

世界學術著作譯本

計算機系統結構

(全一冊)

梅諾 (M. Morris Mano) 原著
張 煒 譯

譯作權所有人 國立編譯館
印 行 者 復 興 書 局

世界學術著作譯本

計算機系統結構

(全一冊)

梅諾(M. Morris Mano)原著
張 焯 譯

譯作權所有人 國立編譯館
印行者 復興書局

版權所有

中華民國六十八年五月初版

計算機系統結構

全一冊（平裝本）~~售價壹佰柒拾肆元~~
一九三元

原著者 梅 諾 (M. Morris Mano)

譯 者 張 燁

著作權人 國立編譯館

印行者 復興書局

臺北市懷寧街四十四號

電話：371-6911・331-5012

發行人 沈 亦 珍

本書局登記證字號：行政院新聞局

版臺業字第〇八一八號

序 言

計算機結構 (*computer architecture*) 是一種術語，通常用來表示數位計算機的組織與設計。這學科與處理計算機系統硬體設計的計算機工程師有關，亦與專心於依靠軟體系統來作硬體設計的計算機科學家有關。計算機系統是包括有軟體與硬體兩者的一種系統。此系統的硬體部分為實體組元及所有有關設備組成。其系統的軟體部分參照運算程式的搜集，其目的是使計算機的用處更為有效。本書大半是與計算機系統的硬體方面有關，但是在計算機結構上的軟體影響未曾略去。

積體電路 (*integrated circuit*，簡寫為 IC) 是構築的方塊，計算機的結構是以這種方塊為根據的。中型積體 (*medium scale integration*，簡寫 MSI) 元件供應數位的功用，大型積體 (*large scale integration*，簡寫 LSI) 元件供應整個計算機模組。這些元件所提供最重要的特性，即是它們被封入在很小的包裝內這一事實。以 MSI 與 LSI 功用之開始，計算機結構已採用一種新的重要性，讓設計工程師自由創作以前認為不切實際或不合經濟的結構。對於計算機設計師熟悉積體電路包裝內所常遭遇到的各種數位功用，這是很重要的。因為這原因，本書內介紹有三十多種不同的 MSI 與 LSI 功用，具有它們內部特性與外部特性的說明。

IC 數位功用間的互連最好以記錄器轉移語言作符號性的敘述。這樣的語言是在本書中詳述着，其關係於數位計算機的硬體組織與設計全予說明。記錄器轉移語言是用在許多場合，以簡明及正確的方式來列舉各種計算機的操作。

2 計算機系統結構

最基本的具備條件要包括主要實情者，是技術上與科學上的背景，以及學習數位計算機內部組織的興趣。數位系統的基本知識對瞭解所提出的一些硬體觀念是必需的。以機器語言或高階語言作程式計畫的一些經驗，將增進硬體觀念的了解。

本書的計畫是首先介紹較簡單的材料，然後引入較高深的主题。前六章包括對瞭解簡單數位計算機的組織，設計，及程式所需要的材料。後六章提出數位計算機的各個功用單位，着重於前半部中未包括的較深入的題目。

第一章介紹數位系統設計所需要的基本知識，此時這數位系統是用個別的閘及正反器組成。它包括布耳代數，組合電路，及順序電路。為使本書內容保持在合理的範圍內，在緒篇中限制這類課題的討論是必需的。就這方法而言，其理由是這樣的，當積體電路功用代替用作個別閘與正反器時，其數位計算機的設計就採用與眾不同的重要性。第一章中所包含的材料，對於瞭解正在提出的數位系統，供應些需要的背景。

第二章以列舉積體電路的一般特性開始。它詳細地論述一些最基本的 MSI 功用，例如記錄器，計數器，解碼器，多元調節器，隨意出入記憶器 (RAM)，僅讀記憶器 (ROM)。這些數位功用為以後各章中較大單位設計時用作建造式樣的方塊。最後一節含有二十種以上不同功用的 MSI 與 LSI 參考目錄，這些資料將在本書中別處論述。

第三章提出數位計算機中所有的各種資料型式，及說明它們如何地被表示成計算機記錄器中的二進位形式。着重於算術計算中所使用的數字表示，與符號的二進寫碼，例如資料處理中所用的字母文字，以及對特種應用所使用的其它各別符號。

第四章闡釋記錄器轉移語言，與說明它如何地被用在數位計算機的記錄器中來表示成微運算順序的符號形式。這些符號係對算術，邏

輯，及移位微運算作定義的，同樣地它亦給創始微運算的控制作定義的。提出配成大的長度來說明和各種不同符號及記錄器轉移敘述有關的硬體含意。

第五章介紹基本數位計算機的組織與設計。雖然在與商用計算機系統作比較時，基本計算機是簡單的，然而它包含足夠的作用能力，證明有已存程式及通用計算機的權力。記錄器轉移語言是用來定義計算機的內部運算及逐一指明對它設計的需求。

第六章係利用第五章中所作基本計算機的二十五種指令之定義，來說明寫計算機程式所常用的許多技巧。對許多初步的資料處理工作，則介紹用符號碼作程式範例。二進位程式，符號碼程式，與高階層語言程式三者間的關係用範例來解釋。這導至對翻譯程式諸如組合程式與編譯程式的需要。組合程式的基本操作係與別的系統程式一併介紹。本章的目的是介紹計算機軟體的基本概念，但不作詳細深入討論。軟體原理的知識與硬體表示聯為一起，可以給讀者包含硬體與軟體一種整個計算機系統的概論。

第七章討論數位計算機的中央處理單位 (central processor unit, 簡寫 CPU)。一種匯流排所組織的 CPU 提出介紹，及一種算術邏輯單位 (arithmetic logic unit, 簡寫 ALU) 作為分析。二進位與十進位的算術單位均予論及。各種指令碼格式則與定址技術一起說明。記憶體堆疊的組織以它的一些用途示教來作解釋。最後一節介紹微處理機，在基本上它是一種附在一個或數個 LSI 包裝內的 CPU。微處理機的發展已經革新了計算機的技術。雖然本書不是專門敘述微處理機的，可是它介紹對微處理機所需的基本觀念，給讀者有技術上的知識背景，以這些知識背景，使用廠商的參考手冊，就可以研究特種的微處理機。

第八章介紹微程式計劃的觀念。發展一特種的微程式控制單位，

4 計算機系統結構

以範例來說明典型的計算機指令組如何產生微式碼。本章對使用控制記憶器來發起控制作用，及決定微式指令地址的順序，討論各種不同方法。最後一節討論微程式計畫原理的優點與其應用。

第九章將專論算術處理機的設計。它介紹對正負號 - 大小表示法中定點型二進位加法，減法，乘法，及除法的演算。然後使用記錄器轉移語言來設計算術處理機。連結算術處理機至計算機的組態在第五章中已設計過。註譯二進位計算器定義，並用它作證明算術操作能以微程式來做的方法。這可與硬線控制製作及相同運算的軟體製作作比較。

第十章介紹其它基本的算術演算法。此演算法是對正負號 - 2 補數的二進位資料，浮點型資料，與十進位資料而發展的，同樣地亦用在十進制至二進制變換及二進制至十進制變換。演算法是以流程圖表示，並以記錄器轉移語言來說明微型運算的順序以及演算製作所需的控制決定。

第十一章解釋一些常用的輸入與輸出裝置之作用。處理機與 I/O 裝置間介面的需求予以解釋，對 I/O 控制轉移可用的各不同組態則予以逐一計算。使用特種範例對異步串列轉移與直接記憶器出入 (direct memory access, 簡寫 'DMA) 來說明介面的需要。其他所包括的論題為優先間斷，I/O 處理機，及資料通信處理機。

第十二章介紹記憶器層次分類的觀念，由主記憶器與磁盤，磁鼓，磁帶等輔助記憶器所組成。對記憶器匯流排組織的各不同組態予以說明。結合記憶器的內部組織與外部操作則予詳細解釋。記憶器管理的觀念則通過虛構記憶器系統的硬體需求表示法，予以介紹。

各章包含一羣問題與一系列參考書。有些問題可作各章中所含內容的習題，其他有較高深性質者，它們為解決有關數位計算機及系統主要範圍的問題，供應一些實際應用而編排。教師需要解答手冊可向出

版書商索取。

本書是適宜於作電機或計算機工程系的計算機設計課程。它可用作計算機科學系的計算機組織課程。本書亦適合於計算機工程師及科學家需要獲得積體電路功用的知識與計算機結構的基本以及高深觀念，自修之用。

我衷心對我妻表示我的最大感謝，她校訂全部手稿，提出所有建議以增進本書的可讀度，以及在全部編纂過程中的鼓勵與支持。

梅 諾 (M. Morris Mano)

目 次

序 言

第 一 章 數位邏輯電路	1
1-1 邏輯閘	1
1-2 布耳代數	6
1-3 圖示簡化	10
1-4 組合電路	17
1-5 正反器	24
1-6 順序電路	30
1-7 結論摘要	40
參考文獻	41
問題	41
第 二 章 積體電路與數位函數	45
2-1 數位積體電路	45
2-2 IC正反器與記錄器	53
2-3 解碼器與多元調節器	58
2-4 二進計數器	63
2-5 移位記錄器	69
2-6 隨意出入記憶器	73
2-7 僅讀記憶器	81
2-8 IC數位函數的參考表	85
參考文獻	86
問題	87

2 計算機系統結構

第 三 章 資料表示法	91
3-1 資料類型	91
3-2 定點表示法	100
3-3 浮點表示法	107
3-4 其他二進碼	109
3-5 偵錯碼	114
參考文獻	116
問題	116
第 四 章 記錄器轉移與微運算	121
4-1 記錄器轉移語言	121
4-2 內部記錄器轉移	122
4-3 算術微運算	133
4-4 邏輯微運算	139
4-5 移位微運算	150
4-6 控制函數	151
4-7 結論摘要	158
參考文獻	159
問題	160
第 五 章 基本計算機組織與設計	165
5-1 指令碼	165
5-2 計算機指令	169
5-3 定時和控制	174
5-4 指令的執行	180
5-5 輸入-輸出與中斷	188
5-6 計算機的設計	195
5-7 結論摘要	201

參考文獻	202
問題	203
第 六 章 計算機軟體	207
6-1 概論	207
6-2 程式語言	209
6-3 組合語言	213
6-4 組合程式	219
6-5 程式迴圈	226
6-6 程式算術運算	229
6-7 次常式	236
6-8 輸入-輸出程式計畫	242
6-9 系統軟體	249
參考文獻	258
問題	258
第 七 章 中央處理機組織	263
7-1 處理機匯流排組織	263
7-2 算術邏輯單位	267
7-3 傳遞延遲	274
7-4 十進算術單位	282
7-5 CPU 指令與格式	288
7-6 堆疊組織	297
7-7 LSI 微處理機	307
參考文獻	317
問題	318
第 八 章 微程式控制組織	325
8-1 控制記憶器	325

4	計算機系統結構	
8-2	位址順序	327
8-3	微程式事例	334
8-4	控制字格式	342
8-5	定時考慮	349
8-6	軟體輔助	351
8-7	優點與應用	354
	參考文獻	358
	問題	359
第 九 章	算術處理機設計	362
9-1	概論	362
9-2	不帶符號二進數字的比較與減法	363
9-3	加法與減法的演算法	369
9-4	乘法演算法	373
9-5	除法演算法	376
9-6	處理機組態	382
9-7	控制的設計	387
9-8	微程式計算器	390
	參考文獻	400
	問題	401
第 十 章	算術演算法	405
10-1	概論	405
10-2	加法溢位	407
10-3	乘法與除法	410
10-4	浮點算術運算	415
10-5	十進算術運算	427
10-6	十進數-二進數變換	434

10-7	結論摘要.....	441
	問題.....	442
第十一章	輸入-輸出組織	447
11-1	外圍裝置.....	447
11-2	I/O 介面.....	451
11-3	異步串列介面.....	459
11-4	直接記憶器出入.....	465
11-5	優先中斷.....	468
11-6	外圍處理機.....	479
11-7	資料通信處理機.....	485
	參考文獻.....	492
	問題.....	493
第十二章	記憶器組織.....	496
12-1	記憶器的階層.....	496
12-2	輔助記憶器.....	499
12-3	記憶器匯流排組織.....	503
12-4	結合記憶器.....	506
12-5	虛構記憶器.....	513
12-6	結論摘要.....	521
	參考文獻.....	522
	問題.....	523
索 引.....		525

第一章 數位邏輯電路

1-1 邏輯閘

顧名思義，數位計算機是一種執行各式各樣計算工作的數位系統。數位 (*digital*) 一字意含計算機內變數所代表的資訊，是採用個別或分立值的有限數目。這些數值由組件在內部予以處理，那就能保持個別狀態的有限數目。例如，十進數位 0, 1, 2, ..., 9 供給十個個別的值，實際上，倘使僅用二個狀態，數位計算機工作起來較為可靠。因為組件的實體限制，又因人類邏輯趨向於二元制（即真或偽，是或非敘述），被限制於採用個別值的數位部分，再受限制僅採用二值，而稱為二進位 (*binary*)。

數位計算機使用二進數字系統，這系統有二個數位，即 0 與 1。一個二進數位稱為一數元 (*bit*)。在數位計算機中的資訊是用數元羣來表示的。依照使用各種編號的技巧，數元羣不僅可被用來表示二進數字，亦可代表任何其他的分立符號，例如十進數位或字母的文字。審慎使用二進排列法及應用各種碼技巧，二進數位或數元羣可以被用來發展成完整的指令集合，作為執行不同類型的計算。

將使用基數以 10 作系統的常用十進數對比，二進數是使用基數為 2 的系統的。例如二進數 101101 代表一數量，將每一數元乘以相關的 2 之指數幂，它可轉換成十進數如下：

$$1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 45$$

六個數元 101101 代表的一個二進數，它的十進等效數為 45。可是，一種六個數元羣亦可代表作為字母的文字二進碼，或在特種數位系統中，作為指定某些決定邏輯的控制碼。換句話說，數位計算機中的數

2 計算機系統結構

元羣可被用來代表許多不同的東西。這情形類似於字母的相同文字可用來組成不同的語言，諸如英文與法文等這種觀念。

在數位系統中，二進資訊由實在數量所表示者稱為信號(*signal*)。在全部數位系統中，電的信號諸如電壓在二個可認出的值中一定任有一個，而代表二進變數等於 1 或 0。例如，某一特種數位系統可用 3 伏信號代表二進制 1，而 0.5 伏為二進制 0。如圖 1-1 所示，每個二進值有離正規可接受的偏差量。兩個允許區域間的中間部分僅在狀態轉變期間才被跨越。數位電路的輸入端接受二進制信號，不出其允許的容許度，電路在輸出端隨同着二進制信號，在指定的容許度範圍內響應着。

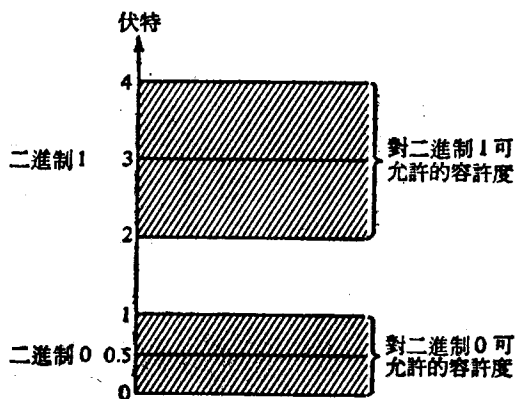
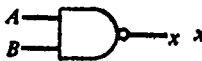


圖 1-1 二進制信號的事例

二進邏輯 (*binary logic*) 討論二進變數，以及具有邏輯意義的運算。以代數或表格形式，用來敘述二進資訊的操作與處理。二進資訊的操作以邏輯電路來工作，這邏輯電路稱為閘(*gate*)。閘為硬體的組件，當其輸入邏輯需求被滿足時，它們產生二進制 1 或 0 的信號。各種邏輯閘 (*logic gate*) 通常在數位計算機中被找得到的。每種閘有

不同的圖形符號，而它的運算係以代數函數來說明的。對每種閘而言，其二進變數的輸入-輸出關係可用真理表 (truth table) 中的表格形式來表示。

八種邏輯閘的名稱，圖形符號，代數函數，及真理表如圖 1-2 所示。每種閘有一個或二個以 A 與 B 標明的二進輸入變數，及一個以 x 標明的二進輸出變數。及閘 (AND gate，為運算明顯起見，本書以後仍以 AND 代及) 產生 AND 邏輯函數，即是當輸入 A 與輸入 B

名稱	圖形符號	代數函數	真理表														
及 (AND)	 $x = A \cdot B$ 或 $x = AB$	<table border="1"> <tr> <th>A</th> <th>B</th> <th>x</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	A	B	x	0	0	0	0	1	0	1	0	0	1	1	1
A	B	x															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
或 (OR)	 $x = A + B$	<table border="1"> <tr> <th>A</th> <th>B</th> <th>x</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	A	B	x	0	0	0	0	1	1	1	0	1	1	1	1
A	B	x															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
反 (inverter)	 $x = A^c$	<table border="1"> <tr> <th>A</th> <th>x</th> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	A	x	0	1	1	0									
A	x																
0	1																
1	0																
緩衝器 (buffer)	 $x = A$	<table border="1"> <tr> <th>A</th> <th>x</th> </tr> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </table>	A	x	0	0	1	1									
A	x																
0	0																
1	1																
反及 (NAND)	 $x = (AB)^c$	<table border="1"> <tr> <th>A</th> <th>B</th> <th>x</th> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	A	B	x	0	0	1	0	1	1	1	0	1	1	1	0
A	B	x															
0	0	1															
0	1	1															
1	0	1															
1	1	0															

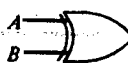
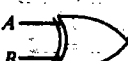
名稱	圖形符號	代數函數	真理表															
反或 (NOR)		$x = (A + B)'$	<table> <tr> <th>A</th> <th>B</th> <th>x</th> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	A	B	x	0	0	1	0	1	0	1	0	0	1	1	0
A	B	x																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
互斥或 (exclusive-OR)		$x = A \oplus B$ or $x = A'B + AB'$	<table> <tr> <th>A</th> <th>B</th> <th>x</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	A	B	x	0	0	0	0	1	1	1	0	1	1	1	0
A	B	x																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
互斥反或 (exclusive-NOR) 或互合 (equivalence)		$x = A \odot B$ or $x = A'B' + AB$	<table> <tr> <th>A</th> <th>B</th> <th>x</th> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	A	B	x	0	0	1	0	1	0	1	0	0	1	1	1
A	B	x																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

圖 1-2 數位邏輯閘

兩者同時為二進制 1 時，其輸出為 1；否則，其輸出為 0。這些條件亦在 AND 閘真理表中載明。此表指出僅當輸入 A 與輸入 B 為 1 時，其輸出 x 為 1。AND 函數的代數運算符號是像通常代數的乘法符號一樣。我們可任用變數間一點或者連結變數在一起沒有它們間的運算符號。AND 閘可有比兩個較多的輸入，此時，依照其定義，若且惟若所有輸入為 1 時，其輸出為 1。

或閘 (OR gate，為運算明顯起見，本書以後仍以 OR 代或) 產生本含或 (inclusive-OR) 函數，即若輸入 A，或輸入 B，或兩者為 1 時，其輸出為 1；否則，其輸出為 0。OR 函數的代數符號為 +，相類於算術的加法。OR 閘可有比兩個較多的輸入，此時，依照其定義，若任何輸入為 1 時，其輸出為 1。