

6800 微處理器與 微電腦

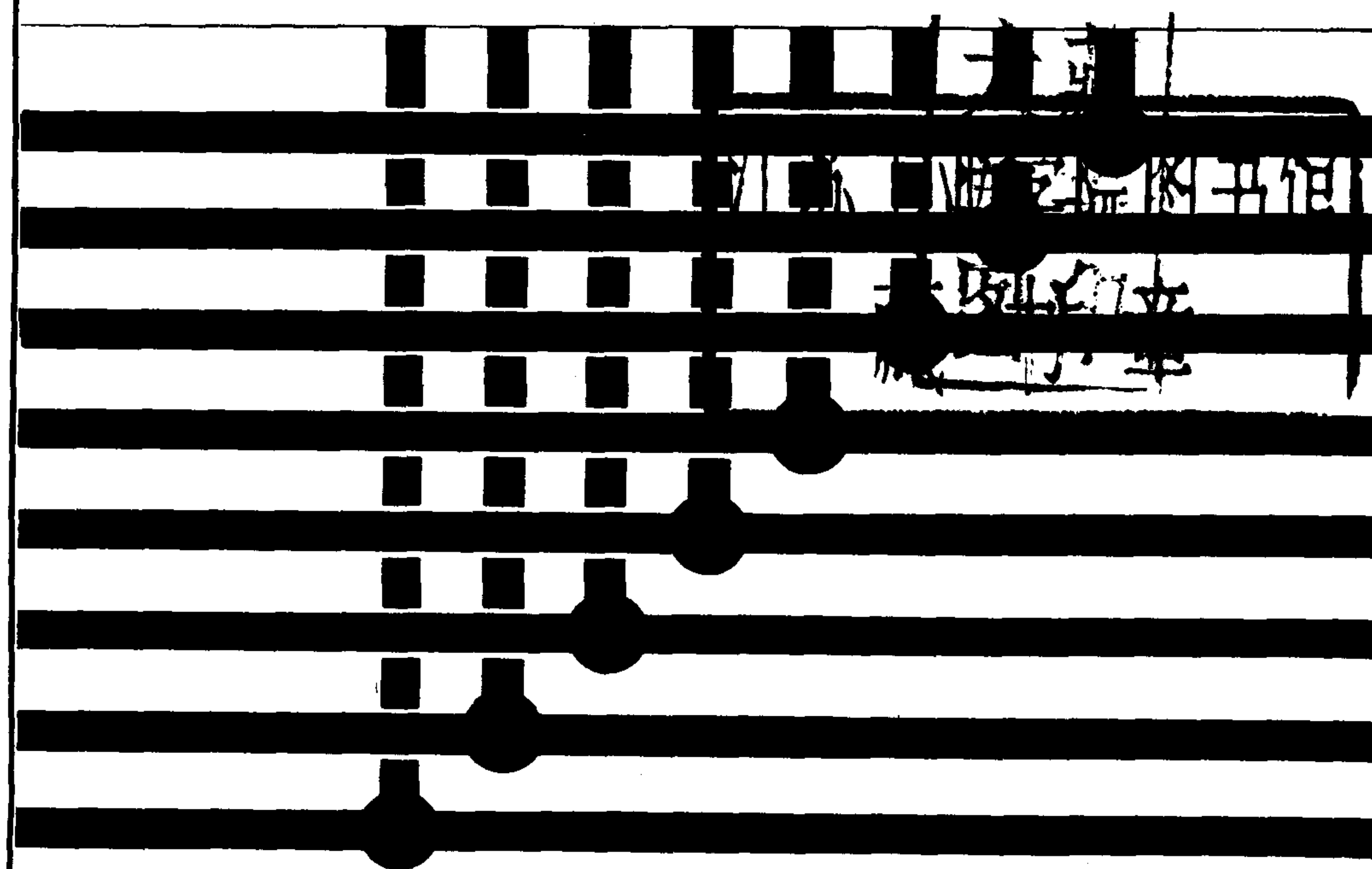
石印滇 編著



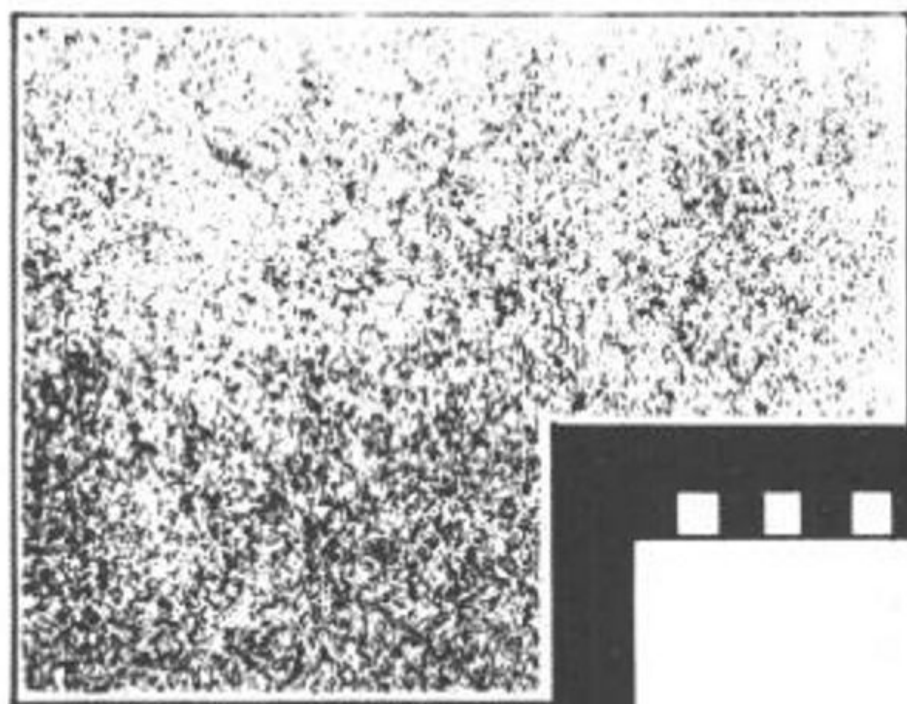
全華科技圖書股份有限公司 印行

6800 微處理器與 微電腦

石印滇 編著



全華科技圖書股份有限公司 印行



微處理器（microprocessor）這個咬不動的洋芋片是目前許多工商用微電腦的心臟。市面上主要微處理器有 6502，6800，及 8080 等。本書的重點是討論廣受使用的 6800 微處理器及有關的微電腦。

現在是電腦時代。許多人爭先恐後地學電腦，都想找碗電腦飯吃，這是可喜的現象，因為目前各行業幾乎都離不開電腦的幫忙。據專家預測，在不久將來，微電腦會成為很多家庭的必需品，用來處理報稅，收支平衡，子女課業輔導，家用電器控制，娛樂等事情。有關微處理器及微電腦的就業機會非常多。希望您能“軟硬兼施”，一定會成為這方面的高手。

本書內容由淺入深。在第一章中，您能初步瞭解微處理器這複雜的電子小玩意會玩些什麼鬼把戲。小小的一塊晶片竟包含成千上萬極小，極小的電晶體等，它們組成許多不同的電路，像暫存器（register），計數器（counter），解碼器（decoder）等。這些重要的電路是如何打成一片去執行工作呢？請看本章就明白。

第二章所談到的程式都很簡單，任何人都能懂。但是，別小看它們簡單，它們可帶給您不少重要的程式概念。研究這些程式後，您能瞭解微處理器是如何去應付更複雜的工作。

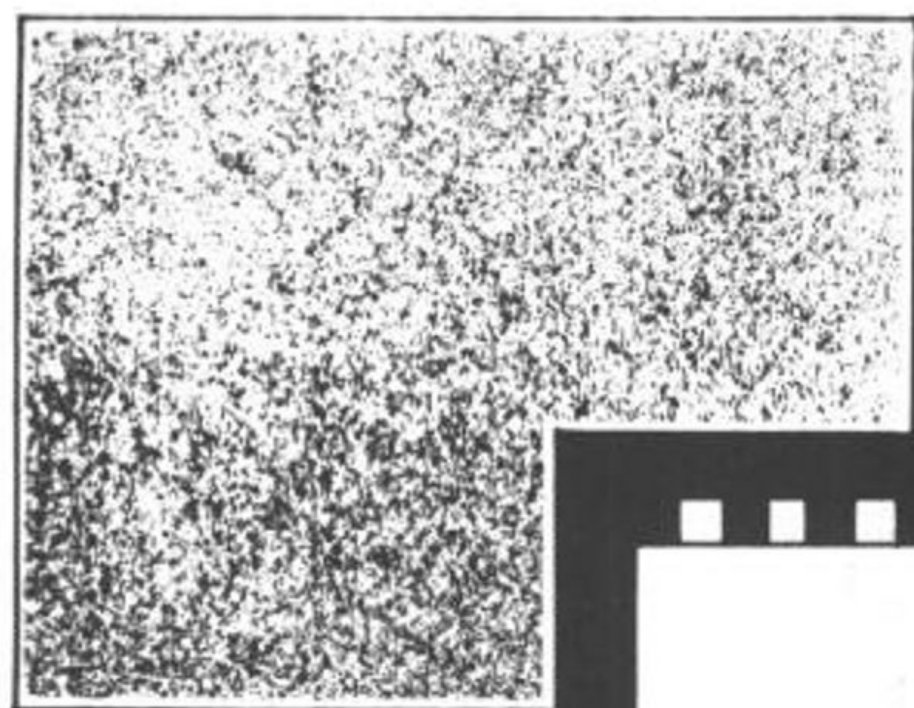
前二章用一個簡單的微處理器模型來使您對微處理器有個粗淺的認識。但在第三章中，要玩真刀實槍了。這一章介紹 6800 微處理器的指令群，結構，及定址方式。美國 MOTOROLA 公司在 1975 年左右首先推出這個迷人的東西。後來，該公司又推出改良的產品，如 6801 及 6802。市面上有許多類似 6800 的產品，所以只要熟悉 6800，就能摸清楚類似產品的脾氣。

微處理器的玩法有二種。第一種玩法是用它發展出許多實用的程式，在前面三章中已談過。第二種玩法是在微處理器與其他外面電路之間裝設適當的電路或器件，來看看它與其他系統能聯手幹些什麼好事。第四章就是談第二種玩法。您最好有點數位電子學基礎，才能瞭解更透徹。

附錄一及附錄二討論美國有名的 HEATH 公司出品的 ET-3400 微電腦的電路工作原理，修理技術，有關介面的電路實驗，及微程式例子。

在鄙人編寫這本書期間，雙親及弟妹不斷鼓勵寫下去，特在此萬分感激他們。鄙人才識有限，雖已盡全力，但難免有錯誤的地方，敬請您不吝指正。

石印滇 謹識



第一章 微電腦基本觀念	1
1.1 專用術語解釋.....	1
1.1.1 儲存程式.....	2
1.1.2 電腦字元.....	3
1.1.3 字元長度.....	3
1.2 一個基本微電腦.....	5
1.2.1 微處理器單元.....	5
1.2.2 記憶體.....	8
1.2.3 提取—執行順序.....	10
1.2.4 程式範例.....	11
1.3 執行一個程式.....	13
1.4 定址方式.....	22
1.4.1 固有或隱含定址.....	22
1.4.2 立刻定址.....	22
1.4.3 直接定址.....	24
1.4.4 用直接定址方式的程式範例.....	26

1.4.5 執行程式範例.....	27
1.4.6 混合定址方式.....	38
第二章 介紹程式規劃	41
2.1 轉 道.....	41
2.1.1 相對定址.....	42
2.1.2 執行轉道指令.....	43
2.1.3 向前轉道.....	47
2.1.4 轉道回頭.....	49
2.2 條件轉道.....	51
2.2.1 條件暗碼.....	53
2.2.2 條件轉道指令.....	56
2.3 演算法.....	57
2.3.1 用重複加法來達到“乘”的目標—積.....	57
2.3.2 利用重複減法來求商.....	61
2.3.3 把二進碼十進數變成二進位數.....	64
2.3.4 把二進位數變成二進碼十進數.....	68
2.4 其他指令.....	71
2.4.1 帶進位加指令.....	71
2.4.2 帶進位減指令.....	73
2.4.3 算術上把累積器內容向左移指令.....	75
2.4.4 十進位調整累積器指令.....	77
第三章 6800微處理器	81
3.1 6800 微處理器單元的結構.....	81
3.1.1 6800 微處理器單元的程式規劃模式.....	83
3.1.2 6800 微處理器單元的方塊圖.....	85
3.2 6800 微處理器單元的指令群.....	87
3.2.1 算術指令.....	87
3.2.2 處理數據的指令.....	87
3.2.3 邏輯指令.....	93

3.2.4	數據測試指令.....	94
3.2.5	指標暫存器及堆疊指示器指令.....	95
3.2.6	轉道指令.....	95
3.2.7	條件碼暫存器指令.....	99
3.3	新定址方式.....	101
3.3.1	延伸定址.....	102
3.3.2	指標定址.....	103
3.4	堆疊操作.....	108
3.4.1	串接堆疊.....	109
3.4.2	記憶型堆疊.....	111
3.5	次常式.....	117
3.5.1	跳越指令.....	118
3.5.2	跳越到次常式指令及從次常式返回指令.....	120
3.5.3	巢型次常式.....	122
3.5.4	轉道到次常式指令.....	123
3.5.5	次常式指令群.....	123
3.6	輸入／輸出操作.....	125
3.6.1	輸出操作.....	126
3.6.2	輸入操作.....	128
3.6.3	輸入／輸出的程式規劃.....	129
3.6.4	用程式來控制輸入／輸出的操作.....	131
3.6.5	用中斷來控制輸入及輸出的操作.....	131
3.7	中斷.....	132
3.7.1	重置.....	132
3.7.2	不可遮罩中斷.....	135
3.7.3	有關中斷的指令群.....	137
3.7.4	從中斷返回指令.....	137
3.7.5	中斷請求.....	139
3.7.6	中斷遮罩指令.....	141
3.7.7	軟體中斷指令.....	141
3.7.8	等待中斷指令.....	141

第四章 介 面	143
4.1 介面基本概念	143
4.1.1 匯流排	143
4.1.2 三態邏輯	145
4.1.3 6800 微處理器單元的介面線路	148
4.1.4 指令的時序	152
4.1.5 用一段程式來說明時序	153
4.2 微處理器與隨機出入記憶器的介面關係	156
4.2.1 靜態隨機出入記憶型儲存胞	156
4.2.2 一個 128 字元 (每個字元有 8 個數元) 的隨機出入記憶器	158
4.2.3 一個有 256 個數元組 (4 數元) 的隨機出入記憶器	163
4.2.4 把隨機出入記憶器接到微處理器單元上	165
4.2.5 位址解碼	166
4.3 微處理器與顯示器的介面關係	168
4.3.1 7 段顯示器	168
4.3.2 推動 7 段顯示器	169
4.3.3 運用可定位址的栓器	173
4.3.4 多工顯示器	177
4.4 微處理器與開關的介面關係	179
4.4.1 介面條件	179
4.4.2 一個典型的鍵盤	181
4.5 週邊介面轉換器	189
4.5.1 輸入 / 輸出圖	190
4.5.2 週邊介面轉換器中的暫存器	191
4.5.3 預置週邊介面轉換器	193
4.5.4 定週邊介面轉換器的位址	195
4.6 週邊介面轉換器的運用	197
4.6.1 推動 7 段顯示器	197
4.6.2 解鍵盤的暗碼	200
4.6.3 解開關矩陣的暗碼	202

附錄 I ET-3400微電腦205

I.1 字鍵的意義及功用205

I.2 ET-3400的電路工作原理209

I.3 ET-3400的修理技術212

I.3.1 初步修理212

I.3.2 深入修理214

(1) 二進位數據 LED 顯示燈的檢修214

(2) 計時脈波產生器的檢修215

(3) 重置的檢修215

(4) 位址線的檢修215

(5) 數據線的檢修218

(6) 控制線的檢修220

(7) 解碼的檢修221

I.4 ET-3400的記憶對應圖、記憶解碼圖及鍵盤、顯示器功能位址圖223

附錄 II ET-3400的實驗電路討論及微程式例子227

II.1 記憶實驗討論227

II.2 位址解碼實驗討論230

II.3 數據輸出實驗討論232

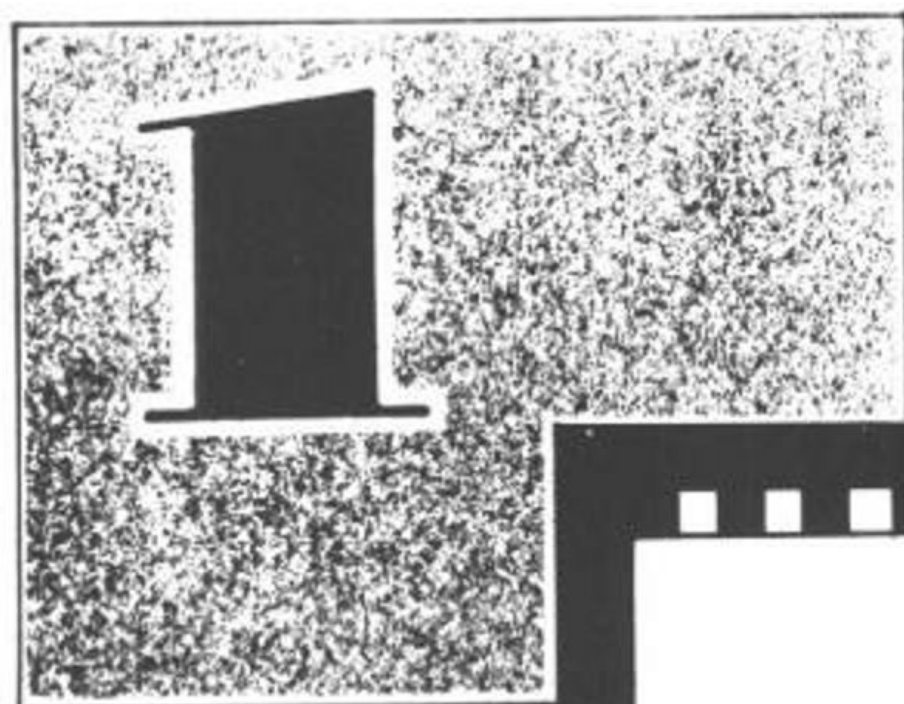
II.4 數據 7 段顯示器輸出實驗討論234

II.5 數據輸入實驗討論235

II.6 週邊介面轉換器實驗討論237

II.7 類比變成數位的轉換線路討論240

II.8 ET-3400 微電腦的應用微程式例子243



微電腦基本觀念

1.1 專用術語解釋

微處理器是一種邏輯器件，用在數位電子系統中。業餘電子愛好者，實驗者及小規模研究團體都喜歡它，因為它的售價不貴，用途又廣泛。不過，微處理器和微電腦之間有顯著的差異。

微處理器單元（縮寫是MPU）能做算術，邏輯及控制的操作。目前的趨勢是儘量想辦法把它縮小又縮小，成為非常小的IC晶片。

微電腦包含微處理器單元，但它也包括其他的電路，像記憶器用來儲存資訊，介面轉換器用來使它和外界發生關係，與計時脈波（clock）產生器用來控制整個系統的時序（Timing）。圖1.1顯示出一個典型微電腦的結構。窄箭頭代表導體，二進位資訊在上面流動。寬箭頭代表並聯的幾個導體。一群並聯又能傳送資訊的導體稱為匯流排（bus）。

微電腦包括大長方形內所有東西。大長方形外的東西稱為外界，所有微電腦一定有某種方法和外界打交道。微電腦從外界收到的資訊稱為輸入數據（data）。微電腦所送到外界的資訊稱為輸出數據。

輸入資訊的方式有紙帶閱讀機、打字機、機械開關、鍵盤，或其他的電腦。輸出資訊的方式有影像顯示器、輸出用印字機、紙帶打孔機，或印刷機。有些器件，像電

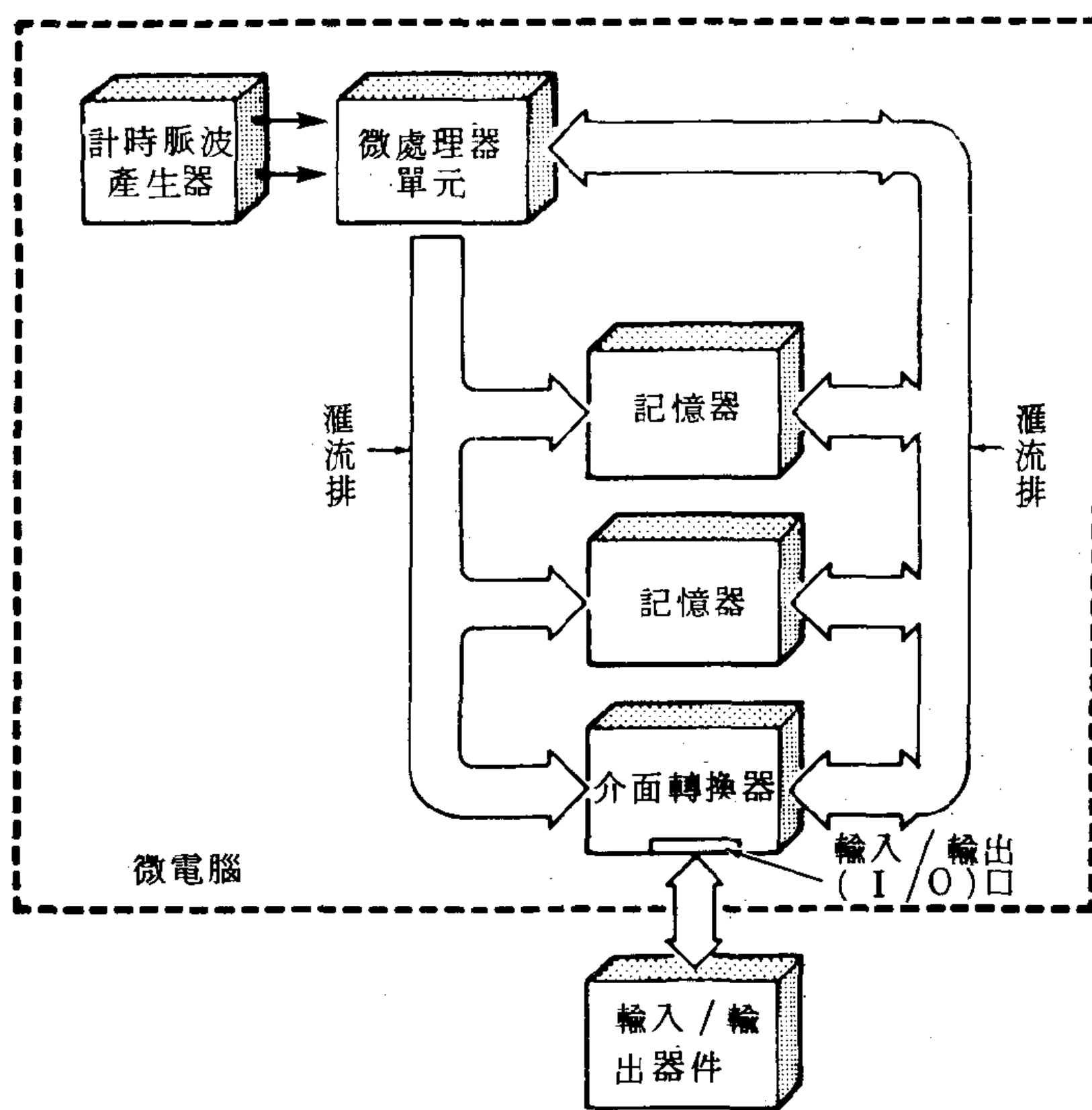


圖 1.1 典型微電腦的結構

傳打字機 (teletypewriter)，可作輸入器件，也可以作輸出器件。這些器件稱為輸入 / 輸出器件。輸入 / 輸出器件和微電腦相連的地方稱為輸入 / 輸出口。

1.1.1 儲存程式概念

微電腦能執行許多不同的操作。它能加、減，也能執行邏輯的操作。它能從輸入用器件讀入資訊，也能把資訊送到輸出用器件那兒。根據所使用的微處理器，微電腦能執行的操作差不多有 100 多種。有時，二個或二個以上的操作能組合起來去執行更複雜的操作。

但是，不管電腦是如何地神通廣大，它自己仍然無法獨立思考。它只能遵命行事。您必須告訴電腦執行什麼樣的操作及執行的順序。電腦受命執行的操作稱為指令 (instruction)。一些典型的指令有加、減、裝入指標暫存器 (index register)，裝入累積器中 (load accumulator)，及暫停 (halt)。

能使電腦去執行某一特定工作的一群指令稱為程式。寫這些指令的人稱為程式師。要想用微處理器來設計電腦，工程師們必須是優秀的程式師。要想修護以微處理器為主的設備，技術員必須熟悉程式。通常，程式的長短和電腦要執行的工作的複雜程

度成正比。加一系列數字的程式或許只要 12 個指令。但是用來控制一個小城市交通燈的程式可能需要成千上萬的指令。

電腦有時和計算機扯在一起，操作者按壓鍵盤就能告訴計算機該做什麼。有些便宜的計算機也能執行一些類似電腦指令的操作。只要鍵按壓的對，就能命令計算機加、減、乘、除，及清除顯示字幕。當然，要先輸入數字才能加、減等。有了計算機，您就能盡您最快的速度來加一系列數字。操作者的按壓速度及準確性決定操作的表現。

從一開始，電腦設計者就知道是人為的操作使計算過程減慢下來。爲了克服這個缺點，儲存程式概念應運而生。這種方法能使程式儲存在記憶器中。例如，有個程式有 100 個指令，要處理 20 個數字。它會產生 10 個答案，在計算開始之前，包括 100 個指令及 20 個數字的程式先存入電腦的記憶器中。還有，10 個記憶位置要保留給 10 個答案用。這時才能開始命令電腦執行整個程式。所花的時間可能不到百萬分之一秒。如果用手輸入指令及數字，一次一個，那麼所花的時間會更長。這一點自動操作的特徵使電腦壓倒計算機。

1.1.2 電腦字元

在電腦專用術語中，字元（word）是指一群二進位數字（binary number），能夠佔據一個儲存位置。雖然字元可能包含幾個二進位數字，但是電腦把每個字元看作是一個單元。

一個字元可能是一個二進位數字，用作數據，也可能是一個指令，告訴電腦該執行什麼操作。也可能是一個美國標準資訊交換暗碼（American standard code for information interchange-ASCII），用來代表一個英文字母。還有，一個字元也可能是一個位址（Address），告訴電腦到什麼地方找一個數據。

1.1.3 字元長度

在過去幾年裏，市面上出現了許多種微處理器。它們的價錢及功能差別很大。微處理器的最重要特徵是它能處理的字元長度。這是指數元（BIT）的長度。數元是最基本的資訊單位。

最常見的字元長度是 8 個數元。數字、位址、指令及數據都是用有 8 個數元的二進位數字代表。最低值的 8 - 數元二進位數字是 0000 0000₂ 或 00₁₆。最高的是 1111 1111 或 FF₁₆。用十進位表示，這個範圍是從 0 到 255₁₀。所以，一個 8 -

數元二進位數字可代表 256_{10} 個不同值中的一個值。如果代表指令的話，它可代表 256_{10} 個指令中的一個指令。也許，8 - 數元字元是一個 ASCII 字母，在這種情形中，它可代表英文字母，標點符號，或其他符號。8 - 數元的字元可代表許多不同的東西，這要看如何解釋它。程式師必須分辨出 ASCII 字母或某些二進位數字不能當作指令。

8 - 數元的字元長度是最普遍的，但是有時會用到其他字元長度。最早期的微處理器用 4 - 數元的字元長度，到現在仍存在，這是因為它的售價便宜。一些 12 - 數元及 16 - 數元的微處理器也已發展成功。

比較長的字元可使我們處理比較大的數字。例如，一個 16 - 數元的字元能代表的最大值是 65535_{10} 。但是，結構更複雜，不易設計，還有價錢更貴。因為大多數的微處理器用 8 - 數元的字元長度，我們只討論這類單位。

有一點要指出，並不因為字元長度是 8 - 數元，我們所能處理的最大值只是 256_{10} 了。我們可以用二個或更多的字元來代表更大的數字。

8 - 數元的字元長度限制了微處理器系統中組件的大小。例如，只能用 8 - 數元的暫存器，記憶器只能儲存 8 - 數元的字元。還有，用來輸送數據字元的匯流排只包括 8 個平行的導體。

有些 16 - 數元的微處理器也用 8 - 數元片斷數據。例如，電傳打字機輸入的資訊經常包括 8 - 數元 ASCII 字母。為了區別它和 16 - 數元（或更長）字元長度的不同，另一個名詞應運而生，就是數元組（byte）。數元組是一群數元，可作為單位。通常，大家都知道一個數元組包含 8 個數元。在 8 - 數元微處理器中，每個字元包括一個數元組。但是在 16 - 數元的機器中，每個字元包括兩個數元組。圖 1.1.3 顯示這點。

圖 1.1.3 也顯示出如何稱呼字元中的數元。最低值數元在右邊，最高值數元在最左邊。在 8 - 數元的字元中，從右到左，數元的排號是 0 到 7。在 16 - 數元的字元中，數元的排號是 0 到 15，如下圖所示。比較低的 8 - 數元稱為低層數元組，比較高的 8 - 數元稱為高層數元組。

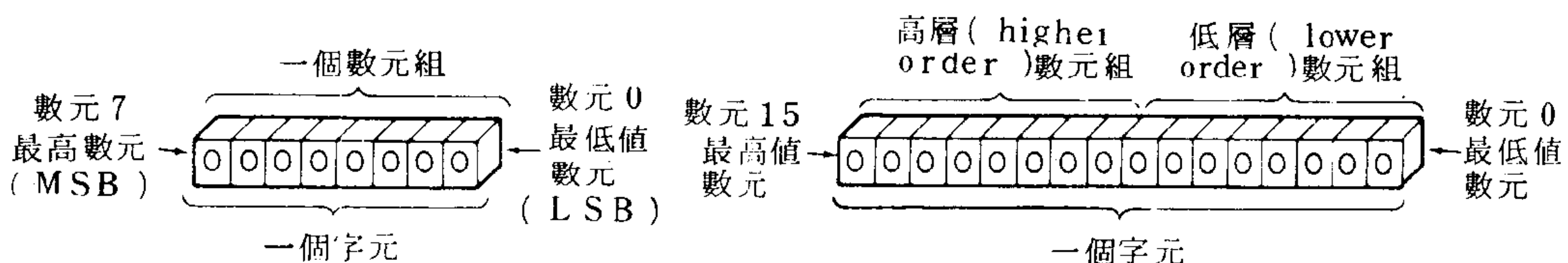


圖 1.1.3 字元與數元組

1.2 一個基本微電腦

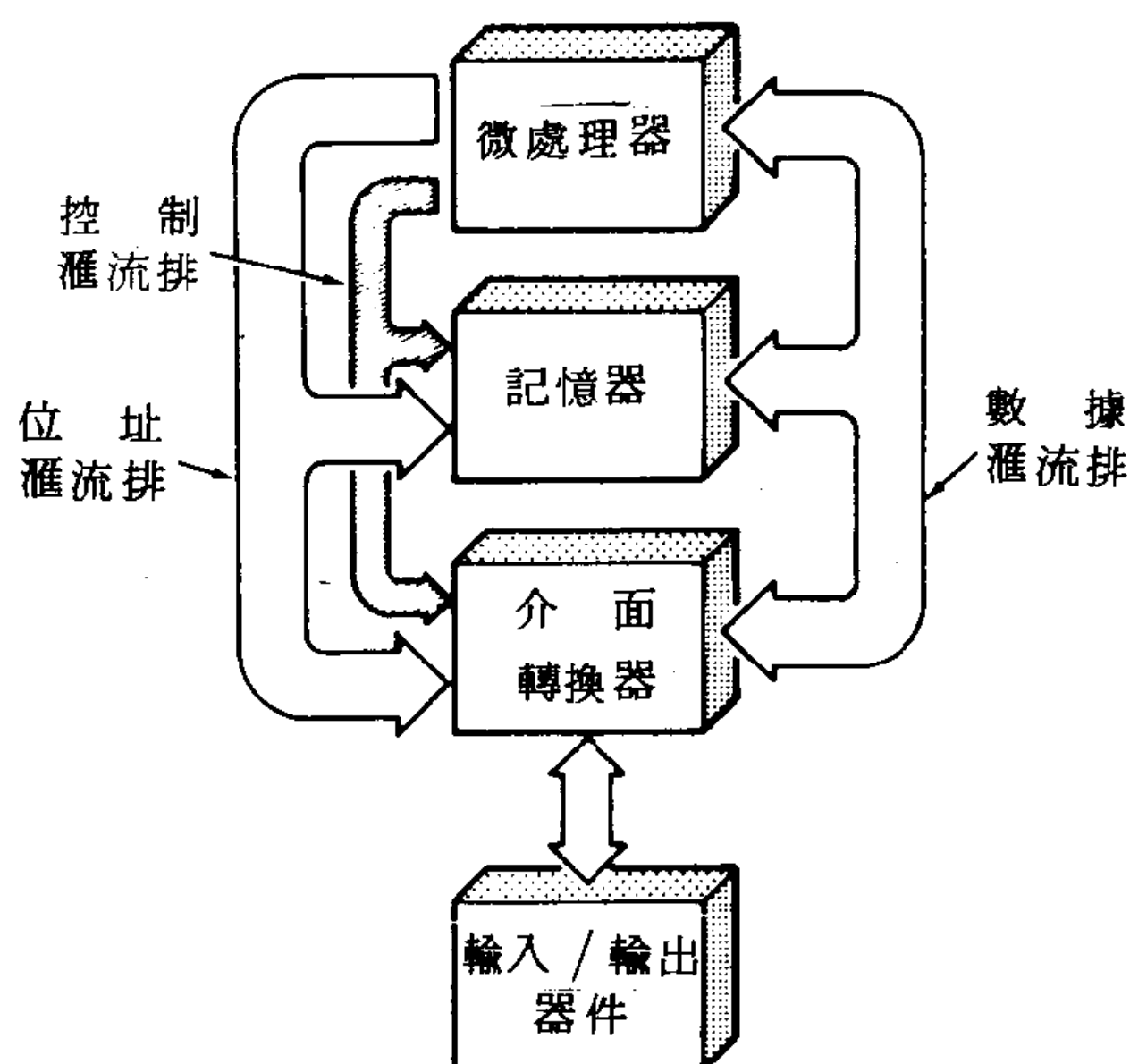


圖 1.2 基本微電腦方塊圖

初次學微電腦的時候，比較困難的問題就是微處理器。通常，它包含 12 個或更多的暫存器，大小不同，從 1 - 數元到 16 - 數元的。它有不少的指令，還有，執行指令的方式可不少。它有數據，位址及控制匯流排。我的天啊！這麼複雜的暗器，怎麼玩啊！

要想玩的起勁，最好把它一層層地剝開來看。先不管微處理器中令人頭痛的法寶，我們來玩一個和真微處理器差不多的模型。別小看它，它的工作原理和目前流行的真微處理器差不多啦！

圖 1.2 顯示出一個基本微電腦的方塊圖，內有微處理器，記憶體，及輸入／輸出電路。爲了簡化起見，先不考慮輸入／輸出電路。這就是說，假設程式及數據已在記憶體中，運算的結果將放在暫存器中或存入記憶體中。當然囉，程式及數據必須來自外界，所得的結果必須送到外界中。在以後幾章裏，會討論這些介面關係。這麼一來，只要對付微處理器及記憶體就好了。

1.2.1 微處理器單元

圖 1.2.1 顯示出微處理器單元的結構。爲了簡化起見，只有主要的暫存器及主要線路亮相。在圖中，計數器、暫存器及匯流排都是 8 - 數元寬。這就是說，他們能夠應付 8 - 數元的字元。

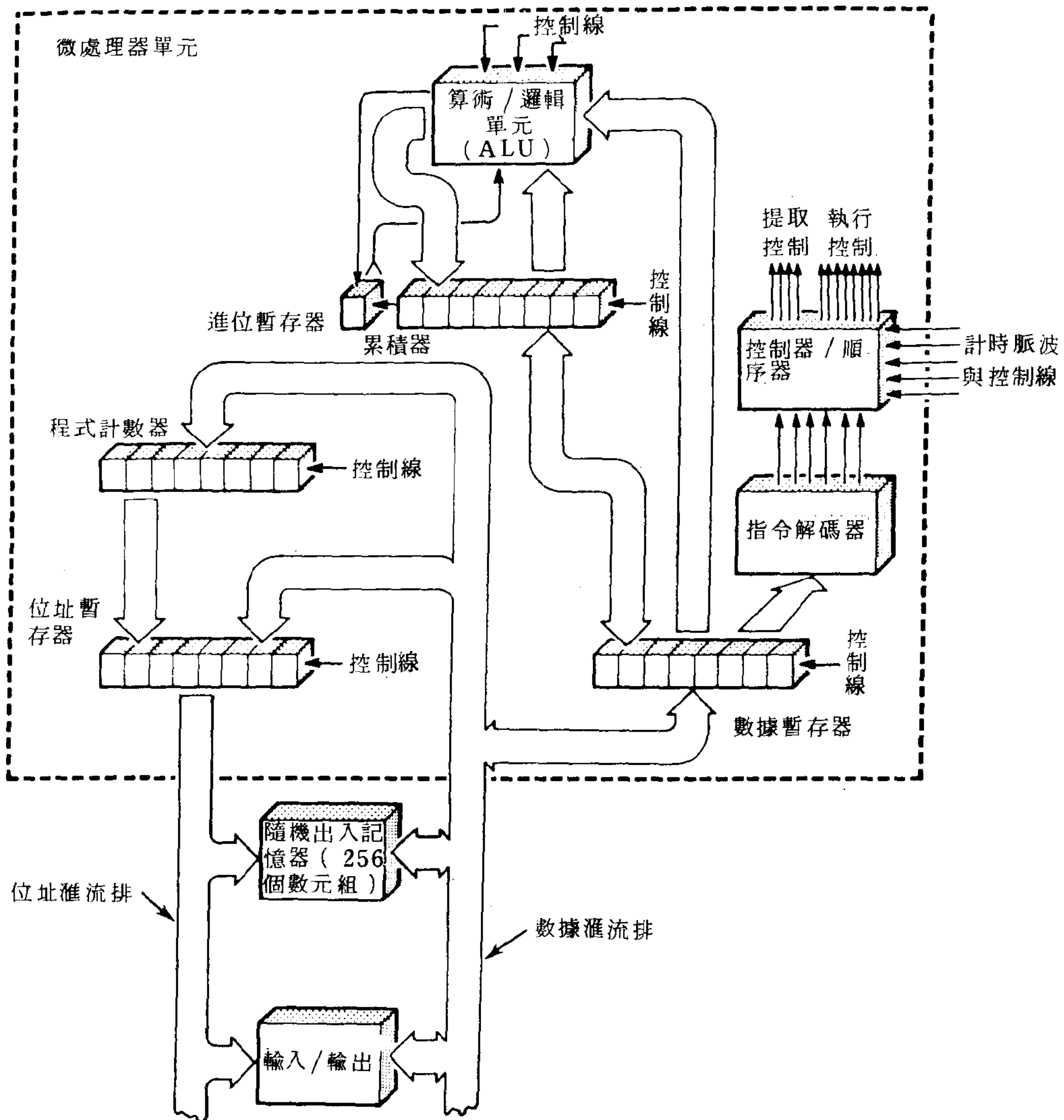


圖 1.2.1 一個基本的微處理器

在微處理器中，最重要的線路之一是算術—邏輯單元（arithmetic logic unit-ALU）。當數據字元送到這個單元時，它就處理有關的算術或邏輯運算。算術—邏輯單元有兩個主要輸入。其中一個來自叫作累積器的暫存器，而另一個來自數據暫存器。算術—邏輯單元能夠把輸入的兩個數據字元加在一起，或者它能夠使一個輸入數據字元減去另一個輸入數據字元。它也能夠處理一些邏輯的運算。算術—邏輯單元所能處理的運算完全由不同控制線上的信號加以控制。

通常，算術—邏輯單元從累積器及數據暫存器那兒，接受兩個 8 - 數元二進位數字，如圖 1.2.1 (a) 所示。這兩個被耍弄的輸入數據字元，叫做運算元 (Operand)。

上述兩個運算元可以相加，相減，或以某種方式相互比較，運算的結果存回累積器中。譬如說，假設兩個數字 (7 與 9) 要加在一起。在加之前，其中一個運算元 (9) 被放入累積器中；另一個 (7) 被放入數據暫存器中。然後，適當的控制線一旦被啟動，就會執行加的操作。算術—邏輯單元把兩個數加在一起，產生“和” (16_{10})，把它送出去。如圖 1.2.1 (a) ② 所示，“和”被存入累積器中，取代了原先儲存在那兒的運算元。注意，所有涉嫌的數字是二進位數字。

累積器是微處理器中最有用的暫存器。在算術及邏輯的運算中，它表演雙重角色。在運算之前，它抓住其中一個運算元。在運算之後，它改抓“和”，“差”，或邏輯答案。平常，在微處理器中，累積器會收到好多個指令。譬如說，“裝入累積器”指令，可使某一特定記憶位址內的數據被轉送到累積器那。“儲存累積器” (store accumulator) 指令，可使累積器的內容被存放在某一特定記憶位址中。

數據暫存器，顧名思義，可知道來往於它和數據匯流排之間的數據，只把它作為暫時落脚休息的地方。譬如說，它會抓住一個要被破碼的指令。它也會抓住一個要被存入記憶器中的數據數元組。

微處理器也包含好多其他重要的暫存器及電路：像位址暫存器，程式計數器，指令解碼器，控制器—順序器。如圖 1.2.1 所示。

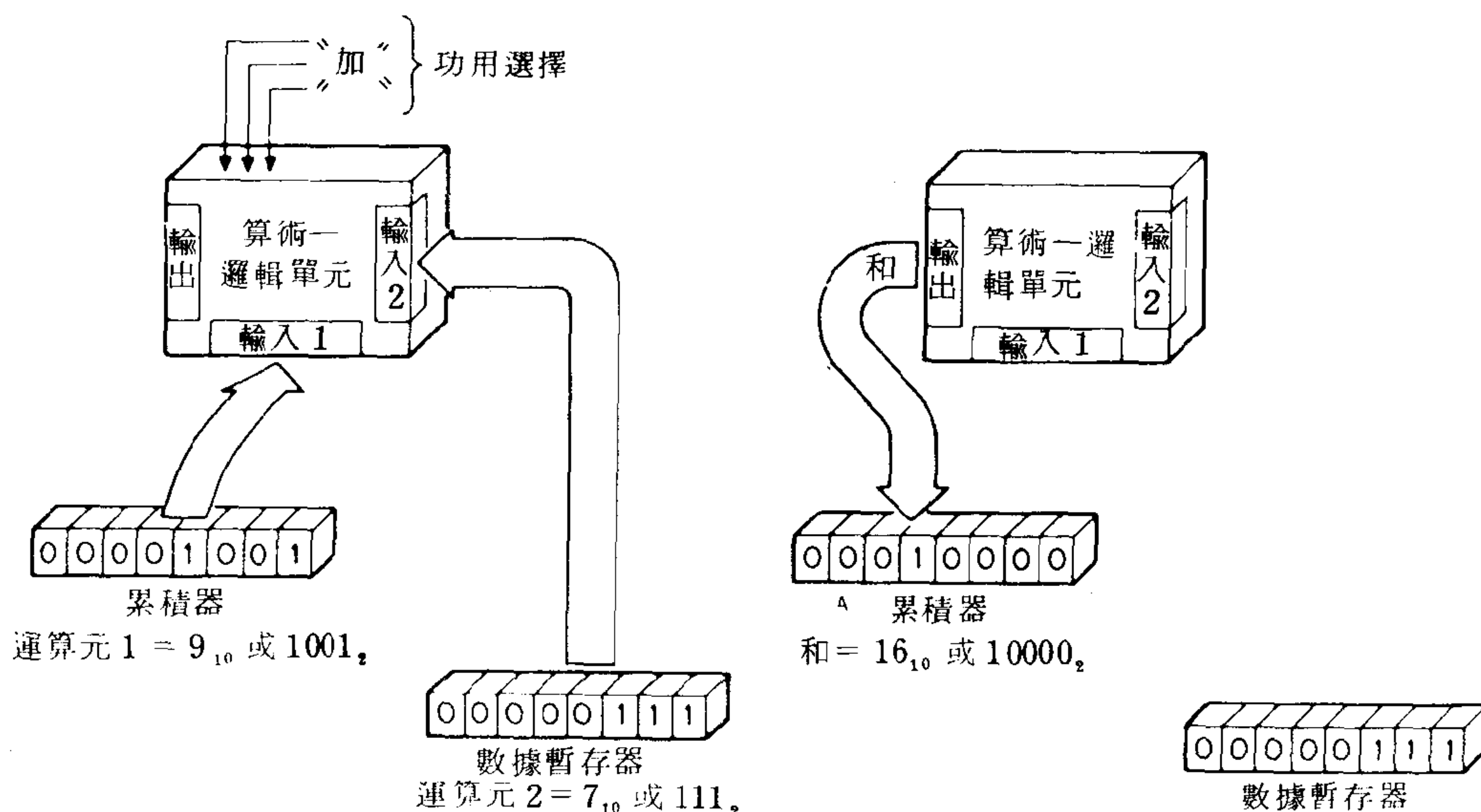


圖 1.2.1 (a) 算術—邏輯單元

位址暫存器是另一個暫時儲存的地點。在運算中所用到的記憶位置或輸入／輸出器件，就是把它們的位址交給位址暫存器暫時保存一下。

程式計數器控制住一個程式中所有指令上台表演的先後順序。通常，它以 0, 1, 2, 3, 4……來計算指令操作的先後順序。在任何指定的一刻，計數器會顯示出資訊中的次一個數元組將會從記憶器中什麼鬼位置滾出來。

指令解碼器的功能，用不着詳細說明，各位一定已瞭解七、八分了。一個指令從記憶器中被拖出來，放入數據暫存器之後，這個指令的暗碼就會被解碼器破解。解碼器檢查 8 - 數元的暗碼後，就決定要執行的操作是什麼。

控制器—順序器 (controller-sequencer) 產生許多不同的控制信號來執行指令。因為每個指令不同，所以控制信號要組合成不同的陣容來執行指令。這個電路決定有關事件的先後順序，來完成指令的運算。

1.2.2 記憶器

在假微電腦模型中，讓我們使用一個具有 256 - 字元，8 - 數元的讀／寫記憶器，如圖 1.2.2 (a) 所示。這個記憶器包含 256₁₀ 個位址，每個位址能夠儲存一個 8 - 數元的字元。這種記憶器通常叫作 256 × 8 記憶器。在讀／寫記憶器中，數據能夠被寫入及讀出，很方便。

兩個匯流排及不少的控制線把記憶器和微處理器單元連接在一起。位址匯流排把 8 - 數元的二進位數字，從微處理器單元輸送到位址解碼器那兒；8 - 數元的二進位

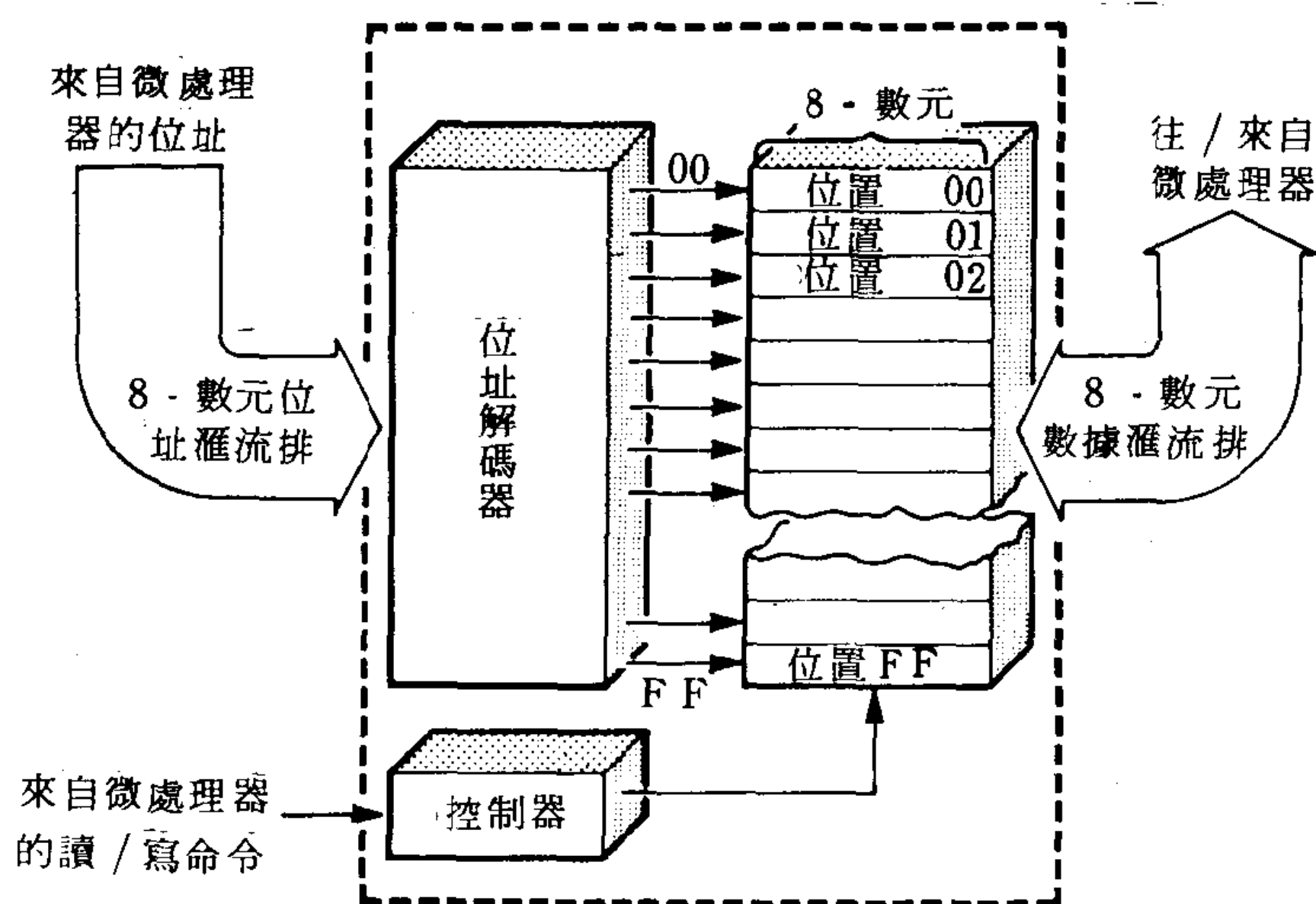


圖 1.2.2 (a) 隨機出入記憶器