個移位指令運算的情形。

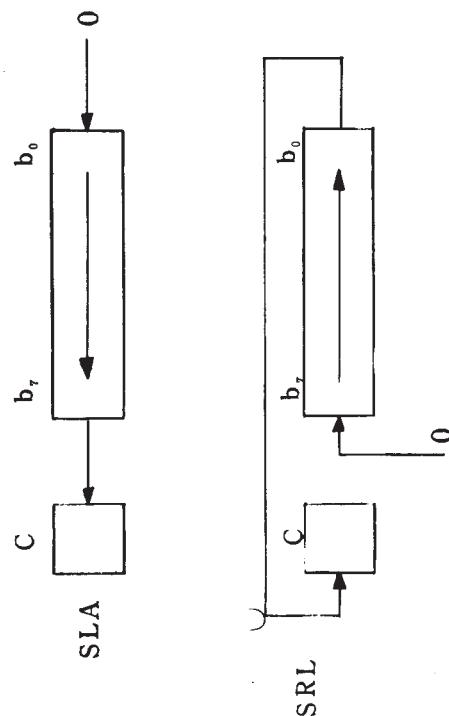


圖 2 - 6 (1) 移位指令(一)

來源 移位目的地 或旋轉	A	B	C	D	E	H	L	(HL)	(1X+d)	(1Y+d)
RLC	CB 07	CB 00	CB 01	CB 02	CB 03	CB 04	CB 05	CB 06	DD CB d 06	FD CB d 06
RRC	CB 0F	CB 08	CB 09	CB 0A	CB 0B	CB 0C	CB 0D	CB 0E	DD CB d 0E	FD CB d 0E
RL	CB 17	CB 10	CB 11	CB 12	CB 13	CB 14	CB 15	CB 16	DD CB d 16	FD CB d 16
RR	CB 1F	CB 18	CB 19	CB 1A	CB 1B	CB 1C	CB 1D	CB 1E	DD CB d 1E	FD CB d 1E
SIA	CB 27	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	CB 26	DD CB d 26	FD CB d 26
SRA	CB 2F	CB 28	CB 29	CB 2A	CB 2B	CB 2C	CB 2D	CB 2E	DD CB d 2E	FD CB d 2E
SRL	CB 3F	CB 38	CB 39	CB 3A	CB 3B	CB 3C	CB 3D	CB 3E	DD CB d 3E	FD CB d 3E
RLD								ED 6F		
RRD								ED 67		

	A
RLCA	07
RRCA	0F
RLA	17
RRA	1F

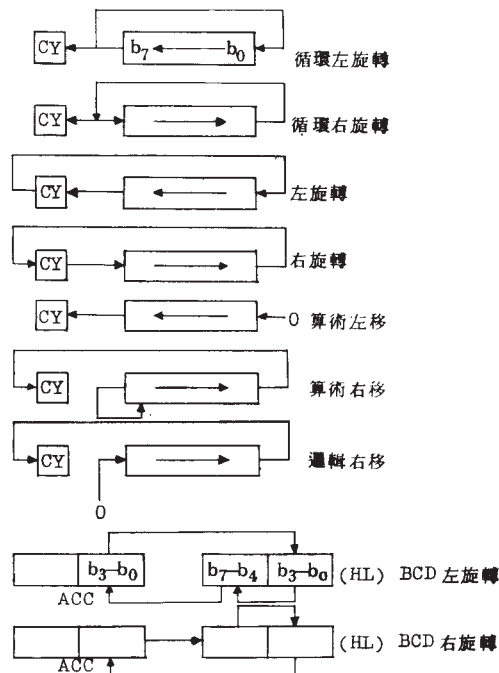
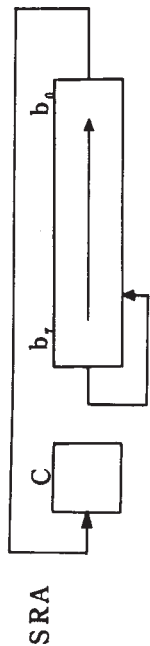


表 2 - 9 移位與旋轉指令

圖 2 - 6 (2) 移位指令 (二)



Z80之旋轉運算有兩種：八位元旋轉與九位元旋轉。圖2-7所示即為九位元旋轉的情形，此一運算，暫存器之八位元加上進位之第九位元，全串聯成一位元串，同時向右或向左旋轉。換言之，移出資料字組之位元進入進位位元，而先前進位位元之值則進入資料字組之另一端。九位元旋轉之兩個指令為：RL (Rotate Left, 左旋轉)與RR (Rotate Right, 右旋轉) 例如，

RL ( HL )

即將HL 所指之記憶位置的內含以及進位，向左旋轉一位元位置，結果存回原記憶位置。

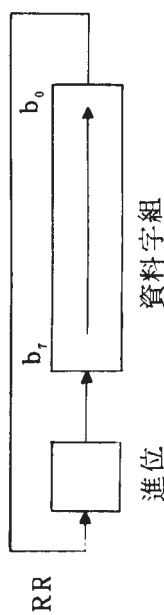
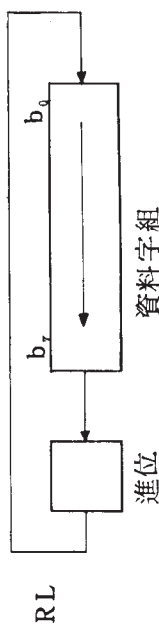


圖 2 - 7 九位元之旋轉

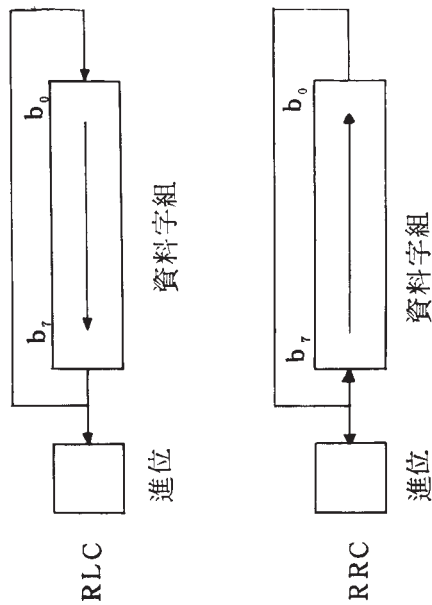


圖 2 - 8 八位元之旋轉

八位元之旋轉運算如圖 2 - 8 所示。左旋轉時，被移出第 7 位元位置之位元進入第 0 位元，同時亦進入進位位元。右旋轉時，移出第 0 位元位置之位元進入第 7 位元，同時亦進入進位位元。兩八位元之旋轉指令分別記為：RLC (Rotate Left Circular, 循環左旋轉) 以及 RRC (Rotate Right Circular, 循環右旋轉)。

### BCD數字旋轉指令

Z80有兩個特殊之BCD數字旋轉指令 (RRD與RLD)，以利BCD算術運算。如圖 2 - 9 所示，這兩指令導至HL暫存器指令及之記憶位置所含的兩位BCD數字，與累加器之四次四位元間的四位元旋轉。

記憶體

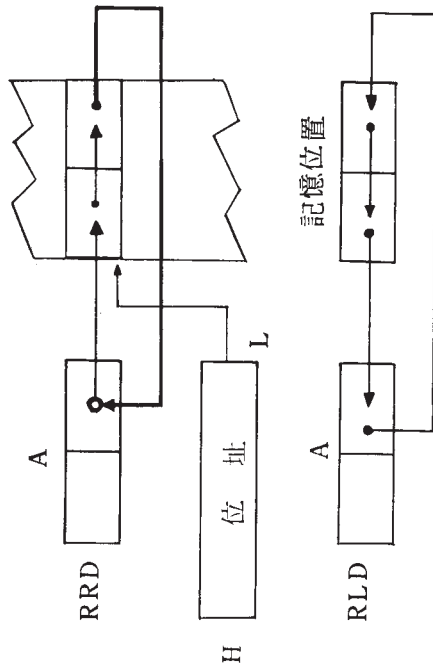


圖 2-9 BCD 數字旋轉指令

### 4-2-3 位元運算指令

許多程式經常需置定，清除，或測試某一暫存器或某一記憶位置之某些位元。前面已提過，邏輯指令可用以將某一特定暫存器之某些位元置定為 1 或清除為 0。不過，若僅一個指令即能將任一暫存器或記憶位置之任意位元置定或清除，那將更方便了。

由於這需要相當多之運算碼，致一般微處理器均未具備。Z80 微處理器則例外，其獨具備廣泛之位元運作指令。此等指令之運算碼如表 2-10 所示。（此一表所含之測試指令將於下節討論）

於位元運作指令，暫存器定址可用以選取累加器或任一 CPU 一般用途暫存器。而記憶位置則可以 HL 暫存器間接或索引定址選取。

例如，

SET 7, B

指令即將 B 暫存器之第 7（最高次）位元置定為 1。

附帶一提的是，表 2-7 中有兩個指令能將進位旗號之值置

表 2-10 位元運作指令組

BIT	暫存器定址							暫存器間接	備 註
	A	B	C	D	E	H	L		
0	CB 47	CB 40	CB 41	CB 42	CB 43	CB 44	CB 45	CB 46	(1X+d)(Y+d) DD CB d 46
1	CB 4F	CB 48	CB 49	CB 4A	CB 4B	CB 4C	CB 4D	CB 4E	DD CB d 4E
2	CB 57	CB 50	CB 51	CB 52	CB 53	CB 54	CB 55	CB 56	DD CB d 56
3	CB 5F	CB 58	CB 59	CB 5A	CB 5B	CB 5C	CB 5D	CB 5E	DD CB d 5E
4	CB 67	CB 60	CB 61	CB 62	CB 63	CB 64	CB 65	CB 66	DD CB d 66
5	CB 6F	CB 68	CB 69	CB 6A	CB 6B	CB 6C	CB 6D	CB 6E	DD CB d 6E
6	CB 77	CB 70	CB 71	CB 72	CB 73	CB 74	CB 75	CB 76	DD CB d 76
7	CB 7F	CB 78	CB 79	CB 7A	CB 7B	CB 7C	CB 7D	CB 7E	DD CB d 7E
0	CB 87	CB 80	CB 81	CB 82	CB 83	CB 84	CB 85	CB 86	DD CB d 86
1	CB 8F	CB 88	CB 89	CB 8A	CB 8B	CB 8C	CB 8D	CB 8E	DD CB d 8E
2	CB 97	CB 90	CB 91	CB 92	CB 93	CB 94	CB 95	CB 96	DD CB d 96
3	CB 9F	CB 98	CB 99	CB 9A	CB 9B	CB 9C	CB 9D	CB 9E	DD CB d 9E
4	CB A7	CB A0	CB A1	CB A2	CB A3	CB A4	CB A5	CB A6	DD CB d A6
5	CB AF	CB A8	CB A9	CB AA	CB AB	CB AC	CB AD	CB AE	DD CB d AE
6	CB B7	CB B0	CB B1	CB B2	CB B3	CB B4	CB B5	CB B6	DD CB d B6
7	CB BF	CB B8	CB B9	CB BA	CB BB	CB BC	CB BD	CB BE	DD CB d BE
0	CB C7	CB C0	CB C1	CB C2	CB C3	CB C4	CB C5	CB C6	DD CB d C6
1	CB CF	CB C8	CB C9	CB CA	CB CB	CB CC	CB CD	CB CE	DD CB d CE
2	CB D7	CB D0	CB D1	CB D2	CB D3	CB D4	CB D5	CB D6	DD CB d D6
3	CB DF	CB D8	CB D9	CB DA	CB DB	CB DC	CB DD	CB DE	DD CB d DE
4	CB E7	CB E0	CB E1	CB E2	CB E3	CB E4	CB E5	CB E6	DD CB d E6
5	CB EF	CB E8	CB E9	CB EA	CB EB	CB EC	CB ED	CB EE	DD CB d EE
6	CB F7	CB F0	CB F1	CB F2	CB F3	CB F4	CB F5	CB F6	DD CB d F6
7	CB FF	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	CB FE	DD CB d FE



定為 1，或將其互補。這兩指令分別是

SCF (Set Carry Flag) : 進位旗號置定為 1。

CCF (Complement Carry Flag) : 進位旗號之值互補 (0 變 1, 1 變 0)。

## 2-3 測試與控制轉移指令

### 2-3-1 旗號

由於測試運算與旗號暫存器之應用關係密切，因此，在這兒，我們再詳細說明 Z80 之旗號暫存器內，每一旗號之功能與其受影響之情形。如圖 2-10 所示，Z80 之旗號暫存器包含六種旗號：S, Z, H, P/V, N, 與 C。除了 H 與 N 的旗號是用於 BCD 算術而且不能測試外，其餘之四個旗號均可以條件跳越、條件叫用、或條件回返指令加以測試。

### 進位旗號 (C)

幾乎於所有微處理器，包括 Z80 在內，進位旗號均具有雙重角色。其不僅用以顯示或加減運算是否產生進位 (或借位)，同時亦用以作為移位與旋轉運算之第九位元。此種雙重角色使得諸如乘法等之某些運算，變得相當容易。

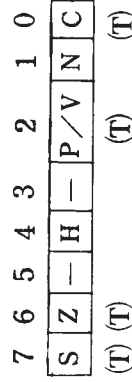


圖 2-10 旗號暫存器

關於進位旗號，有一點必須牢記的是，所有算術均影響進位位元；視運算之不同結果而定，該位元被置定為 1 或清除為 0。同樣地，所有移位與旋轉指令亦皆影響進位位元；視被移出運算元之位元值而定，進位旗號值被置定為 1 或清除為 0。於 Z80

，所有的邏輯運算 (AND, OR, XOR) 恆使進位位元清除為 0。是以，此些指令可專門用以將進位旗號清除為零。

影響進位旗號值之指令可由表 2-11 查出。

### 減法旗號 (N)

此一旗號通常於 BCD 運算時由 Z80 自己使用，程式設計者無法加以利用。由於計算機永遠只能作二進制算術。因此，於 BCD 算術時，為使計算機運算之結果能合乎十進制算術，加或減指令後必須立即緊接一 DAA 指令，將運算結果調整成有效之 BCD 結果。然而，此一“調整”作業對加與減運算有所不同。N 旗號即應此目的而生，用以記錄 Z80 微處理器方才所執行的是加法或減法運算。於加法運算後，N 旗號被清除為 0，於減法運算後，N 旗號被置定為 1。(注意，有些其他微處理器以 N 代表負值旗號，切勿弄混了！)

許多指令執行之結果將 N 旗號值清除為 0，亦有許多指令執行之結果使 N 旗號置定為 1，此些指令可由表 2-11 查出。

### 極性 / 溢位旗號 (P/V)

基本上，極性 / 溢位旗號具有兩種不同之功能。於邏輯，移位與旋轉，DAA，及 IN r, (C) 等運算時，該旗號為極性 (P) 旗號。於所有算術與比較指令時，該旗號則為溢位 (V) 旗號。

所謂極性 (parity)，即為某項資料所含之“1”位元的總個數。若總數為奇數，則稱奇極性；若總數為偶數，則稱偶極性。於作為極性旗號時，若指令運算所得結果為奇極性 (結果所含“1”之總個數為奇數)，則極性旗號值清除為 0。若運算所得結果為偶極性，則極性旗號值置定為 1。極性主要用於文數字 (alphanumeric) 資料之傳輸。於代表文數字之七位元電碼中加上一額外的極性位元，可偵測出資料在傳遞過程中是否發生錯誤。

於算術運算時，P/V 旗號主要作為溢位（V）旗號。該旗號用以顯示算術運算所得之結果，是否越過八位元 2 補數所能表示之最大範圍（-128 至 +127）。若是，則溢位旗號置定為 1；否則，其值為 0。

除以上所述外，Z80 之 P/V 旗號尚有另外兩種功能。

於整批（區段）傳輸與搜尋指令時，該旗號用以顯示計數器暫存器 B 值之是否已為 0。於遞減指令。若位元組計數器之暫存器對內含已遞減至零，則旗號值清除為 0。於遞增指令，若指令開始前  $BC - 1 = 0$ ，則旗號值置定為 1。

此外，當執行 LD A, I 與 LD A, R 兩指令時，P/V 旗號則反映了插斷致能正反器（IFF2）之值。此一特色可用以保留或測試該位元值。

#### 半進位旗號(H)

於算術運算時，半進位旗號顯示第 3 位元（ $b_3$ ）是否產生進位至第 4 位元（ $b_4$ ）。換言之，其顯示低次四位元是否有進位至高次四位元。顯然，這是用於 BCD 算術。特別，DAA 指令可用之作爲調整運算結果之依據。

若加法運算時第 3 位元產生進位至第 4 位元，則半進位旗號值置定為 1。否則，（若無進位）其值清除為 0。反之，於減法運算，若第 3 位元向第 4 位元借位 1，則 H 之值置定為 1。否則，若無借位，則 H 旗號值清除為 0。

半進位旗號受指令影響之情形如表 2-11 所示，注意，H 位元並不受十六位元加與減指令之影響。

#### 零值旗號(Z)

零值旗號用以顯示剛剛被計算或傳輸之位元組是否為零。除此之外，其亦用以顯示比較運算是否已找到匹配（相同）值，以及若干其他之功用。

於資料計算與傳輸時，若最後導致之結果為零，則零值旗號

置定為 1；否則，零值旗號為 0。於比較運算，若比較找到匹配值（即比較成功），則零值旗號亦置定為 1；否則，其值為零。

除此之外，於 Z80，零值旗號尚有三種其他功能：第一，其與 BIT 指定併用，以顯示被測試之位元值是否為零。若是，則其值被置定為 1；否則，其值為 0。第二，於整批輸入／輸出指令（INI, IND, OUTI, OUTD），若  $D - 1 = 0$ ，則 Z 旗號置定為 1；否則，其值清除為 0；若位元組計數器將遞減成 0（INR, INDR, OTIR, OTDR），其值置定為 1。第三，於特殊指令 IN r, (C)，零值旗號被置定為 1，以顯示輸入位元組之值為零。

#### 正負數旗號(S)

該旗號反映指令運算結果或被傳遞位元組之最高次位元值（第 7 位元）。於 2 補數目表示法，最高次位元代表正負號：“0”表正數，“1”表負數。因此，第 7 位元又稱為符號（正負號）位元。

在大多數微處理器，符號位元於輸入／輸出設備溝通時，扮演了相當重要之角色。由於大多數微處理器均未具備能測試某一暫存器或記憶位置之任意位元的指令。以致，符號位元變成一最容易測試之位元。在讀取某一輸入／輸出設備之狀態資訊，以檢查其狀態時，我們可將狀態位元置於狀態暫存器之最高次位元。如此，程式即能輕易測試狀態位元之值。此即為何大多數連至微處理器之輸入／輸出設備的最重要狀態位元，均設於狀態暫存器之第 7 位元的原故。

#### 旗號摘要

微處理器之旗號位元用以自動偵測算術／邏輯單元之某些特殊狀態。這些旗號位元能方便地以各種條件式的控制轉移指令加以測試，以便程式能依據測試結果，決定進一步應採取何種反應。由於程式內所發生之決策大多根據這些旗號。因此，程式設計



X 該旗號無關。

V 運算之溢位結果影響該旗號。

P 運算之極性結果影響該旗號。

r 表 A B C D E H L 等任一暫存器。

S 指令可用之定址法選取之八位元位置。

SS 指令允許之定址法所選取之十六位元位置。

ii IX 或 IY 兩索引暫存器中任一者。

I 插斷向量暫存器。

R 記憶復新計數器。

n 0 至 255 範圍內之八位元值。

nn 0 至 65536 範圍內之十六位元值。

m 指令允許之定址法所選取之八位元位置。

## 2-3-2 控制轉移指令

控制轉移指令即是應用程式控制轉移指令改變程式之正常（循序）執行次序，而令控制轉移至儲存於其他記憶區之程式指令。換言之，此種指令執行之結果，將一新數值存入程式計數器，使次一被執行之指令來自其他記憶位置，而非立即緊接目前指令後之指令。Z80 之程式控制指令可分成跳越（jump），副程式（call），與回返（return）三種。跳越指令將程式控制轉移至另一記憶位置，但並不將程式計數器之內含存起。叫用指令同樣將程式控制轉移至另一記憶位置，但卻將程式計數器之內含推入堆疊器存起，以便副程式結束後能回返。回返指令則自堆疊器拉取先前存起之程式計數器內含，將程式控制傳回至緊接叫用指令後之指令。叫用與回返指令皆用於副程式之處理。

### 跳越指令

Z80 之跳越指令有無條件跳越與條件跳越兩種。無條件跳越指令可使用擴展立即，相對，與暫存器間接等三種定址，其指令

格式分別為

擴展立即： JP nn nn 為一十六位元位址。

相對： JR e 為八位元之位移（2補數形式）

暫存器間接： JP (HL)  
(IX)  
(IY)

例如，若 Z80 微處理器執行 JP nn 指令，則該指令執行完後，緊接被執行的指令將來自位址 nn 之記憶位置。

條件跳越指令是當所指明之條件滿足時，跳越才發生；而條件不滿足時，微處理器繼續執行次一緊接指令之跳越指令。此種指令除必須指明跳越之目的位址外，尚須指明一欲測試之條件（或狀態），致其格式為

JP C, nn

其中，C 即為所指明之條件，而 nn 為跳越之 11 的位址。於 Z80，條件跳越指令可使用相對與擴展立即兩種定址法。其指令格式分別為

擴展立即： JP C, nn

相對： JR C, e

其中，擴展立即定址之條件跳越指令可測試 Z, C, P/V，與 S 等四種旗號，而相對定址之條件跳越指令則僅能測試 Z 與 C 兩種旗號。由於每一旗號有兩種可能狀態，故擴展立即定址之

JP C, nn

指令中，陳述條件 C 可為下列八種情況之任一種：

Z : 零值 ( Z = 1 )

NZ : 非零值 ( Z = 0 )

C : 有進位 ( C = 1 )

NC : 無進位 ( C = 0 )

PO : 奇極性

## 表 2-12 JUMP, CALL, 與 RETURN 指令

條件		有進位	無進位	零值	非零	偶極性	奇極性	負數	正數	BCD 修正
JUMP, JP,	擴展立即	nn	C3 nnn	DA nnn	D2 nnn	CA nnn	C2 nnn	EA nnn	E2 nnn	
JUMP, JR,	相對	PC+e	18 e-2	30 e-2	20 e-2					
JUMP, JP,	暫存器間接	(HL)	E9							
JUMP, JP,		(IX)	DD E9							
JUMP, JP,		(IY)	FD E9							
'CALL'	擴展立即	nn	CD nnn	DC nnn	D4 nnn	CC nnn	C4 nnn	EC nnn	E4 nnn	F4 nnn
B減一，非零則再跳越, DJNZ,	相對	PC+e								10 e-2
回返, RET,	暫存器間接	(SP) (SP+1)	C9 ED4D	D8 D0	C8 C0	E8 E0	F8 F0			
插斷回返, RETI,	暫存器間接	(SP) (SP+1)	ED 4D							
不可罩蓋插斷回返, RETN,	暫存器間接	(SP) (SP+1)	ED 45							

注意：某些旗號有多種用途。

PE:偶極性

P:正值( $S=0$ )

M: 負値 ( $S = 1$ )

而相對定址。

JR  
C,e

指令中之C，則僅能爲Z，NZ，C，與NC四種情況之一種。

例如，

JP  
NZ, LOOP

指令即為一擴展立即定址之條件跳越指令。當 Z80 微處理器執行此指令時，若零值旗號之值為 0，則程式控制將跳至儲存於位址 LOOP 之記憶位置上的指令。否則，若零值旗號為 1（表前一運算之結果為零），則微處理器繼續執行下一緊接指令。

注意無條件跳越指令與條件跳越指令之格式差異：條件跳越指令上多一“陳述條件”C。

於運算碼寫碼時，上述所列之條件的寫碼情形爲：

條件碼	條件
000	NZ
001	Z
010	NC
011	C
100	PO
101	PE
110	P
111	M

圖 2-11 測試條件(狀態)之寫碼

Z80 有一特殊之條件跳越指令：DJNZ。該指令將 B 暫存器



之內含值減一，若結果不為零，則跳越發生；否則，若 B 暫存器之值為零，則跳越不發生。此一指令主要用於程式迴路 (program loop) 之製作，其僅能使用相對定址。

### ○ 副程式叫用指令

Z80 之副程式叫用指令亦有條件式與無條件式之分。該指令只能使用擴展立即定址，其格式為

無條件叫用：CALL nn  
條件叫用：CALL C, nn

其中，陳述條件 C 如擴展立即定址之條件跳越指令。微處理器執行 CALL nn 指令時，首先將現有程式計數器內含推入堆疊器存起，然後將 nn 存入程式計數器，使次一被執行之指令來自定址 nn 之記憶位置。

### 回返指令

同樣地，Z80 之副程式回返指令 (RET) 亦有條件與無條件式之分。RET 指令所能測試之條件回於條件副程式叫用指令。微處理器執行回返指令時，即將堆疊頂端連續兩位元組之資料，拉回至程式計數器。使控制回至緊接叫用指令後之指令。副程式叫用與回返指令之功能如圖 2-12 所示。

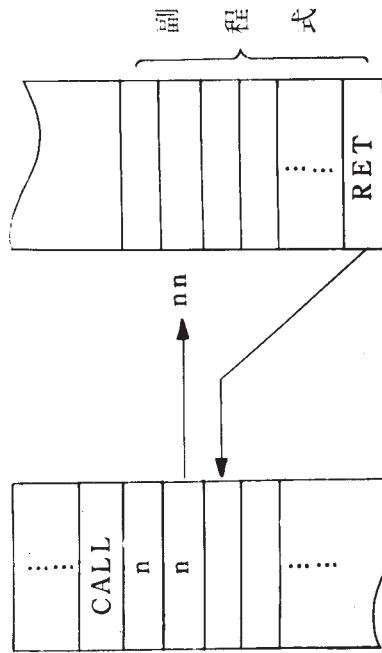


圖 2-12 副程式叫用迴返指令

Z80 另有兩個特殊之回返指令。RET I 指令用於可罩蓋插斷之回返，RETN 指令則用於不可罩蓋插斷 (nonmaskable interrupt) 之回返。此兩指令在後面討論插斷時，會再作更進一步之解釋。

### 重始指令

叫 用 位 址	0000 <sub>H</sub>	運算碼	RST 0 <sub>H</sub>
	0008 <sub>H</sub>	0F	RST 8 <sub>H</sub>
	0010 <sub>H</sub>	D7	RST 16 <sub>H</sub>
	0018 <sub>H</sub>	DF	RST 24 <sub>H</sub>
	0020 <sub>H</sub>	E7	RST 32 <sub>H</sub>
	0028 <sub>H</sub>	EF	RST 40 <sub>H</sub>
	0030 <sub>H</sub>	F7	RST 48 <sub>H</sub>
	0038 <sub>H</sub>	FF	RST 56 <sub>H</sub>

重始指令表

### 重始指令

Z80 具有另一種特殊程式控制轉移指令——重始指令 RST。RST 指令有兩個主要用途：第一，插斷設備用其將八位元之插斷向量置於資料巴士上。第二，其令控制能跳至八個零頁位置中之任一一個。用以叫用零頁位置之重始指令的格式為 RST P，其中 P 為跳越之目的零頁位址。RST 指令之運算碼寫碼情形為

RST	P						
	<table><tr><td>1</td><td>1</td><td>T</td><td>1</td><td>1</td><td>1</td></tr></table>	1	1	T	1	1	1
1	1	T	1	1	1		
	運算碼						

表 2-13 所示即為 RST 指令之運算碼情形

P	T 欄
00H	000
08H	001
10H	010
18H	011
20H	100
28H	101
30H	110
38H	111

表 2-13 RST 指令之運算碼寫碼

### 2-4 輸入／輸出指令

簡言之，輸入／輸出設備可以兩種方式選取：將之當成記憶位置，以前面介紹過之任一指令存取，或以特定之輸入／輸出指令存取。尋常之記憶選取指令必須使用三個位元組：運算碼一個

位元組，位址兩個位元組。結果，由於指令執行需作三次記憶器存取，因此，速度相當緩慢。特定輸入／輸出指令之主要目的即在於提供較短，因此亦較快之指令。但是，輸入／輸出指令亦有兩個缺點。

首先，其浪費了珍貴有限之運算碼寫碼空間（通常微處理器僅以八位元作運算碼之寫碼）。其次，其必須產生一個或一個以上之特別輸入／輸出信號，致浪費了微處理器之有限接腳。由於具有此些缺點，致一般處理器皆無特別之輸入／輸出指令。不過，由於最原始之 8080 微處理器（世界上第一個能幹之八位元一般用途微處理器）上有，故與之匹配之 Z80 上亦有。

固然輸入／輸出指令執行上較迅速，因其僅長兩個位元組，但零頁位址亦可達成同樣效果。致某些微處理器採取了此一方式。

兩個基本之輸入／輸出指令為 IN 與 OUT。IN 指令將資料自某一特定輸入／輸出位置讀進某一 CPU 暫存器，OUT 指令則將某一 CPU 暫存器之資料輸出至某一輸入／輸出設備。Z80 之輸入／輸出有兩種定址方式。立即定址之輸入／輸出指令僅能將資料來回傳遞於累加器 A 與輸入／輸出設備之間，暫存器間接定址之輸入／輸出指令則能將資料來回傳遞於任一一般用途暫存器與輸入／輸出設備間。其指令格式分別為

立即：	IN	A, (n)
	OUT	(n), A
暫存器間接：	IN	r, (C)
	OUT	(C), r

其中，n 為一八位元立即位址，C 為 C 暫存器，r 表任一 Z80 之八位元一般用途暫存器。

於執行立即定址之輸入／輸出指令時，輸入／輸出設備之位址 n，出現於位址巴士之低次八位元（A<sub>0</sub> ~ A<sub>7</sub>），而累加器之

表 2 - 14 輸入指令

輸入目的位置	資料來源之輸入口位置		暫存器間接
		立即	
		(n)	
輸入 IN	暫存器定址	A	ED 78
		B	ED 40
		C	ED 48
		D	ED 50
		E	ED 58
		H	ED 60
		L	ED 68
	暫存器間接	(HL)	

整批輸入指令

目的輸出  
口的位置

表 2 - 15 輸出指令

源			暫存器							暫存器間接	
			A	B	C	D	E	H	L		
來	源	立即	(n)	D3							(HL)
			(c)	ED 79	ED 41	ED 49	ED 51	ED 59	ED 61	ED 69	
			"								ED A3
			"								ED B3
			"								ED AB
目的輸出 口的位置	源	立即	(c)								ED BB
			"								
			"								
			"								
			"								

內含則被置於位址巴士之高次八位元 ( A<sub>8</sub> ~ A<sub>15</sub> )。於暫存器間接定址之輸入／輸出指令時，C暫存器之內含出現於位址巴士之低次八位元，而B暫存器之內含則被置於位址巴士之高次八位元。如表 2-14，2-15 所示，上述之輸入／輸出指令均長兩位元組，並能選取 256 個可能輸入／輸出位置。

除了上述能傳輸八位元資料之輸入／輸出指令外，Z80 亦提

供了整批傳輸之輸入／輸出指令。四個整批傳輸之輸入指令為：INI, INIR, IND, 與 INDR。同樣地，四個整批傳輸之輸出指令為 OUTI, OUTIR, OUTD, 與 OUTDR。於此些自動之整批傳輸，HL 暫存器均作為目的區指示器，暫存器 C 作為輸入／輸出設備之選擇（由 256 個設備中選取一個），而 B 暫存器則作為計數器，其內含值可加一或減一。

INI 為自動之單位元組傳輸。暫存器 C 之內含選取某一輸入設備。資料位元組然後由該設備讀取，並傳至 HL 所指之記憶位置。之後，HL 之內含值加 1，計數器 B 之內含值減 1。

INIR 則為 INI 之重複執行。重複程序一直進行至計數器 B 之值，遞減至零時為止。因此，該指令最高能自動傳輸 256 個位元組。注意，欲作正巧 256 個位元組之傳輸，B 暫存器之值必須先置定為 0。

輸入與輸出指令之運算碼分別如表 2-14 與 2-15 所示。

## 2-5 各種 CPU 控制指令

控制指令乃改變 CPU 之作業型態或經緯 (manipulate) CPU 之內部狀態資訊的指令。Z80 具有七個此種指令。

NOP 指令為無運算指令，於此一指令之機器週期（僅一個期間，Z80 CPU 什麼事都沒做。典型上，該指令有兩種用途：用以產生延遲（4 個 T 週期 = 2us，就 2MHz 之時序言），或填補偵錯結果所造成之程式空缺。傳統上，NOP 指令之運算碼為所有位元皆 0。此乃因為程式執行期間，記憶器通常清除為 0。執行 NOP 指令可確保程式不受損壞，且程式執行不致停止。

HALT 指令用於插斷或重置。其延遲了 CPU 之作業。當 Z80 微處理器因執行 HALT 指令而處於停止狀態時，插斷或重置信號可用以令其重新恢復作業。於停止 (HALT) 狀態期間，Z80 微處理器一直執行 NOP 指令。於偵錯期間，程式之最後指

令經常為 HALT 指令。

表 2-16 各種 CPU 控制指令

·NOP·	00
·HALT·	76
INT 禁能·DI·	F3
INT 致能·EI·	FB
設定插斷型態 0 ·IM0·	ED 46
設定插斷型態 1 ·IM1·	ED 56
設定插斷型態 2 ·IM2·	ED 5E

8080 A 型

叫用 0038H 位置

以 I 暫存器及插斷設備來之八位元作指示器之間接叫用。

EI 與 DI 指令則分別用以將 Z80 CPU 內之插斷旗號置定為 1 與清除為 0，令外部插斷分別致能與禁能。插斷在後面會有更詳盡之討論。

最後，Z80 提供了三種插斷型態（8080 僅有一種）。插斷型態 0 為 8080 型態，插斷型態 1 叫用位址 0038 H 記憶位置之副程式，插斷型態 2 則為間接副程式叫用，此一型態以特殊暫存器 I 之內含，加上插斷設備所提供之八位元，作為指示器，指至記憶器中之插斷處理常式。

## 2-6 常用指令表

我們已將 Z80 微處理器之指令集，分類作了一概要介紹。對一初學者而言，一開始時並不須將每一指令徹底熟記，而只須記得每一指令種類之幾個主要指令就夠了。畢竟學習程式設計最迅速有效的效率法，乃為——做中學——多寫程式需用到什麼指令而不明瞭時多查表格資料；進步就會很神速。最後自然，你就會記得且熟悉整個指令之每一指令——這是當您自行設計程式，且

欲設計出好的程式所必備的！

本篇列出常用指令註解，十六進位轉換表，相對跳越位移表  
十進數與BCD轉換、Z80指令碼。

經常使用的Z-80指令表

指令	意義
ADC A	與進位加入累積器中
ADD	加法
AND	邏輯“AND”
CALL addr	呼叫副程式
CALL cond,addr	有條件的呼叫
CP	比較
DEC	減量
DJNZ	如果不是零，減量且跳越
IN	輸入
INC	增量
JR	相對跳越
JR cond,addr	有條件的相對跳越
LD reg,(HL)	載入暫存器
LD A,(addr)	直接載入累積器
LD data	載入立即數據
LD (HL),reg	儲存於暫存器
LD (addr),A	直接儲存累積器
LD dst,src	暫存器移入暫存器
OUT	輸出
POP	從堆疊頂層讀取
PUSH	寫入堆疊頂層
RET	從副程式回復
RET cond	有條件的回復
RLA	把累積器內容經過進位向左旋轉
RRA	把累積器內容經過進位向右旋轉
SLL	向左算術移位
SRL	向右邏輯移位
SUB	減法

偶爾使用的Z-80指令

指令	意義
BIT	測試位元
CPD,CPDR	比較，減量，(重複)
CPI,CPIR	比較，增量，(重複)
CPL	取累積器的補數
DAA	十進位調整累積器
DI	停止中斷
EI	啟動中斷
EX	互換
HALT	停止
IND,INDR	輸入，減量，(重複)
INI,INIR	輸入，增量，(重複)
JP addr	跳越
JP cond,addr	有條件的跳越
LD A,(BC) or (DE)	二次載入累積器
LD HL,(addr)	直接載入HL
LD reg,(xy+disp)	載入暫存器指標
LD rp,(addr)	直接載入暫存器
LD xy,(addr)	直接載入指標暫存器
LD (BC) or (DE),A	二次存入累積器
LD (addr),HL	HL內含存入addr位置
LD (xy+disp),reg	儲存暫存器指標
LD (addr),rp	直接儲存暫存器對
LD (addr),xy	直接儲存指標暫存器
LD (HL),data	把立即數據存入記憶體
LD (xy+disp),data	把立即數據存入記憶體指標
LDD,LDDR	載入，減量，(重複)
LDI,LDIR	載入，增量，(重複)
NEG	否定(2'補數)累積器
NOP	不動作
OR	邏輯“OR”
OUTD,OTDR	輸出，減量，(重複)
OUTI,OTIR	輸出，增量，(重複)
RES	重置位元
RETI	從中斷回復
RL	經過進位向左旋轉
RLC	向左循環旋轉
RLCA	把累積器內容向左循環旋轉
RR	經過進位向右旋轉

RRC	向右循環旋轉
RRCA	把累積器內容向右循環旋轉
SET	置定位元
SRA	向右算術移位
XOR	邏輯“XOR”

十六進制轉換表

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	00	000
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	0
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	256	4096
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	512	8192
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	768	12288
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	1024	16384
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	1280	20480
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	1536	24576
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	1792	28672
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	2048	32768
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	2304	36864
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	2560	40960
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	2816	45056
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	3072	49152
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	3328	53248
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	3584	57344
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	3840	61440

5	4	3	2	1	0
HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0
1	1,048,576	1	4,096	1	16
2	2,097,152	2	8,192	2	32
3	3,145,728	3	12,288	3	48
4	4,194,304	4	16,384	4	64
5	5,242,880	5	20,480	5	80
6	6,291,456	6	24,576	6	96
7	7,340,032	7	28,672	7	112
8	8,388,608	8	32,768	8	128
9	9,437,184	9	36,864	9	144
A	10,485,760	A	40,960	A	160
B	11,534,336	B	45,056	B	176
C	12,582,912	C	49,152	C	192
D	13,631,488	D	53,248	D	208
E	14,680,064	E	57,344	E	224
F	15,728,640	F	61,440	F	240

較少使用的 Z - 8 0 指令

指令	意	義
ADC HL, rp	把暫存器對與進位加入 HL 中	
CCF	數進位旗號的補數	
EXX	把暫存器對與交互暫存器對內容互換	
IM n	置定中斷模式	
RETn	從無遮蔽中斷回復	
R1D	把累積器的 LSD 和記憶體連起來向左旋轉 4 位元	
RRD	把累積器的 LSD 和記憶體連起來向右旋轉 4 位元	
RST	重新開始	
SBC	同時減去進位旗號 ( 借位 )	
SCF	置定進位旗號	
LD A, I	從中斷向量暫存器載入累積器中	
LD A, R	從更新暫存器載入累積器中	
LD I, A	把累積器內容存入中斷向量暫存器中	
LD R, A	把累積器內容存入更新暫存器中	
LD SP, HL	把 HL 移入堆疊指示器	
LD SP, xy	把指標暫存器移入堆疊指示器	



ASCII 轉換表

HEX	MSD	0	1	2	3	4	5	6	7
LSD	BITS	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SPACE	0	@	P	-	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	,	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	.	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	*	<	L	\	l	~
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	^	n	~
F	1111	SI	US	/	?	O	←	o	DEL

ASCII 字符

NUL	— Null	DLE	— Data Link Escape
SOH	— Start of Heading	DC	— Device Control
STX	— Start of Text	NAK	— Negative Acknowledge
ETX	— End of Text	SYN	— Synchronous Idle
ENO	— End of Transmission	ETB	— End of Transmission Block
ACK	— Acknowledge	CAN	— Cancel
BEL	— Bell	EM	— End of Medium
BS	— Backspace	SUB	— Substitute
HT	— Horizontal Tabulation	ESC	— Escape
LF	— Line Feed	FS	— File Separator
VT	— Vertical Tabulation	GS	— Group Separator
FF	— Form Feed	RS	— Record Separator
CR	— Carriage Return	US	— Unit Separator
SO	— Shift Out	SP	— Space (Blank)
SI	— Shift In	DEL	— Delete

相對跳越位移表

向前相對跳越

LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

回跳相對跳越

LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

十進數與BCD轉換

十進數	BCD	DEC	BCD	DEC	BCD
0	0000	10	00010000	90	10010000
1	0001	11	00010001	91	10010001
2	0010	12	00010010	92	10010010
3	0011	13	00010011	93	10010011
4	0100	14	00010100	94	10010100
5	0101	15	00010101	95	10010101
6	0110	16	00010110	96	10010110
7	0111	17	00010111	97	10010111
8	1000	18	00011000	98	10011000
9	1001	19	00011001	99	10011001

# Z-80 指令碼

工作碼	組合語言述句
8E	ADC A,(HL)
DD8E05	ADC A,(IX+d)
FD8E05	ADC A,(IY+d)
8F	ADC A,A
88	ADC A,B
89	ADC A,C
8A	ADC A,D
8B	ADC A,E
8C	ADC A,H
8D	ADC A,L
CE20	ADC A,n
ED4A	ADC HL,BC
ED5A	ADC HL,DE
ED6A	ADC HL,HL
ED7A	ADC HL,SP
86	ADD A,(HL)
DD8605	ADD A,(IX+d)
FD8605	ADD A,(IY+d)
87	ADD A,A
80	ADD A,B
81	ADD A,C
82	ADD A,D
83	ADD A,E
84	ADD A,H
85	ADD A,L
C620	ADD A,n
09	ADD HL,BC
19	ADD HL,DE
29	ADD HL,HL
39	ADD HL,SP
DD09	ADD IX,BC
DD19	ADD IX,DE
DD29	ADD IX,IX
DD39	ADD IX,SP
FD09	ADD IY,BC
FD19	ADD IY,DE
FD29	ADD IY,IY
FD39	ADD IY,SP
A6	AND (HL)
DDA605	AND (IX+d)
FDA605	AND (IY+d)
A7	AND A
A0	AND B
A1	AND C
A2	AND D
A3	AND E
A4	AND H
A5	AND L

工作碼	組合語言述句
E620	AND n
C846	BIT 0,(HL)
DDC80546	BIT 0,(IX+d)
FDC80546	BIT 0,(IY+d)
C847	BIT 0,A
C840	BIT 0,B
C841	BIT 0,C
C842	BIT 0,D
C843	BIT 0,E
C844	BIT 0,H
C845	BIT 0,L
C84E	BIT 1,(HL)
DDC8054E	BIT 1,(IX+d)
FDC8054E	BIT 1,(IY+d)
C84F	BIT 1,A
C848	BIT 1,B
C849	BIT 1,C
C84A	BIT 1,D
C84B	BIT 1,E
C84C	BIT 1,H
C84D	BIT 1,L
C856	BIT 2,(HL)
DDC80556	BIT 2,(IX+d)
FDC80556	BIT 2,(IY+d)
C857	BIT 2,A
C850	BIT 2,B
C851	BIT 2,C
C852	BIT 2,D
C853	BIT 2,E
C854	BIT 2,H
C855	BIT 2,L
C85E	BIT 3,(HL)
DDC8055E	BIT 3,(IX+d)
FDC8055E	BIT 3,(IY+d)
C85F	BIT 3,A
C858	BIT 3,B
C859	BIT 3,C
C85A	BIT 3,D
C85B	BIT 3,E
C85C	BIT 3,H
C85D	BIT 3,L
C866	BIT 4,(HL)
DDC80566	BIT 4,(IX+d)
FDC80566	BIT 4,(IY+d)
C867	BIT 4,A
C860	BIT 4,B
C861	BIT 4,C
C862	BIT 4,D

工作碼	組合語言述句
C863	BIT 4,E
C864	BIT 4,H
C865	BIT 4,L
C86E	BIT 5,(HL)
DDC8056E	BIT 5,(IX+d)
FDC8056E	BIT 5,(IY+d)
C86F	BIT 5,A
C868	BIT 5,B
C869	BIT 5,C
C86A	BIT 5,D
C86B	BIT 5,E
C86C	BIT 5,H
C86D	BIT 5,L
C876	BIT 6,(HL)
DDC80576	BIT 6,(IX+d)
FDC80576	BIT 6,(IY+d)
C877	BIT 6,A
C870	BIT 6,B
C871	BIT 6,C
C872	BIT 6,D
C873	BIT 6,E
C874	BIT 6,H
C875	BIT 6,L
C87E	BIT 7,(HL)
DDC8057E	BIT 7,(IX+d)
FDC8057E	BIT 7,(IY+d)
C87F	BIT 7,A
C878	BIT 7,B
C879	BIT 7,C
C87A	BIT 7,D
C87B	BIT 7,E
C87C	BIT 7,H
C87D	BIT 7,L
DC8405	CALL M
FC8405	CALL M,n
D48405	CALL NC,n
C48405	CALL NZ,n
F48405	CALL P,n
EC8405	CALL PE,n
E48405	CALL PO,n
CC8405	CALL Z,n
CD8405	CALL nn
3F	CCF
BE	CP (HL)
DD8E05	CP (IX+d)
FD8E05	CP (IY+d)
BF	CP A
B8	CP B
B9	CP C
BA	CP D
BB	CP E
BC	CP H
BD	CP L
FE20	CP n
EDA9	CPD
ED89	CPDR

工作碼	組合語言述句
ED81	CPIR
EDA1	CPI
2F	CPL
27	DAA
35	DEC (HL)
DD3505	DEC (IX+d)
FD3505	DEC (IY+d)
3D	DEC A
05	DEC B
08	DEC BC
0D	DEC C
15	DEC D
18	DEC DE
1D	DEC E
25	DEC H
28	DEC HL
DD28	DEC IX
FD28	DEC IY
2D	DEC L
38	DEC SP
F3	DI
102E	DJNZ e
F8	EI
E3	EX (SPI),HL
DDE3	EX (SPI),IX
FDE3	EX (SPI),IY
08	EX AF,AF
EB	EX DE,HL
D9	EXX
76	HALT
ED46	IM 0
ED56	IM 1
ED5E	IM 2
ED78	IN A,(C)
ED40	IN B,(C)
ED48	IN C,(C)
ED50	IN D,(C)
ED58	IN E,(C)
ED60	IN H,(C)
ED68	IN L,(C)
34	INC (HL)
DD3405	INC (IX+d)
FD3405	INC (IY+d)
3C	INC A
04	INC B
03	INC BC
0C	INC C
14	INC D
13	INC DE
1C	INC E
24	INC H
23	INC HL
DD23	INC IX
FD23	INC IY
2C	INC L
33	INC SP
D820	IN A,(n)

工作碼	組合語言述句
EDAA	IND
EDBA	INDR
EDA2	INI
EDB2	INIR
C38405	JP nn
E9	JP (HL)
DDE9	JP (IX)
FDE9	JP (IY)
DAB405	JP C,nn
FAB405	JP M,nn
O28405	JP NC,nn
C28405	JP NZ,nn
F28405	JP P,nn
EA8405	JP PE,nn
E28405	JP PO,nn
CA8405	JP Z,nn
382E	JR C,e
302E	JR NC,e
202E	JR NZ,e
282E	JR Ze
182E	JR e,HL
O2	LD (BC),A
12	LD (DE),A
77	LD (HL),A
70	LD (HL),B
71	LD (HL),C
72	LD (HL),D
73	LD (HL),E
74	LD (HL),H
75	LD (HL),L
3620	LD (HL),n
DD7705	LD (IX+d),A
DD7005	LD (IX+d),B
DD7105	LD (IX+d),C
DD7205	LD (IX+d),D
DD7305	LD (IX+d),E
DD7405	LD (IX+d),H
DD7505	LD (IX+d),L
DD360520	LD (IX+d),n
F07705	LD (IY+d),A
F07005	LD (IY+d),B
F07105	LD (IY+d),C
F07205	LD (IY+d),D
F07305	LD (IY+d),E
F07405	LD (IY+d),H
F07505	LD (IY+d),L
F0360520	LD (IY+d),n
328405	LD (nn),A
ED438405	LD (nn),BC
ED538405	LD (nn),DE
228405	LD (nn),HL
DD228405	LD (nn),IX
F0228405	LD (nn),IY
ED738405	LD (nn),SP
0A	LD A,(BC)
1A	LD A,(DE)
7E	LD A,(HL)

工作碼	組合語言述句
DD7E05	LD A,(IX+d)
FD7E05	LD A,(IY+d)
3A8405	LD A,(nn)
7F	LD A,A
78	LD A,B
79	LD A,C
7A	LD A,D
7B	LD A,E
7C	LD A,H
ED57	LD A,I
7D	LD A,L
3E20	LD A,n
ED5F	LD A,r
46	LD B,(HL)
DD4605	LD B,(IX+d)
FD4605	LD B,(IY+d)
47	LD B,A
40	LD B,B
41	LD B,C
42	LD B,D
43	LD B,E
44	LD B,H
45	LD B,L
0620	LD B,n
FD488405	LD BC,(nn)
018405	LD BC,nn
4E	LD C,(HL)
DD4E05	LD C,(IX+d)
FD4E05	LD C,(IY+d)
4F	LD C,A
48	LD C,B
49	LD C,C
4A	LD C,D
4B	LD C,E
4C	LD C,H
4D	LD C,L
0E20	LD C,n
56	LD D,(HL)
DD5605	LD D,(IX+d)
FD5605	LD D,(IY+d)
57	LD D,A
50	LD D,B
51	LD D,C
52	LD D,D
53	LD D,E
54	LD D,H
55	LD D,L
1620	LD D,n
ED588405	LD DE,(nn)
118405	LD DE,nn
5E	LD E,(HL)
DD5E05	LD E,(IX+d)
FD5E05	LD E,(IY+d)
5F	LD E,A
58	LD E,B
59	LD E,C
5A	LD E,D

工作碼	組合語言述句
5B	LD LD
5C	LD E,E
5D	LD E,H
1E20	LD E,n
66	LD H,(HL)
DD6605	LD H,(IX+d)
FD6605	LD H,(IY+d)
67	LD H,A
60	LD H,B
61	LD H,C
62	LD H,D
63	LD H,E
64	LD H,H
65	LD H,L
2620	LD H,n
2A8405	LD HL,(nn)
218405	LD HL,nn
ED47	LD I,A
DD2A8405	LD IX,(nn)
DD218405	LD IX,nn
FD2A8405	LD IY,(nn)
FD218405	LD IY,nn
6E	LD L,(HL)
DD6E05	LD L,(IX+d)
FD6E05	LD L,(IY+d)
6F	LD L,A
68	LD L,B
69	LD L,C
6A	LD L,D
6B	LD L,E
6C	LD L,H
6D	LD L,L
2E20	LD L,n
ED4F	LD R,A
DD788405	LD SP,(nn)
F9	LD SP,HL
DDF9	LD SP,IX
FDf9	LD SP,IY
318405	LD SP,nn
EDA8	LD LDD
ED88	LD LDDR
EDA0	LDI
ED80	LDIR
ED44	NEG
00	NOP
B6	OR (HL)
DD6605	OR (IX+d)
FD6605	OR (IY+d)
B7	OR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
F620	OR n
ED88	OTDR

工作碼	組合語言述句
ED83	OTIR
ED79	OUT (C),A
ED41	OUT (C),B
ED49	OUT (C),C
ED51	OUT (C),D
ED59	OUT (C),E
ED61	OUT (C),H
ED69	OUT (C),L
D320	OUT (n),A
EDAB	OUTD
EDA3	OUTI
F1	POP AF
C1	POP BC
D1	POP DE
E1	POP HL
DDE1	POP IX
FDE1	POP IY
F5	PUSH AF
C5	PUSH BC
D5	PUSH DE
E5	PUSH HL
DDE5	PUSH IX
FDE5	PUSH IY
C886	RES 0,(HL)
DDCB0586	RES 0,(IX+d)
FDCB0586	RES 0,(IY+d)
C887	RES 0,A
C880	RES 0,B
C881	RES 0,C
C882	RES 0,D
C883	RES 0,E
C884	RES 0,H
C885	RES 0,L
C88E	RES 1,(HL)
DDCB058E	RES 1,(IX+d)
FDCB058E	RES 1,(IY+d)
C88F	RES 1,A
C888	RES 1,B
C889	RES 1,C
C88A	RES 1,D
C88B	RES 1,E
C88C	RES 1,H
C88D	RES 1,L
C896	RES 2,(HL)
DDCB0596	RES 2,(IX+d)
FDCB0596	RES 2,(IY+d)
C897	RES 2,A
C890	RES 2,B
C891	RES 2,C
C892	RES 2,D
C893	RES 2,E
C894	RES 2,H
C895	RES 2,L
C89E	RES 3,(HL)
DDCB059E	RES 3,(IX+d)
FDCB059E	RES 3,(IY+d)

工作碼	組合語言述句
CB9F	RES 3 A
CB98	RES 3 B
CB99	RES 3 C
CB9A	RES 3 D
CB9B	RES 3 E
CB9C	RES 3 H
CB9D	RES 3 L
CB9E	RES 4 (HL)
DDCB05A6	RES 4 (IY+d)
FDCB05A6	RES 4 (IY+d)
CBA7	RES 4 A
CBA0	RES 4 B
CBA1	RES 4 C
CBA2	RES 4 D
CBA3	RES 4 E
CBA4	RES 4 H
CBA5	RES 4 L
CBAE	RES 5 (HL)
DDCB05AE	RES 5 (IY+d)
FDCB05AE	RES 5 (IY+d)
CBAF	RES 5 A
CBA8	RES 5 B
CBA9	RES 5 C
CBA4	RES 5 D
CBA8	RES 5 E
CBAE	RES 5 H
CBAE	RES 5 L
CB86	RES 6 (HL)
DDCB05B6	RES 6 (IY+d)
FDCB05B6	RES 6 (IY+d)
CB87	RES 6 A
CB80	RES 6 B
CB81	RES 6 C
CB82	RES 6 D
CB83	RES 6 E
CB84	RES 6 H
CB85	RES 6 L
CB8E	RES 7 (HL)
DDCB05BE	RES 7 (IY+d)
FDCB05BE	RES 7 (IY+d)
CB8F	RES 7 A
CB88	RES 7 B
CB89	RES 7 C
CB8A	RES 7 D
CB8B	RES 7 E
CB8C	RES 7 H
CB8D	RES 7 L
C9	RET C
D8	RET M
F8	RET NC
D0	RET NZ
C0	RET P
F0	RET PE
E8	RET PO
E0	RET Z
C8	

工作碼	組合語言述句
ED4D	RETI
ED45	RETN
CB16	RL (HL)
DDCB0516	RL (IY+d)
FDCB0516	RL (IY+d)
CB17	RL A
CB10	RL B
CB11	RL C
CB12	RL D
CB13	RL E
CB14	RL H
CB15	RL L
17	RLA
CB06	RLC (HL)
DDCB0506	RLC (IY+d)
FDCB0506	RLC (IY+d)
CB07	RLC A
CB00	RLC B
CB01	RLC C
CB02	RLC D
CB03	RLC E
CB04	RLC H
CB05	RLC L
07	RLCA
ED6F	RLD
CB1E	RR (HL)
DDCB051E	RR (IY+d)
FDCB051E	RR (IY+d)
CB1F	RR A
CB18	RR B
CB19	RR C
CB1A	RR D
CB1B	RR E
CB1C	RR H
CB1D	RR L
1F	RRR
CB0E	RRC (HL)
DDCB050E	RRC (IY+d)
FDCB050E	RRC (IY+d)
CB0F	RRC A
CB08	RRC B
CB09	RRC C
CB0A	RRC D
CB0B	RRC E
CB0C	RRC H
CB0D	RRC L
0F	RRCA
ED67	RRD
C7	RST 00H
CF	RST 08H
D7	RST 10H
DF	RST 18H
E7	RST 20H
FF	RST 28H
F7	RST 30H
FF	RST 38H
DE20	SBC A,n

工作碼	組合語言述句
9E	SBC A (HL)
DD9E05	SBC A (IY+d)
FD9E05	SBC A (IY+d)
9F	SBC A A
98	SBC A B
99	SBC A C
9A	SBC A D
9B	SBC A E
9C	SBC A H
9D	SBC A L
ED42	SBC H BC
ED52	SBC HL DE
ED62	SBC HL HL
ED72	SBC HL SP
37	SCF
CB06	SET 0 (HL)
DDCB05C6	SET 0 (IY+d)
FDCB05C6	SET 0 (IY+d)
CB07	SET 0 A
CB0C	SET 0 B
CB01	SET 0 C
CB02	SET 0 D
CB03	SET 0 E
CB04	SET 0 H
CB05	SET 0 L
CB0E	SET 1 (HL)
DDCB05CE	SET 1 (IY+d)
FDCB05CE	SET 1 (IY+d)
CB0F	SET 1 A
CB08	SET 1 B
CB09	SET 1 C
CB0A	SET 1 D
CB0B	SET 1 F
CB0C	SET 1 H
CB0D	SET 1 L
CB06	SET 2 (HL)
DDCB05D6	SET 2 (IY+d)
FDCB05D6	SET 2 (IY+d)
CB07	SET 2 A
CB00	SET 2 B
CB01	SET 2 C
CB02	SET 2 D
CB03	SET 2 E
CB04	SET 2 H
CB05	SET 2 L
CB08	SET 3 B
CB0E	SET 3 (HL)
DDCB05DE	SET 3 (IY+d)
FDCB05DE	SET 3 (IY+d)
CB0F	SET 3 A
CB09	SET 3 C
CB0A	SET 3 D
CB0B	SET 3 E
CB0C	SET 3 H
CB0D	SET 3 L
CB06	SET 4 (HL)

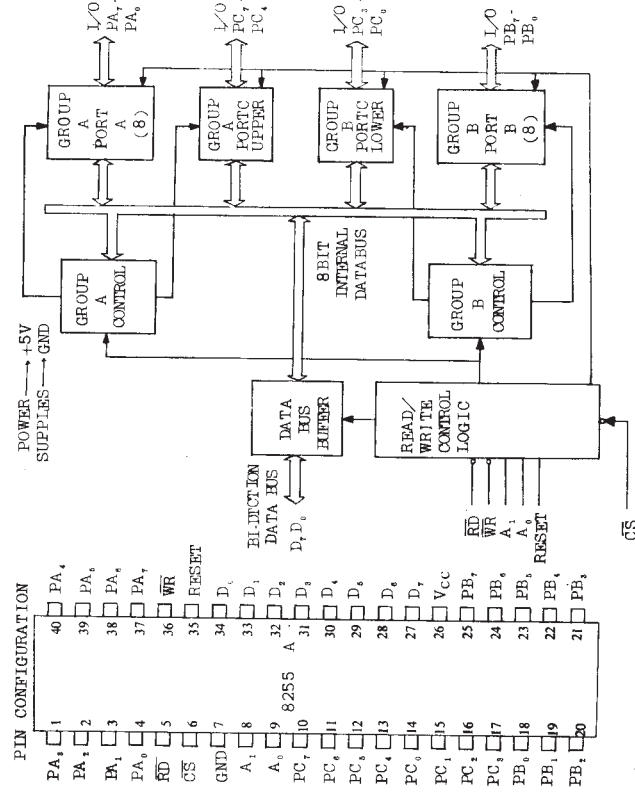
工作碼	組合語言述句
DDCB05E6	SET 4 (IY+d)
FDCB05E6	SET 4 A
CB07	SET 4 A
CB00	SET 4 B
CB01	SET 4 C
CB02	SET 4 D
CB03	SET 4 E
CB04	SET 4 H
CB05	SET 4 L
CB0E	SET 5 (HL)
DDCB05EE	SET 5 (IY+d)
FDCB05EE	SET 5 (IY+d)
CB0F	SET 5 A
CB08	SET 5 B
CB09	SET 5 C
CB0A	SET 5 D
CB0B	SET 5 E
CB0C	SET 5 H
CB0D	SET 5 L
CB06	SET 6 (HL)
DDCB05F6	SET 6 (IY+d)
FDCB05F6	SET 6 (IY+d)
CB07	SET 6 A
CB00	SET 6 B
CB01	SET 6 C
CB02	SET 6 D
CB03	SET 6 E
CB04	SET 6 H
CB05	SET 6 L
CB0E	SET 7 (HL)
DDCB05FE	SET 7 (IY+d)
FDCB05FE	SET 7 (IY+d)
CB0F	SET 7 A
CB08	SET 7 B
CB09	SET 7 C
CB0A	SET 7 D
CB0B	SET 7 E
CB0C	SET 7 H
CB0D	SET 7 L
CB06	SLA (HL)
DDCB0526	SLA (IY+d)
FDCB0526	SLA (IY+d)
CB07	SLA A
CB00	SLA B
CB01	SLA C
CB02	SLA D
CB03	SLA E
CB04	SLA H
CB05	SLA L
CB0E	SRA (HL)
DDCB052E	SRA (IY+d)
FDCB052E	SRA (IY+d)
CB0F	SRA A
CB08	SRA B
CB09	SRA C
CB0A	SRA D

# 附錄

## 周邊設備界面電路—8255

8255 是通用的可規劃 I/O 設備界面電路，具有 24 條 I/O 綫，可執行三類操作。

8255 之 block diagram 如圖 1。



PIN NAMES

D - D	DATA BUS
RESET	RESET INPUT
CS	CHIP SELECT
RD	READ INPUT
WR	WRITE INPUT
A0-A1	PORT ADDRESS
PA7-PA0	PORT A (8 BIT)
PB7-PB0	PORT B (8 BIT)
PC7-PC0	PORT C (8 BIT)
VCC	+5VOLTS
GND	0VOLTS

8255A Block Diagram Showing Group A and Group B Control Functions

### 工作碼 組合語言述句

CB2B	SRA	E
CB2C	SRA	H
CB2D	SRA	L
CB3E	SRL	(HL)
DDC8053E	SRL	(IX+d)
FDC8053E	SRL	(IV+d)
CB3F	SRL	A
CB38	SRL	B
CB39	SRL	C
CB3A	SRL	D
CB3B	SRL	E
CB3C	SRL	H
CB3D	SRL	L
96	SUB	(HL)
DD9605	SUB	(IX+d)
FD9605	SUB	(IV+d)
97	SUB	A
90	SUB	B
91	SUB	C
92	SUB	D
93	SUB	E
94	SUB	H
95	SUB	L
D620	SUB	n
AE	XOR	(HL)
DDAE05	XOR	(IX+d)
FDAE05	XOR	(IV+d)
AF	XOR	A
A8	XOR	B
A9	XOR	C
AA	XOR	D
AB	XOR	E
AC	XOR	H
AD	XOR	L
EE20	XOR	n

(Courtesy of Zilog Inc.)

Data bus buffer 具有 3-state bidirection 之 8-bit buffer, 用來 Interface 8255 與 System data bus, 資料之傳送或接收由 CPU 執行, IN 或 OUT 來進行, Control words 及 status information 也是由此傳送。

Read/Write and Control logic 接受 System 之控制信號

$\overline{CS}$  —— CPU 選擇此一 chip 之信號 ( chip select )

$\overline{RD}$  —— 8255 送資料後, CPU, 亦即 CPU 從 8255 讀取 ( Read ) data 或 status 之控制信號。

$\overline{WR}$  —— CPU 把 data 或 Control words 送給 8255 之控制信號 ( write )

RESET —— RESET 為 H 時會 clear 8255 所有之 registers, 同時使 ports A, B 與 C 都定為 input 。

$A_1, A_0$  —— 用來選擇 port 或 Control word Register, 其選擇方式配合 RD, WR 與 CS 信號後之工作情形如表 1。

8255 BASIC OPERATION

$A_1$	$A_0$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORTA $\Rightarrow$ DATABUS
0	1	0	1	0	PORTB $\Rightarrow$ DATABUS
1	0	0	1	0	PORTC $\Rightarrow$ DATABUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS $\Rightarrow$ PORTA
0	1	1	0	0	DATA BUS $\Rightarrow$ PORTB
1	0	1	0	0	DATA BUS $\Rightarrow$ PORTC
1	1	1	0	0	DATA BUS $\Rightarrow$ CONTROL
					DISABLE FUNCTION
x	x	x	x	1	DATA BUS $\Rightarrow$ 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
x	x	1	1	0	DATA BUS $\Rightarrow$ 3-STATE

## Group A 與 Group B 之控制

8255 之 24 I/O 線分成 A 與 B 兩個 12 I/O 線之 Group。

Group A 包含 port A 及 port C 之高階 4 線, Group B 包含 port B 及 port C 之低階 4 線, 此兩個 Group 分別由 Group A Control 及 Group B Control 來控制這兩個 Control

blocks 接受 CPU 之 Commands, Control word 即寫入

Control word Register 中此 Register 亦能被讀出。

Port A 具有一個 8-bit 之 data output latch/buffer 及一個 8-bit 之 data input latch.

Port B 具有一個 8-bit data input/output latch/buffer 及一個 8-bit data input buffer。

Port C 是 8-bit data output latch/buffer 及一個 8-bit data input buffer, 這個 port 可分成兩個 4-bit ports 此 4-bit port 含有 4-bit latch, 可作為 port B 之 Control signal outputs 及 status signal input 之用。

8255 有三種 Operation modes, 即

MODE 0 Basic input/output

MODE 1 Strobed input/output

MODE 2 Bidirectional bus

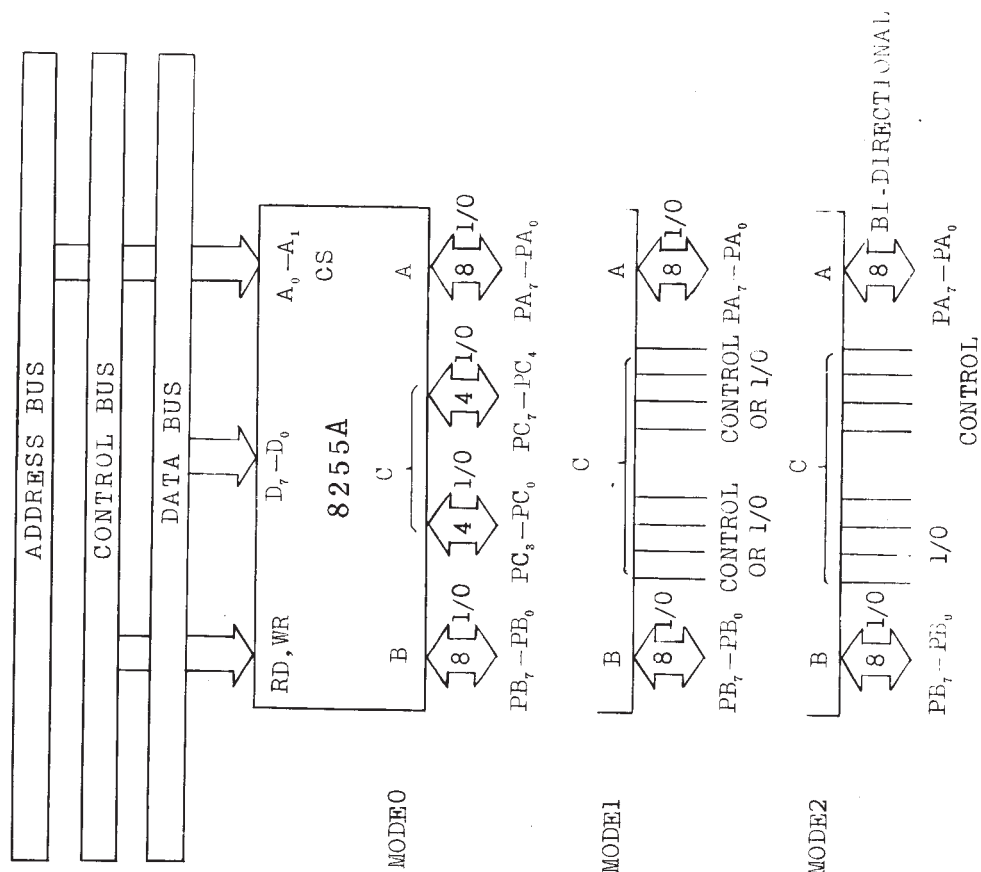
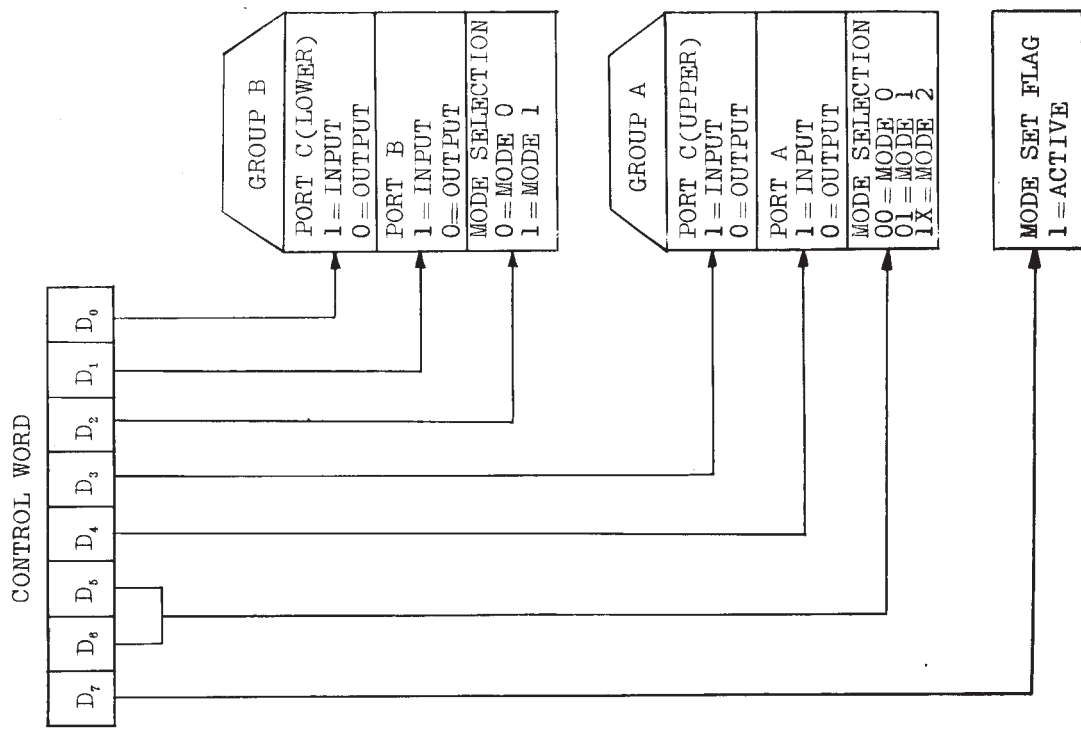
當 Reset 時, 所有 ports 都被定為 input mode。Reset 信號移去之後, 8255 可維持在 input mode。在執行程式時, 其他 mode 可以利用 out 指令來選用, 因此 8255 可作各種不同之操作使用。

port A 與 port B 之 modes 可以分別指定, port C 則分成兩部分, 分別為 port A 與 port B 所定義。

基本之 mode 定義及 bus Interface 如下圖。



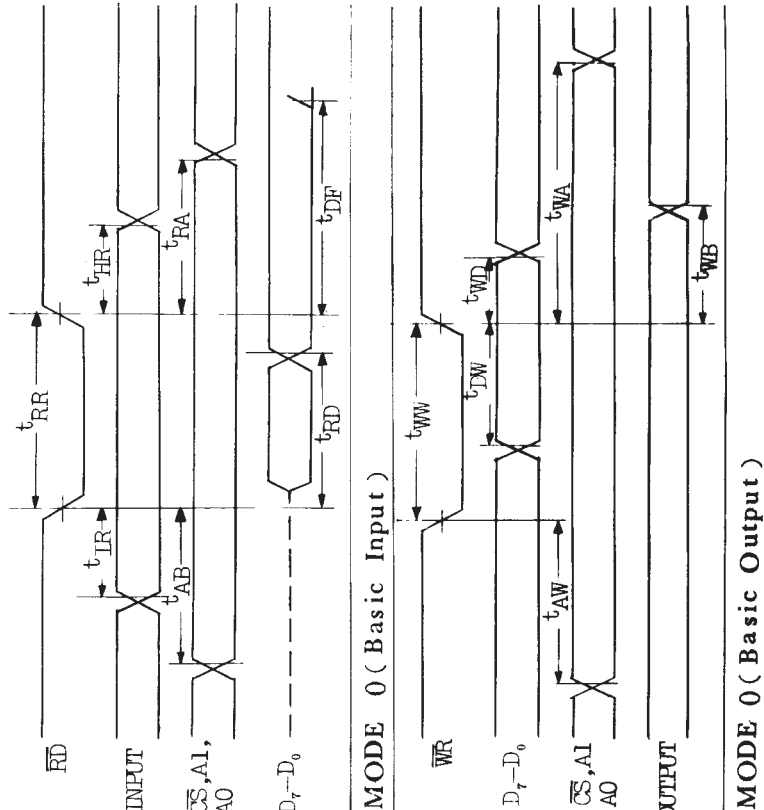
定義mode 之 Control, 其最高階 bit 為 1, 即  $D_7 = 1$ , 其他各 bit 之代表意義如下圖。



Port C 需要 bit Set/Reset 是當它做為 Port A 與 Port B 之 Status/Control 時，有時需要對 CPU 產生 interrupt 信號，這些由 Port C 產生出來的 interrupt request 信號，以由 Set/Reset bit 來 enable/disable 其對應之 INTE flip-flop INTE flip-flop 之動作方式如下：

- BIT-SET — INTE is set — Interrupt enable
  - BIT-RESET — INTE is reset — Interrupt disable
- MODE 0 Basic Input/Output

mode 0 是執行簡單的 I/O 操作，沒有 handshaking 資料只是寫入 port 或從 port 讀出，其 TIMING 如下。



MODE 0 之基本功能可細分成十六種，由 mode control word 決定，其 port 定義之方式如下表

A		B		GROUP A		GROUP B	
D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	PORT A	PORTC (UPPER)	PORT B (LOWER)
0	0	0	0	0	OUTPUT	OUTPUT	0
0	0	0	0	1	OUTPUT	OUTPUT	1
0	0	1	0	0	OUTPUT	OUTPUT	2
0	0	1	1	0	OUTPUT	OUTPUT	3
0	1	0	0	0	OUTPUT	INPUT	4
0	1	0	1	0	OUTPUT	INPUT	5
0	1	1	0	0	OUTPUT	INPUT	6
0	1	1	1	0	OUTPUT	INPUT	7
1	0	0	0	0	INPUT	OUTPUT	8
1	0	0	1	0	INPUT	OUTPUT	9
1	0	1	0	0	INPUT	OUTPUT	0
1	0	1	1	0	INPUT	OUTPUT	11
1	1	0	0	0	INPUT	INPUT	12
1	1	0	1	0	INPUT	INPUT	13
1	1	1	0	0	INPUT	INPUT	14
1	1	1	1	0	INPUT	INPUT	15

當做 output 時，資料會被 latch 住。做 input 時，資料不被 latch 住。

### MODE 1 Strobed Input/Output

在 Mode 1 時 port A 與 Port B 利用 Port C 來產生或接受 handshaking 信號，這時候的 Input 與 output 資料都會被 latch，以下分別說明 Input 與 output 之操作

#### (1) Input Control

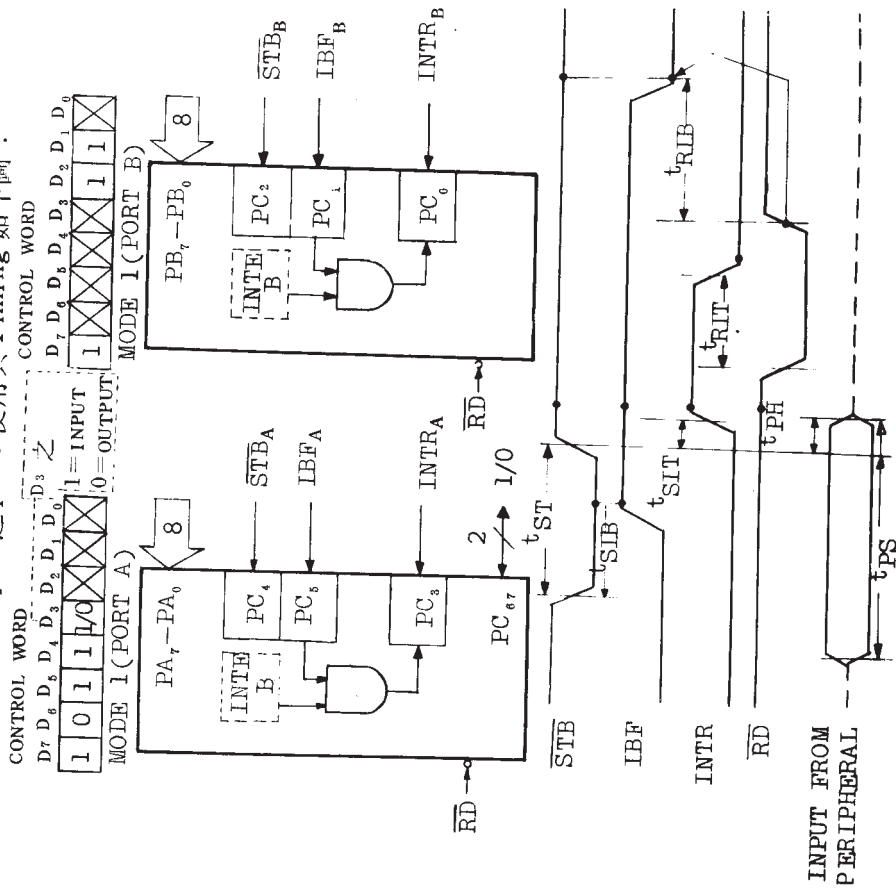
- STB—( strobe input )——將 data 放進 input latch
- IBF—( Input buffer Full )——此 output 信號表示 data 已經放進 input latch，在 read 之 rising edge 時

信號才消失。

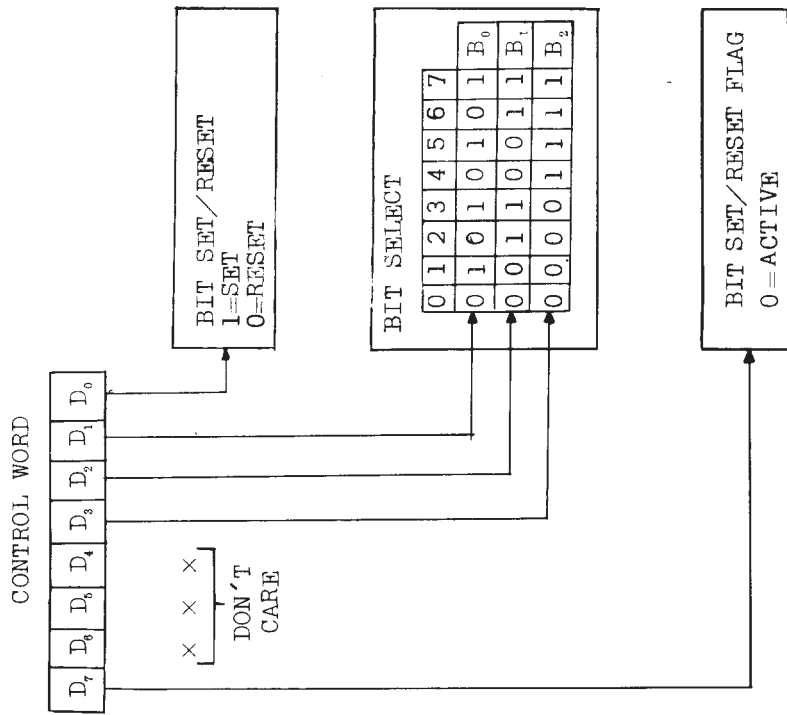
INTR (Interrupt Request)——INTR之產生由 INTE flipflop 管制，產生時間是在 STB 之後，IBF 與 RD 皆有 H 時，此信號用來告訴 CPU 向此 port 讀取資料，因此只要 strobe 進 data，此 port 即向 CPU 要求 service。

INTE 由 PC4 之 Set/Reset 控制，INTE B 由 PC2 之 bit Set/Reset 控制。

mode 1 input 之 port 使用具 Timing 如下圖：



Part C 之每一個 bit 都可以單獨被 set 或 reset，這種 bit set/reset 之 Control word 其最高階 bit 為 0，即 D<sub>7</sub>=0，其他各 bit 之定義如下圖：



Bit Set/Reset Format

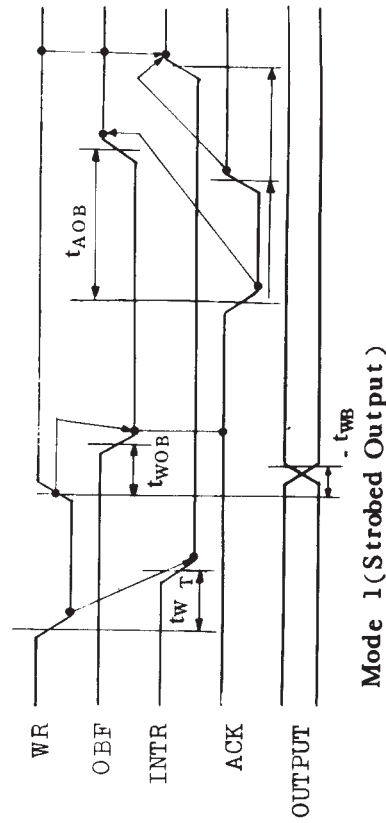
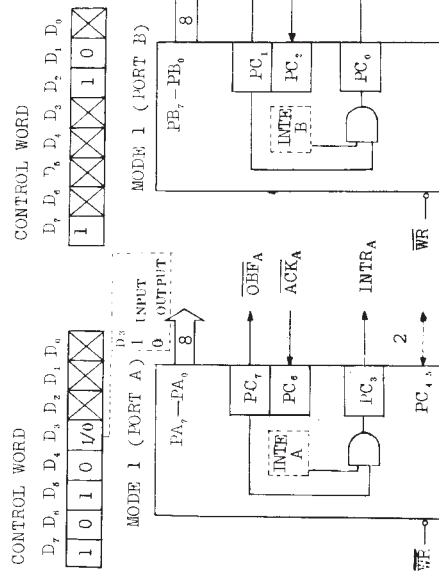
## (2) Output Control

$\overline{\text{OBF}}$  (Output Buffer Full)——此output 信號表示CPU 已經把data 時又回復,亦即由ACK reset。

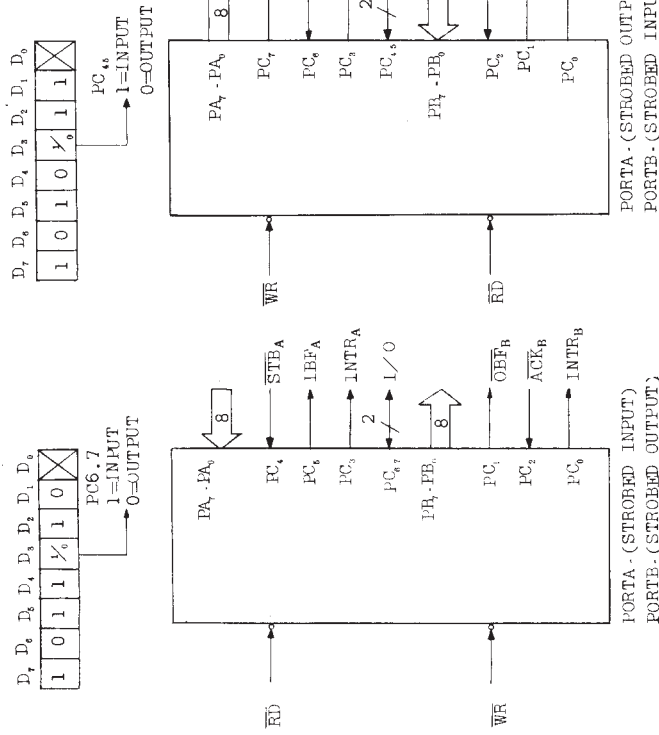
ACK (Acknowledge Input) 外界從此port 取出資料從此信號告知 8255 資料已被取出:

INTR (Interrupt Request)——此信號告訴CPU 外界已得到由CPU送來的資料,在enable interrupt時,INTR由ACK = 1只OBF = 1 set

INTEA 具INTE 分別由PC<sub>6</sub> 與PC<sub>2</sub>之bit Set/Reset 控制。



port A與ports 可混合定義為input 與output , 其Control words 及I/O 定義如下圖:



## MODE 2 Strobed Bi-directional Bus I/O

此mode 適用於只用8-bit 作Input 與output 之周邊設備。

INTR (Interrupt Request) —— 在I/O operation 時中斷CPU 要求Senrice 之用。

$\overline{\text{OBF}}$  (output Buffer Full) —— 表示CPU已將data 寫入Port A。

$\overline{\text{ACK}}$  (Acknowledge) —— 使Port A之output buffer 送出data , ACK為H時output buffer 為H時, output buffer 為high-impedance。

INTE 1 (INTE flip-flop associated with OBF)

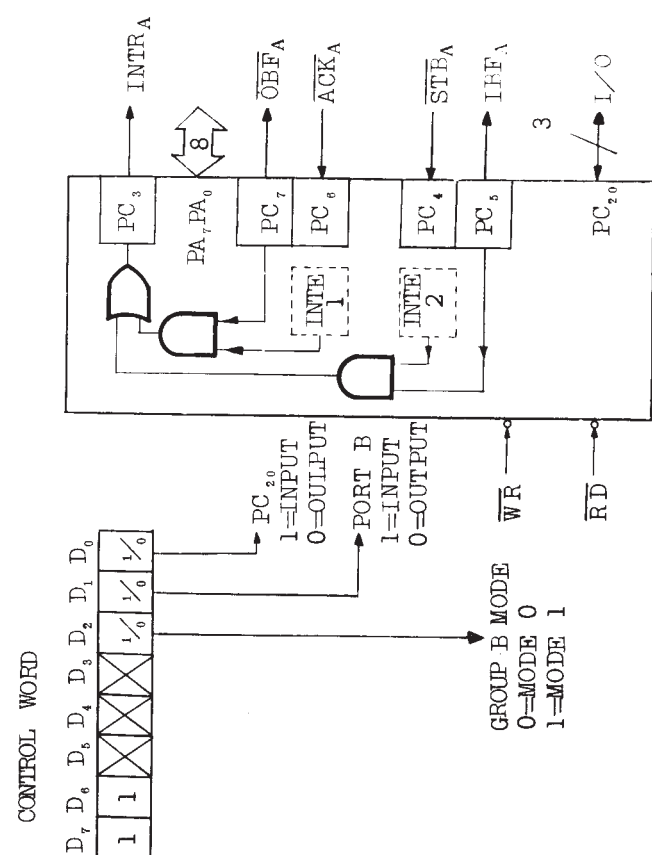
由PC0之bit Set/Reset 控制。

$\overline{\text{STB}}$  (strobe input)——此信號將 data 由外界放進 input latch。

$\overline{\text{IBF}}$  (input Buffer Full)——表示外界已將 data 放進 input latch。

$\text{INTE } 2$  (INTE flip-flop associated with  $\overline{\text{IBF}}$ )——由PC4 之bit Set/Reset 控制。

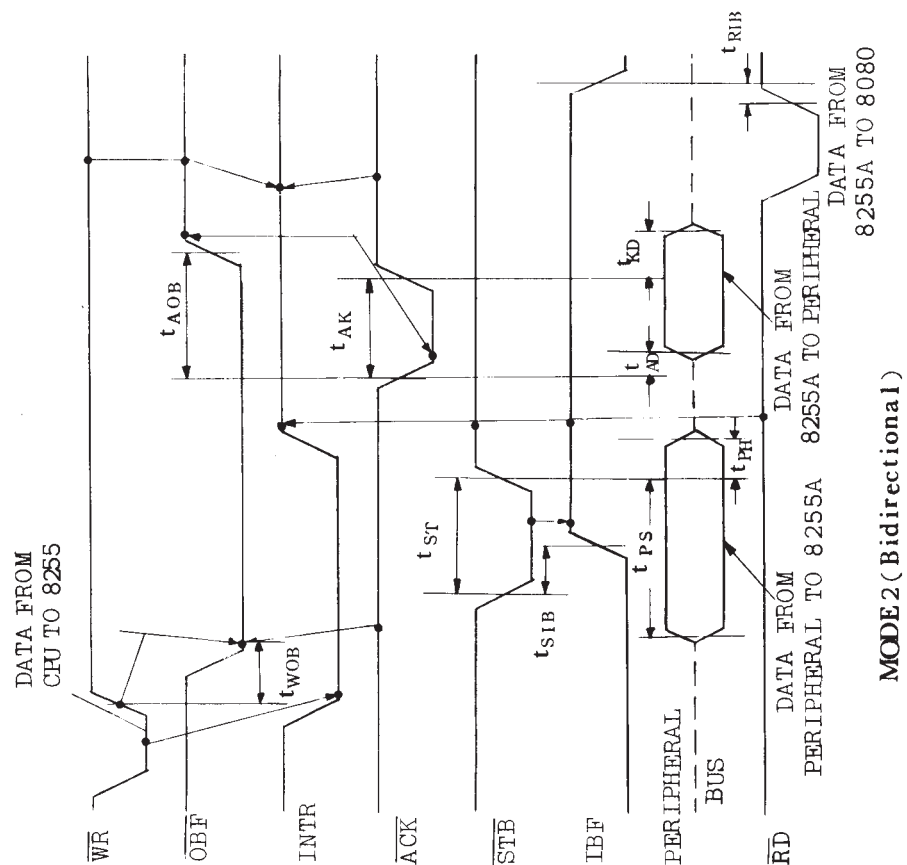
mode 2 之 control word 及 port 定義如下圖。



MODE Control word

MODE 2

mode 2 之 Timing diagram 如下圖：



- mode 2 可以與 mode 0 或 mode 1 混合使用。
- port B 與 port C 之 output buffer 可提供 1 mA 於 1.5 V 之時，故有推動 Darlington 之類的電路及高電壓之 display。

• 上述三種modes 對各 port 之定義歸納如下表：

MODE DEFINITION SUMMARY TABLE

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	MODE 2 GROUP A ONLY
PA <sub>0</sub>	IN	OUT	IN	OUT	↔	↔
PA <sub>1</sub>	IN	OUT	IN	OUT	↔	↔
PA <sub>2</sub>	IN	OUT	IN	OUT	↔	↔
PA <sub>3</sub>	IN	OUT	IN	OUT	↔	↔
PA <sub>4</sub>	IN	OUT	IN	OUT	↔	↔
PA <sub>5</sub>	IN	OUT	IN	OUT	↔	↔
PA <sub>6</sub>	IN	OUT	IN	OUT	↔	↔
PA <sub>7</sub>	IN	OUT	IN	OUT	↔	↔
PB <sub>0</sub>	IN	OUT	IN	OUT	—	—
PB <sub>1</sub>	IN	OUT	IN	OUT	—	—
PB <sub>2</sub>	IN	OUT	IN	OUT	—	—
PB <sub>3</sub>	IN	OUT	IN	OUT	—	—
PB <sub>4</sub>	IN	OUT	IN	OUT	—	—
PB <sub>5</sub>	IN	OUT	IN	OUT	—	—
PB <sub>6</sub>	IN	OUT	IN	OUT	—	—
PB <sub>7</sub>	IN	OUT	IN	OUT	—	—
PC <sub>0</sub>	IN	OUT	INTR <sub>B</sub>	INTR <sub>B</sub>	1/0	1/0
PC <sub>1</sub>	IN	OUT	IBF <sub>B</sub>	IBF <sub>B</sub>	1/0	1/0
PC <sub>2</sub>	IN	OUT	STB <sub>B</sub>	ACK <sub>B</sub>	1/0	1/0
PC <sub>3</sub>	IN	OUT	INTR <sub>A</sub>	INTR <sub>A</sub>	INTR <sub>A</sub>	INTR <sub>A</sub>
PC <sub>4</sub>	IN	OUT	STB <sub>A</sub>	1/0	STB <sub>A</sub>	STB <sub>A</sub>
PC <sub>5</sub>	IN	OUT	IBF <sub>A</sub>	1/0	IBF <sub>A</sub>	IBF <sub>A</sub>
PC <sub>6</sub>	IN	OUT	1/0	ACK <sub>A</sub>	ACK <sub>A</sub>	ACK <sub>A</sub>
PC <sub>7</sub>	IN	OUT	1/0	OBFA	OBFA	OBFA

MODE 0  
OR 1  
ONLY

• 以下是幾個應用 8255 的例子。

- 在mode 1 及 mode 2 時，port C 是產生或接受 hand-shaking 信號，讀取 PC 可知道外界 peripheral device 之狀況，讀取 port C 就只是用一段 I/O read 之操作。
- 在mode 1 及 mode 2 情形下讀取。
- mode 1 與mode 2 之 Status Word Formats 如下圖：

INPUT CONFIGURATION

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1/0	1/0	IBFA	INTEA	INTR <sub>A</sub>	INTE <sub>B</sub>	IBFB	INTR <sub>B</sub>
GROUP A				GROUP B			

OUTPUT CONFIGURATION

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
OBFA	INTEA	1/0	1/0	INTR <sub>A</sub>	INTE <sub>B</sub>	OBFB	INTR <sub>B</sub>
GROUP A				GROUP B			

Mode 1 Status Word Format

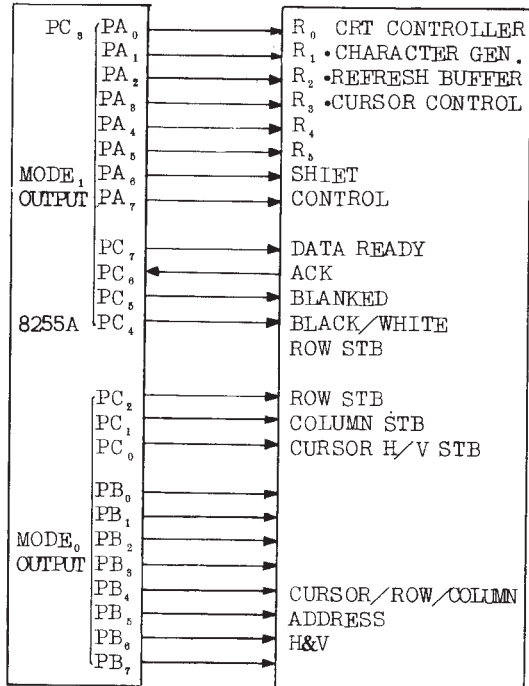
D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
OBFA	INTE <sub>1</sub>	IBFA	INTE <sub>1</sub>	INTR <sub>A</sub>	INTE <sub>2</sub>	IBFB	INTR <sub>B</sub>
GROUP A				GROUP B			

(DEFINED BY MODE 0 OR MODE 1 SELECTION)

Mode 2 Status Word Format

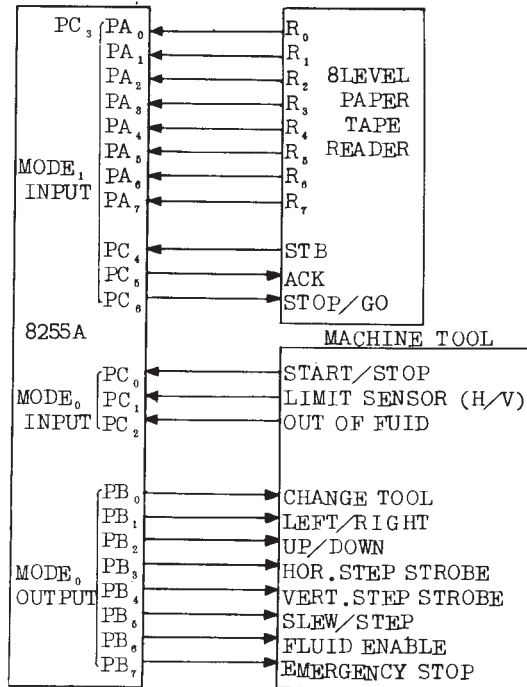


INTERRUPT  
REQUEST



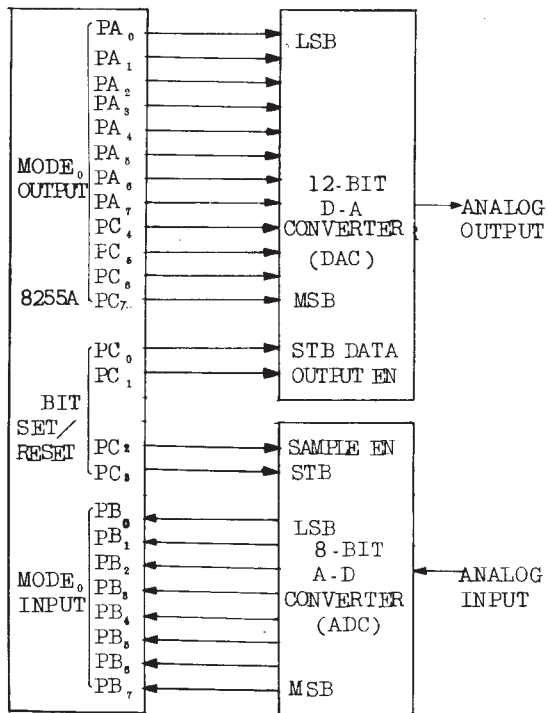
Basic CRT Controller Interface

INTERRUPT  
REQUEST

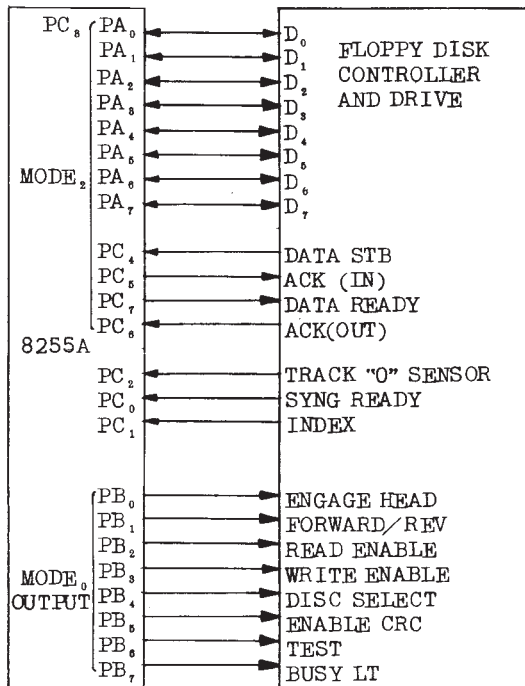


Machine Tool Controller Interface

INTERRUPT  
REQUEST



Digital to Analog to Digital



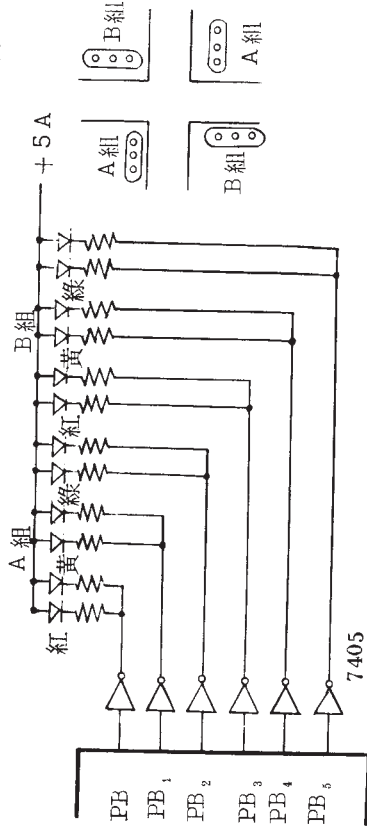
Basic Floppy Disc Interface

# 實習24 紅綠燈

功能：

1. 可程式化紅綠燈。
2. 可任意修改時間。

本程式乃是模擬真正紅綠燈之動作，假設有個十字路口，有A B 兩組紅綠燈，每組之紅黃綠三燈的時間各自獨立。黃燈閃爍次數可調整。由於A組的綠燈時間為B組之紅燈時間，故調整A組綠燈之時間，實際上亦調整了B組紅燈之時間，反之亦然。



程式範例：

1000	3E 90	LD	A, 90H	
1002	D3 83	OUT	(CWR), A	
1004	3E 0C	START	LD	A, 0CH
1006	D3 81	OUT	(PB), A	A組綠燈亮
1008	0E 07	LD	C, 07H	
100A	CD 49 10	CALL	DELAY	
100D	06 03	LD	B, 03H	
100F	3E 0A	LOOP1	LD	A, 0AH
1011	D3 81	OUT	(PB), A	A組黃燈亮

1013	0E 01	LD	C, 01H	
1015	CD 49 10	CALL	DELAY	
1018	3E 08	LD	A, 08H	A組黃燈閃爍
101A	D3 81	OUT	(PB), A	
101C	0E 01	LD	C, 01H	
101E	CD 49 10	CALL	DELAY	
1021	05	DEC	B	
1022	C2 0F 10	JP	NZ, LOOP1	
1025	3E 21	LD	A, 21H	B組綠燈亮
1027	D3 81	OUT	(PB), A	
1029	0E 07	LD	C, 07H	
102B	CD 49 10	CALL	DELAY	
102E	06 03	LD	B, 03H	
1030	3E 11	LD	A, 11H	B組黃燈亮
1032	D3 81	OUT	(PB), A	
1034	0E 01	LD	C, 01H	
1036	CD 49 10	CALL	DELAY	
1039	3E 01	LD	A, 01H	
103B	D3 81	OUT	(PB), A	
103D	0E 01	LD	C, 01H	
103F	CD 49 10	CALL	DELAY	
1042	05	DEC	B	
1043	C2 30 10	JP	NZ, LOOP2	
1046	C3 04 10	JP	START	
1049	11 00 80	LD	DE, 8000H	
104C	1B	DEC	DE	
104D	7A	LD	A, D	
104E	B3	OR	E	
104F	C2 4C 10	JP	NZ, WAT	
1052	0D	DEC	C	
1053	C2 49 10	JP	NZ, DELAY	
1056	C9	RET		

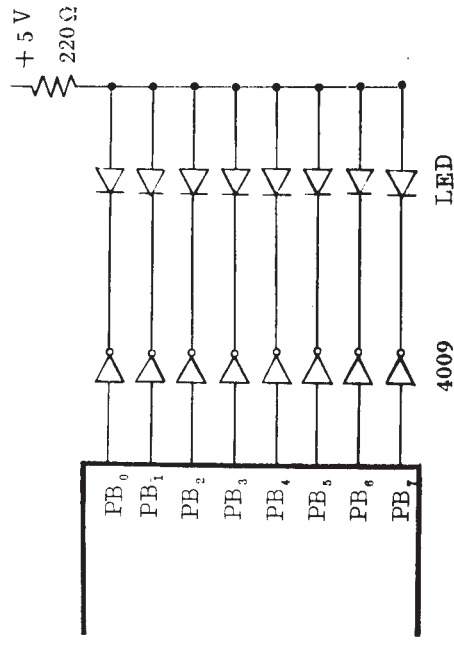
## 實習25 燈光順序控制

燈光順序控制，乃利用PB<sub>0</sub>~PB<sub>7</sub>之輸出，使LED循環順序發亮，程式執行後，8個LED每次只有一個會發亮，經過一段預設之延遲時間後，接著下一個LED發亮。(程式利用HL與DE偶對內容相加，如不為零時，重覆再相加而達到延遲的功能。

```

1000 3E 90      LD  A,90H
1002 D3 83      OUT (CWR),A
1004 3E FE      LD  A,FEH
1006 D3 81      OUT (PB),A
1008 47         NEXT
1009 21 00 00   LD  B,A
100C 11 01 00   LD  HL,0000H
100F ED 5A      LD  DE,0001H
1011 C2 0F 10   JP  NZ,LOOP
1014 78         LD  A,B
1015 07         RLCA
1016 C3 06 10   JP  NEXT

```



## 實習26 圖案

利用D/A轉換使ABC-80在示波器上顯示圖案。

程式範例：

```

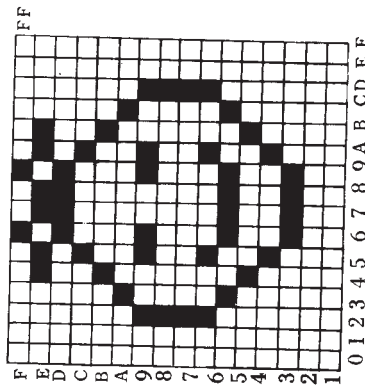
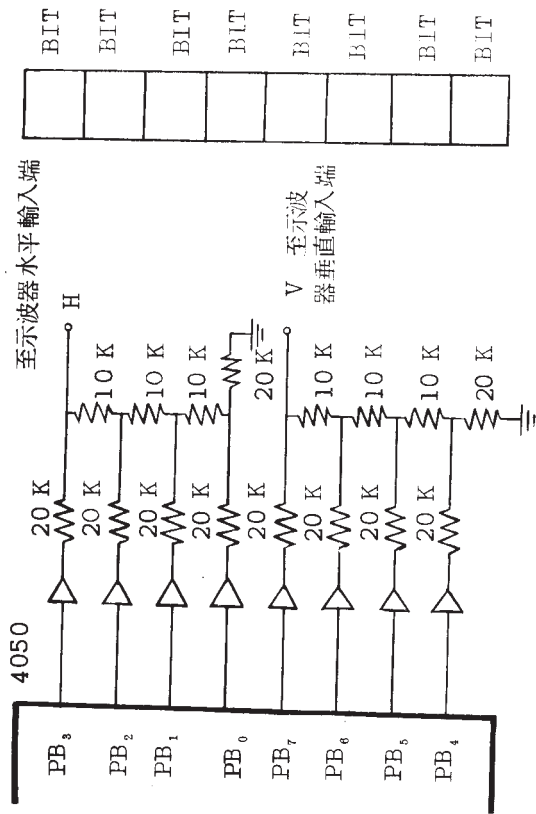
1000 3E 90      LD  A,90H
1002 D3 83      OUT (CWR),A
1004 21 40 10   LD  HL,1040H
1007 7E         DISP
1008 FE 00      CP  00H
100A CA 04 10   JP  Z,BEGIN
100D D3 81      OUT (PB),A
100F CD 16 10   CALL DELAY
1012 23         INC HL
1013 C3 07 10   JP  DISP
1016 0E FF      DELAY
1018 0D         LOOP
1019 C2 18 10   JP  NZ,LOOP
101C C9         RET

```

```

1040 92 82 72 62 A3 53 B4 44 C5 95 85 75 65 35 A6 56
1050 26 D7 27 D8 28 D9 A9 99 69 59 29 CA 3A BB 4B AC
1060 5C 9D 8D 7D 6D 00

```



將圖案用 16x16 的方格紙  
畫好再加以編碼 ( 1 BYTE 的  
上 4 BH 為垂直位置, 下 4BIT  
為水平位置 )。

在程式裏 HL 內放置顯示 DATA 的開始地址, 利用資料 00  
表示 DAT 顯示結束, DELAY 利用 C 暫存器構成, 內部存放之  
數值可構成延時之長短。如果不加以延時, 在示波器的 CRT 上  
將會產生“拖線”的現象, 但延時太長將會因低於視覺暫留時間  
而發生閃爍現象。

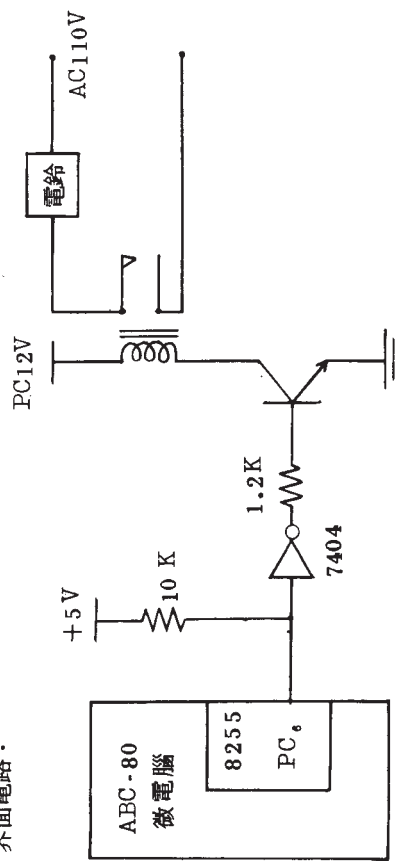
## 實習 27 校 鐘

本實驗是按一般上課時間, 早上 8 點以後每正點敲下課鐘,  
10 分敲上課鐘。本實驗僅是供參考, 其目的在表示微電腦之控制  
功能, 程式範例僅適合 ABC-80 使用。

說明:

1. 本實驗是利用時鐘程式為基礎, 程式中插入判別副程式,  
鐘聲副程式, 早上八點鐘以後每正點或 10 分各發出模擬聲  
響一次, 由 ABC-80 微電腦喇叭發出。
2. 配合電鈴使用敲鐘時設置 8255 之 PC6 輸出 LOW (通常經  
10K 電阻提昇為 Hi)。PC6 經 7404 反相可控制電鈴開  
關, 參考界面電路。
3. 平常顯示幕顯示時間。

界面電路:



程式範例：

1000	F3		DI		104C	10 F9		DJNZ	PUTBF
1001	DD 21 03 12		LD	IX,OUTBF	104E	2B		DEC	HL
1005	06 64	CLOCK	LD	B,64H	104F	2B		DEC	HL
1007	CD CD 04		CALL	SCD&K1	1050	CB F6		SET	6,(HL)
100A	10 FB		DJNZ	LOOP1	1052	2B		DEC	HL
100C	CD 27 10		CALL	TMUPD	1053	2B		DEC	HL
100F	CD 3F 10		CALL	BFUPD	1054	CB F6		SET	6,(HL)
1012	00	LOOP2	NOP		1056	C9		RET	
1013	10 FD		DJNZ	LOOP2	1057	60		DB	
1015	CD 60 10		CALL	TIMET	1058	60		DB	
1018	18 EB		JR	CLOCK	1059	24		DB	
1027	21 57 10	TMUPD	LD	HL,1057	1060	21 00 12	TIMET	LD	HL,1200H
102A	11 00 12		LD	DE,1200	1063	7E		LD	A,(HL)
102D	06 03		LD	B,03H	1064	FE 00		CP	A,00H
102F	37		SCF		1066	20 0A		JR	NZ,RETURN
1030	1A	TMINC	LD	A,(DE)	1068	23		INC	HL
1031	CE 00		ADC	A,00H	1069	7E		LD	A,(HL)
1033	27		DAA		106A	FE 00		CP	A,00H
1034	12		LD	(DE),A	106C	28 05		JR	Z,BALLT
1035	96		SUB	(HL)	106E	FE 10		CP	A,10H
1036	38 01		JR	C,COMPL	1070	28 01		JR	Z,BALLT
1038	12		LD	(DE),A	1072	C9	BALLT	RET	
1039	3F	COMP	CCF		1073	23		INC	HL
103A	23		INC	HL	1074	7E		LD	A,(HL)
103B	13		INC	DE	1075	D6 08		SUB	08H
103C	10 F2		DJNZ	TMINC	1077	D4 7B 10		CALL	NC,BALL
103E	C9		RET		107A	C9		RET	
103F	21 03 12	BFUPD	LD	HL,OUTBF	107B	3E 3F	BALL	LD	A,3FH
1042	11 00 12		LD	DE,1200H	107D	D3 82		OUT	(PC),A
1045	06 03		LD	B,03H	107F	D9		EXX	
1047	1A		LD	A,(DE)	1080	0E 65		LD	C,65H
1048	CD 25 05		CALL	2BYSEG	1082	21 00 04		LD	HL,0400H
104B	13		INC	DE	1085	CD 7C 04		CALL	SOUND
					1088	0E 80		LD	C,80H



108A	21 AA 03	LD	HL,03AAH
108D	CD 7C 04	CALL	SOUND
1090	0E 72	LD	C,72H
1092	21 AA 03	LD	HL,03AAH
1095	CD 7C 04	CALL	SOUND
1098	0E AD	LD	C,ADH
109A	21 AA 03	LD	HL,03AAH
109D	CD 7C 04	CALL	SOUND
10A0	06 FF	LD	B,FF
10A2	0E FF	LD	C,FF
10A4	0D	DEC	C
10A5	20 FD	JR	NZ,LOOP
10A7	10 F9	DJNZ	DELAY
10A9	0E AD	LD	C,ADH
10AB	21 AA 02	LD	HL,02AAH
10AE	CD 7C 04	CALL	SOUND
10B1	0E 72	LD	C,72H
10B3	21 00 04	LD	HL,0400H
10B6	CD 7C 04	CALL	SOUND
10B9	0E 65	LD	C,65H
10BB	21 00 04	LD	HL,0400H
10BE	CD 7C 04	CALL	SOUND
10C1	0E 80	LD	C,80H
10C3	21 AA 04	LD	HL,04AAH
10C6	CD 7C 04	CALL	SOUND
10C9	D9	EXX	
10CA	3E FF	LD	A,FFH
10CC	D3 82	OUT	(PC),A
10CE	3A 00 12	LD	A,(1200H)
10D1	C6 15	ADD	A,15H
10D3	32 00 12	LD	(1200H),A
10D6	C9	RET	

## 實習28 家庭電器電腦化控制

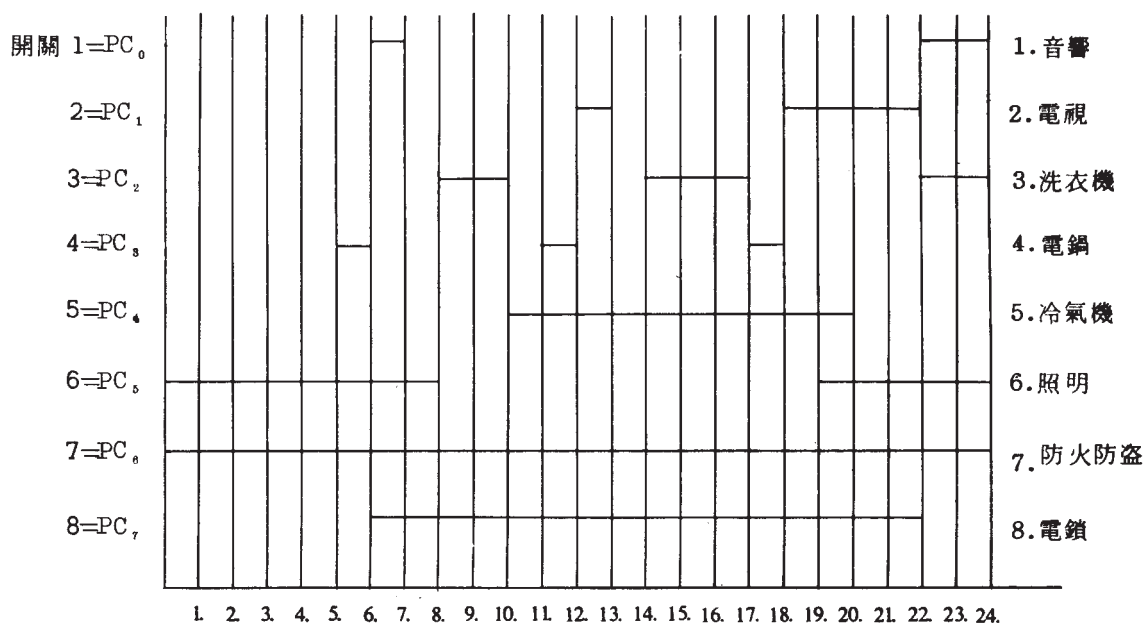
利用微電腦來控制家庭所有電器之電源，視其各種電器使用性質，經過微電腦控制，在其允許使用時間內，才供給電源；非使用時間內其電源被關掉，咱們可隨本身生活習慣，設定各種電器使用時間，使家庭步入現代化、自動化，節省人力，減少浪費，還可防止意外，提高生活品質。

說明：

1. 本實驗利用 ABC-80 微電腦，再擴充一個 8255。其 CS 接於 74 CS 139 的第 11 腳，定址位置為 40 ~ 43。參考第三篇 4 - 6 節輸入／輸出。
2. 8255 有 24 個 I / O 線，本例題只利用 PORTC 8 個，I / O 控制 8 個開關，PC<sub>0</sub> ~ PC<sub>7</sub> 分別控制開關 1 ~ 8。讀者可視需要再增加到控制 24 個開關，只需修改“ESA-VE”副程式即可。
3. 隨個人生活習慣之不同，可改變各開關之 ON、OFF 時間，其方法是改變 1100 ~ 1124 位置之內容每一數元控制一個開關，當數元輸出 H1 表示開關 ON，反之輸出 LOW 表示開關 OFF 如 60 = 01100000 表示第 6、7 開關 ON，其餘 OFF。
4. 讀者可自己設定每個開關之用途。
5. 設定 8 個開關，每個開關之動作情形如表所示，水平表時間以小時為單位，垂直表開關、劃直綫部分代表開關 ON 的時間，沒劃綫部分代表 OFF。視其開關所控制之電器用品之不同，其使用時間也不一樣。

[illegible]

FROM  
LS 139 第11脚



1000	3E 80	LD	A, 80H	1047	1A	PUTBF	LD	A, (DE)
1002	D3 43	OUT	(CWR), A	1048	CD 25 05		CALL	2BYTEG
1004	CB 27	SIA	A	104B	13		INC	DE
1006	D3 42	OUT	(PC), A	104C	10 F9		DJNZ	PUTBF
1008	DD 21 03 12	LD	IX, OUTBF	104E	2B		DEC	HL
100C	06 64	LD	B, 64H	104F	2B		DEC	HL
100E	CD CD 04	CALL	SCD&K1	1050	CB F6		SET	6, (HL)
1011	10 FB	DJNZ	LOOP1	1052	2B		DEC	HL
1013	CD 27 10	CALL	TMUPD	1053	2B		DEC	HL
1016	CD 3F 10	CALL	BFUPD	1054	CB F6		SET	6, (HL)
1019	00	NOP		1056	C9		RET	
101A	10 FD	DJNZ	LOOP2	1057	60		DB	
101C	CD 60 10	CALL	TIMET	1058	60		DB	
101F	18 EB	JR	CLOCK	1059	24		DB	
1027	21 57 10	LD	HL, 1057H					
102A	11 00 12	LD	DE, 1200H	1060	21 00 12	TIMET	LD	HL, 1200H
102D	06 03	LD	B, 03H	1063	7E		LD	A, (HL)
102F	37	SCF		1064	FE 00		CP	A, 00H
1030	1A	LD	A, (DE)	1066	20 06		JR	NZ, RETUR
1031	CE 00	ADC	A, 00H	1068	23		INC	HL
1033	27	DAA		1069	7E		LD	A, (HL)
1034	12	LD	(DE), A	106A	FE 00		CP	A, 00H
1035	96	SUB	(HL)	106C	28 01		JR	Z, ESAVE
1036	38 01	JR	C, COMPL	106E	C9		RET	
1038	12	LD	(DE), A	106F	00		NOP	
1039	3F	CCF		1070	23	ESAVE	INC	HL
103A	23	INC	HL	1071	5E		LD	E, (HL)
103B	13	INC	DE	1072	16 11		LD	D, 11H
103C	10 F2	DJNZ	TMINC	1074	1A		LD	A, (DE)
103E	C9	RET		1075	D3 42		OUT	(PC), A
103F	21 03 12	LD	HL, OUTBF	1077	C9		RET	
1042	11 00 12	LD	DE, 1200H					
1045	06 03	LD	B, 03H					

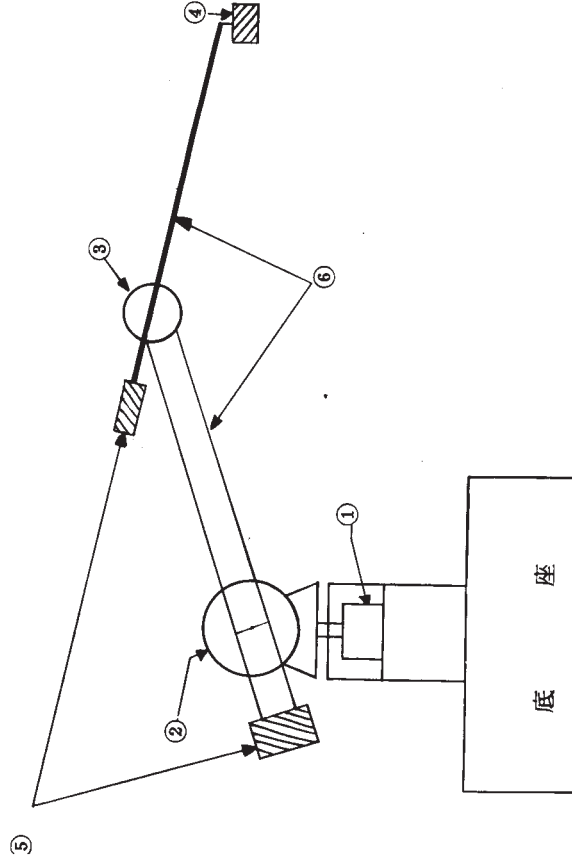
# 實習29 模擬機械手

DATA BYTE

```
1100 60 60 60 60 60
1105 68 E1 E0 C4 C4
1110 D0 D8 D2 D0 D4
1115 D4 D4 D8 D2 F2
1120 E2 E2 65 65 60
```

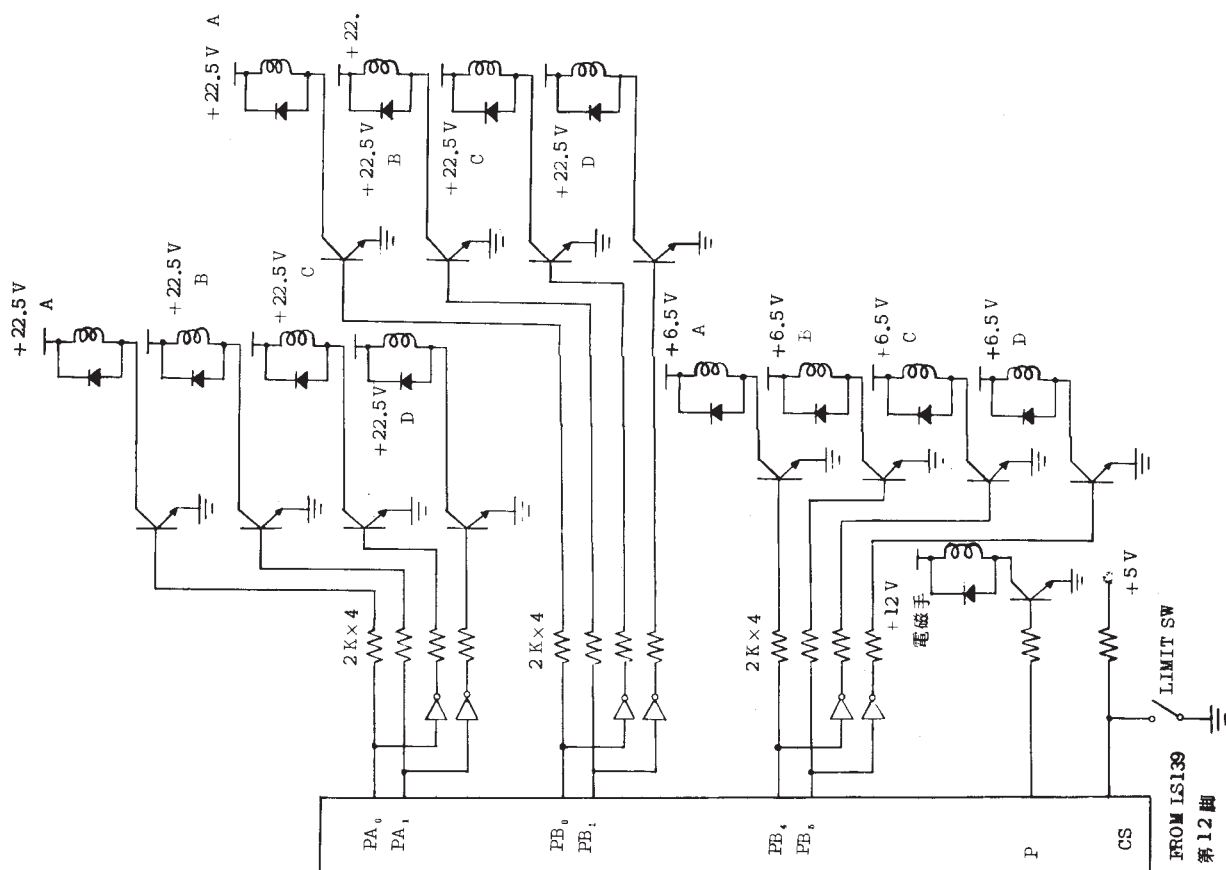
本實驗是利用 ABC-80 微電腦，再擴充一個 8255 來控制三個進步馬達，由進步馬達直接驅動機械手臂，其目的在表示微電腦之控制功能。本例並不代表所有 STEP MOTOP 之控制技術。

參考結構圖：



圖內標示說明：

1. 圖中①為步進馬達 (DC 22.5V, 0.61A, 200 OZ. in), 可做 360° 旋轉, 使手臂能在有效半徑內靈活轉動。
2. 圖中③為步進馬達 (DC 22.5, 0.61A, 200 OZ. in), 通常以 300° 旋轉為限配合①號步進馬達, 已能對四



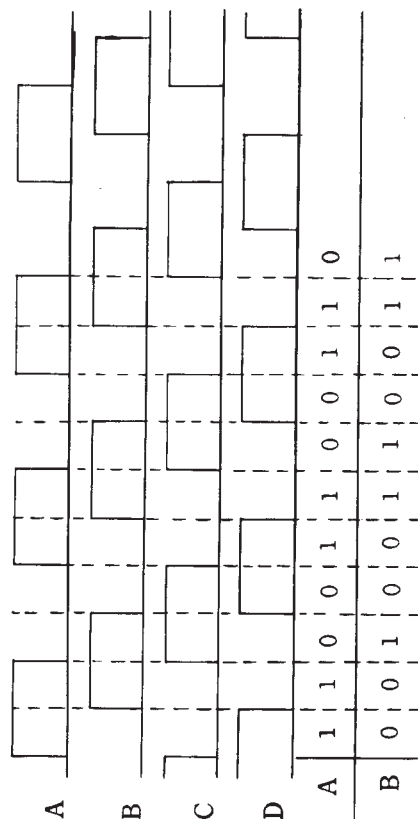
面八方靈活動作。

3. ⑤為步進馬達，可做 360° 旋轉，配合②的步進馬達，對於上、下、遠、近的操作能更為靈活。
4. ④為電磁手 利用電磁力能吸起磁性物質。
5. ⑤為平衡錘，使手臂保持平衡，減輕馬達負擔。
6. ⑥為手臂，其長短視實際需要而定。

界面電路如圖 2：

工作原理：

1. 利用微電腦透過程式設計技巧，送出推動步進馬達時序脈波，經電晶體放大後推動步進馬達線圈組，使步進馬達做順時針方向旋轉，或反時針方向旋轉。其時序如下圖。



※C為A之反相，D為B之反相，A B之間之真值表如下：

※若B為第一位元，A為第0位元，將上表改成16進制。

0 1 3 2 0 1 3 2 0 1 3 2 .....

※若欲將步進馬達反轉，則將時序依相反方向輸入

0 2 3 1 0 2 3 1 0 2 3 1 .....

2. 擴充 8255 之 CS 是接於 74LS139 第 12 脚，因此其定址位置爲 00 01 02 03 ( 參考第三篇 4 - 6 節輸入／輸出 )。

程式範例：

1000	3E 80	LD	A, 80H
1002	D3 03	OUT	(03H), A
1004	3E 00	LD	A, (00H)
1006	D3 00	OUT	(00H), A
1008	D3 01	OUT	(01H), A
100A	D3 02	OUT	(02H), A
100C	21 67 11	LD	HL, NN
100F	3E 00	LD	A, FORDB1
1011	BE	CP	(HL)
1012	CC 49 10	CALL	Z, ONEFOR
1015	3E 01	LD	A, BAKDB1
1017	BE	CP	(HL)
1018	CC 6B 10	CALL	Z, ONEBAK
101B	3E 02	LD	A, FORDB2
101D	BE	CP	(HL)
101E	CC 8E 10	CALL	Z, TWOFOR
1021	3E 03	LD	A, BAKDB2
1023	BE	CP	(HL)
1024	CC B9 10	CALL	Z, TWOBAK
1027	3E 04	LD	A, FORDB3
1029	BE	CP	(HL)
102A	CC E4 10	CALL	Z, THFOR
102D	3E 05	LD	A, BAKDB3
102F	BE	CP	(HL)
1030	CC 18 11	CALL	Z, THRAK
1033	3E 06	LD	A, UPDB
1035	BE	CP	(HL)
1036	CC 4C 11	CALL	Z, UP

1039	3E 07	LD	A, DOWNDB
103B	BE	CP	(HL)
103C	CC 55 11	CALL	Z, DOWN
103F	3E 08	LD	A, STOPDB
1041	BE	CP	(HL)
1042	28 03	JR	Z, STOP
1044	C3 0F 10	JP	LOOP
1047	76	HALT	
1048	00	DEFB	NN
1049	23	INC	HL
104A	46	LD	B, (HL)
104B	3A 48 10	LD	A, (nn)
104E	FE 00	CP	00H
1050	28 0A	JR	Z, G01
1052	FE 01	CP	01H
1054	28 05	JR	Z, RUN1
1056	FE 02	CP	02H
1058	28 01	JR	Z, RUN1
105A	3C	INC	A
105B	3C	INC	A
105C	3C	INC	A
105D	E6 03	AND	03H
105F	32 48 10	LD	(nn), A
1062	D3 00	OUT	(00H), A
1064	CD 5E 11	CALL	DELAY
1067	10 E2	DJNZ	LOOP1
1069	23	INC	HL
106A	C9	RET	
106B	23	INC	HL
106C	46	LD	B, (HL)
106D	3A 48 10	LD	A, (nn)
1070	FE 00	CP	00H
1072	28 09	JR	Z, RUN2



Address	Op Code	Op	Op 2	Op 3	Op 4	Op 5	Op 6	Op 7	Op 8	Op 9	Op 10	Op 11	Op 12	Op 13	Op 14	Op 15	Op 16	Op 17	Op 18	Op 19	Op 20	Op 21	Op 22	Op 23	Op 24	Op 25	Op 26	Op 27	Op 28	Op 29	Op 30	Op 31	Op 32	Op 33	Op 34	Op 35	Op 36	Op 37	Op 38	Op 39	Op 40	Op 41	Op 42	Op 43	Op 44	Op 45	Op 46	Op 47	Op 48	Op 49	Op 50	Op 51	Op 52	Op 53	Op 54	Op 55	Op 56	Op 57	Op 58	Op 59	Op 60	Op 61	Op 62	Op 63	Op 64	Op 65	Op 66	Op 67	Op 68	Op 69	Op 70	Op 71	Op 72	Op 73	Op 74	Op 75	Op 76	Op 77	Op 78	Op 79	Op 80	Op 81	Op 82	Op 83	Op 84	Op 85	Op 86	Op 87	Op 88	Op 89	Op 90	Op 91	Op 92	Op 93	Op 94	Op 95	Op 96	Op 97	Op 98	Op 99	Op 100	Op 101	Op 102	Op 103	Op 104	Op 105	Op 106	Op 107	Op 108	Op 109	Op 110	Op 111	Op 112	Op 113	Op 114	Op 115	Op 116	Op 117	Op 118	Op 119	Op 120	Op 121	Op 122	Op 123	Op 124	Op 125	Op 126	Op 127	Op 128	Op 129	Op 130	Op 131	Op 132	Op 133	Op 134	Op 135	Op 136	Op 137	Op 138	Op 139	Op 140	Op 141	Op 142	Op 143	Op 144	Op 145	Op 146	Op 147	Op 148	Op 149	Op 150	Op 151	Op 152	Op 153	Op 154	Op 155	Op 156	Op 157	Op 158	Op 159	Op 160	Op 161	Op 162	Op 163	Op 164	Op 165	Op 166	Op 167	Op 168	Op 169	Op 170	Op 171	Op 172	Op 173	Op 174	Op 175	Op 176	Op 177	Op 178	Op 179	Op 180	Op 181	Op 182	Op 183	Op 184	Op 185	Op 186	Op 187	Op 188	Op 189	Op 190	Op 191	Op 192	Op 193	Op 194	Op 195	Op 196	Op 197	Op 198	Op 199	Op 200	Op 201	Op 202	Op 203	Op 204	Op 205	Op 206	Op 207	Op 208	Op 209	Op 210	Op 211	Op 212	Op 213	Op 214	Op 215	Op 216	Op 217	Op 218	Op 219	Op 220	Op 221	Op 222	Op 223	Op 224	Op 225	Op 226	Op 227	Op 228	Op 229	Op 230	Op 231	Op 232	Op 233	Op 234	Op 235	Op 236	Op 237	Op 238	Op 239	Op 240	Op 241	Op 242	Op 243	Op 244	Op 245	Op 246	Op 247	Op 248	Op 249	Op 250	Op 251	Op 252	Op 253	Op 254	Op 255	Op 256	Op 257	Op 258	Op 259	Op 260	Op 261	Op 262	Op 263	Op 264	Op 265	Op 266	Op 267	Op 268	Op 269	Op 270	Op 271	Op 272	Op 273	Op 274	Op 275	Op 276	Op 277	Op 278	Op 279	Op 280	Op 281	Op 282	Op 283	Op 284	Op 285	Op 286	Op 287	Op 288	Op 289	Op 290	Op 291	Op 292	Op 293	Op 294	Op 295	Op 296	Op 297	Op 298	Op 299	Op 300	Op 301	Op 302	Op 303	Op 304	Op 305	Op 306	Op 307	Op 308	Op 309	Op 310	Op 311	Op 312	Op 313	Op 314	Op 315	Op 316	Op 317	Op 318	Op 319	Op 320	Op 321	Op 322	Op 323	Op 324	Op 325	Op 326	Op 327	Op 328	Op 329	Op 330	Op 331	Op 332	Op 333	Op 334	Op 335	Op 336	Op 337	Op 338	Op 339	Op 340	Op 341	Op 342	Op 343	Op 344	Op 345	Op 346	Op 347	Op 348	Op 349	Op 350	Op 351	Op 352	Op 353	Op 354	Op 355	Op 356	Op 357	Op 358	Op 359	Op 360	Op 361	Op 362	Op 363	Op 364	Op 365	Op 366	Op 367	Op 368	Op 369	Op 370	Op 371	Op 372	Op 373	Op 374	Op 375	Op 376	Op 377	Op 378	Op 379	Op 380	Op 381	Op 382	Op 383	Op 384	Op 385	Op 386	Op 387	Op 388	Op 389	Op 390	Op 391	Op 392	Op 393	Op 394	Op 395	Op 396	Op 397	Op 398	Op 399	Op 400	Op 401	Op 402	Op 403	Op 404	Op 405	Op 406	Op 407	Op 408	Op 409	Op 410	Op 411	Op 412	Op 413	Op 414	Op 415	Op 416	Op 417	Op 418
---------	---------	----	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

10E9	E6 03	AND	03H	1129	3E 20		LD	A, 20H
10EB	4F	LD	C, A	112B	18 12		JR	OPER2
10EC	3A 8D 10	LD	A, (PORTB)	112D	FE 20	NEXT2	CP	20H
10EF	E6 30	AND	30H	112F	20 04		JR	NZ, G06
10F1	FE 00	CP	00H	1131	3E 30		LD	A, 30H
10F3	20 04	JR	NZ, NEXT1	1133	18 0A		JR	OPER2
10F5	3E 10	LD	A, 10H	1135	FE 30	G06	CP	30H
10F7	18 12	JR	OPER1	1137	20 04		JR	NZ, RUN6
10F9	FE 10	CP	10H	1139	3E 10		LD	A, 10H
10FB	20 04	JR	NZ, G05	113B	18 02		JR	OPER2
10FD	3E 30	LD	A, 30H	113D	3E 00	RUN6	LP	A, 00H
10FF	18 0A	JR	OPER1	113F	B1	OPER2	OR	C
1101	FE 20	CP	20H	1140	32 8D 10		LD	(PORTB), A
1103	20 04	JR	NZ, RUN5	1143	D3 01		OUT	(01H), A
1105	3E 00	LD	A, 00H	1145	CD 5E 11		CALL	DELAY
1107	18 02	JR	OPER1	1148	10 D0		DJNZ	LOOP6
1109	3E 20	LD	A, 20H	114A	23		INC	HL
110B	B1	OR	C	114B	C9	UP	RET	
110C	32 8D 10	LD	(PORTB), A	114C	3E 01		LD	A, 01H
110F	D3 01	OUT	(01H), A	114E	D3 02		OUT	(02H), A
1111	CD 5E 11	CALL	DELAY	1150	CD 5E 11		CALL	DELAY
1114	10 D0	DJNZ	LOOPS	1153	23		INC	HL
1116	23	INC	HL	1154	C9		RET	
1117	C9	RET		1155	3E 00	DOWN	LD	A, 00H
1118	23	INC	HL	1157	D3 02		OUT	(02H), A
1119	46	LD	B, (HL)	1159	CD 5E 11		LALL	DELAY
111A	3A 8D 10	LD	A, (PORTB)	115C	23		INC	HL
111D	E6 03	AND	03H	115D	C9		RET	
111F	4F	LD	C, A	115E	11 00 40	DELAY	LD	DE, 4000H
1120	3A 8D 10	LD	A, (PORTB)	1161	1B	WAI	DEC	DE
1123	E6 30	AND	30H	1162	7A		LD	A, D
1125	FE 00	CP	00H	1163	B3		OR	E
1127	20 04	JR	NZ, NEXT2	1164	20 FB		JR	NZ, WAT
				1166	C9		RET	

1167	00	選擇 1 號步進馬達正轉
1168	0A	; 1 號步進馬達正轉 10 步
1169	02	選擇 ( 令 ) 2 號步進馬達正轉
116A	10	; 2 號步進馬達正轉 16 步
116B	05	; 令 3 號步進馬達逆轉
116C	08	; 3 號步進馬達逆轉 8 步
116D	06	; 令電磁鐵吸起東西
116E	04	; 令 3 號步進馬達正轉
116F	0B	; 3 號步進馬達正轉 11 步
1170	01	; 令 1 號步進馬達逆轉
1171	0E	; 1 號步進馬達逆轉 14 步
1172	07	; 令電磁鐵放下東西
1173	08	; 整個動作完畢

文件名稱：**Z80 指令集詳細說明掃描版**

文件分類	I
文件編號	00027
文件批號	00

製作群	原稿掃描	文稿編輯
	原稿圖文分離	文稿整合
	原稿辨識	文稿校對
	文稿成品輸出	特別感謝名單

文件完成日期	初版	2006-11-23	其他加註
	再版		

文件出處	原圖書名	Z80 微電腦製作
	原圖書者	劉秋良
	原圖書者	欣大出版社
	原圖書日	民國 73 年 02 月三版

## DDSC 文件 版權宣告

本文件版權屬原輸出公司、出版社、圖書公司或原著作人所有，作商業用途者請自行洽上述公司，本文件僅可在非商業上流傳或供私人收集資料用。另由於資料老舊，DDSC 不對原書內的內容負責，且除了更正原書內的錯字、漏字之外一切照原書內容所用的文字顯示。

*Documents Digitize Service Center* 製作  
1998-2006

### 檔名格式說明：

DDSC — 文件分類 — 文件編號 — 文件批號 — 文件名.PDF

以 DDSC 為起頭，加上 1 個字母為分類代碼，再加上以 5 位數由 00001 起的編號，加上 2 位數由 01 起的編號，加上完整的文件名稱而成的。

其中分類代碼詳見下面列表。文件批號指該文件為非合訂版的，可能因書的內容過多而分批完成的，此項可有可無。

文件分類代碼說明	
代 碼	說 明
A	小說／文學類文章類
B	娛樂類
C	天文類
D	科學類
E	古文明事物類
F	自然界類
G	古怪事物類
H	動／植物類
I	電子類
J	電腦類