

Z80 微電腦

軟體，硬體

(上册) 陳金追 編著

羅拔書局印行

Z 80微電腦軟體硬體(上)

編著者：陳金追

出版
發行 羅拔書局

澳門大馬路 381 號 樓 F 座

印刷者：振興印刷公司

澳門龍崙街 152 號地下

定價：H.K.\$ 25.00

目 錄

前 言

上 冊

第〇章 引 言

第一章 基 本 概 念

1-1 何謂程式設計.....	5
1-2 畫流程圖.....	6
1-3 資訊表示.....	7
1-4 計算機內部資訊表示.....	8
1-4-1 程式.....	9
1-4-2 數值資料.....	10
1-4-3 文數字資料.....	32
1-5 外部資訊表示.....	34
1-5-1 組合語言.....	34
1-5-2 八進制與十六進制.....	35
習題.....	39
習題解答.....	40

第二章 Z80硬體結構

2-1 Z80 微電腦系統結構.....	43
2-2 一般微處理器之結構與動作原理.....	46
2-2-1 暫存器.....	47

2-2-2	堆疊器	50
2-2-3	指令執行週期	52
2-2-4	拿取下一指令	54
2-2-5	指令之執行	54
2-2-6	臨界競賽問題	57
2-3	Z 80 微處理器之結構	59
2-3-1	一般用途暫存器	62
2-3-2	旗號暫存器	66
2-3-3	特殊用途暫存器	67
2-4	Z 80 之指令格式	73
2-5	Z 80 之指令執行	77
2-5-1	拿取週期	80
2-5-2	解碼與執行	82
2-5-3	重要練習	85

第三章 定 址 法

3-1	隱含定址	98
3-2	立即定址	99
3-3	擴展立即定址	100
3-4	暫存器定址	101
3-5	暫存器間接定址	102
3-6	擴展定址	104
3-7	修正零頁定址	105
3-8	相對定址	106
3-9	索引定址	110
3-10	位元定址	114

第四章 Z80 指令集

4-1	計算機指令之種類	119
4-2	Z80 指令集	125
4-3	資料傳送指令	126
4-3-1	八位元傳送	126
4-3-2	十六位元傳送	129
4-3-3	交換指令	132
4-3-4	區段(整批)傳輸指令	133
4-3-5	區段搜尋指令	136
4-4	資料處理指令	137
4-4-1	算術與邏輯指令	137
4-4-2	移位與旋轉指令	146
4-4-3	位元運作指令	151
4-5	測試與控制轉移指令	151
4-5-1	旗號	151
4-5-2	控制轉移指令	158
4-6	輸入/輸出指令	164
4-7	各種CPU控制指令	168
摘要		169
習題解答		170
Z80 指令集摘要		172

第五章 Z80 組譯程式

5-1	機器語言	421
5-2	組合語言	421
5-3	組譯程式	423
5-4	組譯程式之特色	424

5-4-1	符號位址	424
5-4-2	組合語言格式	427
5-4-3	數底表示	430
5-4-4	算式求值	430
5-4-5	虛指令	431
5-5	上機	437

第六章 資 料 傳 送

6-1	八位元傳送	440
6-2	十六位元傳送	449
6-3	區段(整批)傳送	455
6-4	資料互換	459
6-5	摘要	461
6-6	副程式	466

附 錄

下 冊

第七章 算術程式

第八章 移位與位元運算

第九章 表列與表格處理

第十章 程式設計技巧

第十一章 常用副程式

目 錄

第七章 算術程式

7-1	加 算	471
7-1-1	八位元加算	471
7-1-2	十六位元加算	474
7-2	減 算	477
7-3	BCD算術	479
7-3-1	八位元BCD加算	479
7-3-2	十六位元BCD加算	481
7-3-3	濃縮BCD減算	482
7-4	乘 算	484
7-4-1	8×8乘算	486
7-4-2	改良之8×8乘算程式	492
7-4-3	16×16乘算	496
7-5	除 算	498
7-6	比較運算	504

第八章 移位、旋轉、與位元運作

8-1	邏輯移位	511
8-2	旋轉型移位	516
8-3	算術移位	520
8-4	BCD數字移位	522

8-5	位元運作	526
第九章 表列與表格處理		
9-1	資料串	
9-2	表格作業	544
9-2-1	表格索引	544
9-2-2	剔除一已知元素	547
9-2-3	加入一新元素	550
9-2-4	二分搜尋	553
9-3	表列作業	557
9-3-1	泡浮排序	559
9-3-2	單端連鎖表列	565

第十章 程式設計技巧

10-1	表格跳越	572
10-2	副程式	578
10-2-1	何謂副程式	578
10-2-2	副程式之叫用與回返	578
10-2-3	副程式之使用及優缺點	583
10-2-4	參數傳遞	587
10-2-5	副程式巢串	588
10-2-6	副程式文書	590
10-3	Z80 副程式之特色	593
10-3-1	條件叫用與回返	593
10-3-2	重始指令	596
10-4	再進入	599

第十一章 常用副程式

11-1	比較副程式	634
11-2	計時副程式	636
11-3	乘除副程式	638
11-4	多段算術常式	642
11-5	ASCII 至基底X之轉換	644
11-6	基底X至ASCII之轉換	646
11-7	資料填補常式	625
11-8	資料串比較常式	625
11-9	找最大值副程式	626
11-10	泡浮排序副程式	629
11-11	表格搜尋常式	632

第十二章 Z80 CPU之界面信號與時序

12-1	位址與資料巴士	634
12-2	巴士控制信號	636
12-3	記憶體控制信號	636
12-4	輸入／輸出信號	637
12-5	其他CPU控制信號	638
12-6	與插斷有關之信號	639
12-7	Z80 CPU之電特性	640
12-8	M1週期	643
12-9	記憶體資料讀取與寫入週期	645
12-10	輸入與輸出週期	648
12-11	巴士請求／認可週期	650
12-12	插斷請求／認可週期	651
12-13	不可罩蓋插斷週期	653
12-14	脫離暫停	654

第十三章 輸入／輸出程式設計

13-1	輸入／輸出指令	656
13-1-1	累加器輸入／輸出指令	657
13-1-2	使用暫存器C之輸入／輸出指令	662
13-1-3	整批輸入／輸出指令	663
13-2	產生脈衝信號與延遲	668
13-3	資訊傳遞之方式	676
13-3-1	並行資訊傳輸	676
13-3-2	串行資訊傳輸	680
13-4	與輸入／輸出設備溝通	685
13-4-1	握手連絡	686
13-4-2	七段LED顯示	686
13-4-3	電傳打字機輸入／輸出	692
13-4-4	印出一串文數字	698

第十四章 輸入／輸出技巧

14-1	取 樣	701
14-2	插 斷	705
14-2-1	何謂插斷	705
14-2-2	插斷之用途	706
14-2-3	插斷處理	708
14-2-4	可罩蓋與不可罩蓋插斷	711
14-2-5	Z80 插斷之致／禁能	712
14-3	Z80 之插斷系列	714
14-4	巴士請求	716
14-5	不可罩蓋插斷	716
14-6	可罩蓋插斷	718

14-6-1	插斷型態 0	719
14-6-2	插斷型態 1	722
14-6-3	插斷型態 2 (向量式插斷)	723
14-6-4	多個設備共用同一插斷線	727
14-6-5	巢串插斷	732
14-6-6	與插斷有關之 Z80 指令	734
14-7	直接記憶器存取 (DMA)	736

第十五章 輸入 / 輸出界面電路

15-1	Z80-PIO	740
15-1-1	特 性	740
15-1-2	接腳說明	741
15-1-3	內部結構	745
15-1-4	作業型態	748
15-1-5	插斷與重置	751
15-1-6	程式規劃	754
15-1-7	插斷服務	757
15-1-8	應 用	761
15-2	Z80-CTC	763
15-2-1	接腳功能	764
15-2-2	內部結構	767
15-2-3	作業型態	774
15-2-4	程式規劃	776
15-2-5	時 序	781
15-2-6	插斷服務	784

第十六章 組成一微電腦系統

16-1	最小之 Z80 系統	789
------	------------	-----

16-2	ROM與RAM之界面	793
16-3	動態記憶體之界面	795
16-4	記憶體速度控制	797
16-5	Z80 PIO 之界面	798
附錄A	十六進制轉換表	802
附錄B	: ASCII 轉換表	803
附錄C	: 相對跳越位移表	804
附錄D	: 十進數與BCD轉換	805
附錄E	: Z80 指令碼	806
附錄F	: Z80 與 Z80 指令對等	813
附錄G	: 8080 與 Z80 指令對等	814

引 言

西元 1971 年，美國 Intel 公司推出世界上第一部字長為四位元之微處理器——Intel 4004。雖然 4004 並非真正的單晶微處理器（single-chip microprocessor），但其確已包含一計算機中央處理單元之大部份電路。Intel 4004 的來臨，使得一塊大型積體電路（LSI）便能取代傳統迷你電腦之數以百計電路。Intel 4004 之 46 個指令的指令集雖說不大，但其却已能滿足可規劃邏輯陣列（PLA）不易達成，需要決策能力但不需廣泛數學計算能力之控制應用。4004 每次能處理四位元之資料（因此稱為四位元微處理器），並且每秒鐘能做十萬次兩個四位元數目之相加。

Intel 推出之第二代微處理器，保留了第一代微處理器之 PMOS 製造技術，但資料巴士（字長）却增為八位元，且指令集亦增為 48 個指令。此一微處理器稱為 Intel 8008。因為不論指令之解碼、執行，或運算元，每次均能八位元一併處理，因此，8008 之指令週期時間比 4004 快速。除此之外，8008 亦具有下列優點：其能選取 16384 個八位元之記憶位置，包括七個八位元之暫存器，具有記憶堆疊能力，以及一單層之插斷能力。8008 每秒鐘能做 80000 次兩八位元數目的相加。但其指令集並不吻合（Compatible）於 4004。

於推出後，8008 與 4004 很快就受到電子工業界廣泛的採用，主要因為當時幾乎沒有其它類似之產品。就像於 4004 後曾推出其改良型 4040 一樣，繼 8008 後，Intel 於 1973 又推出在軟體上與 8008 相吻合之 Intel 8080。此一微處理器除了具有 8008 之所有指令外，又額外增加了 30 個指令。8008 的用者現在可以換上一速度

更快，用途更廣之微處理器，而不必捨棄其原來 8008 之程式。因為 8008 之軟體在 8080 上大概皆能執行。8080 由 NMOS 技術製成，具有比 8008 更高之時序頻率，每秒鐘能做五十萬次八位元數的相加。此外，由於 8080 做於僅有 40 支接腳之晶片，因此，於資訊傳輸與指令執行時，CPU 必須共用資料巴士之時間也隨之縮短。

在硬體方面，8080 亦強於 8008。8080 所能選取之記憶位址高達 65536 (64K) 個，而非 16384 (16K)。8080 之堆疊器亦非有限七層之 CPU 堆疊器，而是設於外部記憶器內，幾乎可“無限”延伸之記憶器堆疊器。此外，8080 亦具有 BCD 算術運算能力，能做 BCD 加法，以及較廣泛之定址法，能做記憶器之直接定址。對許多程式設計而言，8080 之 78 個指令似乎有點不可思議，但此一指令集確無疑地將 8080 之應用範圍，自單獨之控制領域，轉移至更一般化之領域。

1976 年，Intel 又推出了 8080 之多種變樣。新改良的 Intel 8085 不僅本身具有串行輸入／輸出能力，同時亦僅需一單相時序（8008 與 8080 均為雙相時序）與單一正 5 伏特電源（8008 與 8080 分別需要 2 個及 3 個電源）。在支援晶片之數量逐漸增加之際（如可程式化週邊界面、插斷控制器、與 CRT 控制器），Intel 為用者提供了愈來愈迅速之高度計算能力（8085 每秒能作七十七萬次兩八位元數目之相加）。而却保存了與 8008 及 8080 軟體之吻合性。

雖然在許多方面 8085 是 8080 之改良，但其指令集却極類似於 8080。8085 總共才僅增加兩個指令，一個用以讀取串行與插斷（interrupt）資料，另一個則用以寫入串行與插斷資料。8008 與 8080 先天上的許多缺點却仍存在。

1976 年，原來在 Intel 設計 8080 的一批工程師，離開了 Intel 加入 Zilog，設計了一新的八位元微處理器——Zilog Z80。此一微處理器之目標是在軟體與硬體上皆比 8080 強，但在軟體上又能與 8080 吻合。因此，其象徵了 8008/8080 之另一成熟層次。記得，今日的產品總比昨日為佳，Z80 之指令集與能力勝於 8080 的道理，

就如同 8080 優於 8008 一般。Zilog 公司同時也為 Z 80 設計一整套支援電路。此外，Z 80 在軟體上亦與 8080 相吻合，致使現有 8008 與 8080 之軟體，皆能在 Z 80 上執行。雖然 8008 與 8080 之指令與結構上的某些限制，Z 80 仍無法克服，但 Z 80 却提供了新的指令，新的定址法，以及比以前具有更高能力與更廣泛用途之硬體特色。

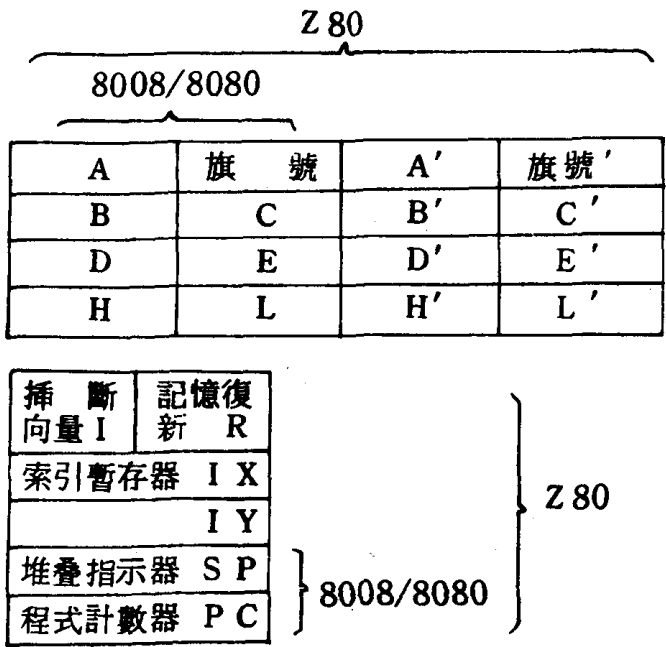


圖 0-1 8008，8080 與 Z 80 之暫存器比較

除了具有 8080 之八個八位元 CPU 暫存器外，Z 80 又再設置了八個同樣之暫存器，使其一般用途之暫存器達十六個。Z 80 還具有兩個 8080 所沒有的十六位元索引暫存器 IX 與 IY。另外，插斷向量暫存器與記憶復新暫存器，分別提供了特殊之插斷功能與動態記憶復新能力。圖 0-1 所示即為 8008，8080，與 Z 80 之基本暫存器結構圖。

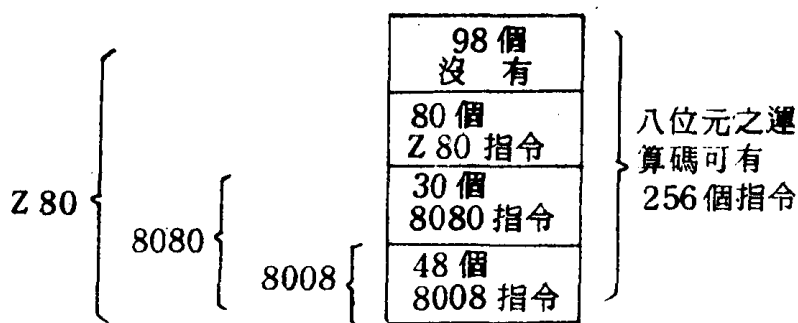


圖 0-2 8008，8080 與 Z 80 之指令集比較

Z 80 總共具有 158 個指令，涵蓋了 8080 之 78 個指令。所增...之指令有許多是 8080 指令之邏輯延伸，但亦有許多是十分能幹而為 8080 所沒有的。圖 0-2 所示即為 8008，8080，與 Z 80 三者之指令集的邏輯差異。

Z 80 保留了 8080 之所有輸入 / 輸出與插斷能力。但却擴充成能對任一 CPU 暫存器作業，以及“整批”(block)之作業方式。換言之，每一程式化(非 DMA)之輸入 / 輸出通道(I/O Channel)，每次能傳輸數個位元組。Z 80 之插斷則除了具有 8080 之標準插斷請求外，尚包括一類似於 Motorola MC 6800 與 MOS Technology MCS 6502 所具備之“不可罩蓋”(nonmaskable)插斷。其它之插斷能力則使插斷向量能指至記憶器之任一位置，而非僅第 0 頁之八個記憶位置。同時，插斷之層次可高達 128 層，而非僅僅 8 層。

基本概念

本章介紹一些與計算機程式設計有關之基本概念與定義。熟悉這些概念之讀者或許想僅大略地翻一下，就跳至第 2 章。可是，作者還是建議您從頭至尾再讀一遍，因為，這裡面含有許多非常重要之概念，如 2 補數、BCD、以及其它之數目表示法等。其中有些或許是您以前沒學過的；其它的或許會增進您之常識與技巧也說不定！

1-1 何謂程式設計

已知一個問題，最重要的就是要找出其解決方案。此一解決方案，表示成一步一步之程序，即稱為演算法(algorithm)。因此，演算法即為一已知問題之解決方案的逐步說明。但其必須在有限步驟之內結束。演算法可以任意語言或符號表示。下面即為演算法之一簡單例子：

1. 鑰匙插入鑰匙孔。
2. 鑰匙向左轉一圈。
3. 握住門之把柄。
4. 將門把柄左旋，且推門。

此時，對門鎖之種類而言，若演算法正確，則門就會開。此一四步程序，夠資格算是一開門之演算法。

在以計算機解題時，問題之解決方案一旦表示成演算法，即可以計算機加以執行。很遺憾的，當前的事實是計算機根本聽不懂，亦無法執行一般之口語英文（或甚至其它任何人類所使用之語言）。這原因主要在於所有一般人類語言之語法均有模擬兩可性(syntatic