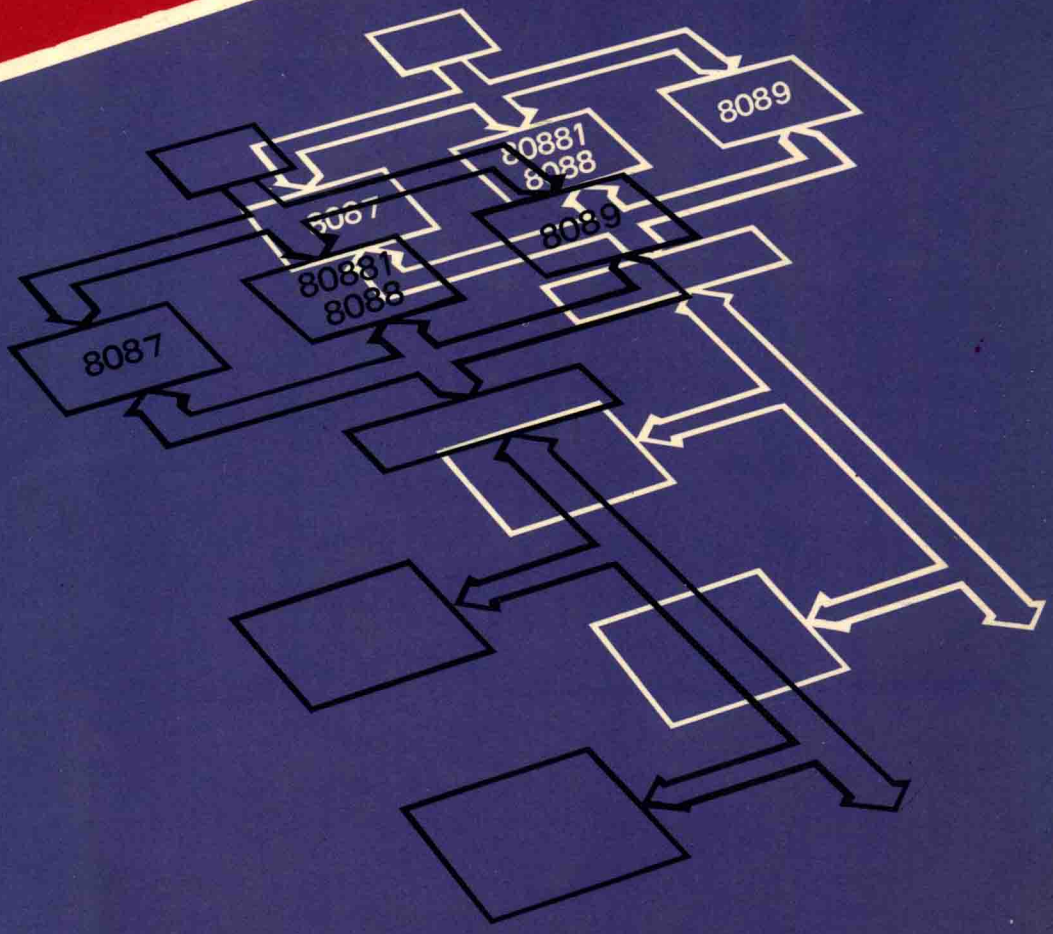
微電腦系統：

8086/8088 系列結構、程式寫作與設計

李中生　譯



微電腦系統：

8086/8088 系列結構、程式寫作與設計

李中生　譯

儒林圖書公司 印行

微電腦系統：8086／8088系列結構、程式寫作與設計

譯　　者：李　中　生
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
郵政劃撥：0106792-1號
吉豐印刷廠有限公司承印
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
中華民國七十四年九月初版
定價新台幣　250　元正

前　言

本書是設計來給時間爲一學期、程度較高的大學課程作爲教本，不過，對於已經進入工業界服務的工程師和技術人員，也頗具參考價值。在此我們假設讀者最少已熟悉一種像 FORTRAN 、BASIC ，或 Pascal 之類的高階語言，並且也已經有了基本的邏輯概念。第一章主要在複習一般的計算機概念，對計算機已經很熟悉的可以很快的翻過去。每章的最後都有許多習題，這些習題是根據該章材料的順序來編排，主要是讓讀者能自行驗證自己對教材瞭解的程度，而有些習題則是設計來擴充讀者對某些領域的認識。

本書的目的，在於深入探討一個微算機系統所包含的硬體和軟體部份。雖然，所考慮的觀念都是一般性的，不過，這些討論都是以一個特殊的微處理器 Intel 8086／8088 ，及其輔助裝置和軟體爲基礎。有關軟體方面的材料集中在 Intel 的ASM-86 組譯程式，但我們並沒有探討其中的每一個細節，以免分散讀者對眞正重點的注意力。

由於以八位元處理器爲代表的簡單控制器領域，已經無法滿足許多應用的需求，因此本書選擇 16 位元的處理器作爲討論的中心。在那麼多種都十分受歡迎的 16 位元處理器中，我們選擇 8086／8088 的主要原因，在於它提供了相當完整的指令集，在這個指令集中，包括相當廣泛的算術操作和字串處理指令，以及幾種設計來支持多元程式規劃和多元處理的先進結構特性。此外， Intel 也發展了幾種輔助裝置，像 8087 數值資料處理器和 8089 I／O 處理器，使得若干重要的多元處理功能，平均分配到整個系統上。因此，我們藉著逐步探討 8086 微處理器每一部份的功能 ，可以相當完整而一致的討論多元程式規劃和多元處理各方面的特性。此外，本書也詳細討論有關系統軟體、字串處理設備，和模組化程式，這些在八位元處理器的書上，都不曾仔細研討過。有關這些領域的考慮，隨著應用日益複雜而愈來愈重

要。

在第一章複習完計算機系統和微處理器的基本概念後，第二章從描述其內部結構和機器語言指令，來開始介紹 8086 。雖然 8086 和 8088 執行同一組指令集，但其內部結構仍有些微差異，這些差異我們在第二章最後一節都會提到。第三章則涵蓋了整個組合語言程式的基本概念。這章首先從介紹組合語言的觀念開始，然後看看基本的 8086 指令，包括算術、邏輯，和控制轉移指令。我們在討論中間加入許多例子，以加強讀者對各指令應用的瞭解。另外也討論到單模組程式所需要的檢定式指令。而這章的最後則以解釋組譯的過程來結束。第四章將組合語言的討論，擴充至多模組程式結構，並考慮較爲進步的程式設計技巧。其中有一節專門探討多模組程式的發展過程。

第五章是 8086 有關字串處理的設備。這一部份的材料也包括在寫一個有效率 I／O 常式所需要的程式碼轉換。第六章則介紹 I／O 程式技巧，包括程式 I／O 、中斷驅動式 I／O ，及直接記憶存取。在此我們經由例子解釋中斷序列和中斷向量的使用。此外也介紹如何來設置緩衝區，以使得讀者對即時作業系統是如何來處理 I／O 有一個較清楚的瞭解。第七章則介紹多元程式規劃。雖然像本書這種型態並不適合對多元程式規劃，以免作過分冗長的討論，不過本章仍然簡單介紹其中重要概念，以及它們是如何影響到微處理機系統的設計。這部份的討論包括資料共用、處理程序共用、可重進入程式碼，和記憶管理等部份。

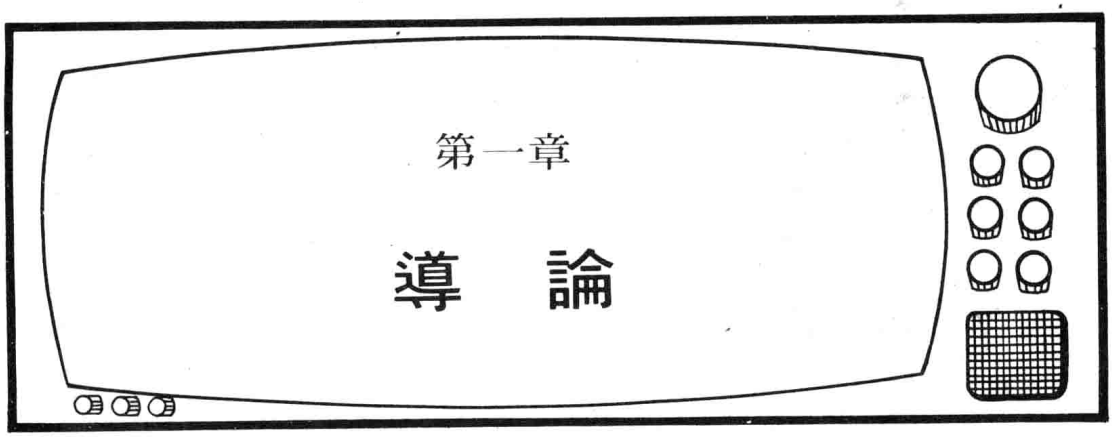
本書剩下的部份則集中在微處理機系統的硬體部份。第八章考慮系統滙流排和微處理器間的界面、中斷管理，及滙流排的活動和時序。第九章是有關將 I／O 裝置及巨量儲存裝置和系統滙流排連接的界面，其中它考慮了序列和平行式界面、直接記憶存取控制器、計時器、巨量儲存體控制器等。第十章討論記憶副系統，第十一章則討論和 Intel 裝置相容的各種多元處理組態。在此我們將介紹 8087 數值資料處理器和 8089 I／O 處理器。最後一章則介紹 Intel 較爲先進的 VLSI 裝置，並簡單討論 80130 、80186 ，和 80286。附錄則包含所有 8086／8088 指令集的整理。

作者們想要感謝德州大學電機工程系的充分配合，以及在寫這本書時，

來自許多人的協助。特別是提供大部份所需要資料及校閱本書 Intel 公司的 Rich Bruns 先生，以及爲本書原稿打字、校閱，和解出許多習題的Mary Lou Gibson 小姐。

Yu-cheng Liu
Glenn A. Gibson

目　錄



第一章

導　論

微電子裝置主要應用在控制和資料處理上面。將微電子裝置應用在簡單的控制器上時稱爲低階層應用（ **low-end application** ）；至於將其用在較複雜的控制器上：如機器人、航空電子裝備、複雜的軍事設備，以及一般用途的資料處理等，則稱作高階層應用（ **high-end application** ）。通常，我們並不將複雜的電路使用在低階層應用上，因爲這樣並不符合成本效益。實際上，就整個微電子的應用領域來說，雖然較低階層的應用仍然不斷有所改進，但是新發展的技術主要還是用在高階層的應用上。在這一階層中，不斷地有過去未盡想過的應用出現，因而不斷的需要更小和更複雜的電路。

我們通常根據微電子裝置的複雜度，將其區分爲：

小型積體電路（**SSI**）— 其邏輯閘的個數少於 10 個。
中型積體電路（**MSI**）— 其邏輯閘的個數介於 10 個和 100 個之間。
大型積體電路（**LSI**）— 其邏輯閘的個數介於 100 個和數千個之間。
超大型積體電路（**VLSI**）—其邏輯閘的個數超過數千個。

目前微電子界尚待研究的是在VLSI這個領域，而目前技術發展則主要集中在單晶十六位元微處理器及其附屬裝置上面。在這個領域中，目前居於領導地位的三個微處理器是 Zilog 公司的 Z 8000、Motorola 公司的M68000 以及 Intel 公司的 8086。雖然這本書主要是討論有關 Intel 8086 及其附屬裝置，但這些理論也同樣適用於其它的 16 位元微處理器。

本章主要是提供一些背景性質的定義和資料。在第一節中將對計算機系

統作一簡單的介紹，而接下來的三節中，則將討論資料是如何的儲存在計算機中、如何的被存取，以及計算機如何的來執行一個程式。在最後一節中將討論一些數位系統設計的準則，並藉著 Intel 微處理器族系的發展來探討這整個領域演化的過程。

1-1　微算機系統概論

微算機系統就和任何其它的計算機一樣，也包括硬體和軟體這兩個主要的部分。所謂硬體（ **hardware** ）當然是指電路、機殼等部分；而軟體（ **software** ）則是指導計算機如何執行其工作的一組程式。

1-1-1　硬　體

所謂計算機的結構（ **architecture** ）是指其主要元件相關位置的安排，以及這些元件是如何的連接在一起。圖 1-1 就是一套典型的微算機結構。在

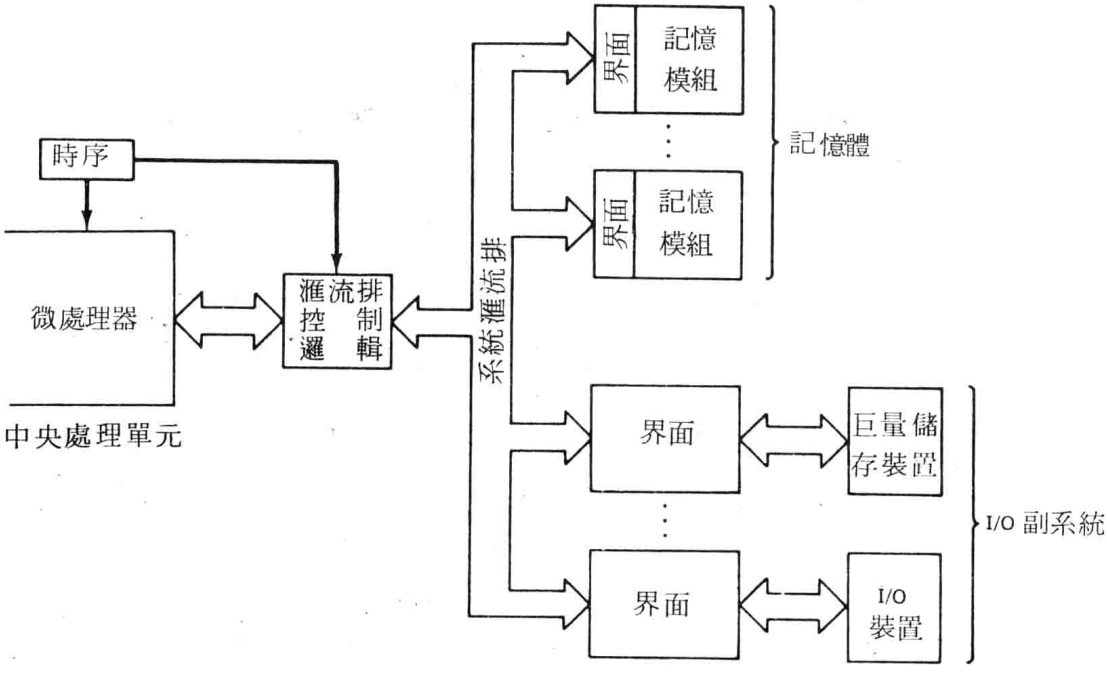


圖 1-1　一套典型微算機的結構

此圖中的元件包括中央處理單元（CPU）、時序電路、記憶體、輸出入副系統、滙流排控制邏輯，以及系統滙流排。在一個微算機中，CPU 是一個微處理器，通常稱作微處理器單元（**Microprocessor Unit，MPU**）。它主要的功能是將指令解碼，然後依照解碼所得的指示來控制系統內的活動，且它也執行算術和邏輯方面的運算。時序電路（**timing circuitry**），或又稱爲時鐘（**clock**），可以產生等間隔的脈波列，來使微處理器和滙流排控制邏輯中所有的活動同步。時序電路所產生的各個脈波列，通常具有相同的頻率，但脈波出現的時間相互錯開，亦即具有不同的相位。一般說來，微處理器需要一到四個相位的脈波列，早期的微處理器常需要一個以上的相位，近期則朝著單相位的方向發展。此外，早期的微處理器常需要單獨的一塊積體電路來產生所需的脈波；最近問市的微處理器除了振盪器之外，其時序電路都已包含在處理器本身的電路中。

記憶體是用來儲存目前正在使用的資料和指令，它通常分爲幾個模組，每個模組包含了數千個記憶位置，每個位置都可以存放資料或是指令，同時用一個記憶“位址”（**memory address**）來識別。而CPU就不斷從記憶體中取來指令，並執行這些指令所指示的工作。

輸出入副系統包括各種和外界通信和儲存大量資訊的裝置。像讀卡機、紙帶閱讀機，以及類比數位轉換器（A/D converter）都是輸入裝置的例子；而像列表機、繪圖機、打卡機、紙帶穿孔機，以及數位類比轉換器（D/A converter）都是輸出裝置的例子。至於像終端機則同時提供輸入和輸出的能力。計算機中用來作爲程式和資料永久儲存的元件稱爲巨量儲存裝置（**mass storage units**）。常見的巨量儲存裝置包括磁帶和磁碟，但由於最近技術上的進步，而使得磁泡記憶體（**magnetic bubble memories，MBM**）和電荷耦合裝置（**charge-coupled devices，CCD**）也開始問市。雖然巨量儲存裝置可以同時存放資料和程式，但是程式部分必須先傳送到記憶體才能被執行。

所謂“系統滙流排”（**system bus**）是將CPU和記憶體以及輸出入裝置連接起來的一組導線。所有資訊的傳輸都必須經過這些導線，至於是如何

傳遞的，則由滙流排的規格來決定。通常滙流排導線可以分爲三組：

1. 資料線—用來傳送資訊。
2. 位址線—用來指示資訊將由何處傳來或將傳送至何處。
3. 控制線—用來調整滙流排中的活動。

滙流排上的信號必須和連接到滙流排各元件的信號相配合，所謂“界面”（ **interface** ）就是連接滙流排和各裝置所需的電路；而“滙流排控制邏輯”（ **bus control logic** ）則是CPU 和滙流排間的界面。大部分廠商都提供各種 IC 裝置來幫助界面和滙流排控制邏輯的設計。滙流排控制邏輯依照系統的複雜度可能部分或甚至全部都包含在CPU的 IC 內。

記憶體界面主要包含用來將記憶位址解碼的邏輯電路，提供資料上或下滙流排的緩衝區，以及執行記憶讀寫的電路。輸出入界面可能相當簡單或是非常複雜。但無論其複雜度如何，這些界面必須能夠提供資料上或下系統滙流排時所需的緩衝區，並從 CPU 接受命令，以及傳送和界面所連接裝置的狀態資訊給CPU；此外， 和巨量儲存裝置連接的界面必須能夠直接和記憶體聯絡，因此需要界面有直接控制系統滙流排的能力。一般說來，輸出入界面和資料滙流排之間的連絡是須透過所謂“輸出入埠”（ **I/O ports** ）的暫存器（ register ）。

1-1-2 軟 體

計算機軟體通常可以區分爲兩大類別—系統軟體和使用者軟體。所謂系統軟體是指用來幫助別的程式產生、準備和執行的程式，而使用者軟體則是指那些應用計算機來解決問題的使用者所寫的程式。

一個計算機到底需要多少系統軟體，完全視其應用來決定。一個在其有生之年，功能非常單純的計算機（ 例如一個控制器，不斷重覆作同一件工作 ）可能只需要執行一個程式，另一方面，一個用來作一般目的資料處理的計算機則可能需要執行許多不同的系統程式。本書大部分都在討論一般目的的情況。在這種情況下，一般來說，整個系統的程式可以形成如圖 1-2的架構。

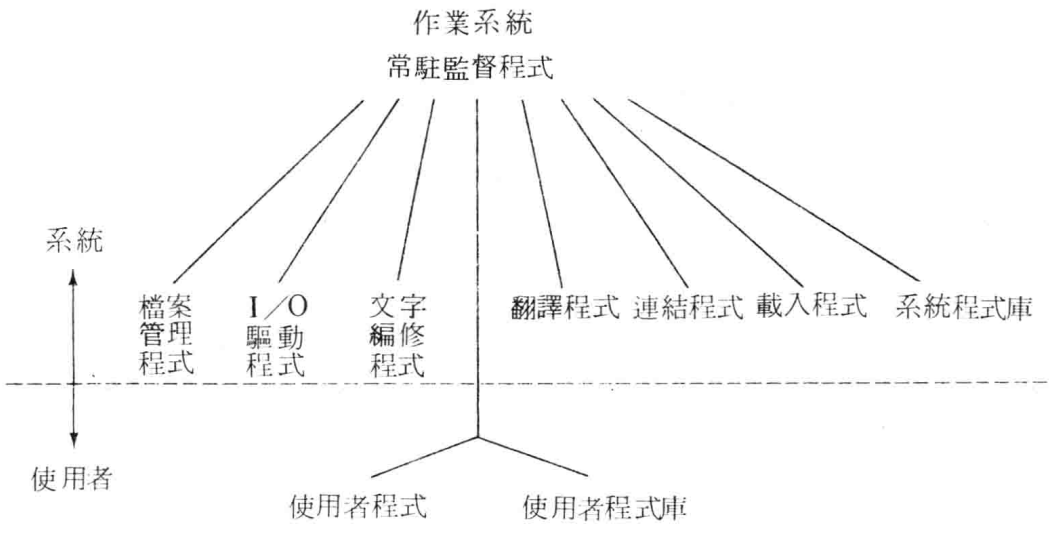


圖 1-2　軟體架構

作業系統（ **operating system** ）是用來提供使用者和機器間界面的系統程式，並且使得機器能夠更有效的被利用。所謂“作業系統”這個名辭並沒有精確的定義，它包括那些系統程式，乃視我們所看的書或技術手册而定。作業系統中最重要的部分是常駐監督程式（ **resident monitor** ），它是作業系統中只要計算機一打開就存在記憶體中的部分。常駐監督程式必須能夠接受使用者的命令，並且爲其啓動作業系統中負責處理這個命令的部分。

作業系統也包括“輸出入驅動程式”（ **I／O driver** ）和“檔案管理程式”（ **file management routine** ），這些程式主要的功能，在於當處理儲存在巨量儲存裝置上的大量資料時，可用來執行輸出入動作。當使用者程式或是其它系統程式需要使用輸出入裝置時，它通常不自己來執行這些動作，而要求作業系統使用輸出入驅動程式來幫它完成這件工作。這樣，一方面可以使得作業系統能將計算機控制在一個較佳的狀態下，另一方面使用者也可以減輕需要自己寫輸出入副程式的負擔。至於檔案管理程式則是和巨量儲存裝置的輸出入驅動程式相互配合，作存取、複製以及其它檔案處理的工作。

此外，系統軟體還包括高階語言的翻譯程式、組合程式、文字編修程式，以及其它幫助使用者的軟體。通常，程式語言可以分爲三個層次，它們是