

微電腦設計 與故障檢修

施 威 銘 譯

香 港 科 技 出 版 社

微電腦設計 與故障檢修

施 威 銘 譯

香 港 科 技 出 版 社

微電腦設計與故障檢修

編譯者：施 威 銘

出版者：香港科技出版社

發行者：香港科技出版社

九龍彩虹道810號六樓

印刷者：永達印刷公司

香港黃竹坑建明工業大廈九樓D座

定 價：H. K. \$

譯者序

微電腦並不只是娛樂性的電視遊樂器，也不只是啓益性的程式練習用機器，更不只是家務、個人的「計算與記憶幫手」而已；這些用途充其量都只是微電腦「應用」上的「成品」罷了。微電腦的真正本質，可以說就是一種「具有強大控制能力」的「工具」。

本書所要介紹給各位的觀念，便是如何去掌握住微電腦的本質與精髓，使各位能恣意發揮，全依照自己的構想去設計、去製造新的微電腦成品，或以微電腦爲核心的控制系統，可再也不必在別人已做死的系統上，就只用其所提供的有限能力去做事，而老是覺得不盡稱意，又苦於不知如何更改成自己所需者。也就是說，本書所討論的並不是如何在給定的系統與功能上做事情，而是在說明如何做出各種系統，以供給出我們所要的功能、完成我們所欲進行的作業。

筆者感於自動化的先決條件，在於強有力的控制系統，而強有力的控制系統，非微電腦無以爲功。因此，不揣淺陋，翻譯此書，祈望有興趣於設計自動化系統者，與有興趣擴大微電腦之應用層面與製、售者，皆能從本書獲得裨益，或是捕捉得些許啓發性的靈感

。也願各位先進聞達之士，能對書中的疏漏誤謬處不吝賜教指正。

原 序

微處理機革命起源於七十年代早期，而短短幾年之間，「微處理機」一詞就已是家喻戶曉的了。可是，儘管其應用產品現在已多得勝枚舉，微處理機的能力却仍還未完全發揮出來，造成這一事實的原因是多方面的。

首先，最急切希望應用到微處理機的人，絕大多數是工程師、科學家、甚至於是非技術人員，這些人通常是沒接受過正規電子電路方面的訓練。就算是有，如果他們所學的是六十年代中期以前的技術，那也很難能完全進入他們所全無接觸過的數位邏輯領域的。

其次，即使是已經有使用微處理機的知識和經驗，真正決定要拿微處理機來應用時，往往也會令人覺得是在下一場很大的賭注。因為微處理機“晶片”本身雖然極為便宜，大約只需美金一、兩塊錢就買得到，但產銷中的發展系統和測試儀器等，往往却都賣到美金一萬至三萬塊不等。

再者，就算是需求和財政上都允許使用微處理機來設計研究專題，最後所得的產品成果，又往往都很欠缺完整的文獻資料。因為這些文獻資料，向來就都是由那些在實驗室裏自己苦學自修出來的人所寫的，所以市面上雖然書籍到處充斥，其內容却常常令人大失所望。

我們可以把微處理機方面的文獻資料，約略歸納成三類——廠商提供的資料、平裝本文獻、和較正式的精裝本教科書。廠商所提供的文獻資料，通常是一些資料手冊和使用說明書，對於專業的電子設計師，這些書籍資料就已經夠完整、夠清楚了。然而，這些廠

商所提供的文獻資料，對於設計能力的提升並沒有多大幫助。對讀者有所助益的設計說明，應該是讀者所未曾想像過的，而且也會鼓勵讀者在設置時，儘可能採用不同“族系”的組件。廠商們當然是不會建議其客戶去使用別家出產的週邊裝置，也更不會告訴客戶去採用別族系的處理機。

平裝本資料佔了這些文獻的絕大部分，其目的是寫給一些業餘嗜好者，以及一些原非電子方面的技術人員看的，但其成效却是令人訝異地好！在銜接廠商文獻和純靠本能的使用者之間技術經驗和知識基礎上的鴻溝，平裝本文獻的成績是很可觀的。其所使用的“食譜式”說明，可以讓任何人只要作一點點判斷，加上少許的專業知能，就可達成其所要的硬體、軟體上的應用。

至於正式的精裝本教科書，其主題自然是在討論設計。可是，很少有書籍是從各個不同的層面來討論設計方法的；或者，有些書是花了四、五百頁的說明，却還捉不住單一層面系統設計的要點。而大多數的作者，則都是花太多時間在討論那些早已可在別的地方“抄”來的設計問題，很少有關於設計方面的書，能加入真正是嶄新、獨創的內容。

本書：「微電腦設計與故障檢修」，可說是唯一以各種層面，從概念上一直討論到實際微電腦系統設計的書。第一章中，我們定義出什麼是微電腦，並提出其內可執行的各個控制器功能。而結束這一章之前，我們也對微處理機的發展史略作介紹，並對現行的幾種微處理機稍作討論。

第二章則定義出微處理機上各必要的單元，而最重要的，是告訴各位讀者如何只花幾百美元，就自行組合出一套頗為妥切的發展系統，通常，買一套現成的發展系統，是要花上好幾千美元的。沒

有發展系統，是無法建造任何微電腦的，但是大部分討論「設計」的書，往往是假設讀者已購置了一套現成的發展系統，因而對發展系統不作任何介紹，這是不應該的。

在第三章中我們介紹了微電腦設計中最重要的一環，亦即讀 / 寫時序；其中我們以 6502 和 8085 這兩種處理機為例作比較。同時，我們也介紹了這兩種處理機與其附接的 ROM、RAM、和通用型裝置其間所需的界面技術。甚至於也討論到某一處理機接上另一族系處理機的元件時，其間的界面處理方法如何。另外，我們也提出在這種使用不同族系的裝置時，如何克服其間無互通性的時序問題。通常，在任何一本討論設計的書中，關於時序問題的討論總是最完整的。

第四章是對整個系統配置作考慮，例如：匯流線之結構、位址之解碼、以及回置和中斷回應、佈線等。另外也討論到低功率技巧、測試與維護要點、雜訊等問題。

第五章的主題是界面處理，也就是討論如何和外部裝置作信息傳遞，以及單純讀 / 寫時序所不能處理者。坊間有許多號稱為“設計”的書，很少有提到界面處理者，但是本書却提出一些足以啓發作簡易 I / O 的概念，也同時介紹了串列式和並行式的技巧（與其標準）。

關於故障的檢修，通常討論設計的書是不提的，而本書却在第六章中詳細討論。由於這一章所提的技巧都很有用，本書的書名也就因此加上了「故障檢修」幾字。請記住，不管所作的設計有多精巧，只要是不能工作，它就一點兒價值也沒有。所幸，大多數的硬體「瑕疵」，都可藉由本章所提的技巧，和一些不算昂貴的測試設備，一一找出。

第七章則敘述軟體在微電腦上所扮演的角色。如果沒有合適的軟體，即使硬體已經設計得非常完整，而且也已能正常工作，同樣是做不了什麼事的。這是因為有許多控制器的功能，可用硬體，也可用軟體來執行，所以在進行硬體設計時，我們就應該先考慮清楚軟體所扮演的角色；這一點也正是其他書所根本未提到的地方。所以，在第七章裏，我們先介紹一些好的文書整理技巧，以便使程式的可靠度更高，也更容易做維護工作。然後，從原理上來討論如何抉擇硬體與軟體所佔的份量，以及何時該介入中斷的使用。另外，我們也特別考慮如何撰寫由即時時鐘驅動的程式。

本書的撰寫，主要是針對高級實驗人員，或是有實際經驗的工程師等類讀者，所以假設各位讀者對微電腦的硬體已有初步認識，而且也都知道什麼是組合語言軟體了。或許有些讀者可以對本書一頁頁地讀下去，也都完全了解書中的內容，不過，總有些讀者可能在某些方面上，想從其他地方找出加強實力的資料，因此在每章之後，我們都列有一些參考資料。

本書的寫法絕非是做食譜式的說明，因此，為了解釋某一觀念或是構想，我們總是繪出其線路圖來說明，但只有在少數幾個例子中，才會把所用零件的編號和腳數都標了出來。各位讀者如果想要自行設計出一些獨創的東西，就應該先收集好各位所用組件的資料手冊或是資料說明書，然後應用本書所提的種種概念，相信各位讀者都可以一步步地自行設計出來；或者，在必要的時候，依自己的意願進行修改。

總之，「微電腦設計與故障檢修」必能使高級實驗人員或工程師們，很成功地設計出一套微電腦系統來。而且，不需要再花多少錢，就能再組合出一套很適用的發展系統。再者，所有的硬體毛

病也都可被診斷檢出，並加以修正，而不需另外使用奇巧、複雜、且價格昂貴的測試儀器。雖然本書並不作食譜式的指導，但書內成千個構想，儘都可併入讀者的設計概念上。當然，我們更希望這些概念，也能啓發、刺激出各位自己所創的構想。

EUGENE M. ZUMCHAK

目 錄

第一章 微處理機之使用	1
計算機系統——微處理機——微處理機與離散邏輯之比較	
——控制器功能——微處理機整體之優點——微控制器範	
例——微處理機之選用——選擇微處理機之準則——總記	
——參考資料	
 第二章 發展系統	 29
問題——現成之發展系統——自製之發展系統——微處理	
機系統——仿真器——仿真器之建造——EPROM 編寫器	
——一般編寫器之軟體——2716 之軟體——完整之系統	
——總記——參考資料	
 第三章 微電腦之設計：讀/寫時序	 99
導言——讀/寫時序——匯流線之結構——關於訊號之用	
語——微處理機時序——記憶體時序——快速處理機或慢	
速記憶體——族系之混用——非族系 I / O 裝置——結論	
——參考資料	

第四章 硬體設計：系統之考慮 175

導言——滙流線之緩衝——位址之解碼——回置與中斷之
回應——中斷之硬體——即時時鐘——低功率技巧——測
試與維護之考慮——其他——總記——參考資料

第五章 硬體設計：界面處理 253

導言——簡易輸入／輸出——可程式 I/O——可程式端口
與離散式 I/O 之比較——非簡易式 I/O：傳輸模式與式
樣——並行式技巧——訊息交換——並行式之標準——串
列式技巧——串列式之標準——多元處理——多元處理之
硬體——總記——參考資料

第六章 硬體之測試與故障檢修 329

導言——測試電路之接線——靜態處理機仿真器——SPE
之使用——靜態測試——動態測試——可有可無之測試儀
器——成品之測試與故障檢修——總記——參考資料

第七章 軟體設計 385

文書報告——軟體原理——軟體技巧——總記——參考資
料

第一章

微處理機之使用

近幾十年來，我們的日常生活中，無處不是使用著電子方式的控制與控制器，促成這些器具興盛的原因，可說是由於電子學上的層層突破。較早期的控制器，都是使用電子真空管，而電晶體（transistor.）推出後，便使控制器的體積大大縮小，可靠度也逐漸提高。隨後發展出的積體電路（integrated circuits），不管是類比（analog）方面，或是數位（digital）方面，其技術現在都已經可以把一整個計算機系統，做成只有一片“晶片（chip）”大小，而且價格也只有幾塊錢美金而已。就是這一種新的裝置——我們稱之為微處理機（microprocessor）——把控制器的技術，帶進了另一新的紀元。許多現在仍使用中的控制器應用，全都可改用微處理機來取代，而一些幾年前尚屬不可思議的應用發展，現在也都可由這一種不平凡的裝置——完成了。

本章裏就先說明到底微處理機是什麼，以及其特性與功能又是如何應用到種種控制器的功能上。最主要的，我們要探討到底怎樣的控制工作，最適於用微處理機來達成。

計算機系統

任何計算機系統〔註：computer，通俗名稱譯為「電腦」，本書則依據學術用語而譯為「計算機」。至於市面上到處買得到的掌上型作加、減、乘、除、開平方等 calculator，則譯為「計算器」。〕，不管型式的大小新舊，一定需包含如下三個基本單元：中央處理單元（central processing unit）、記憶體（memory）、輸入／輸出（input/output）。圖 1-1 於是可以用來代表「任何」計算機系統的方塊圖，也就是說：少掉了其中的任一單元，其結構

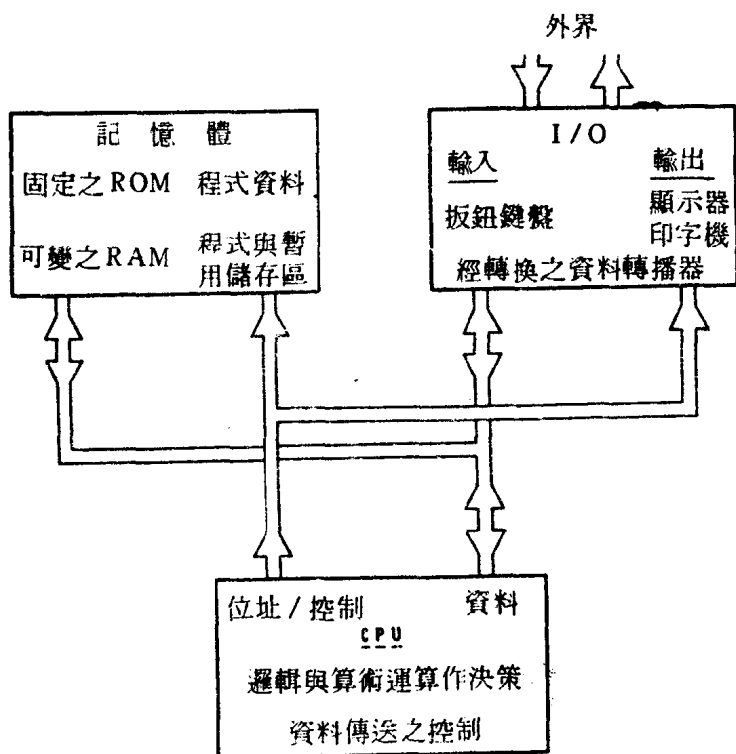


圖 1-1 計算機系統之基本單元

便不能稱為是計算機系統。

CPU（中央處理單元之簡稱）可說是計算機的「頭腦」，其任務是執行計算機內的基本運算，這些基本運算我們稱之為指令（instruction）。能達成某特定功能的一序列指令，我們稱之為“程式（program）”。執行指令時，CPU 就得控制所有的資料傳送，並發出各種必要的硬體時序訊號。

事實上計算機並沒有傳說中的那麼精靈聰敏，就算是近幾年其應用發展上的突飛猛進，它也沒有說就增長了多少「智力」。計算機所能夠做的事，其實也只不過是把二進位數字相加起來、判斷某一數字是否為零、把資料字元（word）搬移或移位等等。計算機之所以顯得功能特強，並不在於其是否夠聰明，而是由於其運算速度奇快；計算機做完一個指令，只需幾個百萬分之一秒。計算機雖然功能愈來愈多，體積愈來愈小，速度愈來愈快，但其心智仍然是很幼稚。

組成程式的一系列指令，是存放在記憶體上。記憶體也同時是儲存資料（data）的有次序性結構。記憶體所儲存的資料，有些是固定不變的，有些則是屬於暫時性的資訊（information），其值必需依情況而改。在傳統的主體計算機（mainframe computer）上，是使用同一記憶媒體——通常是磁蕊（magnetic core）——同時作為固定性（fixed）與可變動（variable）的記憶體。但是在現在，大部分的記憶體則都是由半導體（semiconductor）材料製成，而固定性和可變動的記憶體，則分別採用不同的結構，也分別具有不同的名稱。

記憶體在製造時，可以把預定好的資料型態固定在其上，以後都無法再作更改，這種半導體記憶體，我們就稱做是 ROM（僅讀記

憶體之簡稱)。僅讀記憶體 (read-only memory) 所儲存的内容，通常是一些已固定不再做更動的程式，且其内容很長。如果該程式很短，且所用計算機系統也只是還在研究中的雛形 (prototype) 階段，我們可以改用 PROM (可程式僅讀記憶體之簡稱) 來儲存。可程式僅讀記憶體 (programmable read-only memory) 是一種較特殊的 ROM，在出售時，其内容呈未程式化的無值狀態，可由使用者自行編製其程序内容，但其編製過程是不可逆的 (irreversible)，也就是說，被寫過的地方就不能再寫值進去。另一種較特殊的 PROM，稱為可抹除可程式僅讀記憶體 (erasable programmable read-only memory，簡稱為 EPROM)，約和微處理機同時期推出，可由使用者自行編製其内容，也可以無限次抹除其内容，重新編製。時至今日，其他各種可重新編製程式的 ROM，也正不斷的推出著。

記憶體所儲存的資料，如果能被更改的話，我們就稱這種記憶體為讀 / 寫記憶體 (read/write memory)。從發展過程來說，半導體製成的讀 / 寫記憶體，通常是稱為隨意存取記憶體 (random-access memory，簡稱為 RAM)。雖然現今大多數的讀 / 寫記憶體都是隨意存取的，而所有的 ROM 記憶體也是如此，似乎 RAM 一詞不應就只表示是讀 / 寫記憶體，但是就慣常的用法來說，RAM 已經是一致被用來代表為讀 / 寫記憶體的名詞了，絕不會也用來表示是 ROM 的。

計算機的第三個基本單元是 I/O (輸入 / 輸出之簡稱)，也就是處於計算機和“外界”之間的界面 (interface)。例如，掌上型計算器如果沒有了鍵盤 (作輸入) 和顯示器 (作輸出)，那將是一文不值的。就根本上說，可變資料是經由輸入裝置送進計算機

系統內，而 CPU 依照記憶體上的程式，對資料進行操作，然後把所得的結果，再輸出到外界。

微 處 理 機

微處理機 (microprocessor)，也就是“在晶片上的計算機”，有種種不同的式樣。其最基本的形式，是在一晶片裏頭，裝進了計算機系統 CPU 的最主要部分，所以需要再加接其他的元件，才能構成一完整的 CPU。例如，市面上頗流行的 8080A 微處理機，就必需再接上 8224 時鐘產生器 (clock generator) 晶片，和 8228 狀態門鎖 (status latch) 晶片，才能構成一可工作的 CPU。不過，就算是微處理機晶片已經是一完整的 CPU 時，比如說 Z-80 或 6502 就是，那也必須另外加接一些晶片，才能執行 CPU 與記憶體、輸入 / 輸出裝置之間的界面處理。

在極端的情形中，一片微處理機晶片甚至於可以把一完整的計算機系統全部容納進去，也就是說，除了 CPU 之外，該晶片內還包含有記憶體 (ROM 和 RAM)、輸入 / 輸出，以及一些控制及位址解碼等線路。這種專用性微處理機最典型的例子，就是掌上型的計算器、機械控制、電動玩具等，其所用的微處理機晶片，都是把固定不變的程式寫在 ROM 上，而且只能使用在它們所設計的特定應用中。

在大多數微處理機控制器的應用上，都是使用通用型微處理機，所以都需要另外接上記憶體、I/O、和界面裝置。本書所考慮的系統，也都是以通用型處理機為主，因此，圍繞在處理機周圍的硬體，都不是預先設定不變的，且都很有可能受到不良設計、錯誤