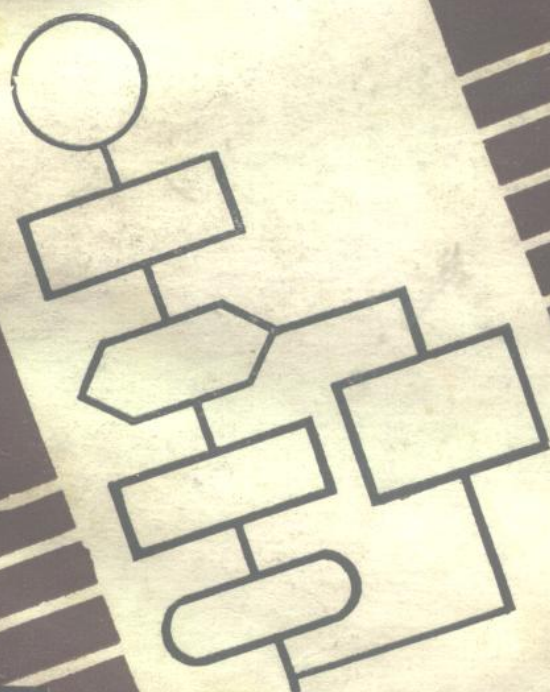


微處理機·微算機硬體 ·軟體應用

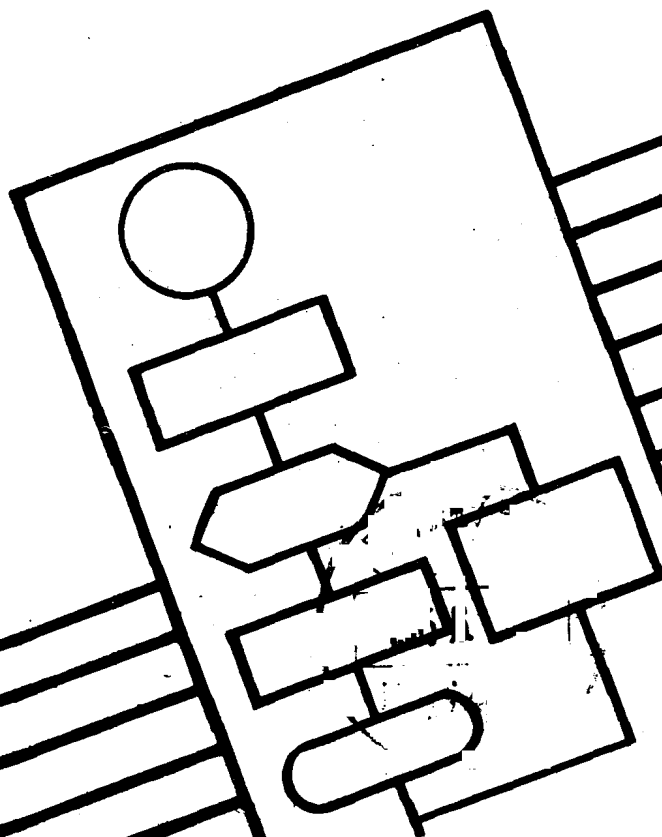
徐茂林 編著



全華科技圖書公司

微處理機・微算機硬體・軟體應用

徐茂林 編著



全華科技圖書公司印行



全華圖書 版權所有 翻印必究
局版台業字第0223號 法律顧問：陳培豪律師

微處理機・微算機
硬體・軟體應用

徐茂林 編著

出版者 全華科技圖書股份有限公司
北市龍江路76巷20-2號
電話：581-1300・564-1819
581-1362・581-1347
郵政帳號：100836

發行者 蕭 而 鄭
印刷者 欣瑜彩色印刷廠
定價 新臺幣 150 元
再版 一九八二年10月

11/11/07

感謝您

感謝您選購全華圖書！

希望本書能滿足您求知的慾望！

圖書之可貴 在其量也在其質

量指圖書內容充實、質指資料新穎够水
準，我們就是本著這個原則，竭心
盡力地為國家科學中文化努力
，貢獻給您這一本全是精
華的全華圖書。

8210306

Dedicated to the memory of Zou.
my best friend.

前 言

微處理機 (microprocessor) 到底是什麼呢？它的定義是「做在一片 IC 上的計算機」 (Computer on a chip) ；所以，它只是一片 IC，一片有 16 只接腳到 48 只接腳的 IC ；接腳的多寡，端視廠牌而定。

一片微處理機實際上是利用近代 LSI 的技術將一部計算機的中央處理系統 (CPU) 做在一片 IC 晶片上。由於能大量的生產，價錢非常便宜。

從 1971 年，Intel 公司推出第一部微處理機後，到處便是一片噓聲，有許多讀者希望能對微處理機有詳盡的了解，並能加以運用。本書，就是爲了要幫助這些讀者達到此目的而寫的；在第一章，介紹基本的數字系統，除了討論到一般的半加法器，全加法器外，還詳細的討論了“進位預知”加法器 (carry-look ahead) ；在第二章，介紹計算機的基本知識，其中內容都是要替讀者打好基礎的；尤其對計算機的一步一步動作及許多很有用的 IC 如開集極 IC (open collector)，三態 IC (tri-

state) 都在本章中有詳細說明；在第三章，先讓讀者對微算機的內部結構有詳盡的了解以便而後幾章運用；在第四章，介紹微算機的程式，指令，及如何設計程式；在第五章，介紹微處理機如何和外界週邊裝置配合，並涉及介面電路的設計；尤其在直接記憶體資料傳送 (DMA, 又常稱為 cycle stealing) 方面及計算機和磁帶控制器配合使用方面，本章都有詳盡的討論；在第六章，舉了兩個實用的例子，俾使讀者能對微算機的應用得心應手。最後在第七章中，則仔細的介紹 8080 微算機系統，使讀者能更進一步熟悉機器的運用。

當我寫這本書的時候，正值炎夏，幸賴好友呂的幫忙，及吾弟徐良華為我一個字一個字的校稿，及全華圖書公司陳繩籌先生的大力支持，才能使這本書順利出版，在此謹向他們一併申謝。

著者徐茂林謹識於台大電研

1977 6.30

目 錄

第一章 數目系統	1
1.1 十進制系統及二進制系統	3
1.2 二進制的計算	6
1.3 進位預知加法器 (Carry-look ahead)	11
1.4 二進制的減法	17
1.5 八進制系統	18
1.6 十六進制系統	19
1.7 資訊交換用的美國標準譯碼 (ASCII Code)	20
第二章 計算機的基礎知識 (基本準備)	23
2.1 微處理機及微算機	25
2.2 計算機的一步一步動作	25
2.3 開集極閘 (open collector gate)及其應用	28
2.4 三態 (tri-state)	32
2.5 解碼器 (decoder)	33
第三章 微算機的結構	35
3.1 概說	37
3.2 指令處理的流程圖	41
3.3 數據處理的流程圖	42
3.4 記憶體	43
3.5 靜態的 RAM	53

3.6	動態的RAM	60
3.7	動態RAM的 refresh	63
3.8	微處理機	64
3.9	指令及選址	71
3.10	指令的執行	86

第四章 軟體 93

4.1	概說	95
4.2	設計一程式例	96
4.3	流程圖	99
4.4	機械語言	100
4.5	符號語言	104
4.6	用組合語言寫作程式的基本知識	111
4.7	附程式	118
4.8	輸入／輸出	123
4.9	Editors 編輯，校訂	128
4.10	assembler 的動作	129
4.11	載入及偵錯	131

第五章 微算機介面設計（軟體及硬體） 133

5.1	概說	135
5.2	程式控制資料傳送	139
5.3	直接記憶體出入	163
5.4	控制語句及狀態語句	172
5.5	各種不同的資料傳送方法之比較	180
5.6	IC 介面元件	181
5.7	週邊裝置（A/D和D/A的用法，和微算機配合法）	195

第六章 典型的微算機系統（軟體及硬體）	203
6.1 微算機系統用以做交通燈號控制器（軟體、硬體）.....	205
6.2 微算機系統用以做計程車計費錶（硬體、軟體）.....	221
第七章 Intel 8080詳細描述	235
7.1 8080 微處理機.....	237
7.2 時間分配及時間延遲的計算.....	245
7.3 記憶器的選址.....	259
7.4 stack pointer的選址.....	261
7.5 狀態 bits 的使用.....	265
7.6 8080 微處理機指令集.....	267
7.7 程式舉例.....	285
7.8 附程式的使用.....	292
7.9 data 傳送到附程式的方法.....	293
7.10 中斷及輸出／輸入.....	302
附表 港、台与内地计算机名词术语对照	B 1

數目系統

(Number system)

- 1.1 十進制系統及二進制系統
- 1.2 二進制的計算
- 1.3 進位預知加法器
- 1.4 二進制的減法
- 1.5 八進制系統
(Carry-look ahead)
- 1.6 十六進制系統
- 1.7 訊交換用的美國標準譯碼
(Ascii Code)

了解數目系統是了解微算機的第一步驟；由於人類的一雙手具有十根手指頭，所以很自然的就發展出了十進制系統，如果，我們假想人類只有八根手指頭的話，或甚至只有兩根手指頭的話，必然是會發展出 8 進制系統或 2 進制系統。

1.1 十進制系統及二進制系統

1.1.1 十進制系統

人類使用最廣的就是十進制系統，這是由 0~9 總共十個數字 (digit) 組成的，(digit 這個字母是由拉丁字來的，它代表的原意是手指頭或腳趾的意思)。由於數字所在的位置不同，代表的大小就不相同；例如：

374.29

代表的是： $(3 \times 10^3) + (7 \times 10^2) + (4 \times 10^1) + (2 \times 10^0) + (9 \times 10^{-1})$

由此可以看出，十進制的系統是以 10 為基底的 (base 或 radix)

1.1.2 二進制系統 (Binary number system) :

由於計算機 (數位計算機) 是利用電路的“開”或“關”(ON或OFF) 來代表數字的，而且只有兩種狀態 (ON 或 OFF)，故計算機中，只能使用最簡單的二進制系統，(*在十進制中，我們由 0, 1, 2, 3……算到 9 時，便須往十位進 1；而在二進制中，我們由 0, 算到 1, 就須往“十”位進 1 了)，故二進制中，只用 0, 1 兩數字，例如：(註腳的 2 代表 2 進制)

1011, 1101,

代表的是： $(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) + (1 \times 2^{-1})$

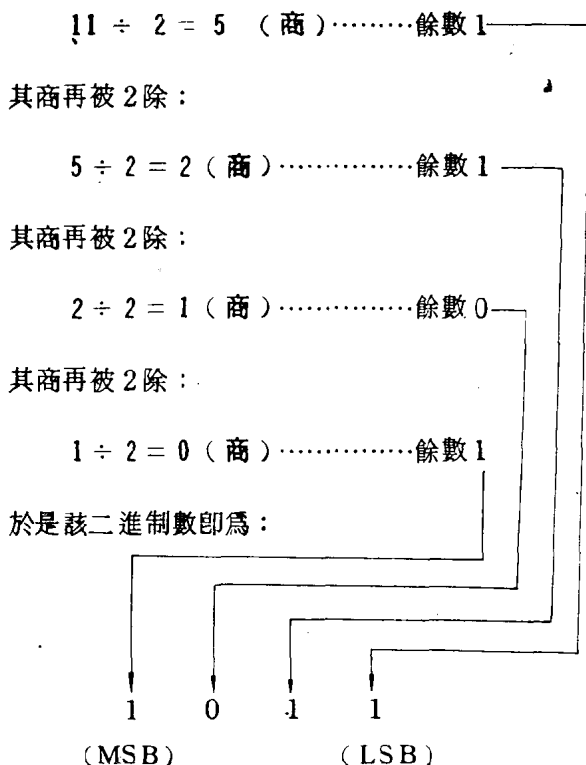
8510394

4 微處理機及微算機硬體、軟體、應用

$+(1 \times 2^{-2}) + (0 \times 2^{-3}) + (1 \times 2^{-4}) = 11.8125_{10}$ 。（註腳的10 代表10 進制）

$$\left[\begin{array}{c} \text{比較:} \\ \begin{array}{ccccccc} & 3 & 7 & 4 & . & 2 & 9 \\ & \swarrow & \downarrow & \swarrow & & \swarrow & \searrow \\ 10^2 & 10^1 & 10^0 & 10^{-1} & & 10^{-2} & \end{array} \end{array} \right]$$

上面的例子也告訴我們如何由二進制變換到十進制，而由10 進制要換到2 進制則比較麻煩。有一種叫做 dabble-dabble 的方法，可以將10 進制的數目變換成任意基底的數目，它的方法是用不斷的除，例如要將 11_{10} 變成2 進制，其方法是這樣的：首先11 被2 除，



最左邊的1 個1，一般稱為此數的最高次數位 (Most significant bit,

1000100

簡稱MSB)，而最右邊的1個1，一般稱為此數的最低次數位(Least significant bit, 簡稱LSB)。

而如果10進制的小數要變成二進制，則必須用不斷的乘法，例如要將 0.8125_{10} ，換成二進制，則必須如下變換：

$$0.8125 \times 2 = 1.6250 \Rightarrow 0.6250 \quad \dots\dots\dots \text{個位有 1, (MSB)}$$

$$0.6250 \times 2 = 1.2500 \Rightarrow 0.2500 \quad \dots\dots\dots \text{個位有 1}$$

$$0.2500 \times 2 = 0.5000 \quad \dots\dots\dots \text{個位有 0}$$

$$0.5000 \times 2 = 1.000 \Rightarrow 0.000 \quad \dots\dots\dots \text{個位有 1, (LSB)}$$

於是，轉換所得變是：

$$0.8125_{10} = 0.1101_2$$

如果一個數，像 11.8125_{10} ，既有整數11，又有小數0.8125，則必須分別轉換，像上面所算的步驟一般，再將其結果，合併起來，即：

$$11.8125_{10} = 1011.1101_2$$

有一點讀者須稍微注意的是，由十進制小數要換到二進制時，我們不斷的用2去乘該數，乘的結果，會得到1.0000的可能性並不大（參看上面由 0.8125×2 連乘到第4次時的結果），即要得到整數1的可能性並不大，往往會一直乘2，而得不到整數。例如 0.8313_{10} 要換成二進制：

於是

$$0.8313 \times 2 = 1.6626 = 0.6626 \quad \dots\dots\dots \text{個位 1 (MSB)}$$

$$0.6626 \times 2 = 1.3252 = 0.3252 \quad \dots\dots\dots \text{個位 1}$$

$$\begin{array}{rcl}
 0.3252 \times 2 = 0.6504 & \cdots & \text{個位 } 0 \\
 0.6504 \times 2 = 1.3008 = 0.3008 & \cdots & \text{個位 } 1 \\
 0.3008 \times 2 = 0.6016 & \cdots & \text{個位 } 0 \\
 \vdots & & \vdots
 \end{array}$$

如此，一直乘下去，不知何年何日才會得到 LSB（最後一位），這時，由個人決定取幾位，如取 5 位數，則到 11010 即可。當然，愈多位，精密度愈高。

1.2 二進制的計算

一般二進制的數目是由一些 1 及 0 組合而成的，如 1101。其基底為 2，故：

$$\begin{array}{cccc}
 1 & 1 & 0 & 1 \\
 \uparrow & \uparrow & \uparrow & \uparrow \\
 2^3 & 2^2 & 2^1 & 2^0
 \end{array}$$

代表十進制的 $(1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = (1 + 8) + (1 \times 4) + (1 \times 1) = 13$ ，表 1.1 中，是將十進制的每個數字用 2 進制的數目來代表。如 9 表示成 $1011 = 1 \times 2^3 + 1 \times 2^0 = 1 \times 8 + 1 \times 1 = 9$ 。

二進制的加法和十進制是相同的，只是當任兩個數字相加必須運用下列法則：

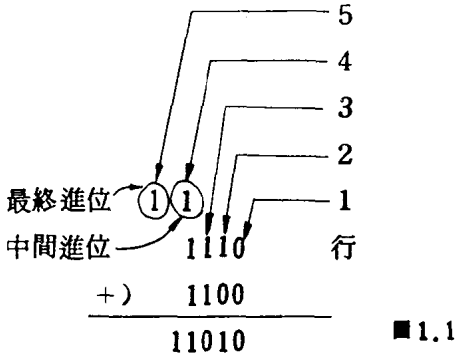
$$(a) 0 + 0 = 0$$

$$(b) 0 + 1 = 1$$

$$(c) 1 + 1 = 0 \cdots \cdots \text{進位 } 1$$

(d) $1 + 1 + 1 = 1 \dots\dots\dots$ 進位 1

例如 1110 要加 1100，如圖 1.1 所示，在第一行， $0 + 0 = 0$ ；第二行



$0 + 1 = 1$ ；第 3 行， $1 + 1 = 0$ 進位有 1，這個進位一般叫中間進位；第 4 行則為 $1 + 1 + 1 = 1$ ，進位有 1，整個最終進位亦是 1。

如果我們考慮兩個數字相加，運用上面的二進制加法法則，可以得知這二數字的和 SUM 應等於：

$$\text{SUM} = (A \wedge \bar{B}) \vee (\bar{A} \wedge B) = A \oplus B$$

而進位則等於：

$$\text{CARRY} = A \wedge B$$

($\oplus$ 代表 exclusive or, $A = 1, B = 1, A \oplus B = 0$ ，而 $A = 0, B = 0, A \oplus B = 0$ ，又 $A = 1, B = 0, A \oplus B = 1$)，正符合上列加法原則。表 1.2 表示此真值表。

圖 1.2 是一可執行這種加法的電路，這電路一般稱為半加法器 (half-adder)，半加法器只能執行兩個數字 (digit) 相加。