

第 15 章

輸入／輸出界面電路

Z80 微處理器本身無法獨立作業。至少，其必須再配有記憶體，以儲存其所欲執行之程式指令與數據（若有的話）。實用上，一部微電腦必須與外界（其所控制之週遭環境或人類）取得溝通。因此，一部完整之微電腦系統除了應具備微處理器與記憶體外，尚須具有輸入／輸出設備。目前而言，由於運轉速度與資料格式上之差異，輸入／輸出設備通常無法直接與微處理器作線連接（即界面），而必須透過中間媒介—輸入／輸出界面電路。此乃為何一般微處理器之製造商會提供一族（family）晶片（微處理器和與其配屬之記憶晶片、輸入／輸出界面晶片的統稱），而非單單僅一個微處理器晶片之原故。

本章，我們之主要目的即在於介紹 Z80 族中之輸入／輸出界面晶片的結構、特性、與用法。

Zilog 公司提供之輸入／輸出界面晶片主要有三種：

- 1 Z80-PIO：並行輸入／輸出界面電路。
- 2 Z80-CTC：計數／計時器電路。
- 3 Z80-SIO：串行輸入／輸出界面電路。

Z80-PIO 之主要功能乃在於提供 Z80 微處理器與並行傳輸週邊設備間之界面。Z80-CTC 之主要功用則在於提供硬體之計時

與計數，此一功能對輸入／輸出作業相當重要。除了用於串行傳輸之週邊設備的界面外，Z80-SIO之角色類同於Z80-PIO，其主要功用即在於提供並行對串行與串行對並行之轉換。雖然名稱各有差異，但任何微處理器製造商所提供之輸入／輸出界面電路，均不外乎以上三種。

本章，我們將介紹Z80-PIO與Z80-CTC兩界面電路。

15-1 Z80-PIO

15-1-1 特性

PIO為Parallel Input / Output之簡寫，其意即並行輸入／輸出界面電路。Z80-PIO為一可規劃之**雙口**(port)(每一口為八條輸入／輸出線)元件，其提供Z80-CPU與並行傳輸週邊設備間之界面。CPU可配組Z80-PIO，使之與廣泛之週邊界面而不需加任何額外電路。能與Z80-PIO完全吻合之典型週邊設備包括大多數之鍵盤，讀／打紙帶機，印字機，與PROM寫程式機(Programmer)等。Z80-PIO之主要特色包括：

1. 有兩個獨立具有“握手連絡”資訊傳輸控制之八位元輸入／輸出口。
2. 使用插斷式之握手連絡，反應迅速。
3. 每一輸入／輸出口均有四種不同之作業型態：
 - 位元組輸出(即一次輸出一個位元組)
 - 位元組輸入(即一次輸入一個位元組)
 - 位元組雙向巴士(只有A口)
 - 位元控制

並且每一種均具有插斷控制之握手連絡。

4. 具有雛菊花環插斷優先順序電路，能作自動插斷指向，而不需外加任何電路。

5. 每一輸出均能推動達靈頓電晶體。
6. 所有輸入與輸出均完全與TTL吻合。
7. 僅需單一正5伏特電源與單相時序。

Z80-PIO所具備的特色之一，就是週邊設備與CPU間之所有資訊傳輸，均完全以插斷之方式控制。其插斷邏輯不僅能於I/O傳輸時有效地利用Z80-CPU之插斷能力，同時亦允許全然之巢串插斷。Z80-PIO之另一特色為，其可規劃成在當週邊設備之某一特定狀態出現時，才插斷CPU。例如，我們可使PIO在當週邊設備之“鬧鐘響”時才插斷CPU。此一插斷能力大量地削減了CPU必須花費在讀取各週邊設備狀態上之時間。

15-1-2 接腳說明

如圖15-1所示，Z80-PIO為一具有40支接腳之積體電路，其為Z80-CPU與接受八位元並行資料之週邊設備間，提供了簡單之界面。每一Z80-PIO具有兩個八位元之輸入／輸出口，分別稱為**A口**(port A)與**B口**(port B)。每一輸入／輸出口均可以程式指令，獨立將之設定成輸入口或輸出口。

Z80-PIO之40支接腳中，除了電源者外，可分為兩大類：一為與Z80-CPU相連接(如圖15-1之左邊所示)，另一為與外部週邊設備相連接(如15-1圖之右邊所示)。底下我們分別介紹這些接腳的功能。

與週邊設備之界面

除了兩組八位元之雙向資料線(A0-A7與B0-B7)外，Z80-PIO與外部輸入／輸出設備間，尚有兩組作**交互握手連絡**(handshaking)用之控制信號。它們是：

A口暫存器備好信號：A RDY

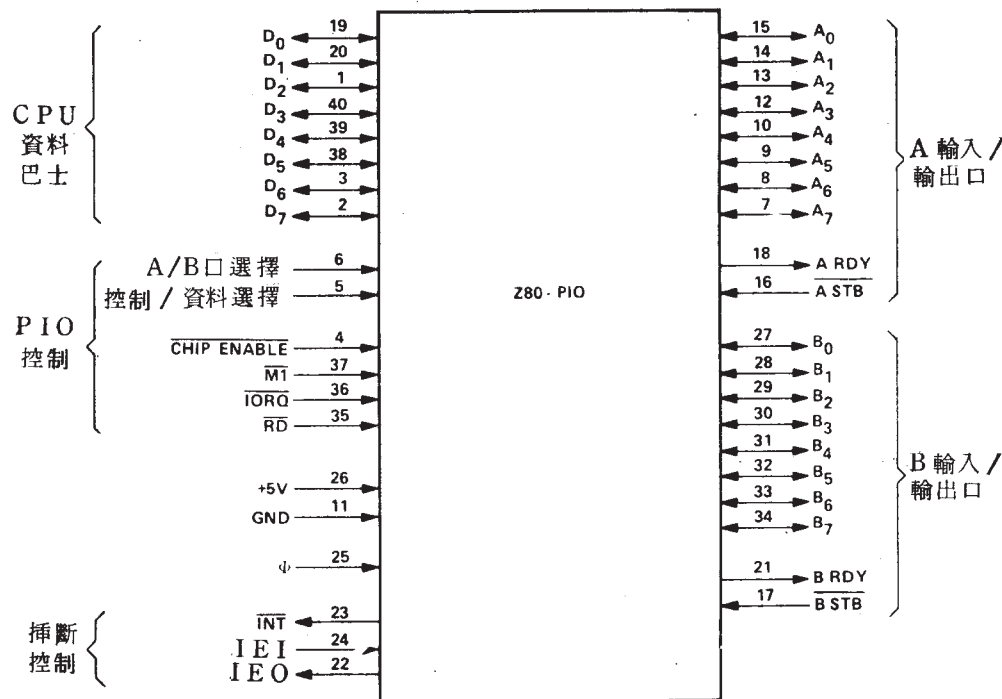


圖 15-1 Z80-PPIO 之外觀接腳圖

A 口擷取信號： $\overline{A \text{ STB}}$

B 口暫存器備好信號： $B \text{ RDY}$

B 口擷取信號： $\overline{B \text{ STB}}$

A 口暫存器備好信號 $A \text{ RDY}$ 為一高電位動作之輸出信號，其意義依 A 口之不同作業型態而異：

- 1 輸出型態時：此信號動作（變至高電位）代表 A 口之輸出暫存器已裝載資料，週邊資料巴士已處於穩定狀態，並可傳輸至外部週邊設備。
- 2 輸入型態時：此信號動作代表 A 口之輸入暫存器是空的（已無資料），並可接受來自週邊設備之資料。

3 雙向型態時：此信號動作代表資料已存至 A 口之輸出暫存器，並可送給週邊設備。

4 控制型態時：此信號失去作用，並被迫降為低電位。

A 口擷取信號 $\overline{A \text{ STB}}$ 為一低電位動作之輸入信號，其意義依 A 口作業型態之不同而異：

- 1 輸出型態時：此信號之正向緣代表週邊設備已收到 PIO 所提供之資料。
- 2 輸入型態時：週邊設備以此一信號將資料裝載入（load）A 口之輸入暫存器。此信號動作時，資料存入 PIO。
- 3 雙向型態時：此信號動作時，A 口輸出暫存器之資料，被送至 A 口之雙向資料巴士。此擷取信號之正向緣函覆資料已收到。
- 4 控制型態時：此信號自動失效。

對 B 口而言，B 口暫存器備好信號 $B \text{ RDY}$ 類似於 A 口之暫存器備好信號 $A \text{ RDY}$ 。唯一的的不同是，於 B 口之雙向型態，此信號於 B 口之輸入暫存器空閒，並可由週邊設備接受資料時，呈高電位。同樣地，B 口擷取信號 $\overline{B \text{ STB}}$ 之功能亦類似於 $\overline{A \text{ STB}}$ ，唯一的差別是，於 B 口之雙向型態，此信號將來自週邊設備之資料取入 B 口之輸入暫存器。

與 Z80-CPU 之界面

Z80-PPIO 與 Z80-CPU 之界面（連接）信號，除了時序信號 ϕ 外，共可分為三類：**資料巴士**、**控制信號**，與**插斷控制信號**。資料巴士線 D0-D7 是 Z80-PPIO 與 CPU 間資料傳輸的孔道。來自 Z80-CPU 之六條控制線，在程式主宰下控制了 Z80-PPIO 之作業。**A/B 口選擇線** $\overline{\text{PORT B/A SEL}}$ 用以從 A 口與 B 口中選擇一個，以作資訊傳輸。若該信號線為低電位，則選取 A 口；反之，選取 B 口。此一信號線通常直接連接至 Z80-CPU 之 A0 位址巴士線。

命令或資料選擇線 $\overline{\text{CONT}}/\text{DATA SEL}$ 用以定義 Z80-CPU 與 PIO 間所正在傳輸之資訊的型態。若 CPU 正寫入 PIO 時，此一控制線為**高電位**，則 B/A SEL 所選取之輸入／輸出，自動將 Z80 資料巴士之內含視為一**控制命令** (Command)。反之，若該控制線為**低電位**，則表 CPU 與 PIO 間所正在傳輸者為**資料**。此一功能經常由 CPU 之 A1 位址線擔當。

晶片致能信號 CHIP ENABLE 即為晶片選取信號。此一信號之低準位使 PIO 在寫入週期時能受取 CPU 輸出之控制命令或資料，或在讀取週期時將某項資料送給 CPU。晶片致能信號通常由 A 口與 B 口之四個輸入／輸出位址解碼而得。

第 1 機器週期信號 $\overline{\text{M1}}$ 用以作為控制數個 PIO 內部作業之同步脈衝。前面說過， $\overline{\text{M1}}$ 與 $\overline{\text{RD}}$ 信號同時動作代表 Z80-CPU 正由記憶體拿取指令。而 $\overline{\text{M1}}$ 與 $\overline{\text{IORQ}}$ 同時動作代表 CPU 正在認知插斷。對 Z80-PIO 而言， $\overline{\text{M1}}$ 信號有兩種功能：(1)、 $\overline{\text{M1}}$ 同步了 PIO 之插斷邏輯。(2)、當 $\overline{\text{M1}}$ 動作，而 $\overline{\text{RD}}$ 與 $\overline{\text{IORQ}}$ 皆不動作時，PIO 進入重置狀態。

輸入／輸出請求 $\overline{\text{IORQ}}$ 信號與 B/A 選取，C/D 選取， $\overline{\text{CE}}$ ，及 $\overline{\text{RD}}$ 等信號合用。此信號動作代表 Z80-CPU 所正在進行的是輸入／輸出作業。若 $\overline{\text{CE}}$ ， $\overline{\text{RD}}$ ，與 $\overline{\text{IORQ}}$ 同時動作，則表 Z80-CPU 正進行輸入（讀取）作業，CPU 自 B/A 選擇線所選取之輸入／輸出口讀取資料。反之，若 $\overline{\text{CE}}$ 與 $\overline{\text{IORQ}}$ 同時動作，而 $\overline{\text{RD}}$ 不動作，則表 Z80-CPU 正在進行輸出（寫入 PIO）作業。CPU 將一項控制或資料訊息寫入 C/D 信號所選取之 PIO 位置。此外，若 $\overline{\text{IORQ}}$ 與 $\overline{\text{M1}}$ 同時動作，則表 Z80-CPU 正在認知（接受）插斷，具最高優先順序之插斷口應自動將其插斷向量置於 CPU 之資料巴士上。

插斷控制信號 $\overline{\text{INT}}$ ， $\overline{\text{IEI}}$ ，與 $\overline{\text{IEO}}$ 在前面討論插斷時已介紹過。INT 線動作代表 Z80-PIO 正對 CPU 提出插斷請求。IEI 與

$\overline{\text{IEO}}$ 主要則用於雛菊花環式之插斷優先順序。 $\overline{\text{IEI}}$ 線輸入高電位表示 Z80-CPU 並未在服務任何其它具有較高優先之設備，此一 PIO 提出之插斷請求可以被接納。 $\overline{\text{IEO}}$ 則惟有當 $\overline{\text{IEI}}$ 輸入高電位，且 CPU 並未在服務此一 PIO 時，方會動作（輸出高電位）。此一信號動作象徵著所有其它較低優先之設備，開始有機會接受 Z80-CPU 之服務。

15-1-3 內部結構

Z80-PIO 之內部構造方塊圖示於圖 15-2。此一內部結構包括 Z80-CPU 之巴士界面，內部控制電路，插斷控制電路，A 輸入／輸出口電路，與 B 輸入／輸出口電路。其中，CPU 巴士界面電路使 PIO 能直接與 Z80-CPU 連接，而不需外加任何額外電路。當然，較大的系統可能需要外加一些位址解碼器與／或線推動器。內部控制電路則使 CPU 之資料巴士與週邊設備之界面（即 A 口與 B 口）取得同步。

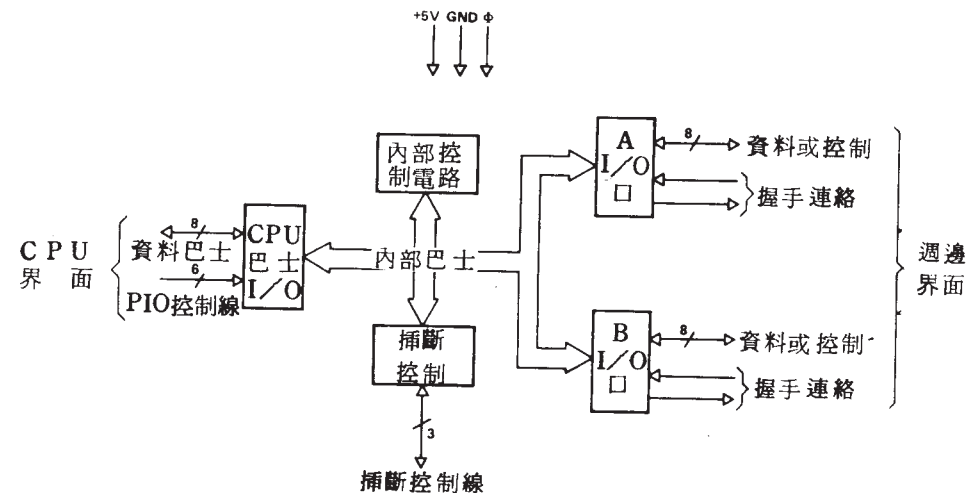


圖 15-2 Z80-PIO 內部結構方塊圖

插斷控制電路負責處理雛菊花環式之巢串插斷。於雛菊花環結構，每一PIO 具有兩條形成此一結構之控制線：插斷致能輸入 IEI 與插斷致能輸出 IEO。IEI 輸入取自前一PIO之 IEO 輸出，而 IEO 輸出則又連接至次一PIO之 IEI 輸入。任何週邊設備之優先順序視其在雛菊花環結構中之實際位置而定，最接近CPU 者具有最高優先。而於每一PIO內，A口又較B口具有更高優先。於位元組輸入、位元組輸出、或雙向型態，不論任何時刻，只要週邊設備要求另一新位元組之傳輸，插斷即可產生。於位元控制型態，插斷於週邊狀態與某一規劃值相吻合時產生。PIO 自動對巢串插斷作完全控制。亦即，CPU 在未服務完較高優先設備時，較低優先設備不能插斷。但較高優先之設備可插斷CPU 對較低優先設備之服務。

由於PIO 直接由CPU 之資料巴士解碼RETI（插斷回返）指令，致系統中每一PIO 隨時知道CPU 是否還在服務它，而不需再藉助任何其它之溝通。

輸入／輸出口之結構

圖 15-3 所示即為Z80-PIO兩輸入／輸出口之內部結構。由圖中可看出，每一輸入／輸出口主要由“握手連絡”控制電路與七個暫存器組成。此些暫存器包括：一八位元之資料輸入暫存器，一八位元之資料輸出暫存器，一八位元之輸入／輸出選擇暫存器，一八位元之面罩暫存器，一兩位元之面罩控制暫存器，一兩位元之型態控制暫存器，與一八位元之插斷向量位址暫存器。

週邊設備與Z80-CPU間之所有資訊傳輸皆經由輸入暫存器與輸出暫存器達成。CPU 可將欲輸出至週邊設備之資料寫入輸出暫存器，亦可自輸入暫存器讀取週邊設備所送來之輸入資料。兩位元之型態控制暫存器則用以選取每一輸入／輸出口之作業型態（位元組輸入，位元組輸出，雙向，或位元控制。）

八位元之面罩暫存器與八位元之輸入／輸出選擇暫存器僅用於位

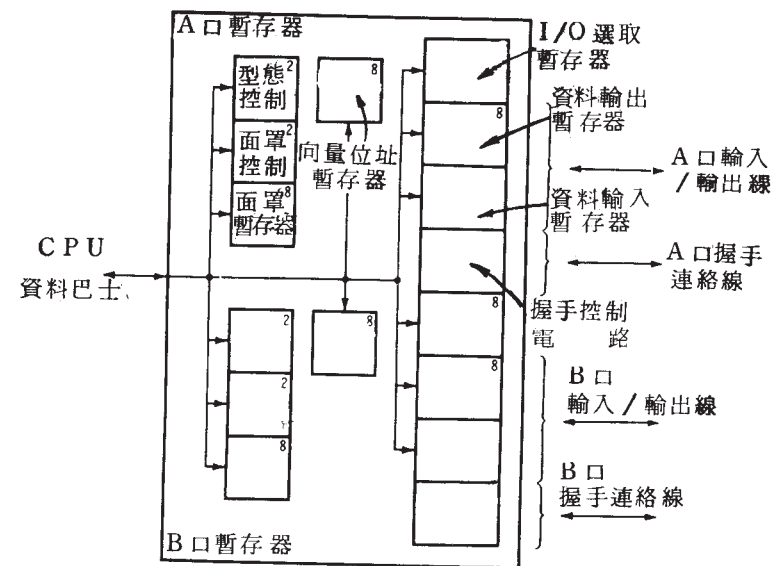


圖 15-3 Z80-PIO 輸入／輸出口之內部暫存器結構

元控制型態。此一型態，週邊資料或控制巴士之八支接腳的每一接腳，皆可個別將之令為輸入或輸出。此一指定由以程式指令將適當內含寫入八位元之選擇暫存器達成。選擇暫存器之每一位元對應於一接腳，若位元值為1，則其對應之接腳即為**輸入**；反之，若位元值為0，則其對應之接腳即為**輸出**。

八位元之面罩暫存器主要與一特殊之插斷特色有關。此一特色使得插斷惟有在任一或所有非罩蓋接腳抵達某一特定狀態（高或低電位）時才發生。兩位元之面罩控制暫存器負責指明此一特定狀態（高或低電位），以及插斷是於**任一**非罩蓋接腳動作（抵特定狀態）時發生，或當**全部**非罩蓋接腳皆動作時才發生。此一特色使週邊狀態抵達某一特定狀態時，插斷能自動發生。於Z80-PIO，面罩控制暫存器之兩位元的意義為：

高次位元 $\begin{cases} 1 : \text{全部 (AND)} \\ 0 : \text{任一 (OR)} \end{cases}$

低次位元 $\begin{cases} 1 : \text{高電位} \\ 0 : \text{低電位} \end{cases}$

當 Z80-CPU 於插斷型態 2 接受一插斷時，請求插斷之設備必須提供 CPU 一八位元之插斷向量。此一八位元向量與 CPU 之 I 暫存器的內含形成一十六位元指示器，該指示器指至設備插斷服務常式之起始位址所儲存的記憶位置。每一輸入／輸出口之向量位址暫存器，即用以儲存此一欲提供給 CPU 之八位元向量。由於八位元向量作為指示器之低次八位元，且指示器每次皆指至連續兩相鄰記憶位置，故 PIO 之八位元向量位址暫存器的最低次位元，自動設定為 0。

15-1-4 作業型態

第 1 節即提過，Z80-PIO 可有四種不同作業型態。此四種型態由 CPU 將適當資料寫入輸入／輸出口之控制暫存器的最高次兩位元（即兩位元之型態控制暫存器）後選取。此兩位元值所選取之作業型態如圖 15-4 所示。

7	6	5	4	3	2	1	0
型態選取	×	×	1	1	1	1	
0 0	型態 0 (位元組輸出)						
0 1	型態 1 (位元組輸入)						
1 0	型態 2 (雙向型態)						
1 1	型態 3 (位元控制)						

圖 15-4 Z80-PIO 作業型態選擇之控制字組

特別注意，型態數之符號具有某種實質意義：0 = Out (輸出)，1 = In (輸入)，2 = 雙向。

Z80-PIO 之 A 口可動作於型態 0, 1, 2, 3 等四種不同型態，而 B 口則僅能動作於型態 0, 1, 3 等三種型態。緊接，我們分別介紹此四種不同型態之作業情形。

PIO 型態 0

Z80-PIO 之型態 0 為位元組輸出型態。於此一型態，八位元之資料輸出暫存器（或稱鎖住器）動作，而輸入暫存器不動作。CPU 可以 OUT 指令將八位元之資料寫入位址線所選取之輸出口的輸出暫存器。此一資料亦可以 IN 指令將之重讀回 CPU，不過，通常我們並不這麼做，因為，程式通常已知道被寫出之資料是什麼。CPU 可於任何時刻執行另一 OUT 指令，將輸出暫存器之內含予以改變。

當資料由 CPU 寫入 PIO 時，該口之 RDY 信號立即變為高電位，以告知外部設備，資料已抵達（備好於）輸入／輸出口線上，可以讀取。外部週邊設備於讀取（拿走）此項資料後， $\overline{\text{STB}}$ 輸入線變為低電位，此一低電位信號令 RDY 線回復至低電位（告訴 PIO 輸出暫存器之資料已被讀走），並且產生一插斷（若 PIO 規劃成欲產生插斷的話）。

PIO 型態 1

Z80-PIO 之型態 1 乃位元組輸入型態。於此一型態，資料輸入暫存器動作，而資料輸出暫存器不動作，CPU 每次可由 PIO 讀取（輸入）一八位元之資料。由外部週邊設備輸入一八位元資料至 PIO 之動作系列為：

- 1 外部設備監聽 PIO 之輸出線 RDY。若該信號線動作，則外部設備即將八位元資料置於輸入／輸出口線，並使 $\overline{\text{STB}}$ 線瞬間變為低電位。

- 2 資料被 $\overline{\text{STB}}$ 信號鎖入口資料輸入暫存器。此一動作使 RDY 線回復至低電位（不動作狀態），並引起插斷（若 PIO 如此規劃）。
- 3 Z 80-CPU 送出輸入／輸出位址，並以 IN 指令自 PIO 讀取資料。
- 4 讀取動作使 RDY 線又變成高電位（動作狀態），通知外部設備，準備下一資料位元組。然後動作又回至步驟 1。

於型態 1，每輸入一位元組之資料，上述四步驟就需重複一次。整個輸入作業最初由 Z 80-CPU 執行一 IN 指令開始。此一指令執行之目的主要在使 PIO 之 RDY 輸出線動作，以通知週邊設備輸入作業即將開始，故此次所讀得之資料被忽略。

PIO 型態 2

Z 80-PIO 之型態 2 乃使用全部四條握手連絡線之雙向資料傳輸型態。由於使用四條握手連絡線，故只有 A 口可用於此一型態。於型態 2，A 口之握手連絡線用於輸出作業，而 B 口之握手連絡線用於輸入作業。 $\overline{\text{A STB}}$ 變為低電位時，A 口之資料輸出暫存器的資料被送至輸入／輸出口線。 $\overline{\text{A STB}}$ 高電位時，資料可以 $\overline{\text{B STB}}$ 信號自週邊輸入至 A 口之資料輸入暫存器。注意，A RDY 與 B RDY 兩信號可同時動作，以顯示 PIO 同時有兩項輸出資料可輸出，並且 PIO 已準備好由週邊設備輸入資料，型態 0 與型態 2 輸出部份作業之唯一差別為，為了達成雙向能力，A 口輸出暫存器之資料惟有在 $\overline{\text{A STB}}$ 信號動作時，才能送出至口資料巴士上。

PIO 型態 3

於型態 3，每一口（A 口或 B 口）之八條輸入／輸出口線不必一律清一色皆作為輸入或作為輸出。八條口條中之每一口線，可個別被令為輸入或輸出線。

Z 80-PIO 型態 3 之作業主要針對狀態及控制應用，並且不使用握手連絡信號線。此一型態於 Z 80-CPU 以 OUT 指令送出選取型態 3 之控制字組後，再送出第二個八位元之控制字組後設定。如圖 15-5 所示，此第二控制字組之每一位元對應於一輸入／輸出口線。若某一位元值為 0，則其所對應之口線即為輸出線；反之，若位元值為 1，則其所對應之口線即為輸入線。

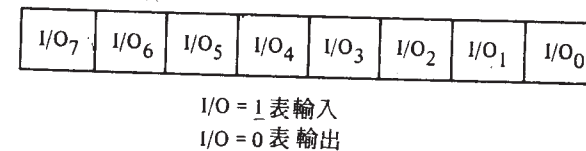


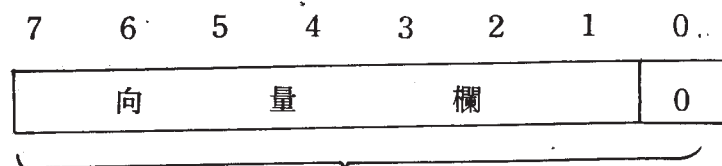
圖 15-5 Z 80-PIO 型態 3 之輸入／輸出方向選擇字組

第二控制字組是寫入前面提過之八位元輸入／輸出選擇暫存器，而非型態控制暫存器。型態 3 一旦設定，資料即可隨時寫入選取口或自選取口讀取。所有握手連絡信號皆不動作；STB 信號不用，RDY 信號恒保持低電位。雖然資料輸出僅影響那些被令為輸出之口線，但輸入資料將讀取所有之口線，包括那些被令為輸出者。

15-1-5 插斷與重置

一、插斷

於輸入或輸出作業時，PIO 之每一口皆可規劃成使之對 Z 80-CPU 提出插斷。當然，此時 Z 80-CPU 必須使用型態 2 之插斷反應。此一插斷型態要求插斷設備（PIO）提供一八位元之插斷向量，以與 CPU 之 I 暫存器的內含組成一十六位元指示器。此一插斷向量，格式如圖 15-6 所示，於使用前必須事先存（寫）入 PIO 輸入／輸出口之插斷向量位址暫存器。



此向量爲一指至插斷服務常式起始位址之十六位元指示器的低次八位元。

圖 15-6 Z80-PIO之插斷向量字組

與插斷向量位址暫存器有關的，每一PIO輸入／輸出口尚具有一兩位元之插斷控制字組暫存器，與一八位元之插斷面罩暫存器。如圖15-7所示，插斷控制字組暫存器儲存與插斷控制有關之訊息。Z80-CPU可藉著送出適當的口位址，與執行OUT指令，將適當的控制字組寫入此一控制字組暫存器，以界定各控制位元之內含。插

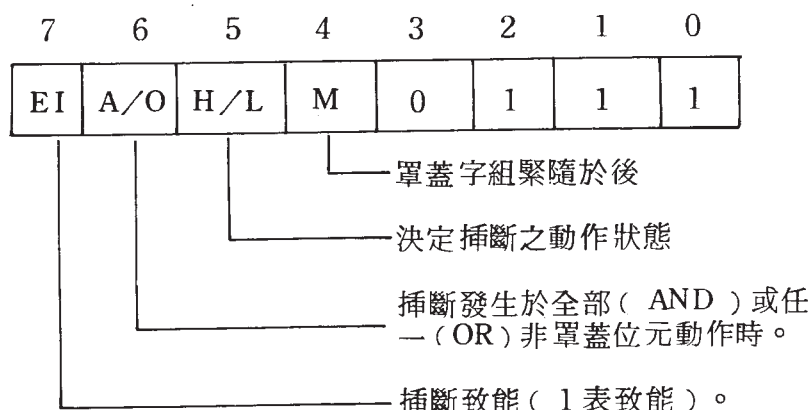


圖 15-7 Z80-PIO之插斷控制字組

斷控制字組之第7位元控制了PIO之插斷能力。若該位元值爲1，則輸入／輸出口可以產生插斷；若該位元值爲0，輸入／輸出口就不會產生插斷。前面說過，型態0，1，與2時，插斷發生於 \overline{STB} 信

號之上升緣。

第4，5，6位元僅用於PIO型態3。第5與6位元即爲兩位元之面罩控制暫存器，其功能在前面已談過。第5位元定義輸入／輸出口線之動作狀態；若1則高電位動作，若0則低電位動作。第6位元指明插斷是於任一非單蓋位元動作時發生，或於所有非單蓋位元皆動作時才發生。該位元值0時代表前一情況，1時代表後一情況。那幾位元爲非單蓋位元則進一步由八位元之面罩暫存器決定。若控制字組之第4位元爲1，則於PIO接到插斷控制字組後，次一送給PIO之字組必須是欲存入插斷面罩暫存器之控制字組。若面罩暫存器之位元值爲1，則其所對應之口線即爲非單蓋位元。

二、重 置

每當電源一打開，Z80-PIO自動進入重置（即起始）狀態。此一重置狀態爲：

- 1 雙口之插斷面罩暫存器自動清除爲零，以將所有輸入／輸出口線禁能。此外，輸出暫存器與插斷致能正反器亦清除爲零。
- 2 口資料巴士線被設定於高阻抗狀態，且RDY握手連絡信號不動作（低電位）。
- 3 PIO型態1自動設定。
- 4 插斷向量位址暫存器並未清除爲零。

除了電源一打開時自動重置外，在 $\overline{M1}$ 信號出現而 \overline{RD} 與 \overline{IORQ} 信號皆未出現時，Z80-PIO亦可重置。若 $\overline{M1}$ 信號動作期間沒有任何 \overline{RD} 或 \overline{IORQ} 信號發生，則 $\overline{M1}$ 信號恢復不動作狀態時，Z80-PIO自動進入重置狀態。此一重置使得不必將電源關掉且不需對PIO加任何額外信號，而仍能達成重置PIO之目的。

15-1-6 程式規劃

Z80-PIO各外部接腳之功用，內部暫存器之構造與功能，至此皆已介紹過。圖15-8所示即為如何以PIO作Z80-CPU與輸入／輸出設備間之界面的情形。PIO於連接在系統後，緊接的工作就是如何將之規劃——作程式設計。此節，我們從一程式設計者之觀點，探討Z80-PIO如何使用。

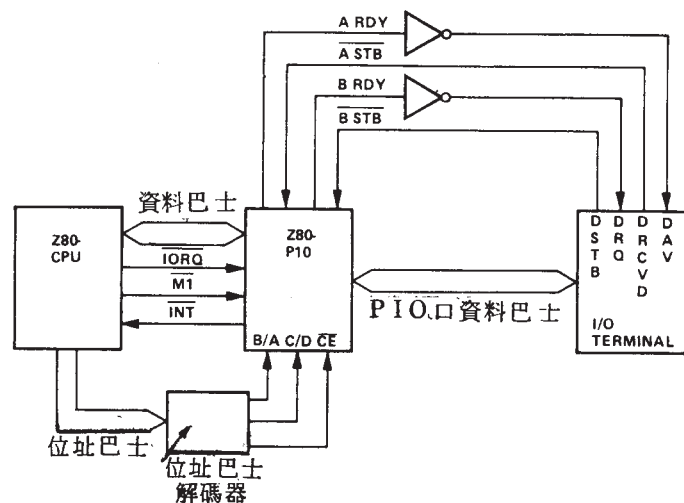


圖15-8 Z80-PIO連接於系統之情形

PIO 一經重置後，即可開始使用。使用Z80-PIO有下列三個步驟：

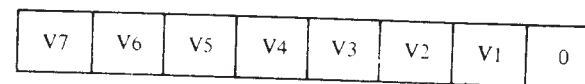
1 設定插斷向量

前面提過，當Z80-PIO所提出之插斷被接受時，其必須對CPU 提供一八位元之插斷向量。此一向量為指至設備插斷服務常式之起始位址之十六位元指示器的低次八位元，且其最低次位元恒為零。PIO 開始使用前，Z80-CPU必須將此一八位元向量寫入選定

PIO 口之插斷向量位址暫存器。

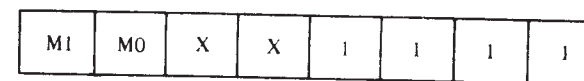
例如，若我們欲以插斷法從A口作輸入，輸入設備之插斷服務常式的起始位址存於位址EFC2之記憶位置，且PIO A口之插斷向量位址暫存器的位址為PAVEC。則於作業前，除CPU之I暫存器必須先存EF外，程式尚須以下列指令，將C2寫入PIO之A口的向量位址暫存器。

```
LD      A, C2
OUT     (PAVEC), A
```



2. 選定作業型態

第二個步驟就是選定PIO之作業型態。每一Z80-PIO有四種不同作業型態（其中B口僅有三種），重置後自動設定於第1型態。使用前，程式設計者必須先將適當之控制字組，寫入輸入／輸出口（即A口或B口）之型態控制暫存器，以選取所要之作業型態。此一控制字組之格式如下所示，其中X部份表示0或1都沒關係。



M ₁	M ₀	型態
0	0	輸出
0	1	輸入
1	0	雙向
1	1	位元控制

譬如，若 PIO A 口之控制暫存器的位址為 PACON，則在以 A 口作輸入口時，程式可以下列指令選取 A 口作輸入口：

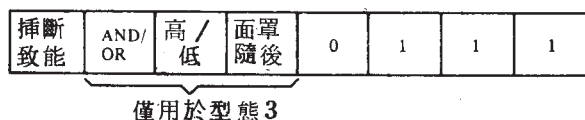
```
LD      A, 4F
OUT     (PACon), A
```

其中，取入 A 暫存器之立即值亦可為 5F，6F，或 7F。此些數值之最低次四位元皆為 1，以指示此一字組是一選取作業型態的控制字組。同時，四個數值之最高次兩個位元皆為 01₂，此一數值即為輸入型態之選取碼。

若此一控制字組選取的是作業型態 3，則程式必須緊接再寫入輸入／輸出選擇暫存器，以決定那一口線欲作為輸入線，那一口線欲作為輸出線。

3. 設定插斷控制字組

最後一個步驟就是設定插斷控制字組，此一插斷控制字組之格式如下所示。



第 7 位元用以將插斷致能 ($b_7 = 1$) 或禁能 ($b_7 = 0$)。第 4，5，6 位元則僅用於型態 3，其功能在前一節皆已說過。最低的四位元恒為 0111₂，以顯示這是一個插斷控制字組。

若上述寫入之插斷控制字組的第 4 位元，即“面罩隨後”位元，為 1，則程式必須緊接寫入 PIO 口之面罩暫存器，以決定那些口線之內含欲再作為進一步決策之依據。

順便一提的是，下面之控制字組可用以將插斷致能正反器清除為 0 或置定為 1，而不致影響插斷控制暫存器之其它位元的原有內含值。

插斷 致能	X	X	X	0	0	1	1
----------	---	---	---	---	---	---	---

15-1-7 插斷服務

時 序

在 PIO 提出插斷請求過後某段時間，若能接受插斷請求，則 CPU 會送出一插斷認知信號—— $\overline{M1}$ 與 \overline{IORQ} 同時動作。此時，PIO 之插斷邏輯會選出提出插斷請求之諸設備中，具有最高優先次序者。（事實上，這就是 IEI 輸入高電位，IEO 輸出低電位之設備。）為了確保雛菊花環線（IEI 及 IEO）確實達到穩定狀態，在 $\overline{M1}$ 動作時，設備之插斷請求狀態禁止改變。於插斷認知期間，最高優先之插斷請求設備將其插斷向量暫存器之內含置於 Z80 之資料巴士。

圖 15-9 所示即為此一插斷請求之時序圖。Z80-CPU 於每一指令之最後 T 週期時讀取插斷請求輸入線 INT 之狀態。並於三個 T 週期後發出插斷認知信號。 $\overline{M1}$ 期間，新的插斷請求不得發生。這使

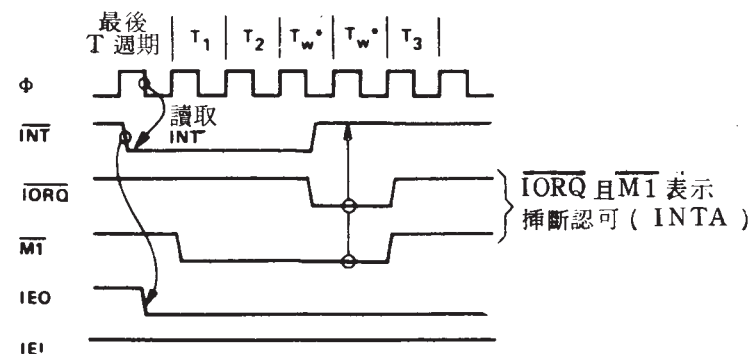


圖 15-9 插斷認可時序

插斷致能信號有時間越過最多四個PIO電路。插斷認知信號 \overline{INTA} (即 $\overline{M1}$ 且 \overline{IORQ}) 期間, IEI 高電位且 IEO 低電位之PIO, 將八位元之插斷向量置於資料巴士上。

若PIO之插斷請求被認知(接納), 則發生請求之口就接受服務。該口之IEO 將一直保持低電位至CPU 執行至插斷回返指令時為止。若插斷請求未被認可, 則於PIO 解碼“ED”之運算碼後一個 $\overline{M1}$ 週期, IEO 被迫變為高電位。此舉保證兩位元組之 RETI 指令為適當之PIO 口所解碼, 圖15-10所示者即為此一情形。

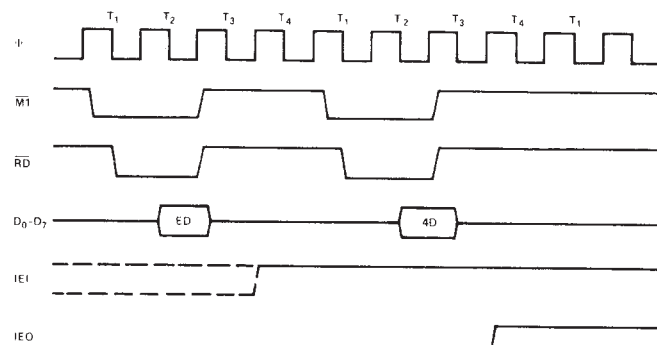


圖15-10 插斷回返週期

雛菊花環結構

圖15-11舉例說明了一具有雛菊花環連接之四個PIO口所能發生之典型巢串插斷系列。(讀者知道, 於此一連接結構, 1A口具有最高優先, 因其位置最近CPU, 1B口次之, 2A口再次之, 2B口殿後)。於此一系列中, 假設2A口先提出插斷, 由於此時1A口與1B口均無正接受CPU之服務, 亦未提出插斷請求, 故2A口立即接受服務。此時, 2A口之IEO 輸出低電位, 使2B口暫時無法插斷。

假設當2A口尚在接受Z80-CPU之服務時, 1B口又提出插

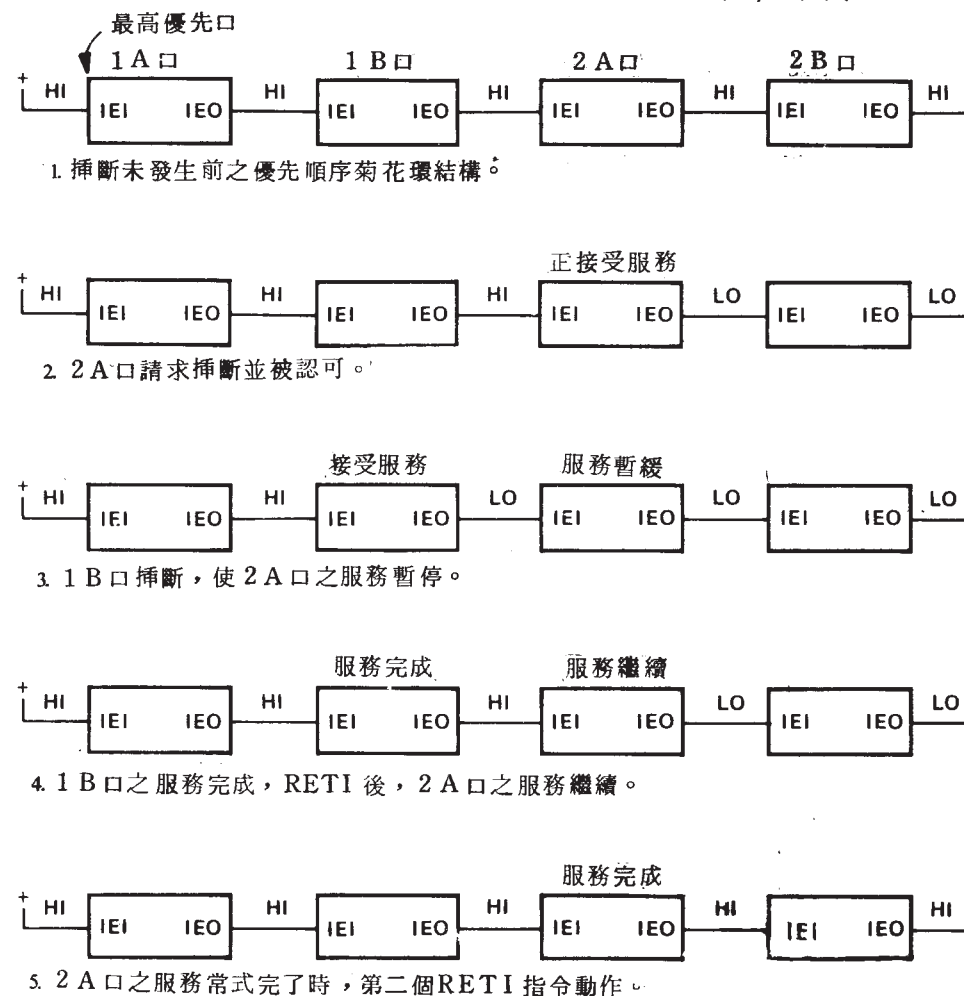


圖15-11 雛菊花環式插斷服務

斷。則由於1B口較2A口具更高優先, 致此時CPU 先暫停對2A口之繼續服務, 並轉而開始服務1B口。此時, 1B口之IEO 亦輸出低電位, 使2A口與2B口均暫時無法接受CPU之插斷服務。若CPU 服務1B口時, 1A口一直未提出插斷, 則CPU 將一直執行完1B口之插斷服務常式。

在服務完1B口後, 1B口之IEO 輸出恢復高電位, 致使2A

口能繼續接受 CPU 之服務。此時，CPU 將從先前暫停處，繼續往下服務 2 A 口。若於此一服務期間，1 A 口與 1 B 口均未提出插斷，故 CPU 將一直執行 2 A 口之插斷服務常式，直至碰及 RETI 指令時為止。此時，2 A 口之插斷服務亦結束，其 IEO 輸出恢復高電位，將 2 B 口之插斷致能。

雛菊花環結構之擴接

若不外加任何電路，則最多有四個 Z80-PIO 設備可以雛菊花環結構連接成具有優先順序之插斷結構。由於在 $\overline{M1}$ 信號開始與 \overline{IORQ} 信號開始間之插斷認可週期內，插斷致能狀態 (IEO) 必須馳越整個環結構 (若最高優先者提出插斷，則正巧為 4 級)，故此一限制是必要的。由於插斷致能狀態在 $\overline{M1}$ 信號動作期間無法改變，故 CPU 所收到之向量位址，可確定必來自具最高優先之插斷設備。

若系統必須容納四個以上之 PIO 設備，則可採用如圖 15-12 所示之“前望”(look-ahead)結構。使用此一技巧，就有 30 個以上之 PIO 可以標準 TTL 電路連結在一起。

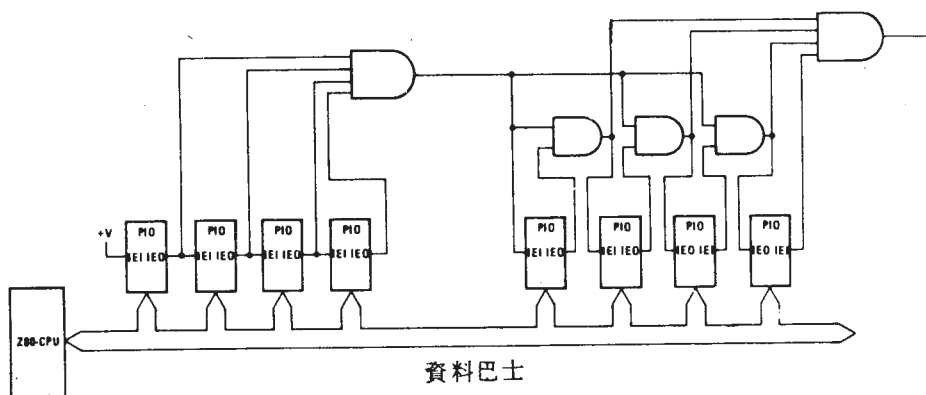


圖 15-12 雛菊花環優先插斷結構的擴充方法之一。

15-1-8 應用

在從各方面討論過 Z80-PIO 後，最後，我們舉一 Z80-PIO 之應用例子。圖 15-13 所示即為一典型之控制型態應用。假若 Z80-CPU 欲監聽某一工業程序。任何非正常之運轉情況發生時，Z80-CPU 構成之控制系統必須獲得通知。則此一程序控制及狀態字組之格式如下：

D7	D6	D5	D4	D3	D2	D1	D0
特殊 測驗	打開 電源	停電 警鈴	暫停	溫度 警鈴	打開 加熱器	壓力 系統	壓力 警鈴

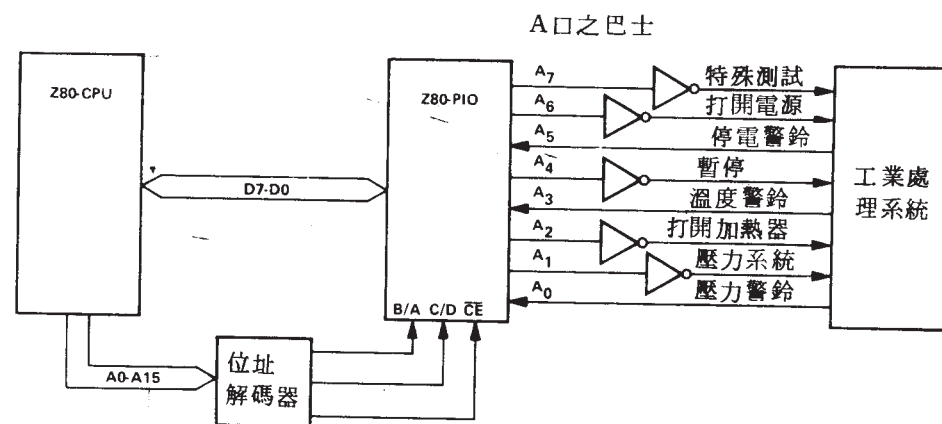


圖 15-13 Z80-PIO 之控制型態應用

此時，PIO 可如下地使用。首先，我們以下列控制字組將 A 口設定於作業型態 3。

D7	D6	D5	D4	D3	D2	D1	D0
1	1	X	X	1	1	1	1

只要型態 3 一選定，次一輸出至選取口之控制字組必須恒為輸入／輸出選擇字組。於此例，我們希望選 A 5，A 3，與 A 0 等口線為輸入，而其它口線為輸出，故此一緊接控制（輸入／輸出選擇）字組的內含必須為

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	0	1	0	0	1

其次，有關之插斷向量必須存入向量位址暫存器。

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	0

再次是輸出插斷控制字組。

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	1	0	1	1	1

致能 C R 高電位面罩
插斷邏輯動作隨後

插斷控制

緊接插斷字組後的是插斷面罩字組。

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	1	0	1	1	0

選取 A 5，A 3，與 A 0 接受監聽

此時，若有一感應器將高電位置於 A 5，A 3，或 A 0 資料線，則插斷請求就會產生。插斷字組可選擇任一輸入或輸出組合引起插斷。例如，若上述之面罩字組為

D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	1	1	0

則若輸出暫存器之第 7 位元（特殊測試）為 1，插斷請求亦會發生。

若 PIO 之口位址指定如下：

$E0_H = A$ 口資料暫存器。

$E1_H = B$ 口資料暫存器。

$E2_H = A$ 口控制暫存器。

$E3_H = B$ 口控制暫存器。

則於圖 15-13 之連接中，A 0 位址線即可用作 A / B 口之選擇線，A 1 位址線用作控制／資料之選擇線。同時，晶片致能（ \overline{CE} ）可由解碼 A_7 至 A_2 （1110 00）等位址線獲得。注意，若系統所用之週邊設備相當有限，則晶片致能解碼甚至皆可不必，而可直接採用一位址位元。

15-2 Z80—CTC

本章第二部份介紹 Z80 族中之另一元件——**Z80 計數／計時器**。本書爾後對此一元件皆直稱 Z80—CTC。

CTC 乃計數／計時器之英文 Counter Timer Circuit 的縮寫。

Z80—CTC 為一具有**四個獨立通道**（Channel）之**可程式化**元件，其為 Z80—CPU 組成之微電腦系統提供**硬體計數與計時**之功能。CPU 可配組 CTC，使其動作於各種不同之型態與狀況下，以期與各種週邊設備達成界面，而不需外加太多電路。Z80—CTC 之主要特色包括：

- 1 所有輸入與輸出均具 TTL 吻合性。
- 2 每一通道皆可選擇動作於計數器型態或計時器型態。
- 3 於任一型態，CTC 內均有一倒數計數器，逐漸遞減計數至零為止。CPU 可隨時讀取此一計數器之內含。
- 4 當倒數計數器計數至零時，時常數暫存器能自動將時間常數值再重新存入計數器。
- 5 於計時器型態，程式可選擇正向或負向觸發緣起始計時作業。於計數器型態，同一輸入用以計數外部發生事件之次數。

6. 三個通道具有零計數／到時輸出，能推動達靈頓電晶體。
7. 任一通道皆可規劃成於零計數時產生插斷。
8. 含雛菊花環優先插斷邏輯，能作自動插斷指向而不需外加額外電路。

15—2—1 接腳功能

Z80-CTC 使用 N 通道矽晶片做成，且包裝成 28 支接腳之 DIP（雙並列）形式。其僅需單一正 5 伏特電源，與一單相 5 伏特時序，其外觀接腳圖如圖 15-14 所示。茲將每一接腳之功能介紹如下：

D7-D0 Z80-CPU 之資料巴士（雙向，三態）

此巴士用以傳遞來回於 Z80-CPU 與 Z80-CTC 兩者間之任何資料與控制命令。

CS1-CS0 通道選擇（輸入，高電位動作）

如下面表格所示，此兩支接腳代表一兩位元之位址碼，用以由 CTC 之四個獨立通道中選取一個，以作輸入／輸出讀取或寫入。

	CS1	CS0
第 0 通道	0	0
第 1 通道	0	1
第 2 通道	1	0
第 3 通道	1	1

\overline{CE} 晶片致能（輸入，低電位動作）

此接腳上之低電位，使 CTC 能接受 CPU 之讀寫。換言之，唯有當此接腳輸入低電位時，CTC 方有“知覺”——輸入／輸出寫入週期時，CTC 能接受 CPU 所送來之控制字組、插斷向量、或時間常數；輸入／輸出讀取週期時，CTC 能將倒數計數器之內含送給 Z80-CPU。通常，此一信號即為對應於 CTC 四個通道之四個輸入

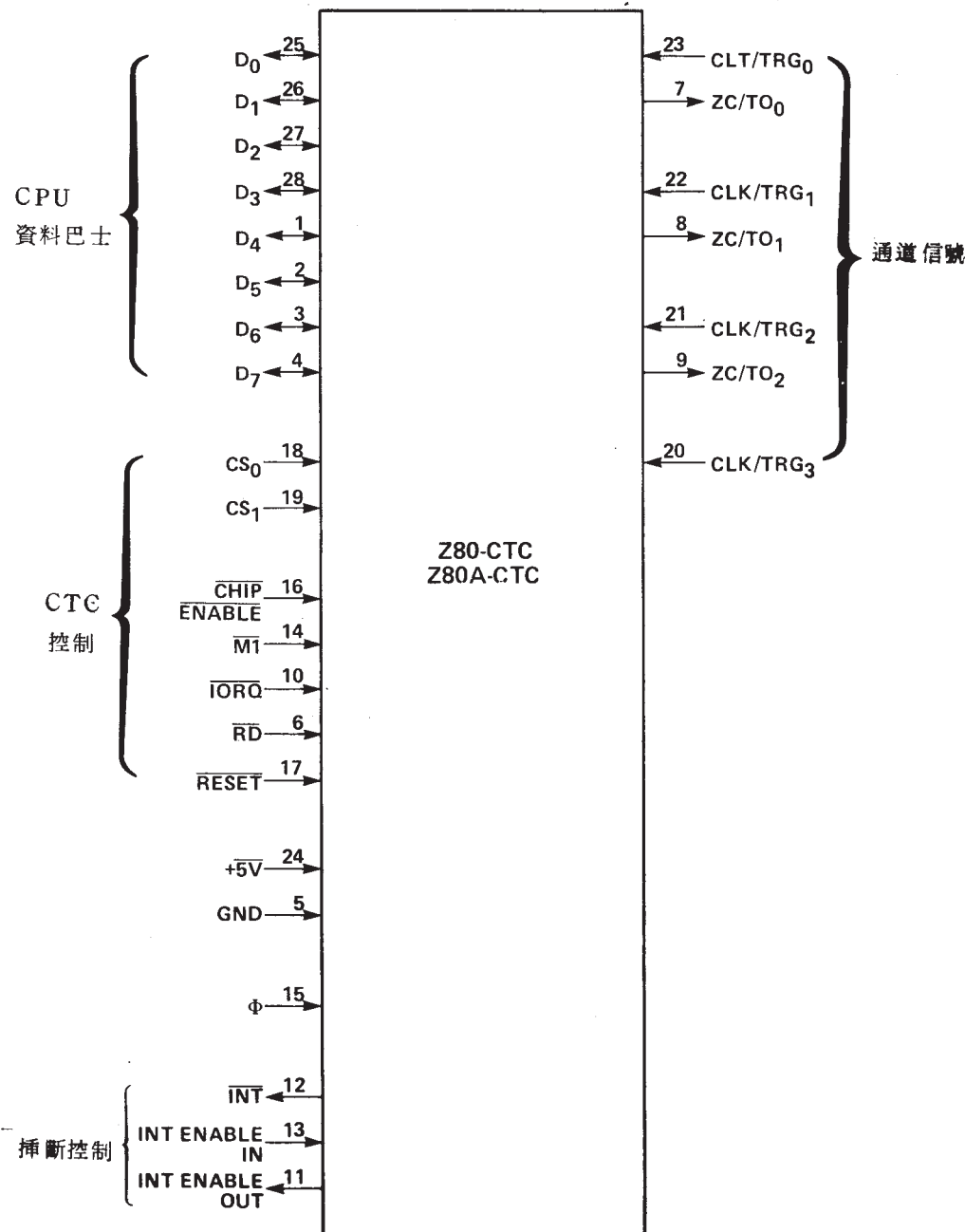


圖 15-14 Z80-CTC 之外觀接腳圖

／輸出口位址的解碼輸出。

ϕ 系統時序（輸入）

CTC 應用此一單相時序，同步了某些內部之信號。

$\overline{M1}$ CPU 之第 1 機器週期信號（輸入，低電位動作）

$\overline{M1}$ 與 \overline{RD} 同時動作表示 CPU 正由記憶器拿取指令。 $\overline{M1}$ 與 \overline{IORQ} 表示 CPU 正在認可插斷，且於 CTC 具有雛菊花環優先順序，而其中之一通道又提出插斷請求時，使 CTC 將插斷向量置於 Z80 之資料巴士上。

\overline{IORQ} CPU 之輸入／輸出請求（輸入，低電位動作）

\overline{IORQ} 與 \overline{CE} 及 \overline{RD} 信號併用，以傳遞來回於 CPU 與 CTC 間之資料與通道控制字組。在 CTC 寫入週期時， \overline{IORQ} 與 \overline{CE} 必須同時動作而 \overline{RD} 不動作。CTC 雖未收到一明確之寫入信號，但却於內部私下將 \overline{RD} 信號反態，產生自己之寫入信號。當 CTC 讀取週期時， \overline{IORQ} ， \overline{CE} ，與 \overline{RD} 同時動作，以使 CTC 之倒數計數器的內含置於 Z80 之資料巴士上。若 \overline{IORQ} 與 $\overline{M1}$ 同時動作，則表 CPU 正在認可插斷請求，具最高優先之插斷通道，必須將其插斷向量置於 Z80 之資料巴士。

\overline{RD} CPU 之讀取週期（輸入，低電位動作）

此一信號動作代表 Z80-CPU 欲讀取資料。

\overline{IEI} 插斷致能輸入（輸入，高電位動作）

此一信號用以幫忙形成雛菊花環之插斷結構，使系統擁有一個以上有插斷能力之週邊設備時，設備彼此之間有一優先順序可循。此接腳高電位表示 Z80-CPU 並未正在服務任何具有更高優先之插斷設備。

\overline{IEO} 插斷致能輸出（輸出，高電位動作）

\overline{IEO} 與 \overline{IEI} 信號一併用以形成優先插斷之雛菊花環結構。

\overline{IEO} 唯有當 \overline{IEI} 輸入高電位，且 CPU 並未正在服務此 CTC 之任一通道時，才會輸出高電位。當 \overline{IEI} 輸入低電位（CPU 正在服

務其它更高優先之插斷設備），或 CPU 正在服務 CTC 自己（任一通道）時， \overline{IEO} 均輸出低電位。致使其它較低優先之插斷設備無法立即接受 CPU 之服務。

\overline{INT} 插斷請求（輸出，源極開路，低電位動作）

此信號動作表示有一經規劃成插斷致能（能提出插斷）之 CTC 通道，其倒數計數器已計數至零。

\overline{RESET} 重置（輸入，低電位動作）

此信號停止所有通道之計數，並將所有控制暫存器內之通道插斷致能位元清除為零，使 CTC 無法產生插斷請求。 $\overline{ZC/TO}$ 與 \overline{INT} 輸出均處於不動作狀態， \overline{IEO} 反映 \overline{IEI} 之狀態，並且 CTC 資料巴士之輸出變成高阻抗狀態。

$\overline{CLK/TRG3-CLK/TRG0}$ 外部時序／計時觸發（輸入，高或低電位動作由用者選定）

Z80-CTC 總共有四支 $\overline{CLK/TRG}$ 接腳，每一此種接腳對應於一個別 CTC 通道。於計數器型態，此接腳信號之每一動作緣，使倒數計數器之內含值減 1。於計時器型態，此接腳信號之動作緣起始計時功能。用者可選擇正向緣或負向緣，作為動作緣。

$\overline{ZC/TO2/ZC/TO0}$ 零計數／到時（輸出，高電位動作）

每一 Z80-CTC 具有三支 $\overline{ZC/TO}$ 接腳，分別對應於第 0，1，2 通道（由於接腳數受限制，致第 3 通道無 $\overline{ZC/TO}$ 接腳）。不管在計數器型態或計時器型態，每當倒數計數器遞減至零時，此一接腳即會輸出一高電位。

15-2-2 內部結構

Z80-CTC 之內部結構方塊圖如圖 15-15 所示。由圖中可看出，每一 CTC 由一 CPU 巴士界面，一內部控制電路，一插斷控制電路，與四個計數／計時通道所組成。四個獨立之通道分別稱為第 0，1，2，與 3 通道。對於每一通道，CTC 均能產生一獨特之插斷

向量（以自動指至一插斷服務常式）。四個通道可連接至標準 Z80 優先順序環之四個相鄰節，而以第 0 通道為最高優先。CPU 巴士界面電路使 CTC 能直接連接至 Z80-CPU，而不需外加任何電路。不過，於大系統可能需要一些額外之口位址解碼電路與／或線緩衝器。

下面，我們進一步探討每一 CTC 通道之內部結構，以及插斷控制電路之功能。

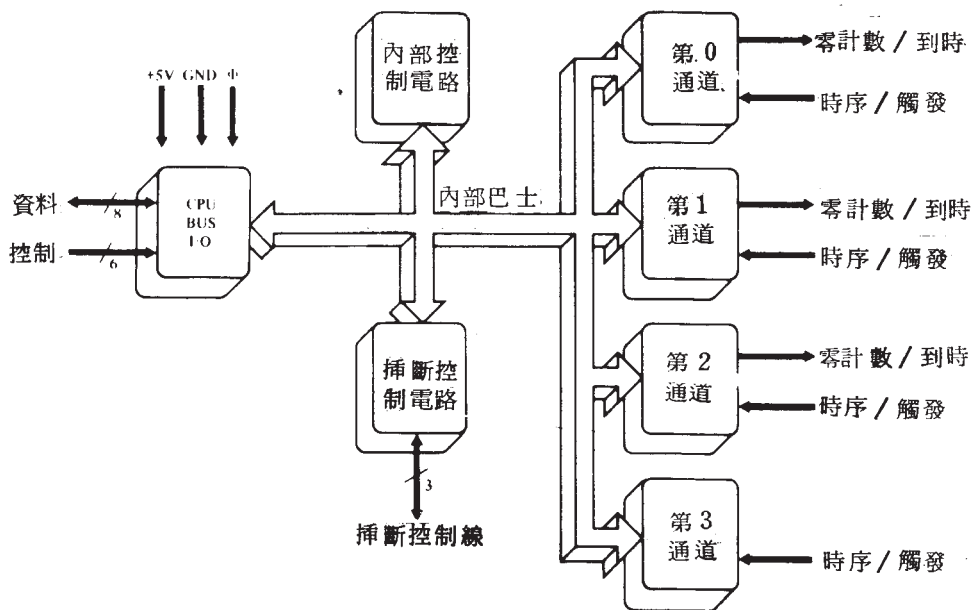


圖 15-15 Z80-CTC 之內部結構

一、通道之結構

每一 CTC 通道之內部結構如圖 15-16 所示。顯然，每一計數／計時通道包括 2 個暫存器，2 個計數器，與控制電路。其中，暫存器為八位元之時常數（time constant）暫存器與八位元之通道控制暫存器，而計數器為 CPU 可讀取之八位元倒數計數器，與一八

位元之比率器（prescaler）。

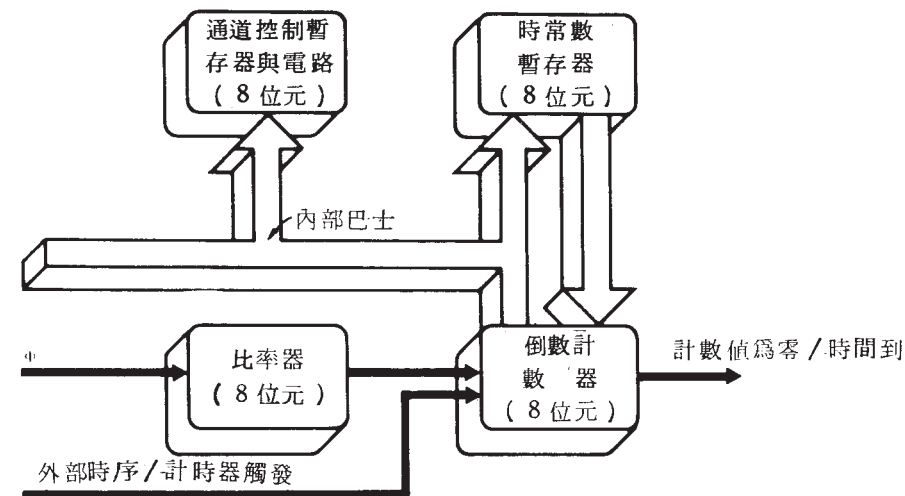


圖 15-16 Z80-CTC 通道之內部結構

通道控制暫存器及電路

Z80-CPU 藉著寫入通道控制暫存器（八位元）與電路，選定了通道之作業型態與參數。整個 CTC 內總共含有四個此種暫存器，每一通道一個。CPU 寫入何者，則需視兩通道選擇輸入 CS0 與 CS1（通常連接至 CPU 位址巴士之最低次兩位元：A0 與 A1）之寫碼而定。下表所舉即是一例。

寫入 CTC 通道控制暫存器之字組，第 0 位元恒為 1，而其它位元則用以選定通道之作業型態與參數，其詳細功能將留待後面作業型態與程式設計兩小節再作討論。

	CS 1	CS 0
第 0 通道	0	0
第 1 通道	0	1
第 2 通道	1	0
第 3 通道	1	1

比率器

此一電路僅用於計時器型態。CPU 可透過通道控制暫存器規劃比率器，將比率器之輸入——系統時序（ ϕ ）——除以 16 或 256。比率器之輸出然後作為倒數計數器之時序。在最開始與每次倒數至零時，時常數暫存器之內含均自動重新存入倒數計數器。實效上，這再度將系統時序除以時間常數值那麼多倍。每當倒數計數器計數至零，其輸出（ZC/TO）即變為高電位。

時常數暫存器

時常數暫存器長八位元，用於計數器與計時器兩種型態。CPU 一寫入通道控制字組後，立即將 1 至 256 之常數值寫入此一暫存器，而完成對此一暫存器之規劃。當 CTC 一起始或每次倒數計數器計數至零時，時常數暫存器會將此一規劃值自動存入倒數計數器。若通道正在計數或計時時，CPU 對時常數暫存器寫入新值，則該新值將一直等到現有之倒數計數完成（數至零）時，才會存入倒數計數器。

倒數計數器

倒數計數器長八位元，同時用於計數器與計時器兩種型態。一開始與每次計數至零時，時常數暫存器會自動將計數初值存入此一暫存器。於計數器型態，每一外加時序邊緣將該計數器之值減一；於計時

器型態，比率器之時序輸出將該計數器之值減一。任意時刻，只要對選取之 CTC 通道的口位址作輸入／輸出讀取，CPU 即可存取此一暫存器之內含，並獲得已數至零之次數。程式可規劃任一 CTC 通道，令其在每次倒數計數器計數至零時產生插斷。

於第 0，1，與 2 通道，每當抵達零計數情況時，其相對之 ZC/TO 接腳就會出現一信號脈衝。不過，由於接腳數受限制，致第 3 通道並無此一接腳，故其僅能用於不需此一脈衝之應用。

二、插斷控制電路

Z 80-CTC 之插斷控制電路確保 CTC 之動作與 Z 80 系統之插斷作業取得一致。每一 CTC 有兩條用以形成雛菊花環結構之信號線（IEI 與 IEO）。系統中任一設備之插斷優先順序取決於其在雛菊花環結構中所處之位置。在系統內，最靠近 CPU 之設備擁有最高優先；在 CTC 內，每一通道之優先順序則取決於其編號，第 0 通道最優先，第 1 通道次之，……。正如其它週邊設備一般，CTC 產生插斷的目的乃在迫使 CPU 開始執行某一插斷服務常式。依據 Z 80 系統之插斷作法，低優先之設備或通道無法插斷 CPU 對較高優先之設備或通道的服務。不過，較高優先之設備或通道却可插斷 CPU 對較低優先設備之服務。

程式設計者可規劃 CTC 通道，令其於倒數計數器數至零時對 CPU 產生插斷請求。（注意，使用此一特色時，Z 80-CPU 必須設定於插斷型態 2。）插斷請求提出過後一段時間，若能接受此一請求，CPU 就會送出一插斷認可信號，且 CTC 之插斷控制電路會找出 CTC 內提出插斷請求中，具最高優先之通道。此時，若 CTC 之 IEI 輸入動作（高電位），CTC 即會將八位元之插斷向量置於系統之資料巴士上。此向量之最高次五位元於先前 CTC 之起始規劃過程中，即已事先寫入 CTC；其次兩位元則由 CTC 之插斷控制電路提供，其值即等於最高優先插斷通道的號碼；最後，最低次位元恒為

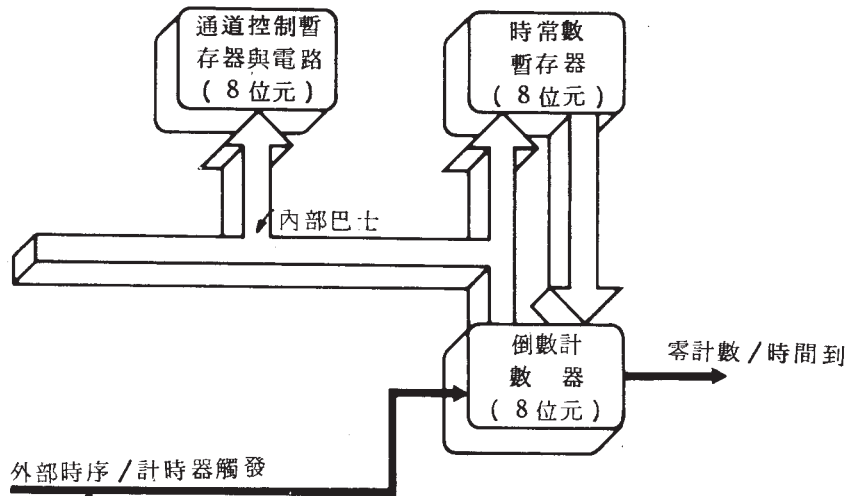


圖 15-19 通道之計數器型態

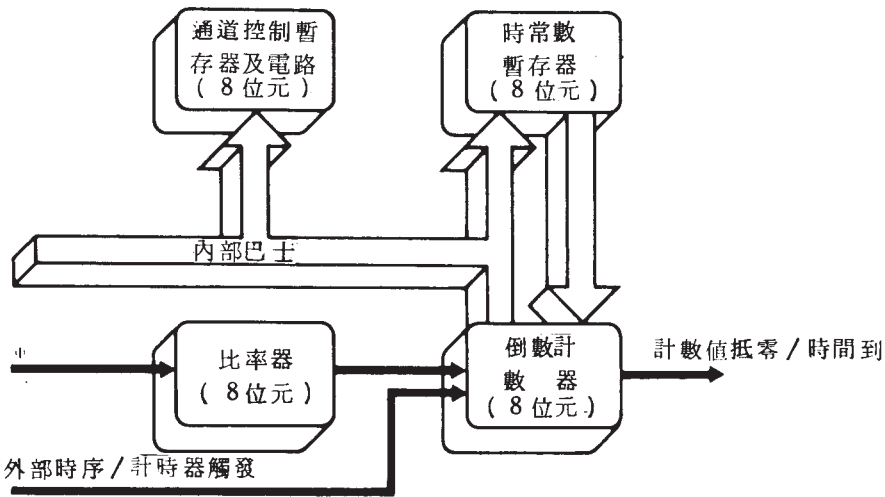


圖 15-20 通道之計時器型態

零，其理由如下。

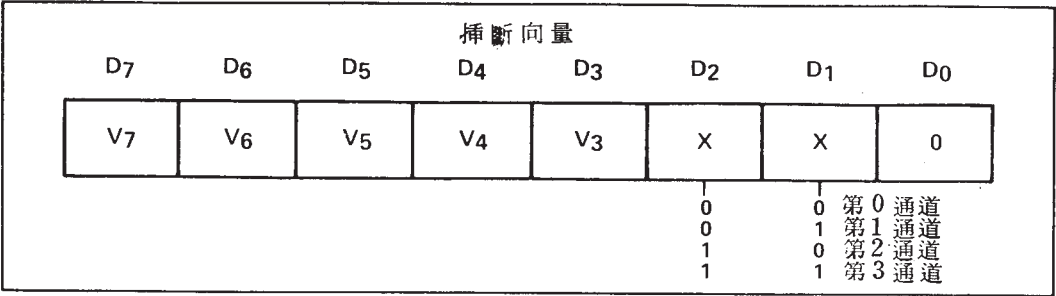


圖 15-17 Z80-CTC 所提供之插斷向量

CTC 所提供之插斷向量用以形成一十六位元指示器，該指示器指至一儲存插斷服務常式之起始位址的記憶位置。如圖 15-18 所示，該十六位元指示器之低次八位元即為 CTC 所提供之八位元插斷向量，

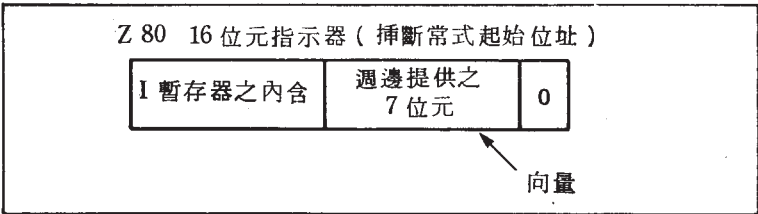


圖 15-18 Z80 之十六位元插斷向量的組成

八位元則來自 Z80-CPU 之 I 暫存器。十六位元指示器所指之記憶位置儲存插斷服務常式起始位址的低次八位元，而次一緊接位置儲存該位址之高次八位元。此一起始位址所指之記憶位置即儲存插斷服務常式之第一指令的運算碼。

Z80 系統有一習慣，插斷服務常式之起始位址表格中的每項位址，其低次八位元均需儲存於偶數位址之記憶位置，且高次八位元儲

Z80 微電腦軟體硬體

存於次一記憶位置。此乃為何插斷設備所提供之八位元插斷向量的最低次位元皆恒為 0 之原故。

每一插斷服務常式之最後指令均為 RETI 指令，該指令用以起始菊花環之插斷致能線 IEO。CTC 會隨時監視著資料巴士，並於該指令出現時將之解碼。因而，CTC 之通道控制電路知道 CPU 何時完成插斷服務，而毋需再進一步與 CPU 溝通。

15—2—3 作業型態

當電源一打開時，Z80—CTC 之狀態未定。按下重置鍵可使 CTC 處於某一已知狀態。在任何通道能開始計數或計時前，CPU 必須先將一控制字組以及一時常數資料字組，寫入該通道之適當暫存器。再者，若程式曾使任一通道之插斷致能，則插斷向量必須寫入 CTC 之插斷控制電路。當 CPU 將此些字組都寫入 CTC 後，所有通道即可規劃，使之動作於計數器型態或計時器型態。

一、CTC 計數器型態

若寫入之通道控制字組第 6 位元為 1，則通道作業於計數器型態。於計數器型態，CTC 計數由 CLK/TRG 接腳輸入之脈衝緣。計數電路會自動計數自通道之外部時序接腳 (CLK/TRG) 所輸入之一連串觸發緣；每一觸發緣後，於緊接系統時序 (ϕ) 之上升緣時，倒數計數器之值減一。(倒數計數器之初值已由先前寫入之時常數資料字組界定。)雖然於外部時序觸發緣與系統時序上升緣間無所謂的準備 (set-up) 時間，但除非緊接之 ϕ 脈衝來到，否則，倒數計數器將不會減一。通道控制字組之第 4 位元決定究竟是外部時序輸入之正向緣為觸發緣，或負向緣為觸發緣。

於上述系列進行過程中，每當倒數計數器倒數至零，時常數暫存器所存之原有時常數字組，即會自動再存入倒數計數器。然後，計數動作繼續不間斷。若通道被規劃成能產生插斷，則在每次倒數至零時

，CTC 之插斷線亦同時動作。當倒數計數進行過程中，若 CPU 對時常數暫存器寫入新的時常數 (字組)，則此一新時常數將在下一次計數時生效。

二、CTC 計時器型態

若寫入通道之通道控制字組的第 6 位元為 0，則 Z80—CTC 設定於計時器型態。於計時器型態，CTC 產生一為系統時序週期之整數倍的時限。如此，根據系統時序週期，通道即可用以測量時限。於計時器型態，系統時序加諸兩連續之計數器：比率器與倒數計數器，視先前寫入之通道控制字組的第 5 位元值而定，比率器可將系統時序頻率除以 16 ($b5=0$) 或 256 ($b5=1$)。比率器之輸出然後再用以遞減倒數計數器，倒數計數器之計數值亦可以程式將之規劃於 1 至 256 之間。

正如計數器型態一般，每當倒數計數器計數至零，時常數即自動由時常數暫存器再取入，並且計數繼續不斷。此外，於倒數至零時，通道之 ZC/TO 輸出 (亦即為倒數計數器之輸出) 亦輸出一脈衝，此一接腳輸出之脈衝系列的週期顯然為

$$t_c \times P \times TC$$

，式中， t_c 為系統時序週期， P 為比率器之比率 (16 或 256)，而 TC 為時常數值。

先前寫入之通道控制字組的第 3 位元值，決定計時程序究竟是自動起始，抑是由通道之 CLK/TRG 輸入接腳上之觸發緣起始。若第 3 位元值為 0，則於時常數字組寫入之機器週期後的 CPU 週期一開始，計時器即自動開始作業。若第 3 位元值為 1，則計時作業於緊接時常數字組取入之計時器觸發緣過後的第 2 個 ϕ 上升緣時開始。若無時常數字組緊隨於後，則計時於緊接控制字組寫入週期後之計時器觸發邊緣，過後的第 2 個系統時序上升緣時開始作業。通道控制字組的第 4 位元則用以選擇計時器觸發是對上升緣或下降緣反應。雖然計時器觸發之動作緣與次一 ϕ 上升緣間毋需準備時間，但是，若觸發緣至

上升緣間小於某一特定準備時間，倒數計數器將一直等到次一 ϕ 上升緣時，才會開始遞減。

若通道控制字組之第 7 位元值亦為 1，則倒數至零之狀態除了使通道之 ZC/T0 接腳輸出一脈衝外，亦起始了插斷請求系列。

15-2-4 程式規劃

— Z80-CTC 通道在能開始計數或計時之前，CPU 必須先對通道寫入控制字組與時常數字組。此兩字組分別儲存在通道之通道控制暫存器以及時常數暫存器。除此之外，若 CTC 四個通道之中有任一通道之控制字組的第 7 位元為 1，致使插斷致能，則 CPU 亦須將插斷向量寫入適當之暫存器。此一插斷向量可供四個通道共同。

茲將 CTC 程式規劃之程序，以及每一寫入字組之各位元的意義，分別介紹如下：

一、寫入通道控制暫存器

欲寫入通道控制暫存器，CPU 必須對應於所要之 CTC 通道的口位址，執行一正常之輸入／輸出寫入系列。CTC 之 CS0 與 CS1 兩輸入接腳用以從四個通道中選取一者。一般之作法，這兩輸入接腳均連接至 CPU 位址巴士之最低次兩位元：A0 及 A1，致使 CTC 所佔之四個位址成連續。寫入 CTC 通道之字組將被視為通道控制字組，其格式如圖 15-21 所示。注意，最低次位元恒為 1。通道控制字組之其它 7 位元所代表的意義如下：

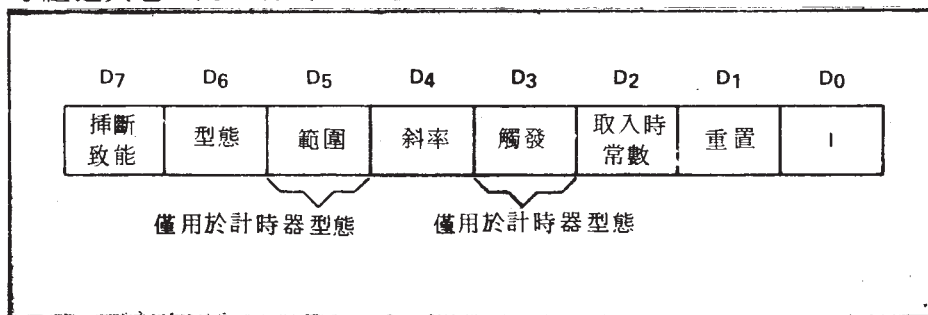


圖 15-21 Z80-CTC 之通道控制字組

第 7 位元

該位元值為 1 時，通道之插斷致能。每當倒數計數器倒數至零時，通道即產生一插斷請求系列。若四個通道控制暫存器中任一者之第 7 位元為 1，則作業開始前，插斷向量亦須寫入 CTC。CTC 之計數器型態與計時器型態皆可產生插斷。若於作業過程中，新的通道控制字組又寫入，則新字組第 7 位元的作用立即生效。

當第 7 位元值為 0 時，通道之插斷被禁能。倒數計數器倒數至零時，通道不產生插斷請求，CTC 之 $\overline{\text{INT}}$ 輸出線不動作。

第 6 位元

該位元值 1 時選取計數器型態，外部時序 (CLK/TRG) 輸入接腳上之每一觸發緣，令倒數計數器之內含值減一。比率器不用。

第 6 位元值 0 時選取計時器型態。系統時序 ϕ 作為比率器之時序。比率器之輸出緊接又作為倒數計數器之時序輸入。倒數計數器之輸出 (即通道之 ZC/T0 輸出) 為一均勻之脈衝系列，其週期為

$$t_c \times P \times TC$$

。式中， t_c 為系統時序週期，P 為比率器因子 16 或 256，而 TC 為時常數。

第 5 位元

該位元僅用於計時器型態。若該位元值為 1，則比率器因子為 256。否則，(該位元值為 0 時)，比率器因子為 16。

第 4 位元

第 4 位元用於兩種型態，其意義如下：

$$b_4 = 1$$

計時器型態：正向觸發緣起動計時作業。

計數器型態：正向緣令倒數計數器之值減一。

$$b_4 = 0$$

計時器型態：負向觸發緣起動計時作業。

計數器型態：負向緣令倒數計數器之值減一。

第3位元

該位元僅用於計時器型態。第3位元值為1時，寫入時常數之次一機器週期的 T_2 上升緣之後的外部觸發緣，起動計時作業。若準備時間符合，則比率器於2個時序週期後遞減，否則，於3個時序週期後遞減。

第3位元值為0時，寫入時常數之次一機器週期的 T_2 上升緣時，計時器開始作業。

第2位元

第2位元值為1時，時常數暫存器之時常數字組，乃為寫入通道之次一字組。若於通道正在作業時，更新之通道控制字組與時常數資料字組又寫入，則新的時常數字組將一直等到倒數計數器遞減至零時，才會再取入倒數計數器。

若第2位元值為0，則緊接通道控制字組之後即無時常數資料字組。因此，寫入一第2位元為0之通道控制字組，表示程式欲改變正在動作中之通道的狀態。因為，若時常數暫存器內無一經正確規劃之資料字組，通道即無法作業，而通道控制字組之第2位元值為1，乃是寫入時常數暫存器之唯一途徑。

第1位元

該位元值為1代表重置通道。此時，通道停止計數或計時。但這並非是一記憶性狀況。一寫入此一位元時，重置脈衝即會中斷目前之通道作業，不過，通道控制暫存器之所有位元值並不改變。若第1與第2位元均為1，則在一取入時常數後，通道即又立即恢復作業。

若第1位元值為0，則通道繼續目前之作業。

二、寫入時常數暫存器

除非CPU 將一時常數資料字組寫入時常數暫存器，否則，通道將無法開始計數或計時之作業。祇要通道控制字組之第2位元為1，則時常數字組必須是緊接通道控制字組後寫入通道的字組。時常數字組可為1至256之間的任意值。所有位元皆0代表256。若時常數資料字組是寫入一正在作業中之通道，則此一新數值將一直等到倒數計數器遞減至零時，才會自時常數暫存器存入倒數計數器。圖15-22所示即為時常數資料字組之情形。

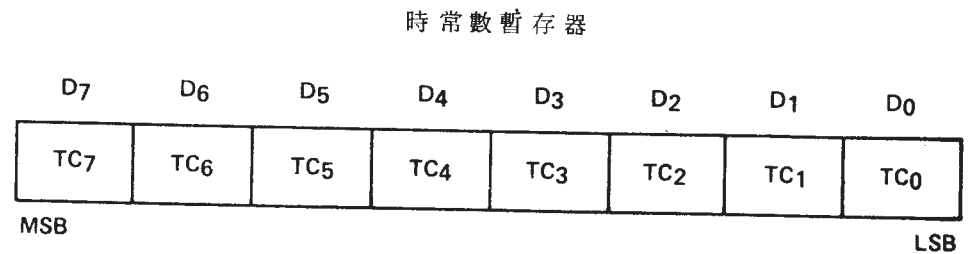


圖15-22 Z80-CTC之時常數暫存器

三、寫入插斷向量暫存器

Z80-CTC特別設計成能與Z80-CPU之插斷型態2一起作業。於Z80-CPU之插斷型態2，當CTC 通道提出插斷請求且被認可時，其必須有一十六位元指示器，以便自記憶器之表格中，獲得有關之插斷處理常式的起始位址。如圖15-23所示，此一十六位元指示器即由CPU I 暫存器之內含，與CTC 所提供之八位元插斷向量兩者組合而成，前者當高次八位元，後者為低次八位元。

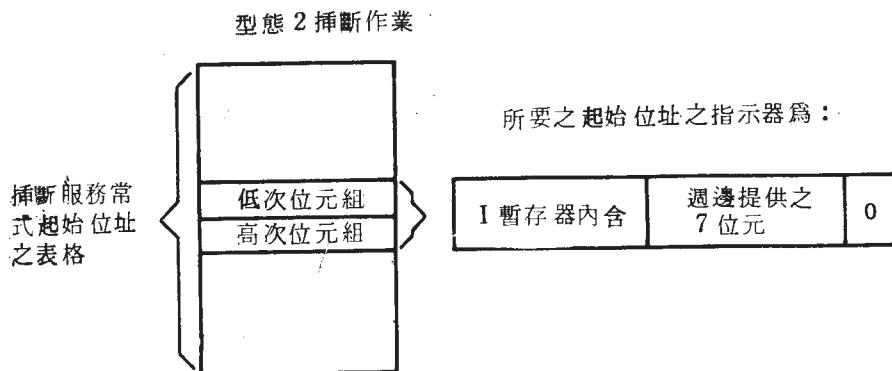


圖 15-23 Z80-CPU 插斷型態 2 之作業

八位元插斷向量之高次五位元，於最初規劃序列時必須事先寫入 CTC。為此，CTC 必須宛如寫入通道控制暫存器般地，寫入第 0

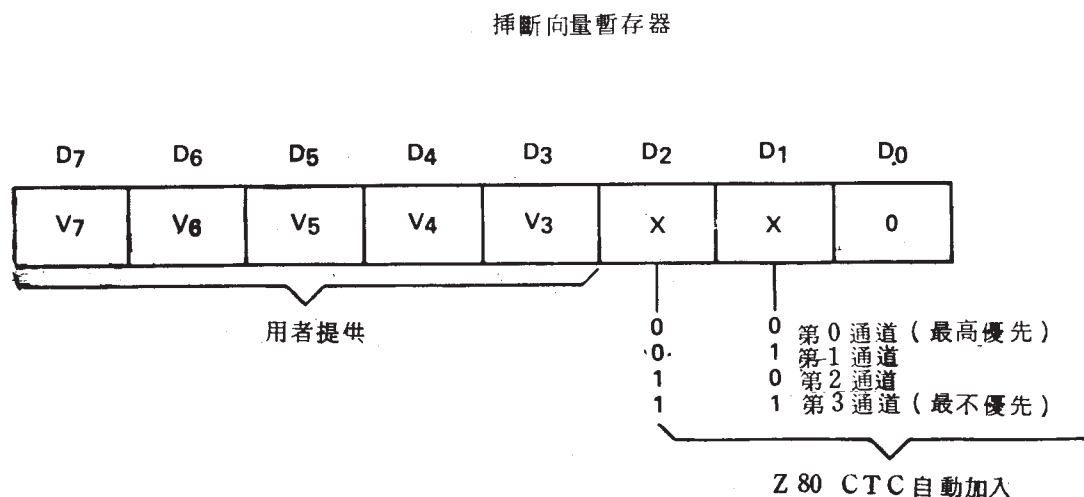


圖 15-24 Z80-CTC 之插斷向量字組

通道之輸入／輸出口位址，唯一的不同是，此時第 0 位元必須為 0。（該位元值為 0 之原因在前面已說明過）不過，此時第 1 與第 2 兩位元值並未使用。當提出插斷之通道必須將插斷向量置於 Z80 之資料巴士時，CTC 之插斷控制電路將自動提供一代表插斷通道之位址的兩位元電碼，作為該兩位元之值。

圖 15-24 所示即為 Z80-CTC 之插斷向量字組的組合情形。

15-2-5 時序

此一小節舉例說明下列各種作業：寫入 CTC，讀取 CTC，計數，與計時，時，CTC 有關接腳之時序關係。

一、CTC 寫入週期

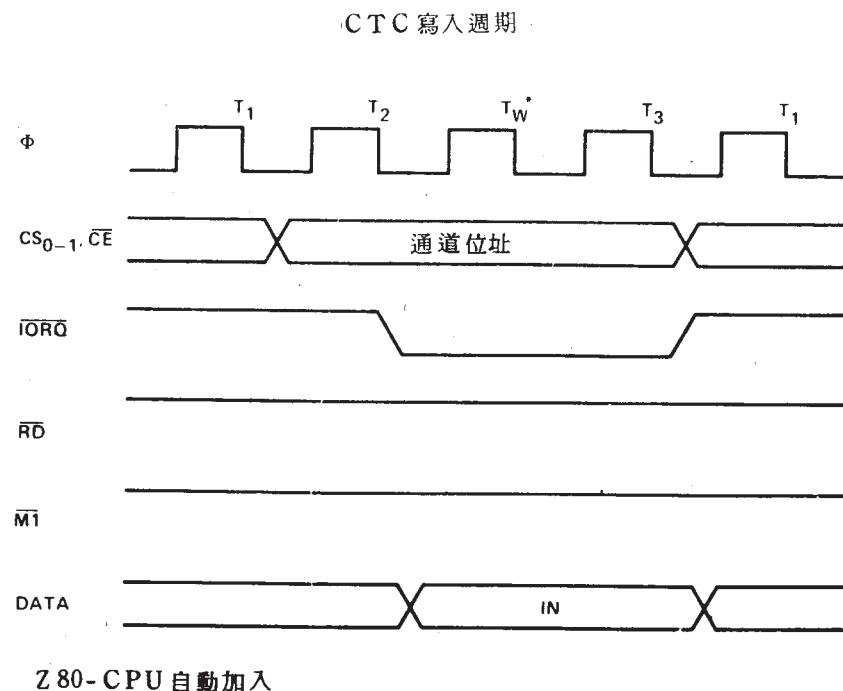
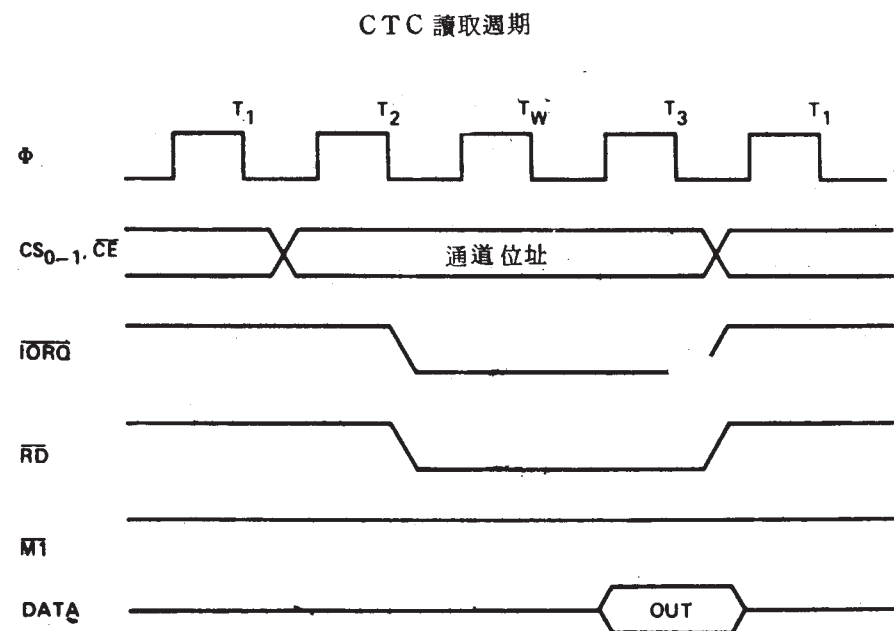


圖 15-25 CTC 寫入週期

圖 15-25 所示即為有關 CTC 寫入週期之時序。此一系列對寫入通道控制字組、時常數資料字組、或插斷向量，皆同樣有效。

如圖所示， T_1 時序週期時，Z80-CPU 準備作寫入作業，CTC 之 \overline{RD} (讀取) 輸入接腳不動作 (維持高電位)。由於 CTC 無個別之寫入信號輸入，故其由此一不動作之 \overline{RD} 信號，私自產生一寫入信號。 T_2 時序週期期間，CTC 之 \overline{IORQ} 與 \overline{CE} 兩輸入接腳同時動作，Z80-CPU 之寫入週期開始。(注意，此時 $\overline{M1}$ 信號無法動作，以避免與插斷認可混淆。) 此時，CTC 之 $CS0$ 與 $CS1$ 兩通道輸入上，出現一選取欲寫入之通道的位址碼。同時，欲寫入之字組亦出現於 Z80 之資料巴士上。在第 T_3 時序週期開始之上升緣時，字組被鎖入適當之 CTC 內部暫存器。

二、讀取週期



* Z80-CPU 自動加入

圖 15-26 CTC 讀取週期

圖 15-26 所示即為 Z80-CTC 之讀取週期時序。此一系列用於 Z80-CPU 讀取倒數計數器內含時。在 T_2 時序週期時， \overline{RD} ， \overline{IORQ} ，與 \overline{CE} 三個信號同時動作，讀取週期開始。此時，CTC 之 $CS0$ 與 $CS1$ (通道選擇輸入 0 與 1) 亦出現指明欲讀取之通道的位址碼。 T_3 時序週期上升緣時，倒數計數器之有效內容出現於 Z80 之資料巴士上。

三、計數與計時

Z80-CTC 之計數器與計時器型態的時序圖如圖 15-27 所示。

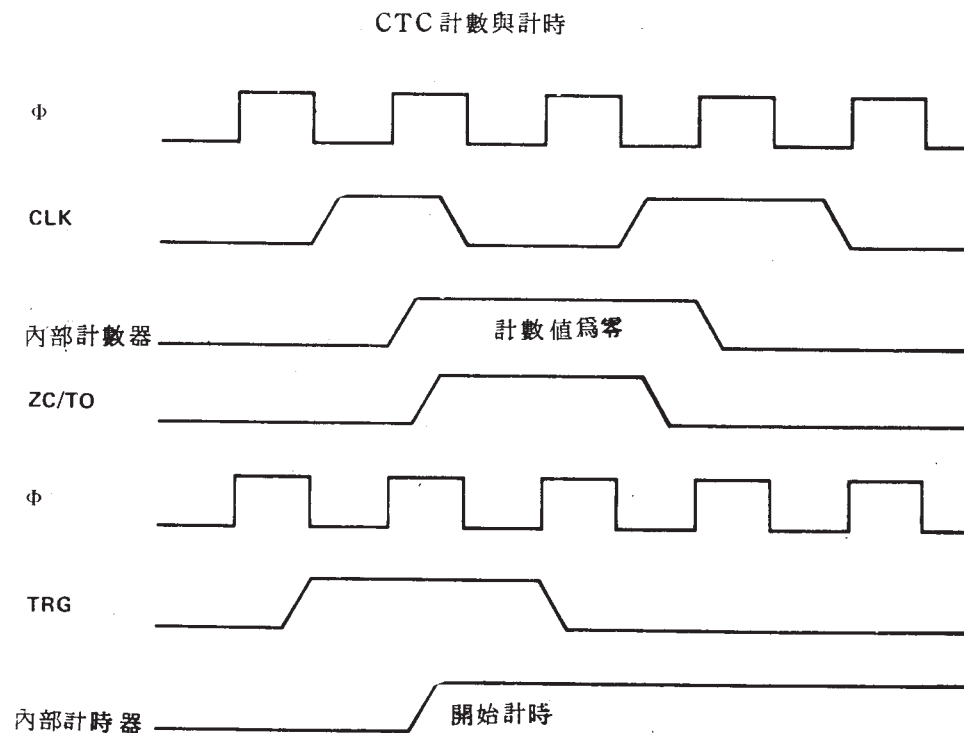


圖 15-27 Z80-CTC 之計數與計時時序

於計數器型態，來自 CLK/TRG 接腳所接之外部硬體的脈衝緣（此例為上升緣動作），在與系統時序 ϕ 取得同步之情況下，令倒數計數器之值減一。此一 CLK/TRG 脈衝必須具有一最低寬度，而其最低週期不可小於系統時序週期之兩倍。雖然 CLK/TRG 動作緣與 ϕ 上升緣間無準備時間之要求，但倘若 CLK/TRG 動作緣太接近於 ϕ 上升緣，則倒數計數器之減一將延遲一個 ϕ 週期。倒數計數器一遞減至零，ZC/TO 接腳即輸出一脈衝。

於計時器型態，CLK/TRG 接腳上之脈衝觸發緣（用者可選擇低電位或高電位動作），在第二個 ϕ 上升緣時，將計時功能致能。正如於計數器型態一般，觸發脈衝以非同步之方式測知，且必須具有一最低寬度。計時功能之起始與系統時序 ϕ 取得同步，且 CLK/TRG 動作緣與次一 ϕ 上升緣之間必須有一最低準備時間。若 CLK/TRG 動作緣太靠近 ϕ 上升緣，則計時器功能之起始將延誤一個 ϕ 週期之時間。

15—2—6 插斷服務

Z80-CTC 之每一通道可個別規劃，令之於每次倒數計數器計數至零時產生插斷。正如其它任何週邊設備一般，CTC 產生插斷之目的即在迫使 CPU 執行一插斷服務常式。使用此一特色時，Z80-CPU 必須設定於插斷型態 2。此一插斷型態要求外部設備在提出插斷請求並且被認可後，插斷設備必須提供一八位元插斷向量，以與 I 暫存器之內含，組合成一指至記憶器之位址表格的十六位元指示器。從該指示器所指之位置（以及其次一位置），CPU 拿取屬於插斷設備之插斷處理常式的起始位址，然後依之跳去執行插斷處理常式。於 CTC，雖然四個通道共有一插斷向量暫存器，但八位元之插斷向量必須是通道所獨有的。

CTC 之插斷控制電路使其插斷作業能與 Z80 系統之插斷處理方式相謀合。每一 CTC 具兩條插斷控制線（IEI 與 IEO），以

期與系統中之其它週邊形成一雛菊花環之優先插斷結構。此種結構之插斷優先順序的決定在前面已討論過，此地不再重覆。此節，我們先看看時序圖，然後再舉一例子說明雛菊花環之插斷服務。

一、插斷認可週期

圖 15-28 舉例說明了插斷認可週期之時序。CTC 提出插斷請求過後一段時間，CPU 送出插斷認可信號： $\overline{M1}$ 與 \overline{IORQ} 同時動作，表示接受插斷請求。為確保菊花環致能線能達穩定， $\overline{M1}$ 動作時，通道之插斷請求狀態保持不變。 $\overline{M1}$ 信號比 \overline{IORQ} 信號約提早兩個時序週期動作，而 \overline{RD} 信號不動作，以避免與指令拿取週期相混淆。此時，CTC 之插斷控制電路將找出提出插斷請求中，具有最高優先之通道。（還記得 CTC 各通道之優先順序吧！）

若 CTC 之 IEI 輸入動作，則 \overline{IORQ} 信號動作時，CTC 內最高優先之插斷設備把插斷向量置於 Z80 之資料巴士上。為確保雛菊

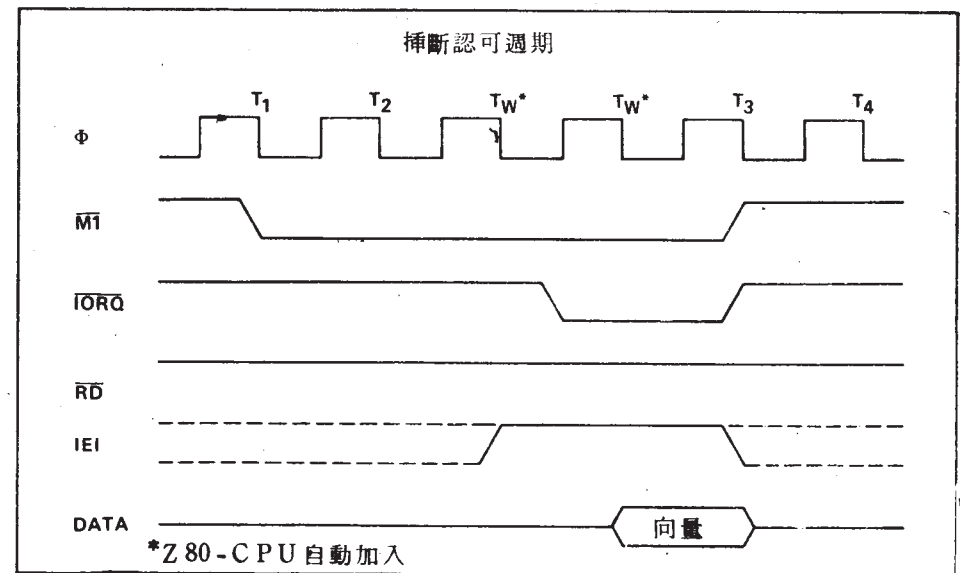


圖 15-28 插斷認可週期

花環達穩定，每一插斷認可週期於此時自動加入兩個等待 T 週期 (T_w)。若不夠，還可以再加上其它之等待週期 (狀態)。

二、插斷回返週期

圖 15-29 所示即為與 RETI 指令有關之時序。RETI 指令置於每一插斷服務常式之末了，以起始雛菊花環結構之插斷致能線，適當控制巢串優先順序之插斷。CTC 於內部自己解碼兩位元組之 RETI 指令，並決定其是否針對目前正在被服務之通道。

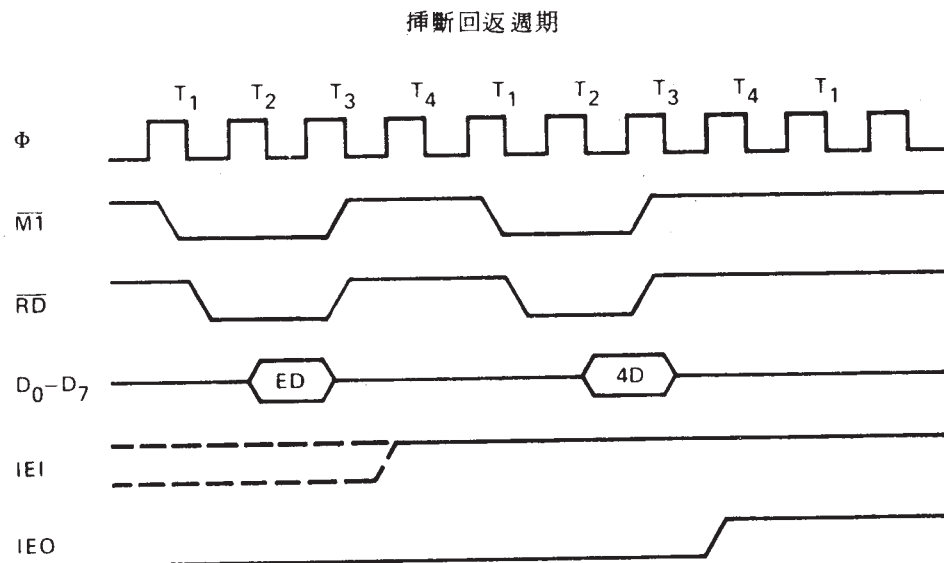


圖 15-29 插斷回返週期

當雛菊花環中有數個 Z80 週邊晶片時，刻在被 CPU 服務之晶片上的 IEI 線於解碼到 EDH 時開始動作。若緊接之運算碼為 4DH

，刻在被服務之週邊會再起始，且其 IEIO 開始動作。時序可加入額外之等待週期。

三、雛菊花環插斷服務

圖 15-30 所示即為 CTC 之內可能發生之一典型巢串插斷系列。於此例中，第 2 通道首先提出插斷請求，並且獲得服務。在其被服務的過程中，具較高優先之第 1 通道亦提出插斷。因第 1 通道具有較高優先，故 CPU 暫停對第 2 通道之服務 (真正的處置是將 CPU 環境推入堆疊器，並獲得新的插斷服務常式之起始位址)，並轉而服務第 1 通道。若服務過程中一直無更高優先之通道或設備提出插斷，則

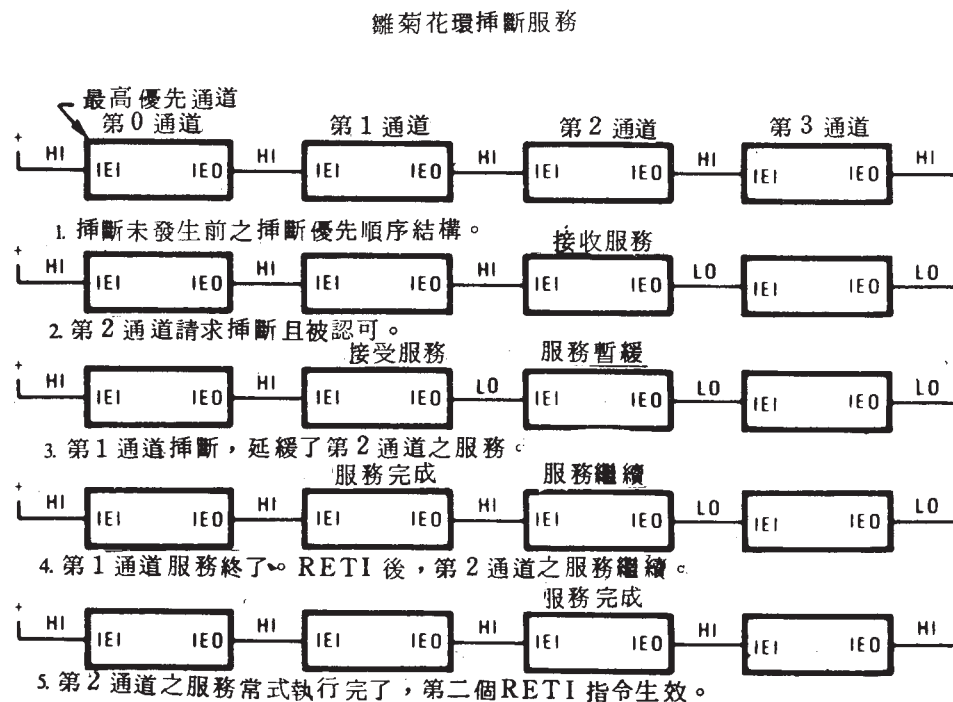


圖 15-30 雛菊花環式之巢串插斷服務舉例。

CPU 對第1通道之服務將順利結束，直至碰上RETI指令為止。此時，CPU 會自動再度拾起對第2通道之繼續服務，直到完成為止。

結 論

Z80 輸入／輸出界面晶片之介紹就至此告一段落。在了解每一界面晶片之功能，內部結構，與如何規劃後，我們即可將之用於實際之Z80 系統。下一章，我們即介紹如何以上述所介紹之界面電路，構成一最低限度的Z80 微電腦系統。

第 16 章

組成一微電腦系統

從一開始到現在，我們一直躲在微處理器與輸入／輸出界面電路之內部，窺其結構與學習如何對之作程式設計。現在，讓我們走出此一象牙塔，看看更廣闊的美麗真實世界。

前面提過，將一微處理器，連接上幾些記憶器與輸入／輸出設備，並加上電源、時序，與必需之軟體，即可構成一運轉之微電腦系統。Z80 由於僅需一單相時序與單一正5伏特電源，致只要加上少數元件，即可構成一簡單之Z80 系統。本章，我們將討論欲構成一Z80 微電腦系統所必先完成之硬體接線工作：Z80 微處理器與ROM 記憶器、靜態RAM記憶器、動態RAM記憶器，以及Z80--PIO 間之簡單連接。

16-1 最小之Z80系統

組成一最小Z80 系統所需之元件如下：

- 1 一正5伏特電源。
- 2 一單相TTL 吻合之時序。
- 3 一種重置（重始）系統之方法。
- 4 ROM或PROM以儲存程式。
- 5 輸入／輸出界面與設備。

6. Z 80 CPU。

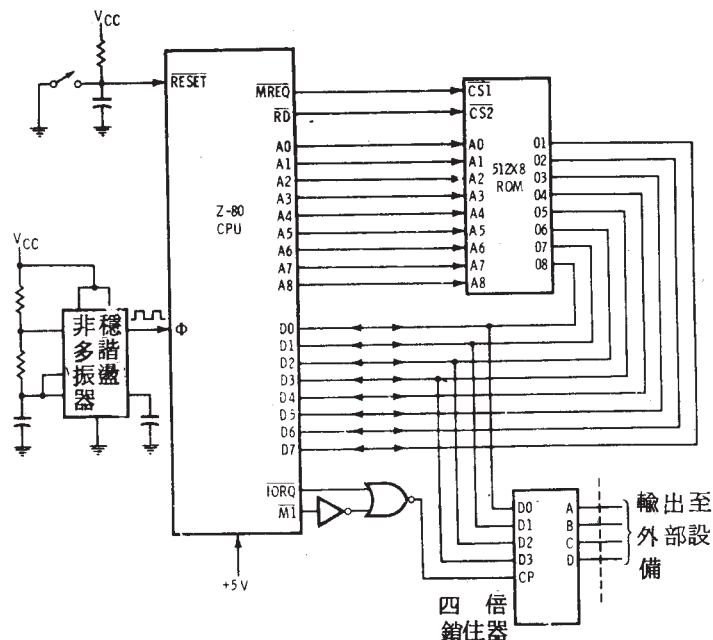


圖 16-1 最小 Z80 系統

圖 16-1 所示即為以上元件所構成之一最小 Z 80 微電腦系統。圖之最左上方，一瞬間開關使 $\overline{\text{RESET}}$ 信號變為邏輯 0 電位，重置 Z 80 CPU，並令其開始執行位址 0 起之程式。正如前一章界面信號所提過者， $\overline{\text{RESET}}$ 信號使插斷禁能，I 與 R 暫存器清除為 0，設定插斷型態 0，以及令程式計數器內含為 0。圖左下方為一能產生頻率 2.5 至 4.0 MHz 方波輸入之時序電路。該時序電路維持連續不斷地產生 -5 伏特之方波。不動作於最高速之系統可使用 RC 振盪器。但若 CPU 動作於近乎最高速，系統就必須使用晶體振盪器，因為系統無法容忍 RC 網路所產生之頻率飄移。晶體振盪目前可由反向器及一些分離元件或單晶積體電路組成。

圖右上方所示之 ROM 記憶器，為一快速存取（存取時間大於

250 ns），容量為 512×8 之 ROM。Z 80 每次以九條位址線（A0 ~ A 8）選取其中之一記憶位置。即使對 4 MHz 之時序頻率而言，由於記憶器永遠能及時對 CPU 之讀取反應，故不需有等待（WAIT）狀態。ROM 之輸出為三態輸出，在沒被選取時，其所有輸出線均處於高阻抗狀態。此八條輸出線直接接至 Z 80 之資料巴士線 D₀ 至 D₇。輸出設備則為一四倍鎖住器，於輸出作業時，Z 80 CPU 透過 D₀ 至 D₃ 等資料線，將資料寫入其四個正反器。此一設備由 $\overline{\text{M1}}$ 且 $\overline{\text{IORQ}}$ 信號選取。

當重置開關按下時， $\overline{\text{RESET}}$ 輸入低電位，CPU 起始，且開始執行位址 0 之 ROM 位置上的程式。讀取程式指令時，Z 80 CPU 對 ROM 送出 $\overline{\text{MREQ}}$ 與 $\overline{\text{RD}}$ 兩控制信號。（當然，其它尚有位址資訊。）此兩控制信號使圖中 ROM 之兩晶片選擇輸入（ $\overline{\text{CS1}}$ 與 $\overline{\text{CS2}}$ ）同時變低電位，且令 A₀ ~ A₈ 所選取之記憶位置的內含輸出至資料巴士上。由於 A₉ 至 A₁₅ 等位址線沒接，致其內含可為任意數值。（更明確說，所有位址必須落於 0 至 511₁₀ 之範圍。）程式可能需要少量記憶以儲存變數值，此些記憶由 CPU 內之 14 個暫存器提供。由於沒有 RAM 記憶器，致系統不具記憶堆疊作業。

資料輸出由鎖住器提供。由於其為系統唯一僅有之輸入 / 輸出設備，致帶任意輸入 / 輸出位址之任意輸入 / 輸出指令，都會選取此一鎖住器。並於輸入 / 輸出週期 $\overline{\text{IORQ}}$ 與 $\overline{\text{M1}}$ 信號發生期間，將資料巴士線 D₀ ~ D₃ 之內含鎖住於鎖住器內。注意，如同 ROM 一般，輸入 / 輸出亦無 $\overline{\text{RD}}$ 或 $\overline{\text{WR}}$ 之解碼。鎖住器之輸出線 A，B，C 及 D 直接與外界連接。

毫無疑問地，上述例子代表一種相當侷限之應用情況。其真正僅能作為最簡單 Z 80 系統結構舉例之用。不過，即使以此一簡單系統，我們仍可以各種不同之程式，完成諸如下列之應用：

- 1 以輸出鎖住器播放音樂。
- 2 做簡單之數位對類比輸出（加上某些額外元件）。

3 產生任意期限之時序。

4 做自動電話撥號(加些額外之邏輯)。

16-2 ROM與RAM之界面

圖 16-2 所示為一具有 ROM (或 PROM) 與 RAM 記憶器, 以及有限輸入 / 輸出能力之一更有用系統。ROM 之容量增大為 1024 個位元組 (1K × 8)。兩個 256 × 4 之高速 RAM (毋需等待狀態) 提供了 256 位元組之動態變數讀寫記憶。其中一 RAM 提供低次四位元, 另一 RAM 提供高次四位元。RAM (與系統之所有組件) 均為三態元件, 致所有輸入與輸出均可以“結合 OR”(wire-OR) 之方式連接至資料巴士線。輸出同樣以一四倍之鎖住器接收來自 CPU 之四位元輸出, 而來自各部設備之輸入則由 G 1 至 G 4 四個閘攫取。

此一結構之記憶分佈圖如圖 16-3 所示。ROM 記憶區位於位址 0000 H 至 03 FF H 等記憶位置。RAM 記憶區則位於位址 FF 00 H 至 FFFF H 之記憶位置 (256 個位置)。A 10 至 A 14 之位址線不用。只要 A 15 位址線為 0, 則選取 ROM 記憶; 反之, A 15 = 1 則選取 RAM 記憶。由於 Z 80 採用隔離式輸入 / 輸出, 故輸入 / 輸出位

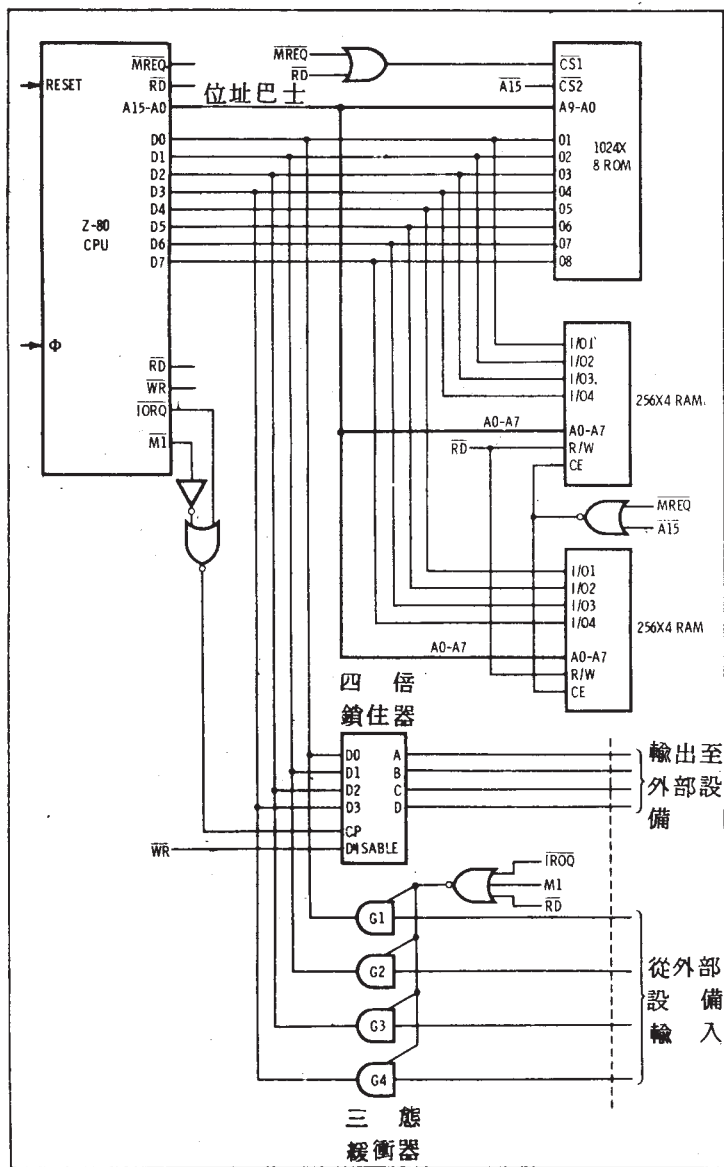


圖 16-2 Z 80 CPU 與 ROM 及 RAM 之界面

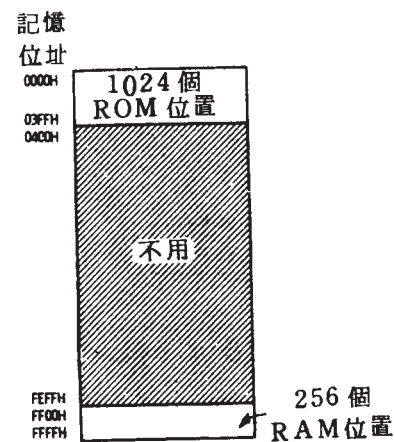


圖 16-3 ROM 及 RAM 記憶分佈圖

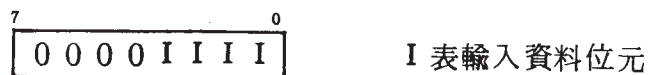
址與記憶位址分開。正如前述例子一般，任一輸入 / 輸出位址若不選取輸入設備，就會選取輸出設備。

同理，當重置開關按下時，Z 80 CPU 開始執行位址 0 之 ROM 位置上的指令。運算過程中，程式可存取位址 FF00H 至 FFFFH 之 RAM 位置。同時，藉著適當設定堆疊指示器之內含值，程式可劃定某一部份 RAM 記憶區作為堆疊器。由於 ROM 記憶器之所有存取必定皆為讀取，故 RD 信號連接至 ROM 可算是多餘的。RAM 之讀 / 寫輸入則來自 CPU 之 \overline{WR} 信號。

輸入 / 輸出之處置與前一節之例子類似。當 Z 80 CPU 執行輸出指令時， \overline{IORQ} ， $\overline{M1}$ ，與 \overline{WR} 信號同時動作。資料巴士 D0 至 D3 輸出於輸出週期內被寫入輸出鎖住器。當 Z 80 CPU 執行輸入指令時

\overline{IORQ} ， $\overline{M1}$ ，與 \overline{RD} 信號線動作，使程式能攫取輸入資料線 I 0 至 I 3 上之輸入資料。輸入與輸出之資料格式如圖 16-4 所示。輸出時，第 4 至第 7 資料位元被忽略。輸入時，第 4 至第 7 資料位元恆為 0。

輸入資料格式



輸出資料格式



圖 16-4 輸入 / 輸出資料格式

即使僅具有最低數量之記憶，但圖 16-2 所示之系統却為一十分能幹之系統。由於其同時具有輸入與輸出能力，故此一系統可用以：

- 1 解碼 BCD 輸入，並於處理後輸出 BCD 結果。
- 2 以開關作偵測盜竊或火警之輸入，然後以顯示燈或警鈴作輸出，製作一“看家”系統。
- 3 輸入代表諸如瞬間速度或其它類比量之數位信號，並加以處理。
- 4 提供數位對類比輸入之解碼，並產生類比對數位之輸出。

16-3 動態記憶之界面

若僅就資料與運算元之讀寫而言，動態 RAM 記憶器之界面與靜態 RAM 記憶器大體相同。不過，由於電路特性的關係，動態 RAM 記憶器之每一記憶槽的記憶內含必須不斷地加以復新 (refresh)，以便確保其記憶內含不因時間而消失。此即所謂的記憶復新 (memory refresh)。事實上，記憶復新就是每隔 2 毫秒左右，便對一記憶槽作一次讀取，但並不真正讀取其內含。

藉著使用 R 暫存器，Z 80 有一套自動產生記憶復新週期之辦法。在每一指令週期之 $\overline{M1}$ 期間， $\overline{M1}$ 信號與記憶存取請求 \overline{MREQ} 信號同時動作，以告知外部動態 RAM 記憶器，可能有一復新週期要發生。RAM 然後利用由 R 暫存器送至資料巴士之現有位址，進行記憶復新。由於 R 暫存器之內含會不斷地由 0 增至 255，故動態 RAM 記憶器亦可不斷地獲得新的位址。

圖 16-5 所示即為一以 8 個 4096×1 動態 RAM 組成之 4096×8 動態 RAM 記憶器。每一 RAM 具有 12 個位址輸入，分成六個列輸入與六個行輸入。記憶復新的要求是，於 2 毫秒之週期內，64 列的每一列都必須要選到 (亦即都要做一次復新)。由於在程式執行期間，資料任意存取，無法確保必能達此要求，故這必須有系統加以計劃與執行。為達此目的，我們乃以 \overline{RFSH} 與 \overline{MREQ} 兩信號作 AND，如圖所示。若兩信號同時動作，通知 \overline{CE} 輸入，則晶片記憶復新致能 (開始動作)，根據 A 0 至 A 5 位址線之列位址，八個晶片之每一晶片均被讀取。由於總共需 64 個復新週期，方能將一晶片內之所有記

憶槽均復新完畢，故作一次完全之復新平均需時 $64 \times N$ 之時間，其中， N 為 Z 80 之平均指令執行時間。若以 $N = 2.5$ 微秒計，則所有 4K 位元組均復新一次需時 160 微秒。注意，當指令執行期間，Z 80 CPU 在存取某一位元組之資料時，一般之非復新讀取或寫入週期，同樣會使晶片選取輸入 CE 致能（動作）。

於 16-5 圖中，RAM 選擇線 RAM SELECT 事實上就是其中一條位址巴士線，如 A_{12} 。若系統欲加上額外之記憶器，則此一選擇線就必須先經過解碼器。

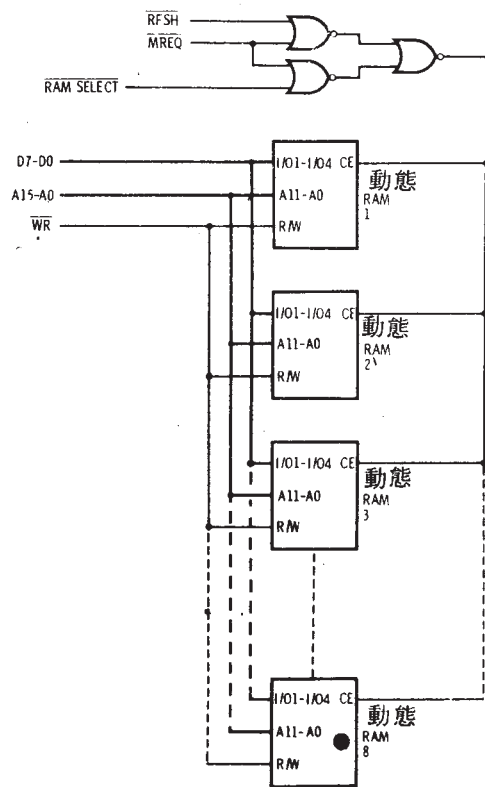


圖 16-5 動態 RAM 記憶復新

16-4 記憶器速度控制

在許多應用上，爲了降低系統之製造成本，經常會採用反應速度較慢之記憶器。Z 80-CPU 之 WAIT 接腳，使其能與任何速度之記憶器相組合。參閱 CPU 時序一節即可發現，記憶器存取時間（反應速度）之要求，於指令拿取之 M1 週期內最嚴厲。爲此，於前述之情況下，M1 週期通常需加一等待週期，以使慢速之記憶器能配合 CPU 之作業。圖 16-6 所示即爲一能達成此項任務之一簡單電路範例。此一電路可經修改成如圖 16-7 所示，對任意記憶器存取週期加上一等待週期之電路。

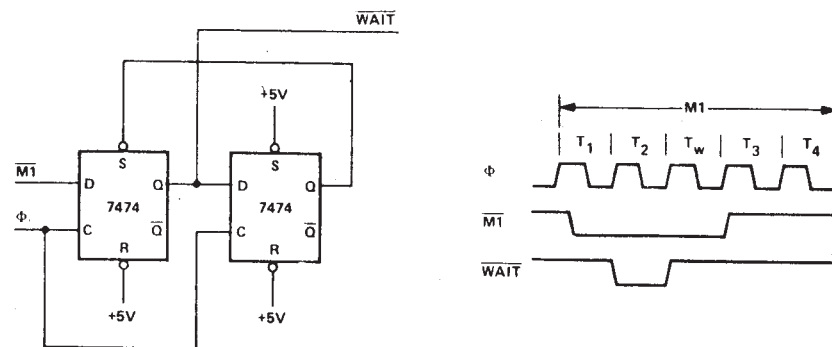


圖 16-6 對 M1 週期加一等待週期

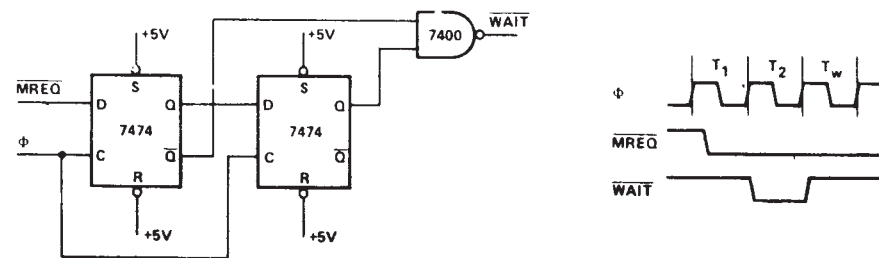


圖 16-7 對任意記憶週期加一等待週期

16-5 Z80 PIO之界面

Z 80 PIO 之結構在前面已介紹過，圖 16-8 所示即為以一 PIO 取代圖 16-2 系統中之四條資料輸出線與四條輸入線的情形，此一取代不僅雙倍了 Z 80 系統之輸入 / 輸出能力，同時亦提供了完整之控制邏輯。Z 80 之資料巴士線連接至 PIO 之資料線輸入。A 口被令為輸出口，CPU 用之將資料寫至外部設備。B 口為輸入口，用以由外部設備輸入（讀取）資料。兩組握手連絡控制線之用法相同。Z 80 PIO 輸出之 INT 信號直接輸入至 CPU。由於系統中僅有一 PIO，故毋需有雛菊花環式之插斷優先順序結構，並且 IEI 設定成 V_{CC} ，而 IE0 不用。M1，IORQ，與 RD 等輸入則直接連接至對等之 Z 80 CPU 接腳。

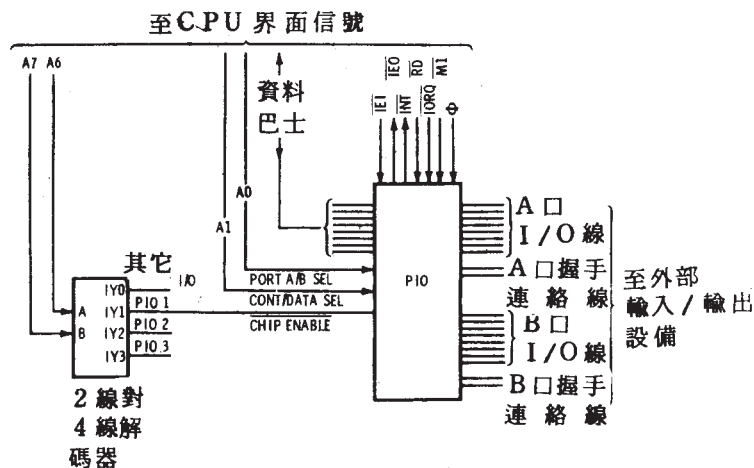


圖 16-8 PIO 於最小 Z 80 系統中之配置

由於 PIO 為系統唯一之輸入 / 輸出設備，故不需解碼輸入 / 輸出位址。不過，16-8 圖中仍顯示一可用於較大系統之解碼電路。如圖所示，PIO 之位址決定於輸入 / 輸出位址之兩最高次位元：A 6 與 A 7。是故，系統最多共可使用四個 PIO。16-8 圖假設系統具有

三個 PIO 以及其它位於 00000000_2 至 $00XXXXXX_2$ 間之輸入 / 輸出位址。任何具有 $01XXXXXX_2$ 形式之位址將使解碼器輸出選取第一個 PIO。A / B 口選擇線連接至位址線 A 0，且控制 / 資料選擇線連接至位址線 A 1。此一位址配置所造成之位址分佈情形為：

輸入 / 輸出位址 (資料巴士之低次八位元)	所選取之設備 (意義)
0 0 0 0 0 0 0 0	非 PIO 位址
0 0 1 1 1 1 1 1	
0 1 X X X X 0 0	A 口，資料
0 1 X X X X 0 1	B 口，資料
0 1 X X X X 1 0	A 口，控制
0 1 X X X X 1 1	B 口，控制
1 0 0 0 0 0 0 0	其它可擴展之 PIO
:	:
:	:
1 1 1 1 1 1 1 1	

欲以非插斷控制之方式自輸入 / 輸出設備輸入或輸出資料，所必須採取的步驟為：

- 1 重置 PIO (電源打開)。此舉清除插斷致能狀態以及 PIO 插斷向量暫存器。
- 2 將插斷控制字組或 07 H 取入 CPU 之 R 暫存器。並以輸出指令輸出至 01000010_2 與 01000011_2 兩位址位置。此舉將 A 口及 B 口之 PIO 插斷禁能。
- 3 將作業型態控制字組 00001111_2 取入 CPU 之 R 暫存器，並以輸出指令輸出至 01000010_2 之位址位置。此舉將 A 口設定為輸出口。

- 4 將作業型態控制字組 01001111₂ 取入 Z 80 CPU 之 R 暫存器，然後以輸出指令輸出至位址 01000011₂ 之位置。此舉將 B 口設定為輸入口。
- 5 自位址 01000001₂ 之設備輸入資料。此一輸入資料來自 B 口。最初之資料被捨棄，不過，B 口之 RDY 線變成動作狀態，通知外部設備，CPU 已準備好可開始讀取資料。
- 6 此時，A 口已可以開始輸出資料，且 B 口可以開始輸入資料。由於未採用插斷作業，故輸出必須定好時間，以使外部設備有充裕之時間對輸出採取反應並提供輸入資料。此 PIO 之讀取與寫入推動器上，均必須設有計時迴路。

若欲以插斷方式履行輸入 / 輸出，則首先插斷向量暫存器之內含必須輸出至 PIO，然後必須再輸出適當之控制字組。假設輸出與輸入設備之插斷處理常式的位址分別儲存於 FF00H 與 FF02H 之位置，則在輸入 / 輸出活動之前，Z 80 CPU 之 I 暫存器必須先存 FFH。其次，有時在第一個輸入 / 輸出活動之前，PIO 之插斷向量暫存器必須先如下地取入：

將插斷向量控制字組 00H 取入 CPU 之 R 暫存器。並以輸出指令輸出至位址 01000010₂ 之設備。此舉將 B 口之插斷向量暫存器內含設定為 02H。

最後，插斷控制字組 80H 必須輸出至設備位址 01000010₂ 以及 01000011₂。此舉使 A 口及 B 口之插斷致能。每當外部設備取走輸出資料時（A STB 瞬間變為低電位），A 口之插斷就會發生。同理，每當外部設備將資料送入 PIO 之輸入暫存器時（B STB 瞬間變為低電位），B 口之插斷亦會發生。A 口之插斷將使微處理器自 FF00H 及 FF01H 兩記憶位置拿取插斷處理常式之起始位址，並開始執行常式。B 口之插斷將使微處理器自 FF02H 與 FF03H 兩記憶位置拿取插斷處理常式之起始位址，並開始執行該位址起之插斷處理

常式的指令。

以上的說明舉例了 Z 80 PIO 之界面。圖中所示之結構可作極廣泛之用途，電傳打字機之輸入 / 輸出，鍵盤解碼，數位對類比或類比對數位之輸入 / 輸出，與 16 線之程序控制應用等。利用類似以上所述之程序，讀者諒必明白如何將多個 PIO 以及額外之 ROM，PROM，或 RAM 連接至系統中。

附 錄 A

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	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15

附 錄 B

ASCII 轉換表

HEX	MSD	0	1	2	3	4	5	6	7
LSB	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	DEL
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
EOT	- End of Transmission	ETB	- End of Transmission Block
ENQ	- Enquiry	CAN	- Cancel
ACK	- Acknowledge	EM	- End of Medium
BEL	- Bell	SUB	- Substitute
BS	- Backspace	ESC	- Escape
HT	- Horizontal Tabulation	FS	- File Separator
LF	- Line Feed	GS	- Group Separator
VT	- Vertical Tabulation	RS	- Record Separator
FF	- Form Feed	US	- Unit Separator
CR	- Carriage Return	SP	- Space (Blank)
SO	- Shift Out	DEL	- Delete
SI	- Shift In		

附 錄 C

相對跳越位移表

向前相對跳越

LSD 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 MSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	128	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113
9	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97
A	96	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81
B	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65
C	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
D	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
E	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
F	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

附 錄 D

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

附 錄 E

Z80 指令碼

機器碼	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

機器碼	組合語言述句
E620	AND n
CB46	BIT 0,(HL)
DDCB0546	BIT 0,(IX+d)
FDCB0546	BIT 0,(IY+d)
CB47	BIT 0,A
CB40	BIT 0,B
CB41	BIT 0,C
CB42	BIT 0,D
CB43	BIT 0,E
CB44	BIT 0,H
CB45	BIT 0,L
CB4E	BIT 1 (HL)
DDCB054E	BIT 1,(IX+d)
FDCB054E	BIT 1,(IY+d)
CB4F	BIT 1,A
CB48	BIT 1,B
CB49	BIT 1,C
CB4A	BIT 1,D
CB4B	BIT 1,E
CB4C	BIT 1,H
CB4D	BIT 1,L
CB56	BIT 2,(HL)
DDCB0556	BIT 2,(IX+d)
FDCB0556	BIT 2,(IY+d)
CB57	BIT 2,A
CB50	BIT 2,B
CB51	BIT 2,C
CB52	BIT 2,D
CB53	BIT 2,E
CB54	BIT 2,H
CB55	BIT 2,L
CB5E	BIT 3,(HL)
DDCB055E	BIT 3,(IX+d)
FDCB055E	BIT 3,(IY+d)
CB5F	BIT 3,A
CB58	BIT 3,B
CB59	BIT 3,C
CB5A	BIT 3,D
CB5B	BIT 3,E
CB5C	BIT 3,H
CB5D	BIT 3,L
CB66	BIT 4,(HL)
DDCB0566	BIT 4,(IX+d)
FDCB0566	BIT 4,(IY+d)
CB67	BIT 4,A
CB60	BIT 4,B
CB61	BIT 4,C
CB62	BIT 4,D

機器碼	組合語言述句
CB63	BIT 4,E
CB64	BIT 4,H
CB65	BIT 4,L
CB6E	BIT 5,(HL)
DDCB056E	BIT 5,(IX+d)
FDCB056E	BIT 5,(IY+d)
CB6F	BIT 5,A
CB68	BIT 5,B
CB69	BIT 5,C
CB6A	BIT 5,D
CB6B	BIT 5,E
CB6C	BIT 5,H
CB6D	BIT 5,L
CB76	BIT 6,(HL)
DDCB0576	BIT 6,(IX+d)
FDCB0576	BIT 6,(IY+d)
CB77	BIT 6,A
CB70	BIT 6,B
CB71	BIT 6,C
CB72	BIT 6,D
CB73	BIT 6,E
CB74	BIT 6,H
CB75	BIT 6,L
CB7E	BIT 7,(HL)
DDCB057E	BIT 7,(IX+d)
FDCB057E	BIT 7,(IY+d)
CB7F	BIT 7,A
CB78	BIT 7,B
CB79	BIT 7,C
CB7A	BIT 7,D
CB7B	BIT 7,E
CB7C	BIT 7,H
CB7D	BIT 7,L
DC8405	CALL C,nn
FC8405	CALL M,nn
D48405	CALL NC,nn
C48405	CALL NZ,nn
F48405	CALL P,nn
EC8405	CALL PE,nn
E48405	CALL PO,nn
CC8405	CALL Z,nn
CD8405	CALL nn
3F	CCF
BE	CP (HL)
DDBE05	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
EDB9	CPDR

機器碼	組合語言述句
EDB1	CPIR
EDA1	CPI
2F	CPL
27	DAA
35	DEC (HL)
DD3505	DEC (IX+d)
FD3505	DEC (IY+d)
3D	DEC A
05	DEC B
0B	DEC BC
0D	DEC C
15	DEC D
1B	DEC DE
1D	DEC E
25	DEC H
2B	DEC HL
DD2B	DEC IX
FD2B	DEC IY
2D	DEC L
3B	DEC SP
F3	DI
102E	DJNZ e
EB	EI
E3	EX (SP),HL
DDE3	EX (SP),IX
FDE3	EX (SP),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
DB20	IN A,(e)

機器碼	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)
DA8405	JP C,nn
FA8405	JP M,nn
D28405	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 Z,e
182E	JR e,HL
02	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
FD7705	LD (IY+d),A
FD7005	LD (IY+d),B
FD7105	LD (IY+d),C
FD7205	LD (IY+d),D
FD7305	LD (IY+d),E
FD7405	LD (IY+d),H
FD7505	LD (IY+d),L
FD360520	LD (IY+d),n
328405	LD (nn),A
ED438405	LD (nn),BC
ED538405	LD (nn),DE
228405	LD (nn),HL
DD228405	LD (nn),IX
FD228405	LD (nn),IY
ED738405	LD (nn),SP
0A	LD A,(BC)
1A	LD A,(DE)
7E	LD A,(HL)

機器碼	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
ED488405	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
ED589405	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

機器碼	組合語言述句
5B	LD E,E
5C	LD E,H
5D	LD E,L
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
ED788405	LD SP,(nn)
F9	LD SP,HL
DDF9	LD SP,IX
FDf9	LD SP,IY
318405	LD SP,nn
EDA8	LDD
ED88	LDDR
EDA0	LDI
ED80	LDIR
ED44	NEG
00	NOP
B6	OR (HL)
DD8605	OR (IX+d)
FD8605	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

機器碼	組合語言述句
EDB3	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
CB86	RES 0,(HL)
DDCB0586	RES 0,(IX+d)
FDCB0586	RES 0,(IY+d)
CB87	RES 0,A
CB80	RES 0,B
CB81	RES 0,C
CB82	RES 0,D
CB83	RES 0,E
CB84	RES 0,H
CB85	RES 0,L
CB8E	RES 1,(HL)
DDCB058E	RES 1,(IX+d)
FDCB058E	RES 1,(IY+d)
CB8F	RES 1,A
CB88	RES 1,B
CB89	RES 1,C
CB8A	RES 1,D
CB8B	RES 1,E
CB8C	RES 1,H
CB8D	RES 1,L
CB96	RES 2,(HL)
DDCB0596	RES 2,(IX+d)
FDCB0596	RES 2,(IY+d)
CB97	RES 2,A
CB90	RES 2,B
CB91	RES 2,C
CB92	RES 2,D
CB93	RES 2,E
CB94	RES 2,H
CB95	RES 2,L
CB9E	RES 3,(HL)
DDCB059E	RES 3,(IX+d)
FDCB059E	RES 3,(IY+d)

機器碼	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
CBA6	RES 4,(HL)
DDCB05A6	RES 4,(IX+d)
FDCB05A6	RES 4,(IY+d)
CBA7	RES 4,A
CBA0	RES 4,B
CBA1	RES 4,C
CBA2	RES 4,D
DBA3	RES 4,E
CBA4	RES 4,H
CBA5	RES 4,L
CBAE	RES 5,(HL)
DDCB05AE	RES 5,(IX+d)
FDCB05AE	RES 5,(IY+d)
CBAF	RES 5,A
CBA8	RES 5,B
CBA9	RES 5,C
CBAA	RES 5,D
CBA8	RES 5,E
CBAC	RES 5,H
CBAD	RES 5,L
CBB6	RES 6,(HL)
DDCB05B6	RES 6,(IX+d)
FDCB05B6	RES 6,(IY+d)
CBB7	RES 6,A
CBB0	RES 6,B
CBB1	RES 6,C
CBB2	RES 6,D
CBB3	RES 6,E
CBB4	RES 6,H
CBB5	RES 6,L
CBBE	RES 7,(HL)
DDCB05BE	RES 7,(IX+d)
FDCB05BE	RES 7,(IY+d)
CBBF	RES 7,A
CBB8	RES 7,B
CBB9	RES 7,C
CBBA	RES 7,D
CBBB	RES 7,E
CBBC	RES 7,H
CBBD	RES 7,L
C9	RET
D8	RET C
F8	RET M
D0	RET NC
C0	RET NZ
F0	RET P
E8	RET PE
E0	RET PO
C8	RET Z

機器碼	組合語言述句
ED4D	RETI
ED45	RETN
CB16	RL (HL)
DDCB0516	RL (IX+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 (IX+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 (IX+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	RRA
CB0E	RRC (HL)
DDCB050E	RRC (IX+d)
FDCB050E	RRC (IY+d)
CB0F	RRC A
CB08	RRC B
C309	RRC C
CB0A	RRC D
CB0B	RRC E
CB0C	RRC H
CB0D	RRC L
OF	RRCA
ED67	RRD
C7	RST 00H
CF	RST 08H
D7	RST 10H
DF	RST 18H
E7	RST 20H
EF	RST 28H
F7	RST 30H
FF	RST 38H
DE20	SBC A,n

機器碼	組合語言述句
9E	SBC A,(HL)
DD9E05	SBC A,(IX+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 HL,BC
ED52	SBC HL,DE
ED62	SBC HL,HL
ED72	SBC HL,SP
37	SCF
CB06	SET 0,(HL)
DDCB05C6	SET 0,(IX+d)
FDCB05C6	SET 0,(IY+d)
CB07	SET 0,A
CB00	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,(IX+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,E
CB0C	SET 1,H
CB0D	SET 1,L
BD06	SET 2,(HL)
DDCB05D6	SET 2,(IX+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,(IX+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
CBE6	SET 4,(HL)

機器碼	組合語言述句
DDCB05E6	SET 4,(IX+d)
FDCB05E6	SET 4,(IY+d)
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,(IX+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
CB0E	SET 6,(HL)
DDCB05F6	SET 6,(IX+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,(IX+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
CB26	SLA (HL)
DDCB0526	SLA (IX+d)
FDCB0526	SLA (IY+d)
CB27	SLA A
CB20	SLA B
CB21	SLA C
CB22	SLA D
CB23	SLA E
CB24	SLA H
CB25	SLA L
CB2E	SRA (HL)
DDCB052E	SRA (IX+d)
FDCB052E	SRA (IY+d)
CB2F	SRA A
CB28	SRA B
CB29	SRA C
CB2A	SRA D

附 錄 F

Z80 與 8080 對等

機器碼	組合語言述句	
CB2B	SRA	E
CB2C	SRA	H
CB2D	SRA	L
CB3E	SRL	(HL)
DDCB053E	SRL	(IX+d)
FDCB053E	SRL	(IY+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	(IY+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	(IY+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.)

Z80	8080	Z80	8080	Z80	8080
ADCA, (HL)	ADCM	EX (SP), HL	XTHL	OR n	ORI [B2]
ADCA, n	ACI [B2]	HALT	HLT	OR r	ORA r
ADC A, r	ADC r	IN A, (n)	IN [B2]	OR (HL)	ORA M
ADD A, (HL)	ADD M	INC BC	INX B	OUT (n), A	OUT [B2]
ADD A, n	ADI [B2]	INC DE	INX D	POP AF	POP PSW
ADD A, r	ADD r	INC HL	INX H	POP BC	POP B
ADD HL, BC	DAD B	INC r	INR r	POP DE	POP D
ADD HL, DE	DAD D	INC SP	INX SP	POP HL	POP H
ADD HL, HL	DAD H	INC (HL)	INR M	PUSH AF	PUSH PSW
ADD HL, SP	DAD SP	JP C, nn	JC [B2] [B3]	PUSH BC	PUSH B
AND n	ANI [B2]	JP M, nn	JM [B2] [B3]	PUSH DE	PUSH D
AND r	ANA r	JP NC, nn	JNC [B2] [B3]	PUSH HL	PUSH H
AND (HL)	ANA M	JP nn	JMP [B2] [B3]	RET	RET
CALL C, nn	CC [B2] [B3]	JP NZ, nn	JNZ [B2] [B3]	RET C	RC
CALL M, nn	CM [B2] [B3]	JP P, nn	JP [B2] [B3]	RET M	RM
CALL NC, nn	CNC [B2] [B3]	JP PE, nn	JPE [B2] [B3]	RET NC	RNC
CALL nn	CALL	JP PO, nn	JPO [B2] [B3]	RET NZ	RNZ
CALL NZ, nn	CNZ [B2] [B3]	JP Z, nn	JZ [B2] [B3]	RET P	RP
CALL P, nn	CP [B2] [B3]	JP (HL)	PCHL	RET PE	RPE
CALL PE, nn	CPE [B2] [B3]	LD A, (DE)	LDAX	RET PO	RPO
CALL PO, nn	CPO [B2] [B3]	LDA, (nn)	LDA [B2] [B3]	RET Z	RZ
CALL Z, nn	CZ [B2] [B3]	LD DE, nn	LXID, [B2] [B3]	RLA	RAL
CCF	CMC	LD SP, nn	LXI SP, [B2] [B3]	RLCA	RLC
CP r	CMP r	LD (BC), A	STAX B	RRA	RAR
CP (HL)	CMP M	LD (DE), A	STAX D	RRCA	RRC
CPL	CMA	LD (HL), r	MOV M, r	RST P	RST P
CP n	CPI [B2]	LD (nn), A	STA [B2] [B3]	SBC A, (HL)	SBB M
DAA	DAA	LD (nn), HL	SHLD [B2] [B3]	SBC A, n	SBI [B2]
DEC BC	DCX B	LD A, (BC)	LDAX B	SBC A, r	SBB r
DEC DE	DCX D	LD BC, nn	LXIB, [B2] [B3]	SCF	STC
DEC HL	DCX H	LD HL, (nn)	LHLD [B2] [B3]	SUB n	SUI [B2]
DEC r	DCR r	LD HL, nn	LXI H [B2] [B3]	SUB r	SUB r
DEC SP	DCX SP	LD r, (HC)	MOV I, M	SUB (HL)	SUB M
DEC (HL)	DCR M	LD r, n	MVI r, [B2]	XOR n	XRI [B2]
DI	DI	LD r, r'	MOV r1, r2	XOR r	XRA r
EI	EI	LD SP, HL	SPHL	XOR (HL)	XRA M
EX DE, HL	XCHG	NOP	NOP		

附 錄 G

8080 與 Z80 對等

版權所有

翻印必究

8080	Z80	8080	Z80	8080	Z80
ACI [B2]	ADC A, n	IN [B2]	IN A, (n)	POP H	POP HL
ADC M	ADC A, (HL)	INR M	INC (HL)	POP PSW	POP AF
ADC r	ADC A, :	INR r	INC r	PUSH B	PUSH BC
ADD M	ADD A, (HL)	INX B	INC BC	PUSH D	PUSH DE
ADD r	ADD A, r	INX D	INC DE	PUSH H	PUSH HL
ADI [B2]	ADD A, n	INX H	INC HL	PUSH PSW	PUSH AF
ANA M	AND (HL)	INX SP	INC SP	RAL	RLA
ANA r	AND r	JC [B2] [B3]	JP C, nn	RAR	RRA
ANI [B2]	AND n	JM [B2] [B3]	JP M, nn	RC	RET C
CALL	CALL nn	JMP [B2] [B3]	JP nn	RET	RET
CC [B2] [B3]	CALL C, nn	JNC [B2] [B3]	JP NC, nn	RLC	RLCA
CM [B2] [B3]	CALL M, nn	JNZ [B2] [B3]	JP NZ, nn	RM	RET M
CMA	CPL	JP [B2] [B3]	JPP, nn	RNC	RET NC
CMC	CCF	JPE [B2] [B3]	JP PE, nn	RNZ	RET NZ
CMP M	CP (HL)	JPO [B2] [B3]	JP PO, nn	RP	RET P
CMP r	CP r	JZ [B2] [B3]	JP Z, nn	RPE	RET PE
CNC [B2] [B3]	CALL NC, nn	LDA [B2] [B3]	LD A, (nn)	RPO	RET PO
CNZ [B2] [B3]	CALL NZ, nn	LDAX B	LD A, (BC)	RRC	RRCA
CP [B2] [B3]	CALL P, nn	LDAX D	LD A, (DE)	RST	RST P
CPE [B2] [B3]	CALL PE, nn	LH LD [B2] [B3]	LD HL, (nn)	RZ	RET Z
CPI [B2]	CP n	LXI B [B2] [B3]	LD BC, nn	SBB M	SBC A, (HL)
CPO [B2] [B3]	CALL PO, nn	LDID [B2] [B3]	LD DE, nn	SBB r	SBC A, r
CZ [B2] [B3]	CALL Z, nn	LXI H [B2] [B3]	LD HL, nn	SBI [B2]	SBC A, n
DAA	DAA	LXI SP [B2] [B3]	LD SP, nn	SHLD [B2] [B3]	LD (nn), HL
DAD B	ADD HL, BC	MOV M, r	LD (HL), r	SPHL	LD SP, HL
DAD D	ADD HL, DE	MOV r, M	LD r, (HL)	STA [B2] [B3]	LD (nn), A
DAD H	ADD HL, HL	MOV r1, r2	LD r, r1	STAX B	LD (BC), A
DAD SP	ADD HL, SP	MVI M	LD (HL), n	STAX Q	LD (DE), A
DCR M	DEC (HL)	MVI r [B2]	LD r, n	STC	SCF
DCR r	DEC r	NOP	NOP	SUB M	SUB (HL)
DCX B	DEC BC	ORA M	OR (HL)	SUB r	SUB r
DCX D	DEC DE	ORA r	OR r	SUI [B2]	SUB n
DCX H	DEC HL	ORI [B2]	OR n	XCHG	EX DE, HL
DCX SP	DEC SP	OUT [B2]	OUT (n), A	XRA M	XOR (HL)
DI	DI	PCHL	JP (HL)	XRA r	XOR r
EI	EI	POP B	POP BC	XRI [B2]	XOR n
HALT	HLT	POP D	POP DE	XTHL	EX (SP), HL

Z 80微電腦軟體硬體 (下)

編著者：陳 金 追

發行人：楊 鏡 秋

出版者：儒林圖書有限公司

地址：台北市重慶南路一段111號

電話：3812302 3110883 3140111

郵撥帳號：106792號

吉豐印製廠有限公司承印

板橋市三民路二段正隆巷46弄7號

行政院新聞局局版台業字第1492號

中華民國七十年八月初版

定價新台幣 180 元正

文件名稱： Z80 微電腦軟體硬體第 1 5、1 6 章(掃描版)

文件分類	I
文件編號	00028
文件批號	07

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	2007-02-11	其他
	再版		加註

文件出處	原圖書書名	Z80 微電腦軟體硬體
	原圖書作者	陳金迫
	原圖書出版者	儒林圖書有限公司
	原圖書出版日期	民國 70 年 8 月

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。
-------------	---

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
K	教育教學類