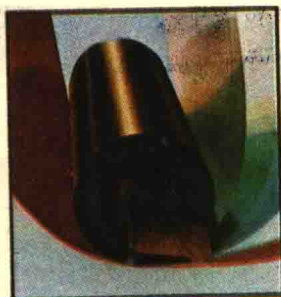
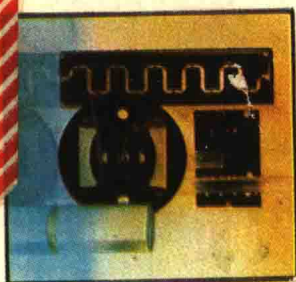
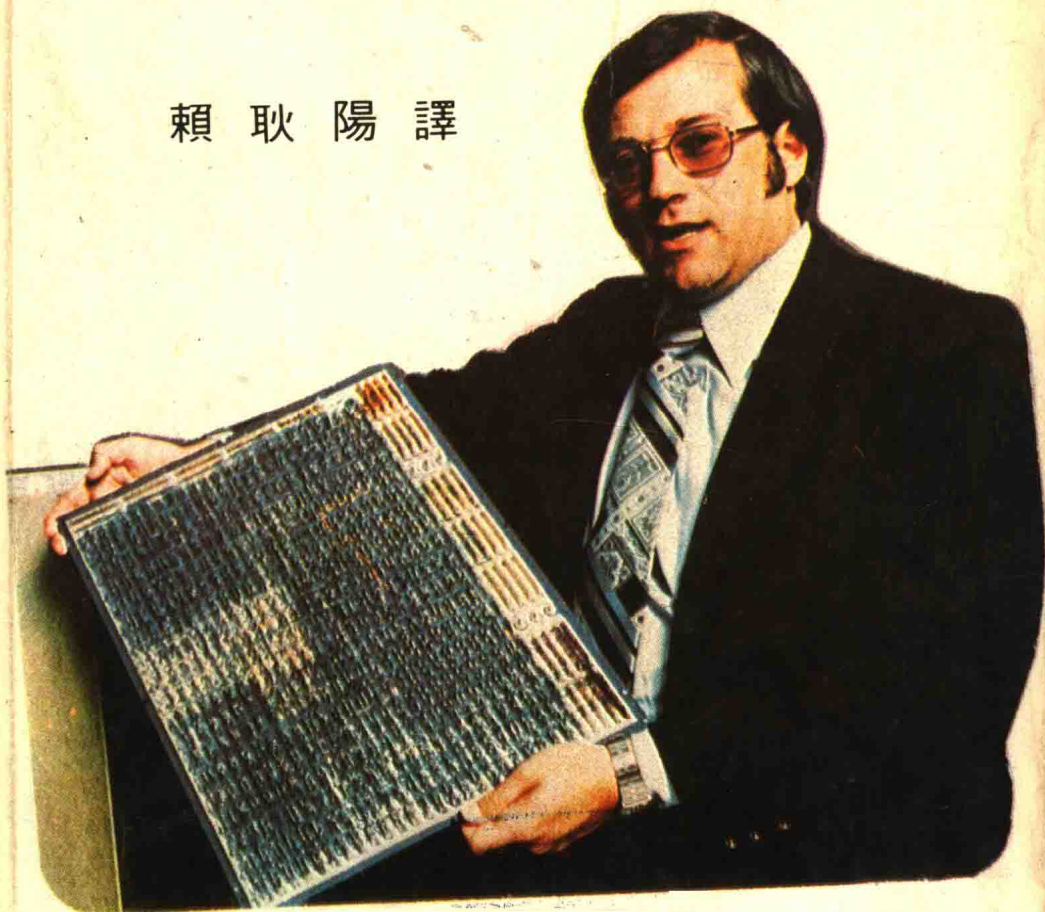


微電腦的開發技巧

賴 耿 陽 譯



國 際 I D M 電 腦 編 譯 社

微電腦的開發技巧

賴 耿 陽 譯

國際 I D M 電腦編譯社

微電腦的開發技巧

編譯者：賴 耿 陽

出版者：國際IDM電腦編譯社

發行者

香港北角堡壘街843號七樓D座

印刷者：華泰印刷公司

香港仔昌業大廈八樓C座

定價港幣：\$15.00

序

本書的目的有二，第1目的是對正使用或將使用微型計算機（微電腦）的硬體技術者敘述微型計算機系統—特別是軟體的開發方法；第2目的是在解說上述方法的過程中，也概述代表性微處理機8080型及周邊系統的知識。

亦即以系統開發方法為經、微型計算機本身為緯而構成全體，但本書的主要目的還是方法論，未必着重處理機的入門體系。

微型計算機在原理上為計算機，但應用上是成為LSI（大規模體積回路）組編入回路，因而，微型計算機的使用者大都為硬體技術者；但原理上既為計算機，軟體仍占很重要的地位，其製作費神費時；使用閘（gate）、起伏（flip-flop）、計數器（counter）等的IC進行邏輯設計的技術者未必熟悉監視程式、組合程式、編譯程式等概念，本書以實例平易解說程式開發用軟體、硬體的各種工具及它們的關係。

最近程式設計員也加強對微型計算機的關心，本書以程式設計者的慣用語敘述微型計算機的技巧。

本書的開發工具或技術的基本觀念也共通於8080以外的任何微處理機，其他公司的開發用工具詳附錄II，本書的內容將有助於8080同等製品的使用者。

本書以Intel公司的8080為主體，所以各種軟體的規範也盡量配合該公司的規定。

著者接受稿約後，赴美國加州聖地牙哥出席「1975年微處理機利用座談會」，順道考察微處理機製造廠、顧問公司。

此趟旅行體會美國在微處理機或記憶體的開發上領先一兩步，但應用方面仍是日本領先，原因似是日本在某方面較進步，其實，與其說是日本因有必要而採用，不如說是外國發表某種裝置後，日本人就會積極試用。

對於近年電子界罕見的寵兒—微處理機與其產業，吾人須力圖使之有健全的育成，但也須避免「殺雞用牛刀」。

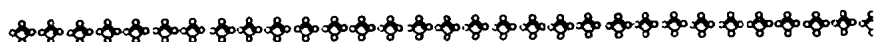
在技術方面，floppy disk 的廉價性與可靠性的提高很受重視，各廠都傾力開發使用 floppy disk 的操作系統。

本書執筆於著者忙於參加微處理機應用展的時期，難免有未盡滿意之處，但願多少能達成前述的編書宗旨，能對有志於開發微型計算機系統的人士稍有助益。

1979 年元月

石田 芳

目 錄



第 1 章 微型計算機的開發程序與技術

1.1 微型計算機與其特色.....	9
1.2 微型計算機系統的規範決定.....	12
1.3 微型計算機的採用基準.....	12
1.4 微型計算機的選定.....	15
1.5 微型計算機系統的設計.....	19
1.6 副常式與擠入.....	21
1.7 微型計算機的程式作成與去錯.....	24

第 2 章 微型計算機開發用基本系統

2.1 微處理機 8080 概要.....	29
2.2 8080 的定時.....	33
2.3 微型計算機的構成.....	38
2.4 計算機模組.....	41
2.5 I/O 模組、記憶模組、其他.....	49
2.5.1 I/O 模組	49
2.5.2 RAM 模組	50
2.5.3 PROM 模組	51
2.5.4 其他的模組	51
2.6 開發用微型計算機的操作.....	52
2.6.1 前控制板與控制板控制模組	52

4 目 次

2.6.2	程式的讀入	53
2.6.3	程式的實行與單步進行	56
2.6.4	核對點	57
2.6.5	程式計數器的定置	58
2.6.6	擠入要求	59
2.6.7	利用 I / O 命令的資料入口與表示	62
2.6.8	電傳打字機的操作	62

第3章 系統監視程式

3.1	系統監視程式與其起動	65
3.2	系統監視程式的指令用法	68
3.3	系統監視指令的種類	70
3.4	指令的錯誤	73

第4章 文章編輯程式

4.1	文章編輯程式與其起動	75
4.2	來源紙帶的作法與修正法	77
4.3	文章編輯指令的種類	80
4.4	來自文章編輯程式的通告	81

第5章 組合程式

5.1	8080 的命令	83
5.1.1	從軟體的立場看8080的構成	83
5.1.2	記憶部中的程式	83
5.1.3	位址方式	85
5.1.4	堆疊位址方式	86

5.1.5 flag	87
5.1.6 8080 命令的種類	88
5.2 組合程式的機能	90
5.3 組合語言的形式	92
5.4 組合程式的動作	92
5.4.1 標號	93
5.4.2 命令碼	94
5.4.3 運算元	94
5.4.4 註解	95
5.5 擬似命令與集體命令	95
5.5.1 擬似命令	96
5.5.2 集體命令	96
5.6 組合程式的用法	98
5.6.1 例題與其來源紙帶的作成	98
5.6.2 組合程式的讀入	100
5.6.3 來源程式的組合	101
5.7 程式的實行	102
5.8 組合程式的錯誤通告	104

第6章 FDOS與高速周邊裝置

6.1 何謂FDOS	107
6.1.1 高速開發系統的必要性	107
6.1.2 FDOS的操作	108
6.1.3 FDOS 與基本系統的時間比較	110
6.2 FDOS的構成	110
6.2.1 FDOS resident driver	111

6 目 次

6.2.2	FDOS executive	111
6.2.3	FDOS編輯程式	111
6.2.4	FDOS組合程式	111
6.2.5	FDOS軟體的裝置	112
6.3	FDOS的動作	112
6.3.1	FDOS的起動	112
6.3.2	檔案的構成	112
6.4	FDOS指令	113
6.4.1	FDOS指令的用法	113
6.4.2	FDOS指令的種類	117
6.4.3	FDOS錯誤通告	118
6.5	FDS	119
6.5.1	FD與驅動裝置	119
6.5.2	FDS	123
6.5.3	FDS用指令	128
6.5.4	FDS的介面	131
6.6	高速紙帶解讀器.....	132

第7章 PROM與其寫入

7.1	PROM	137
7.1.1	PROM的意義	137
7.1.2	PROM的種類	137
7.2	PROM寫入模組	140
7.2.1	PROM寫入模組的機能	141
7.2.2	PROM寫入模組的操作	141
7.3	PROM寫入器	142

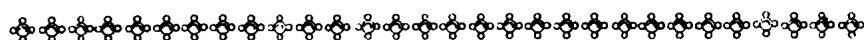
7.4 PROM消去用燈	143
--------------------	-----

第8章 PL/M與交叉軟體

8.1 resident 軟體與交叉軟體	145
8.2 編輯程式語言與PL/M編譯程式	146
8.3 PL/M的操作	148
8.3.1 PL/M的實行流程	148
8.3.2 編譯程式控制與檔案系統	149
8.3.3 PL/M的實行例	149
8.4 實行PL/M時的動作	151
8.4.1 記憶部的分配	151
8.4.2 副常式與堆疊操作	152
8.4.3 擠入處理	152
8.5 PL/M語言	153
8.5.1 PL/M來源程式例	153
8.5.2 註解與文章	155
8.5.3 宣言與procedure	
8.5.4 DO group	157
8.5.5 預先宣言的 procedure	159
8.5.6 reserve 語	160
8.6 PL/M編譯程式的效用與效率	160
8.6.1 PL/M編譯程式的效用	160
8.6.2 PL/M編譯程式的效率	162
8.7 交叉組合程式與交叉模擬程式	164
8.7.1 交叉組合程式	164
8.7.2 交叉模擬程式	164

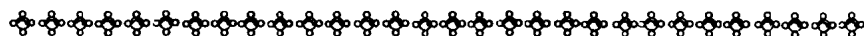
8 目 次

8.8 交叉軟體與分時系統 (TTS).....	167
附錄1 8080 命令一覽表	172
附錄2 主要微處理機與開發工具.....	175



第 1 章

微型計算機的開發程序與技術



1.1 微型計算機與其特色

人們使用微型計算機（微電腦）與微處理機一語已過數年之久，各半導體製造廠發售各種微處理機晶方（chip）或記憶、周邊回路等大規模積體回路（LSI），也有稱為 system house 的公司發售將它們組合而成的各種微型計算機，讀者大半已熟悉微型計算機或微處理機的名詞，但微型計算機的性格稍異於大、中型計算機或迷你計算機，當初的定義曾混淆不清，因而在進入本論前，有必要先弄清微型計算機（微電腦）的概念，以免誤解。

微型計算機（微電腦）的起源是美國 Intel 公司為日本電泉公司設計的 LSI，Intel 公司後來將之命名為 4bit 的「microcomputer」而發售，此系統稱為 MCS-4，以稱為 4004 的 CPU 為中心添加記憶、輸入出晶方、漂移暫存器（shift register）等。

但一般的計算機是指可演算或處理各種數據的機械，演算或處理的程序預先存放於記憶裝置，依序取出而實行，並可依演算或處理的結果改變次後的程序。

指定記憶裝置中的各個演算或處理的情報稱為命令（instruction），指定一系列命令形成的全體工作流程的情報稱為程式（program）。

10 1.1 微型計算機與其特色

計算機是將程式內藏於記憶裝置，有時稱為程式內藏（stored program）方式等。

依序取出收存於記憶裝置的程式，對演算裝置或數據的輸入出裝置指令次一動作的部份稱為控制裝置，這成為計算機全體機能的中心，圖 1.1 為一般計算機的構成。

前述MCS-4的構成又如何？圖 1.2 為MCS-4 的方塊圖，CPU 晶方 4004 將上述的控制裝置與演算裝置合而為一，有時稱為中央處理裝置，CPU 也是同義詞的簡稱，因而記憶裝置分為 2 部份，亦即程式收存於控制記憶裝置，演算等必要的資料貯存於資料記憶裝置；在MCS-4 中，輸入出用部份不獨立，與記憶部一起集積於半導體的晶方上。

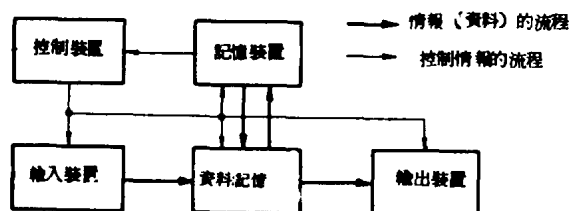


圖 1.1 計算機的構成

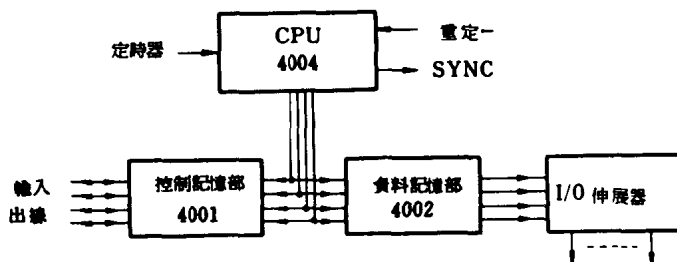


圖 1.2 MCS-4 的方塊圖

如此看來，可知MCS-4構成一個計算機系統，稍異於一般計算機之處是記憶部分為兩部份。

一般的計算機常以一部機械做各種工作，所以程式用控制記憶部也常與資料記憶部同樣裝入自由讀寫的記憶部（RAM），但微型計算機是當作系統要素而埋入使用，用於電桌、電子暫存器，特定控制裝置等進行某一特定的工作，通常程式不大有必要變更，因而這收存於讀出專用記憶部（ROM）。

MCS-4時，CPU固不待言，記憶部也用特定的裝置（4001或4002），最近的微型計算機常將市售的RAM或ROM組合於CPU；本書以8bit微型計算機用的CPU 8080為例，但也可組合各種RAM或ROM、PROM（可寫入的ROM）。

可見微型計算機未必收存於一盒中成為計算形，常為半導體LSI構成的零件集合形，但其構造或原理，軟體方面在本質上同一般的計算機，本書的主題是微型計算機用為一種計算機時的開發技術，不過讀者仍應認清微型計算機的本體及應用上的觀念。

微型計算機系統中，以CPU部份為中心集積的LSI稱為微處理機，加上記憶部、周邊回路，形成計算機者稱為微型計算機；此CPU部份通常裝入1晶方，故也稱為1晶方CPU等，但有時因語長、機能的大小、擴張性、半導體元件的利用率等而成2晶方以上。

微型計算機與微處理機的特色如下：

- (1) 由1個～數個半導體LSI構成的計算機CPU部份稱為微處理機。
- (2) 以微處理機為中心而加必要之記憶部、輸入出、其他周邊回路稱為微型計算機。
- (3) 微型計算機的程式原則上裝入ROM。
- (4) 微型計算機常裝為系統或機器的一要素而使用。

1.2 微型計算機系統的規範決定

設計用微型計算機的系統（機器）時，最先須決定系統的規範，此處所謂的規範是偏重清楚定義機能方面的規範，稱為機能規範（Functional specification）。

一般開始工作之際，未必充分瞭解規範，亦即技術者的頭中只想使用何一微處理機，以何種IC解決等純粹的技術事項，常不有系統瞭解「顧客要求什麼機能一亦即要達成什麼？」，當然，也許會在途中變更規範，不過，開始工作前，若清楚定義機能規範，則即使有變更，也不會有任何混亂。

1.3 微型計算機的採用基準

決定機能規範之後，須開發滿足該規範的確實而經濟的系統。

第一個問題是對象系統要用微型計算機作成或用其他手段一亦即有線的邏輯回路或顧客用品方，若用微型計算機，則須檢討選擇何種處理機？從半導體公司購入LSI而裝配？或用模組式控制台型微型計算機？要用何種開發道具？

1.4 項敘述微型計算機的種類或購入形態的選擇，開發的工具為本書的主題，各種工具的長處、缺點、特色、用法等將詳述於第2章以下，在此只說明判定否使用微型計算機的查核重點。

採用微型計算機的優點如下：

- (1) 零件數減少，所以製造成本降低。
- (2) 製品上市的速度快，相對增大銷售量或市場占有率。
- (3) 可以低價格使製品達成高機能。
- (4) 減少開發成本和時間。
- (5) 提高製品可靠性，減少保養或保證費用。

(1)所謂減少零件數，減少製造成本乃屬事實，1個閘（gate）相當於8～16 bit 記憶部，若1個IC約存入10個閘，則ROM記憶部可置換的IC封裝（package）數，閘數如表1.1所示。

表1.1 ROM置換的IC數（Intel公司資料）

ROM記憶大小 (bit)	置換的 gate 數	置換的 IC 數
2048	128—256	13—25
4096	256—512	25—50
8192	512—1024	50—100
16384	1024—2048	100—200

此表最左方的數字對應於ROM 1個份的bit數，故由此表可知減少很多IC數；減少IC的封裝數後，IC的進貨檢查、基板、裝配、基板試驗和調整、連接器、電晶體等個別零件、配線、電源、框體等的費用也成比例減少，設IC的價格為1，則包含IC的這些成本為3.2。

至於(2)，在某公司內組合硬體的模組（module）或軟體的副常式（subroutine）等，固可將製品儘快上市，但實際常採用微型計算機應用製品，研製新產品時就重新設計，如此會增高時間、成本，若用既製的模組和軟體的標準常式，常較有利。

(3)對硬體是事實，例如追加ROM時，如表1.1所示，宛如附加很多邏輯，軟體開發費用年年增加，不過，一旦作成軟體，可無限制複製，所以製作部數愈多愈有利，實際上，軟體與硬體的分攤是一大問題，所以設計系統時，須慎重檢討。

(4)約與(2)同內容，學習計算機程式計畫的技術者幾無例外，爲了

實現同一機能，程式計畫比邏輯設計容易、方便。

(5)與(1)有關，可提高可靠性，減少保養費用。

那麼，使用微型計算機，是否全無問題？其實不然，問題如下：

(1) CPU及微型計算機用IC有一定的機能，記憶用IC也有最低限度的bit數，在小規模系統反增高成本。

(2) 動作全為程序式（時間秩序），很難像有線邏輯那樣同時動作。

(3) ROM的mask製作頗需費用和時間，用PROM時，零件成本上昇，用某種PROM時，可靠性稍有疑問。

(4) 設計微型計算機時，要求硬體，軟體雙方面的素養，但滿足此種要求的技術者並不多見。

ROM問題也是今後半導體製造廠須解決的大問題。

一般的宣傳是微型計算機可減少開發時間與成本，但ROM的mask製作時間約8～12週，事後若發見錯誤，不只增多費用，也延誤時間。

爲了避免如此，製作mask前，須以各種工具充分去錯（debug），但程式難免有錯，設計完了後，也常變更若干規範，所以mask製作時間長達2～3個月（PROM詳後）。

表1 2比較有線邏輯與微型計算機的系統途徑，微型計算機確是開拓一大新紀元，但應用時，須瞭解本節所述的事項，確立採用基準。