

672121

B 11.20

科學圖書大庫

微算機原理與設計

著者 張時春

徐氏基金會出版

科學圖書大庫

微算機原理與設計

著者 張時春



徐氏基金會出版

徐氏基金會科學圖書編譯委員會

監修人 徐銘信 發行人 王洪鎧

科學圖書大庫

版權所有



不許翻印

中華民國六十八年四月三十日初版

微算機原理與設計

基本定價 2.80

著者 張時春 清華大學電機電力工程研究所副教授

本書如發現裝訂錯誤或缺頁情形時，敬請「刷掛」寄回調換。謝謝惠顧。

(67)局版臺業字第1810號

出版者 財團法人 臺北市徐氏基金會 臺北市郵政信箱53-2號 電話 7813686 號
7815250 號

發行者 財團法人 臺北市徐氏基金會 郵政劃撥賬戶第 1 5 7 9 5 號

承印者 大興圖書印製有限公司 三重市三和路四段一五一號 電話 9719739

序

微算機發明問世迄今六、七年而已，然其影響衝擊已有「無微不至、無遠弗屆」的趨勢。略其功能不談，僅就公司今年推出銷售的 TRS-80 微算機系統，包含微處理器、小銀幕終端顯示器、按鍵式輸入裝置、卡式錄音機、4 K 記憶單元、BASIC 程式及完整電源設備等，僅售美金 599 元折合台幣兩萬多元。試問以一部彩色電視機的價錢，即能擁有一套自用的電子計算機，執行以前惟有計算機中心才能完成的工作，其影響不能謂深遠？尤以今日大型積體電路（LSI）的價格日降，很有可能一年以後，祇需以目前一半的價錢，即可購妥更完整、更強而有力的微算機系統。以國內今日經濟發展的方針、及科學紮根的努力，這一門新興的學科，實極值得吾人開發探研、注意急起直追。期許若干年後，我們或將成為微算機成品的主要輸出國，為國人賺取更多的外匯與榮譽。

這本書純粹是針對初學入門者而編。今年初有機會在清華大學電機系試開「微算機導論」，以電機系二年級為主，以物理、數學、化工、動機等系三、四年級為輔。由於大部分選課同學對於計算機的知識僅止於 FORTRAN, COBOL 等語言，對於邏輯電路、計算機結構、組合語言等均無觀念，一般的微算機教科書派不上用場。因此改採講義，循序以進地介紹與微算機相關的以備瞭解微算機的內部結構，及進一步地應用於數位設計；介紹記憶元件，以加深認識 LSI 的應用彈性及影響。這兩部分均有相當齊全的設計實例，如乒乓球遊戲及電子琴等，導引同學對實際設計有深刻、具體的感受。此階段主要是灌輸基本概念。第二階段，包含硬體、軟體與應用設計，才真正涉及微算機系統。由於清華大學採用 INTEL 8085 微算機，因此實際的程式規劃與電路設計，均

以 INTEL 80 系列為主。硬體部分，當推利用 INTEL 8080 晶片及其輔助元件設計組合小型系統為高潮，軟體應用則以電子鐘、電子琴、乒乓球遊戲等的程式模擬為主。希望以淺顯的例子，幫助完成基礎的設計訓練。這門課大體上均能符合理想，完成指定的計畫，尤其是電二同學們更利用暑假完成進一步的設計工作，誠出乎意料。

鑒於國內中文參考書刊的欠缺，尤其沒有系統性的入門指導。期望更多的微算機著作能激發問世，倉促改編講義，輯印出書，期收「拋磚引玉」之效。奇望本書的刊行，能有助於提高大家對微算機的興趣。

會後要感謝清華同學的協助與促成，才能使此書於極短的時間內完稿問世。更要感謝電機系二年級祝華健、許德懿、郭琛、黃世明、林南昌等的編寫程式，設計實驗，及林華君的繪圖工作。他們實是本書的真正催生劑！作者見少識淺，昌昧為文，錯誤定屬難免，尚祈指正，使之完善可讀！先此致謝！

張 時 春

67年 11月 14日 麻省安城

目 錄

序

第一章	微處理器與微算機系統簡介.....	1
第二章	微算機基本數學概述.....	14
第三章	數位邏輯電路原理與設計.....	27
第四章	半導體記憶元件與系統設計.....	72
第五章	微算基本結構與工作原理.....	100
第六章	微算機實例介紹 (INTEL 80 系列)	129
第七章	微算機語言與程式編寫.....	138
第八章	邏輯設計與程式模擬.....	218

第一章 微處理器與微算機系統簡介

微處理器問世至今不過六年，然而它的成長速率卻是令人驚訝莫測！根據美國 INTEL 公司 ANDREW S. GROVE 副總裁的預估，1980 年代，全球三千萬新車，大約需要一億以上的微處理器及相關的大型積體電路晶片。可以預見的——所有的工業對於微處理器的需求量也將與日益增。要配合這項計算機工業的突破，確實有必要對微處理器及微算機系統深入探討、詳加研究。

什麼是微處理器？簡言之，它是微計算機的中樞，亦即「中央處理單元」。由於電子元件微縮的結果，通常祇需一個晶片就能勝任。它的大小大約 0.05 平方英寸，而電晶體零件可能上萬。INTEL 8085 的晶片大小約為 0.164 × 0.222 英寸，電晶體數目為 6200 左右，（圖 1-1）。

除了微處理器之外，我們尚需記憶元件，輸入及輸出裝置，才能組合完成完整的微算機，（圖 1-2）。利用輸送帶居間聯絡，資料、指令才能適切地執行、演算。今日微算機由於高度微縮化的結果，整個微處理器、記憶元件、輸入及輸出裝置幾乎全濃縮置於一個晶片上，如 INTEL 8048 或 8748 單晶片微算機，（圖 1-3）。

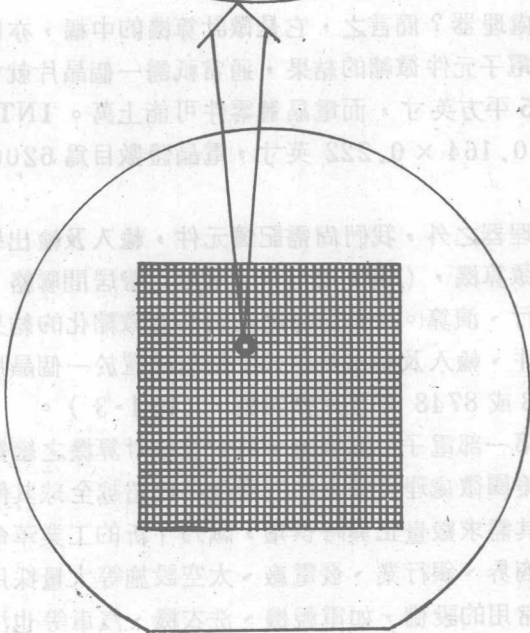
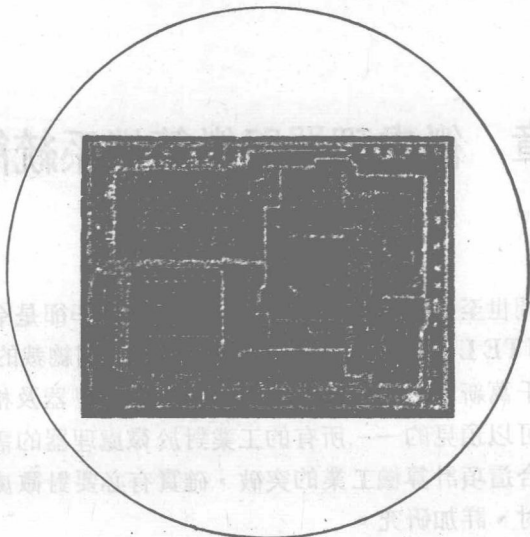
當年設計第一部電子計算機時，估計全球計算機之總需要量，四部足矣！而今，美國微處理器每月的生產量早已超越全球其他類型的計算機總數，而且其需求數量正與時俱增，誠乃「新的工業革命」的前驅！目前不啻是工商界、銀行業、發電廠、太空設施等大量採用微處理器，就連日常生活常用的設備，如電視機、洗衣機、汽車等也注入微處理器的新血脈。

微處理器以演算為主；接收外界資訊（以 0、1 二數元表達），執

852 32
42

介紹

第一章



25 B

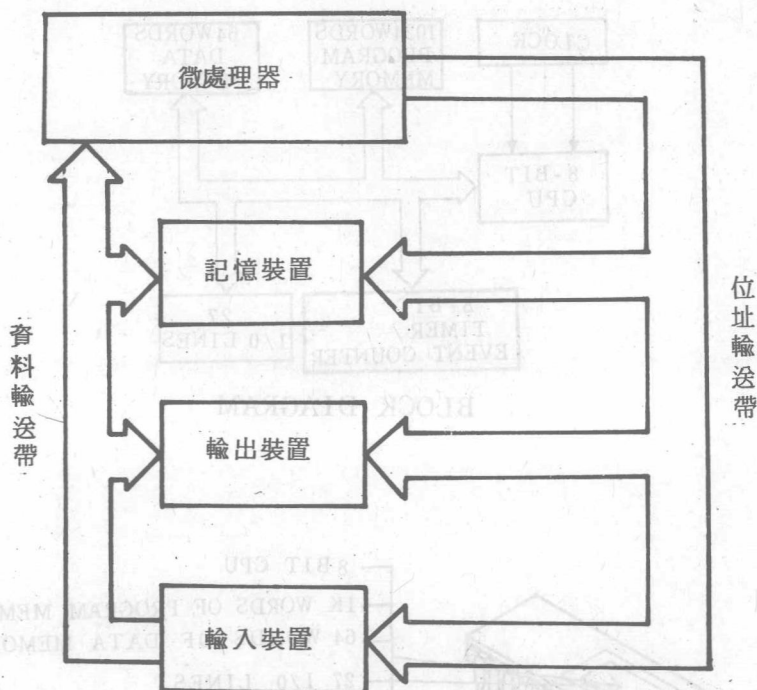
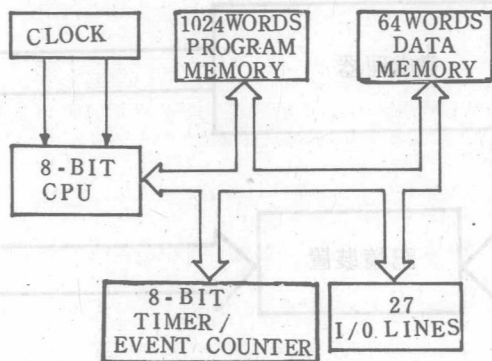


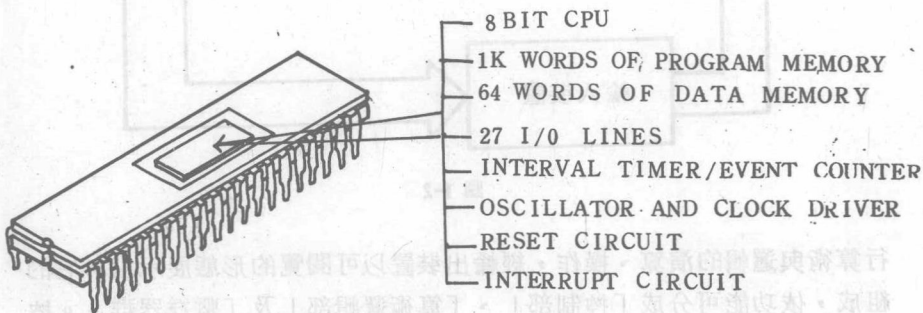
圖 1-2

行算術與邏輯的演算、操作，經輸出裝置以可閱覽的形態展示出。它的組成，依功能可分成「控制部」、「算術邏輯部」及「暫存器群」。控制部闡釋指令，循序輸送控制信號；算術邏輯部執行指令之作業，以演算為主題；暫存器群存放資料、程式及中間結果等，（圖 1-4）。

上述者乃最原始的微處理器模式。詳細的說明留待以後的章節敘述。一般除了前述的單晶片微處理器外，尚有由多個相同的「基片」（modular chip）組合成的「數元分割式」微處理器（bit-sliced microprocessor），如 INTEL 3000 系列分割式組合，（圖 1-5）。它採用雙極性技術（bipolar technique）製作的，與一般單晶片微處理器採用 MOS 技術者大有區別。它的優點簡言之乃速度快、適用性高。例如：3000 型，指令週期約 0.165 微秒，8085 型約



BLOCK DIAGRAM



CHIP FEATURES

圖 1-3

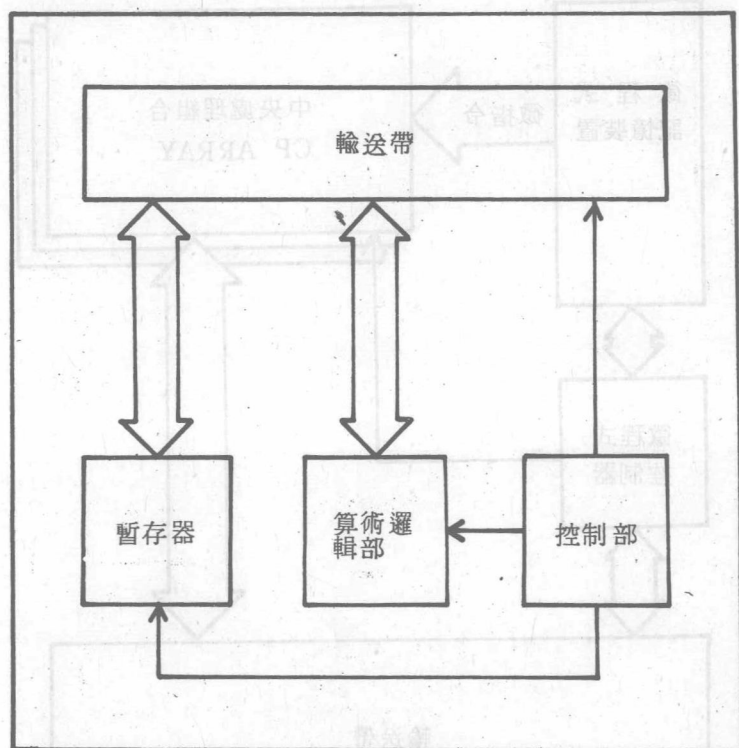


圖 1-4 微處理器

為 1.3 微秒，二者相差幾達十倍。

回顧 1960 年代，IBM 650 型計算機，每月的租金約一萬美金，全部指令數目 60 個，而指令週期為兩萬微秒；而今 INTEL 8085 完整套件，約為美金兩百元，指令總數 80 個，指令週期 1.3 微秒。二者相較，真是不可同日而語！

由於微處理器的發展是這樣迅速，我們實在無法確知它進步的方向，光看 INTEL 公司一系列微處理器的改進，多少能令我們對於微處理器的挑戰有觀念上的啓示，（圖 1.6）。

雖然我們無法肯定地畫分微處理器的類別及預測其未來的發展趨勢

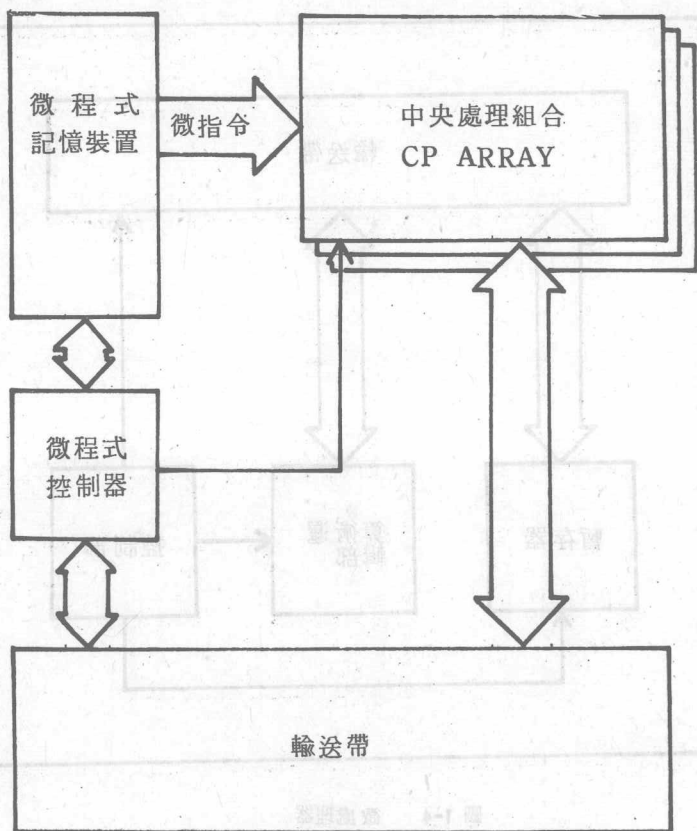


圖 1-5 分割式處理器組合

，不過我們仍可依據各階層的功能關係，從硬體與軟體的觀點討論整個微算機系統。「晶片」，（圖 1-7），是硬體體系的基礎，除了微處理器晶片之外，其他的記憶元件大型積體電路晶片等屬於這一階層。利用多個晶片組合而成「基模」（module），如 INTEL SDK-85，即包含 8085 微處理器，8155 RAM，8355 ROM 及簡單的輸入鍵、LED 顯示燈等，構成最簡單微算機，（圖 1-8）。「小型微算系統——這種系統包含面板、電源及相關的附屬裝置，如 INTEL 80/10

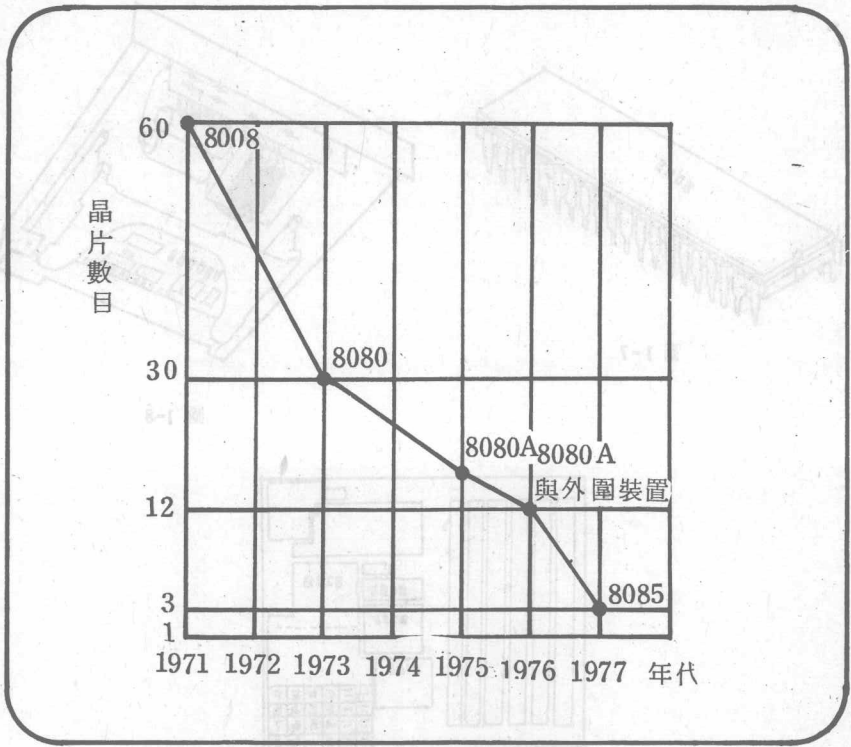


圖 1-6

系統，（圖 1-9）。最後的微計算機系統，如 INTELLEC MDS-800，包含主機、軟碟儲存裝置、CRT、讀紙帶機、印字機等構成完整的體系，相當於有規模的計算機中心，（圖 1-10）。

實用上，可能將多個微處理器聯合應用，而成為「多處理器」系統，取代大型計算機作業；或分層負責、控制各部門之工作。

介紹過硬體體系，接著要說明的是軟體方面的階層關係。軟體，主要是指「程式」——利用程式控制微計算機工作。每部微計算機均有特定的指令，執行基本運算。最原始的指令形式是以 0、1 數列構成的，稱為「機器碼語」（machine code）。例如，INTEL 8085 具備 80

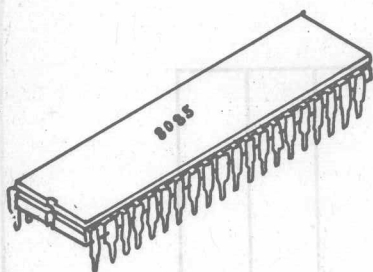


圖 1-7

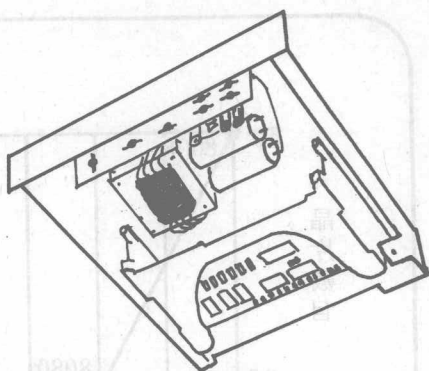


圖 1-8

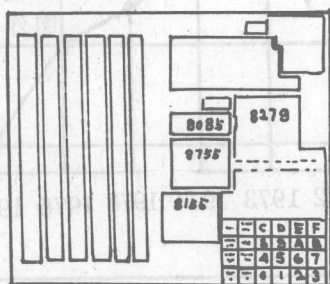


圖 1-9

個基本指令，每個指令的作業碼共有八個0或1數元（bit），如加法運算ADD M的機器碼語是（10000110）。採用機器碼語，既容易錯誤又花費時間，因此實際應用均不採用低階的機器碼語，而以其他的方式取代。「組合語言」（assembly language）就是利用易記的文字代替機器碼語所形成的微算機語言。每一個「組合指令」，均可經由「組程式」（assembler），將其編譯成對應的機器碼語，例如前述的加法運算，ADD M 即為組合指令，經編碼之後即為（10000110）的機器碼語，（圖 1-11）。要編寫組程式指令，使用者必須對微算機內部結構有深切的認識。為了更簡化程式編寫過程，

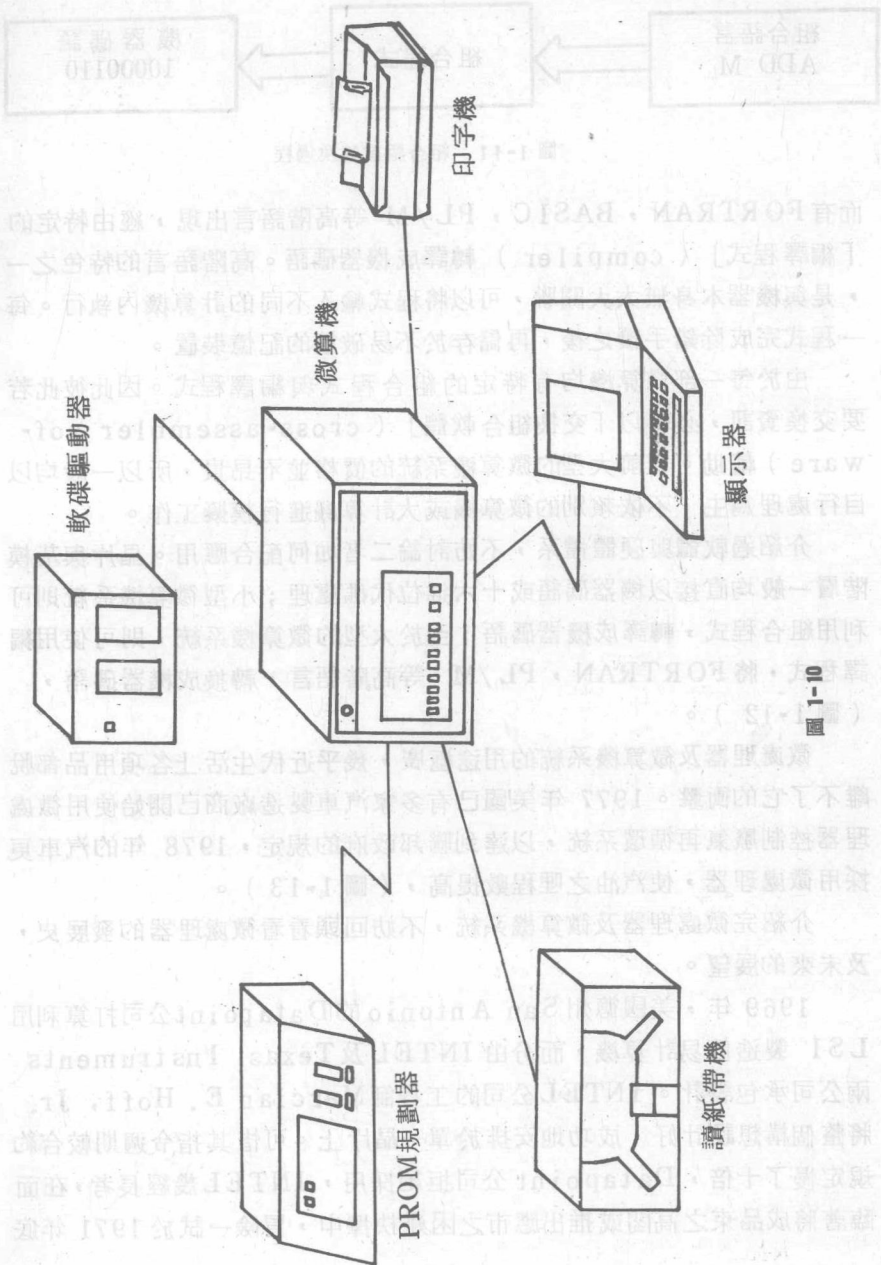


圖 1-10

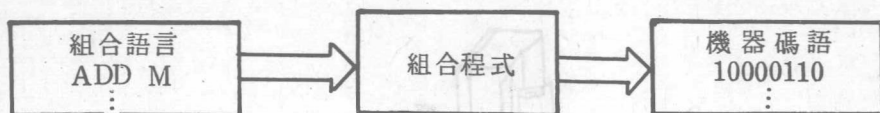


圖 1-11 組合語言轉換過程

而有 FORTRAN, BASIC, PL/M 等高階語言出現，經由特定的「編譯程式」(compiler) 轉譯成機器碼語。高階語言的特色之一，是與機器本身無太大關聯，可以將程式輸入不同的計算機內執行。每一程式完成除錯手續之後，再儲存於不易破壞的記憶裝置。

由於每一部微算機均有特定的組合程式與編譯程式。因此彼此若要交換資訊，必得以「交換組合軟體」(cross-assembler software) 輔助。目前大型的微算機系統的價格並不昂貴，所以一般均以自行處理為主，不依賴別的微算機或大計算機進行模擬工作。

介紹過軟體與硬體體系，不妨討論二者如何配合應用。晶片與基模階層一般均直接以機器碼語或十六進位代碼處理；小型微算機系統則可利用組合程式，轉譯成機器碼語；至於大型的微算機系統，則可使用編譯程式，將 FORTRAN, PL/M 等高階語言，轉換成機器碼語，(圖 1-12)。

微處理器及微算機系統的用途極廣，幾乎近代生活上各項用品都脫離不了它的衝擊。1977 年美國已有多家汽車製造廠商已開始使用微處理器控制廢氣再循環系統，以達到聯邦政府的規定，1978 年的汽車更採用微處理器，使汽油之哩程數提高，(圖 1-13)。

介紹完微處理器及微算機系統，不妨回頭看看微處理器的發展史，及未來的展望。

1969 年，美國德州 San Antonio 的 Datapoint 公司打算利用 LSI 製造簡易計算機，而分由 INTEL 及 Texas Instruments 兩公司承包設計。INTEL 公司的工程師 Marcian E. Hoff, Jr. 將整個構想設計好，成功地安排於單一晶片上。可惜其指令週期較合約規定慢了十倍，Datapoint 公司拒絕採用，INTEL 幾經長考，在面臨著將成品束之高閣或推出應市之困難抉擇中，冒險一試於 1971 年底

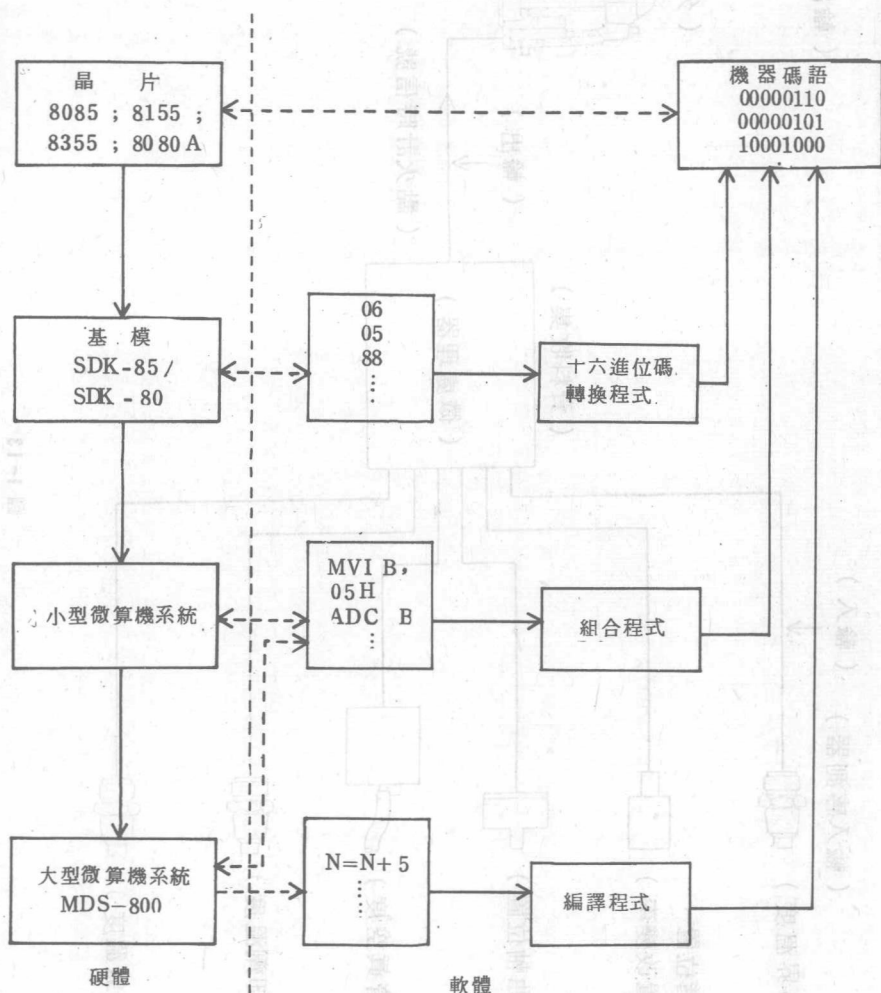


圖 1-12 軟體與硬體配合關係