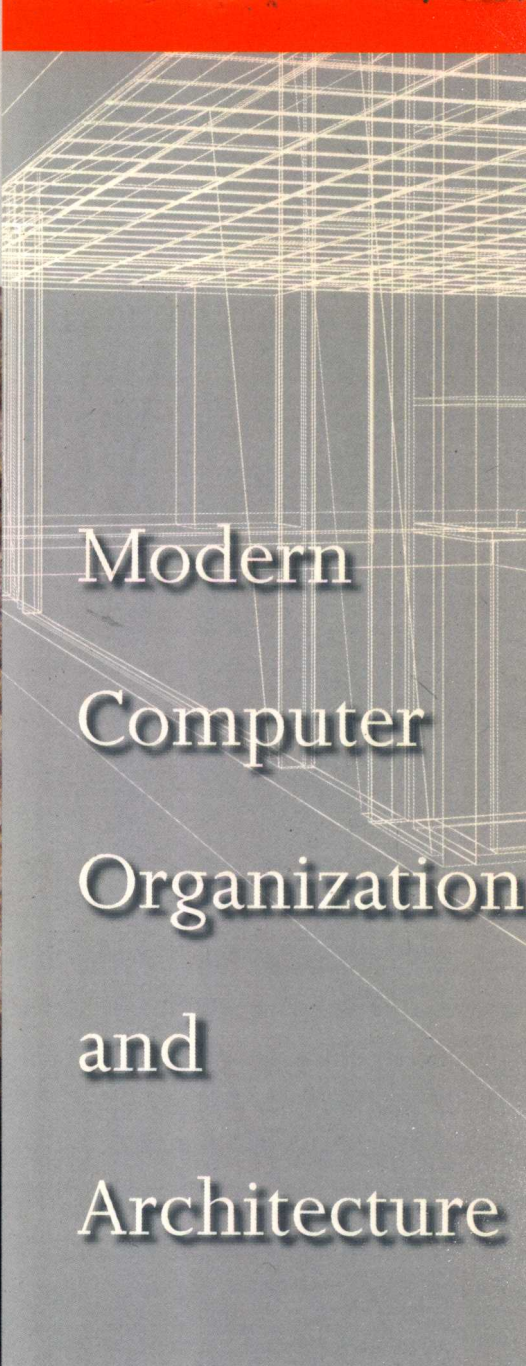


Second ² Edition



Modern
Computer
Organization
and
Architecture

最新

計算機組織與結構

解名仁◎編著

TP303
20106

港台書

Second Edition
ORGANIZATION



Modern
Computer
Organization
and
Architecture



最新

計算機組織與結構

解名仁◎編著

國家圖書館出版品預行編目資料

最新計算機組織與結構 / 解名仁編著. --
二版. -- 臺北縣中和市 : 新文京開發, 2008
(民 97.04)
面 ; 公分

ISBN 978-986-150-852-8(平裝)

1. 電腦結構

312.122

97005618

免無
費作
贈者
閱章



最新計算機組織與結構

(書號 : D052e2)

編著者	解名仁
發行者	新文京開發出版股份有限公司
地址	台北縣中和市中山路二段 362 號 8 樓 (9 樓)
電話	(02) 2244-8188 (代表號)
F A X	(02) 2244-8189
郵撥	1958730-2
初版	西元 2005 年 9 月 15 日
二版	西元 2008 年 6 月 15 日

有著作權 不准翻印

建議售價 : 600 元

法律顧問 : 蕭雄淋律師

ISBN : 978-986-150-852-8

MODERN COMPUTER
ORGANIZATION
AND ARCHITECTURE

序
再版

本書出版方兩年半，承蒙諸位先進前輩的厚愛，全國已超過二十多所專科、學院與大學選用為教課書，而書坊門市每年平均也有六十多本零售，個人深感惶恐，也為這樣的課程，學界能用中文書做教材，心生敬然。

由於初版倉促，總有些字誤、說明不清或內容艱澀之處，多方賜教，適於此再版時做校正、補述，並以感謝先進前輩們的美意。然而值此多核心系統、多處理器控制、平行處理與嵌入式系統設計與應用等科技發展蓬勃的今天，有許多電腦硬體與系統的執行概念，確有其必要的論述與探討，故文中難免有其深入複雜之處，在教學說明上有所不易；但本課程即屬高階，就不應去難談易，這是個人在編撰本書時，最是再三推敲之處，教課書應以詳盡周全為主，以待學者之用心，有些問題實不便於簡化，謹望諒宥。

感謝新文京開發出版公司的支持，洪成賀經理的熱心推動與奔波，使本書逕於二版，編輯部的詹鎧欣與美工的吳小玉小姐，認真辛苦的協助，讓書煥然一新，特予致謝。

從事教學多年，學海無崖，總是愈教愈覺自己多有不足之處，還望先進前輩們不吝賜教，使個人多有所長。也希望學界能深體計算機組織與結構課程，對目前科技發展的重要性，積極推動以期將來。

解名仁 敬致

97/4/3 於永和

序

MODERN COMPUTER ORGANIZATION AND ARCHITECTURE

教結構的課十餘年了，為了要表達其中的內部電路與設計原理，每天都在黑板上畫圖，歲月就在這各色的粉筆灰中飄逝而去。計算機組織與結構已是高階的課程了，它的內容既抽象又龐雜；涉及電學、邏輯、微算機、介面，以及程式、資料結構、系統等基礎，這個課是不容易上的。長年來愈教問題愈多，每一年都有新的變化；而最讓我頭痛的是教學的表達，「如何用最簡單的話語，或一個示意圖來說明，這個觀念或技術！」，精確的描述所要表達的事物，是做老師的基本功。而面對課程時數有限、學生程度有限、教材又多又廣的情況下；要將內容說明得詳細又要完整，真是教學上的挑戰。

思考著用一幅什麼樣的圖？來敘述課堂上的觀念，這已是我生活上的習慣；隨身帶著扎記本，偶有會意就隨筆畫下。但是實際動筆整理這些講義，已是三年多前的事了；每天寫著一些，想著一些，慢慢的彙整至今總算是完成了；因為篇幅過多，所以控制單元的微程式設計就放棄了，有些技術的討論也都簡化了，希望再版時能添加。

這個領域的前輩很多，個人出書其實非常惶恐。僅是想在教學上有一套適合的教材，新的觀念真是一日千里，教書愈久愈覺得自己學淺。但是總希望能為這個課付出一點心力，其實從 AMD64、雙核心 CPU、雙管道記憶體、南北橋晶片、SIMD 架構、L2 Cache、VLIW 等等，這些名詞常出現在報張雜誌時，這個課已經真實的在我們生活中了。但卻只有很少數的人上過這個課，而且很多本行的學生連十年前的管線都不清楚，很顯然這個課被忽視了，不是沒有編排這個課而是內容不務實。每個人對科技的觀念可能有錯，可能過時了，但這一行就是要不斷的面臨新知，並傳授新知，才能真正帶動國家的科技的成長；個人常秉持著求新務實於教學，但若有謬誤之處，還請不吝指正賜教。

寫這本書的過程是充實的，驀然回首真懷念那沉浸在研究撰述的時光。一生獲得別人照顧的多，付出的鮮少；而這本書也是個人些許的奉獻，在本行計算機結構的領域，也算有個註腳；至少這本書可以使得上課時，更輕鬆而貼近現代的電腦技術，能多教一些內容，使得學生們徜徉在這精妙的電腦世界，而獲得啟示與快樂。

感謝長年來，許多同事給我的鼓勵，以及學術上的寶貴意見；有時連學生都給我很多支持。在此特別要感謝我的指導教授李良德老師，對我的包容，及給我的啟蒙與教誨，他的教導對我的一生，都有很大的影響；感謝文京出版機構的協助，徐秀菁小姐的奔波幫忙，與李毓菁小姐的細心編輯，使本書順利付梓。本書謹獻給所愛的內人鄭品秀女士，以及小女怡蕙與怡凡，有個溫暖幸福的家，才能有好的發展；我們共享這份喜悅。

計算機組織與結構，已經是很生活化的課了，希望同好與前輩們能將內容以淺顯的方式，讓它普及到社會上；至少讓大家都能看得懂報紙及雜誌的介紹；讓書本裡的知識，能在生活中用上，學習才有興趣。

解名仁 謹識

94年5月27日

廿一年婚紀于深坑東南

第一章 計算機系統的主要架構 1

第一節 輸入與輸出系統 3

第二節 處理系統 4

第三節 儲存系統 5

第四節 程式的編輯與執行 7

第五節 指令的執行 10

習題研討 16

第二章 儲存系統 19

第一節 快取記憶體與主記憶體之間的關係 21

一、快取讀中(Cache Hit) 22

二、快取錯失(Cache Miss) 23

三、寫回錯失(Write Miss) 24

第二節 主記憶體(Main Memory) 26

一、唯讀記憶體(Read Only Memory, ROM) 27

二、隨機存取記憶體(Random Access Memory, RAM) 30

三、記憶體的資料讀寫動作(Memory Read and Write) 33

四、記憶體讀寫控制的時序 34

五、刷新週期(Refresh Cycle) 35

第三節 其他類型的記憶體與其技術 35

一、共用位址與資料匯流排 36

二、記憶體的擴充 37

三、交錯式記憶體(Cross Memory) 41

四、長字組記憶體(Long Word Memory) 44

五、區塊存取記憶體(Block Access Memory) 44

六、封包存取記憶體 47

七、比對式記憶體 48

八、記憶體庫(Memory Bank) 48

第四節 快取記憶體(Cache Memory) 53

一、各種形式的Cache資料存取設計 54

二、多層Cache的設計 59

第五節 Cache與主記憶體的資料更改 61

第六節 外部儲存體的資料使用 64

一、記憶體管理程式的分頁技術 64

二、實際位址與虛擬位址的存取方式 68

三、對照表 73

四、資料查詢的流程 75

第七節 記憶體資料的除錯 76

習題研討 85

第三章 輸出入裝置的控制 87

第一節 資料傳輸 91

一、資料傳輸線路(Data Transfer Lines) 91

二、傳輸控制線路(Transfer Control Lines) 93

三、同步脈波線路(Synchronous Clock Line) 96

四、資料傳送速率(Data Transfer Rate) 100

五、資料格式(Data Format) 101

六、單一線路傳送(Single Line Transfer) 102

七、無線傳送(Lineless Transfer) 103

第二節 裝置的介面電路(Interface Circuit) 108

一、CPU要讀取裝置的資料 109

二、CPU要寫出資料存入裝置 109

三、CPU對主記憶體的控制 109

四、I/O指令與I/O命令 110

五、狀態表示 110

六、	介面與裝置間的連接	111
七、	多個I/O裝置的連接	111
第三節	CPU與輸出入裝置的資料傳送	112
一、	分時(Time Sharing)作業系統	112
二、	介面電路控制式的資料傳送方式	114
第四節	CPU如何處理同時發生的多個中斷要求	125
一、	中斷訊息的優先順序編碼	125
二、	主動服務的輪詢式(Polling)	127
三、	菊輪式(Daisy Chain)的裝置要求之詢問服務	128
四、	裝置競爭式的中斷服務要求	133
第五節	CPU與輸出入裝置、記憶體的连接方式	139
一、	記憶體圖示式(Memory Mapping)裝置連接	140
二、	分離式(Isolation)裝置連接	141
三、	輸出入處理器(I/O Processor)	143
四、	功能處理器(Function Processor)	143
第六節	橋接器(Bridge)	146
第七節	網路介面(Network Interface)	150
第八節	錯誤檢查碼(Error Check Code, ECC)	153
一、	同位元核對(Parity Check)	153
二、	餘數核對(Remainder Check)	155
三、	循環運算核對(Cyclic Operating Check)	155
習題研討		158

第四章 中央處理單元的運算系統 161

第一節	暫存器的設計	163
第二節	暫存器的資料傳送	171
第三節	算術邏輯運算單元	179

- 一、算術運算電路 179
- 二、邏輯運算電路 194
- 三、移位電路 197
- 四、比較電路 201
- 五、狀態旗號 205
- 六、數值的擴展 206

第四節 實數運算處理 208

- 一、浮點格式 210
- 二、程式中數值的格式轉換 213
- 三、浮點格式的加減法計算 223
- 四、浮點格式的乘除法計算 226
- 五、浮點格式轉換回十進制值 226

第五節 運算電路的整合 229

習題研討 233

第五章 指令的執行系統 235

第一節 硬體架構 236

- 一、計算暫存器 237
- 二、特殊暫存器 238
- 三、狀態位元 239
- 四、ALU電路 239
- 五、記憶體規格 239
- 六、輸出入資料 240
- 七、中斷處理 240

第二節 記憶體規劃 241

第三節 計算電路規劃 246

第四節 指令規劃 247

一、	暫存器計算指令	248
二、	資料搬移指令	249
三、	跳躍指令	250
第五節	硬體電路的控制單元設計	251
一、	電腦的控制頻率	251
二、	外部裝置的不同設計概念	253
三、	外部裝置的大量資料傳遞	256
四、	控制單元的主要電路	257
五、	系統運作的指令週期	259
六、	讀取與解碼指令的階段	260
七、	計算指令執行的階段	262
八、	計算指令的控制電路設計	265
九、	資料搬移指令執行的階段	268
十、	資料搬移指令的控制電路設計	273
十一、	跳躍指令的執行階段	280
十二、	副程式的呼叫	282
十三、	跳躍指令的控制電路設計	285
第六節	中斷處理	286
一、	沒有中斷要求的指令執行	286
二、	中斷的程序	287
三、	中斷所需要的控制電路	290
四、	狀態暫存器	291
五、	暫存器與狀態的保存	291
六、	中斷的指令執行	293
第七節	CPU內部電路設計探討	294
一、	CLR指令的電路設計	294
二、	計算電路的執行時間各有不同	295
三、	計算資料的同時讀入	297
四、	不同的計算指令可分開同時完成	299

	五、段暫存器的值可以改變	299
	六、指令的精簡化	300
	七、呼叫副程式與中斷的暫存器規劃	301
	八、多個中斷要求產生要如何處理	302
第八節	CPU的效能評估	303
第九節	高階語言的執行	308
	一、編譯高階語言為低階語言	309
	二、組譯低階語言為機械碼	313
	三、指令碼的執行	316
第十節	精簡與複雜指令集電腦	318
	一、指令的種類	319
	二、指令的格式	321
	三、指令中運算元的表示	322
習題研討		325

第六章 管線處理 327

第一節	何謂管線處理	328
第二節	指令的分割	334
第三節	CPU內單一管線設計	336
	一、管線執行技術	337
	二、電路障礙(Circuit Hazard)	339
	三、跳躍障礙(Jump Hazard)	344
	四、資料障礙(Data Hazard)	359
第四節	管線的中斷處理	365
	一、指令中斷的管線處理技術	365
	二、系統中斷的管線處理技術	366
	三、外部中斷的管線處理技術	368

第五節 多條管線的設計 370

- 一、指令的非依序(Out of Order)執行 371
- 二、紀錄板(Scoreboard)技術 374
- 三、預留站(Reservation Station)技術 377
- 四、個人電腦的CPU設計 382
- 五、超級管線的中斷處理 383
- 六、超長指令字(Very Long Instruction Word) 384

習題研討 387

第七章 多處理器系統 389

第一節 單一指令單一筆資料(SISD)的電腦架構 393

第二節 單一指令多筆資料(SIMD)的電腦架構 393

- 一、向量處理(Vector Process) 402
- 二、多輸入的運算電路 406
- 三、浪潮陣列處理器(Wave Front Array Processor) 410
- 四、心臟收縮陣列處理器 (Systolic Array Processor) 413

第三節 多個指令單一資料(MISD)的電腦架構 416

- 一、控制流與資料流 418
- 二、資料流的圖形語言 419
- 三、資料流電腦的特色 423

第四節 多個指令多筆資料(MIMD)的電腦架構 429

- 一、多處理器系統 431
- 二、多處理器系統的啟動與執行 435
- 三、多處理器系統的資料管理 441
- 四、多處理器間的連接方式 451
- 五、多核心的CPU時代 458

習題研討 461

第一章

計算機系統的主要架構

電腦資訊已普及到每週報紙的專版及大量的雜誌書籍，其相關的最新產品及技術，廣泛的被介紹與討論，可是國內對這些知識的瞭解與汲取能力是不足的；簡易的計算機概論已遠不能說明目前 CPU、記憶體及相關輸出入裝置；在知識經濟的現代社會中，電腦的基礎教育面要再延伸擴大，最新的科技觀念需要更積極而全面的傳遞開來。

計算機硬體架構，一般為輸入(Input)、儲存(Storage)、處理(Processing)與輸出(Output)四個部份。而系統的流程，是將：

- 1. 指令或資料，透過鍵盤、滑鼠、磁碟等輸入裝置，用來書寫、編輯或下達命令，或是載入檔案。
- 2. 實際的指令（程式）與資料會載入至儲存裝置，如記憶體；使用者在其內部編輯、設定後，再下達執行命令。
- 3. 這些指令或資料會逐一進入處理，即在中央處理單元內做運算，再將結果在寫回儲存裝置。
- 4. 如有需要再將結果，存入磁碟、列表機等輸出裝置。

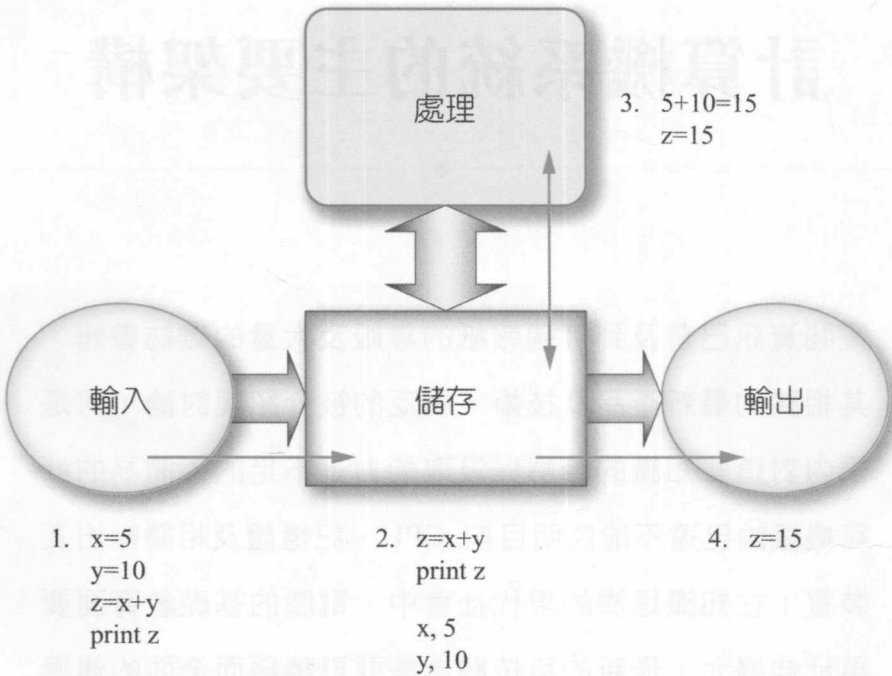


圖 1-1

計算機的硬體結構圖。我們使用輸入裝置，做載入或編輯程式；整個程式放置於儲存系統中修改。然後要求處理系統加以讀取與執行，最後將結果輸出。

在設計觀念裏，如果只討論計算機的硬體；它的輸入、儲存、處理、輸出裝置的設計，一般我們稱為是計算機組織(Computer Organization)，它的領域主要是討論硬體設計觀念。如果同時探討系統與指令的執行，那就稱為計算機結構(Computer Architecture)，它除了討論硬體的設計外，並同時要考慮系統的運作與指令的執行。

早期的自動控制機器並沒有儲存系統，所有自動化的工作必需用輸入系統，直接操縱處理裝置，經由控制然後產生輸出。由於沒有儲存系統，相同的工作仍然必須重新的，透過輸入系統控制，進行指揮、調度，所以儘管是相同的工作，每一次的輸入控制都必須重新做過，而不得有誤。若是換個新的工作程序，就必須把整個輸入的操作，全面更改，非常麻煩。所以當電腦自動化時，儲存系統的設立是非常重要的；它不僅提供一個可以反覆的去做相同工作的能力，並提供了一個可以編輯、修改或開啟新的控制程式的能力。這樣大大的提供了自動化的控制與處理效能，同時也具備了基本電腦的雛型。

第一節 輸入與輸出系統

輸出入系統包含著，輸出入裝置(Input/Output Devices)與各部份連接的匯流排(Bus)，以及配合信號轉換與傳送速率調整的介面裝置(Interface Devices)。

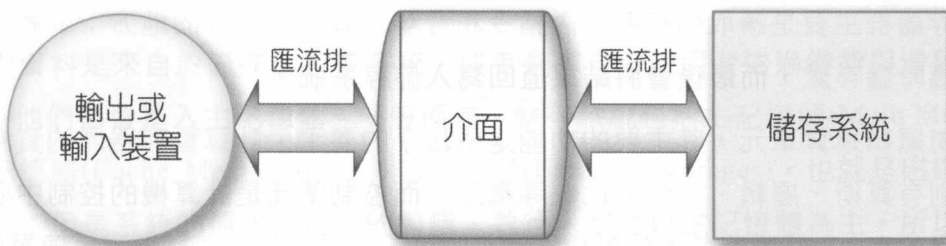


圖 1-2

輸出入系統與儲存系統連結圖。一般輸入裝置有滑鼠和鍵盤，而輸出裝置有螢幕和列表機等等，也有一些像磁碟、光磁或磁帶；它們既是輸入裝置也是輸出裝置。而這些產品在設計與計算機連線時，廠商必須為它們設計其介面與驅動程式，以供硬體與系統的切實連結。

一般而言，輸出入裝置多為機械或電磁控制信號與計算機的二進制碼，是迥然不同的。所以在資料或控制狀態轉換與傳遞時，需要介面裝置，一方面針對儲存系統與輸出入裝置的資料格式做轉換；另一方面緩衝雙方速度上的差異。

而匯流排是短距離的並列傳輸(Parallel Transfer)設計，若輸出入裝置屬遠距設備，則與介面的連接應為串列式單線傳送(Series Transfer)。然而資料在傳送時都必須是同步的，所以在設計上必須考量傳送速率、資料格式等問題。

第二節 處理系統

計算機的處理樞紐是中央處理單元(Central Processing Unit, CPU)，而在硬體電路上，它本身是一個電路晶片(Chip)，又稱為處理器(Processor)，但是處理器除了中央處理單元外，有的內含快取記憶體(Cache Memory)；其負責指令的運算與執行，所有的指令與資料在儲存系統內。程式執行時，處理器逐一讀入指令，並予以解碼，瞭解指令所要求的命令動作後，切實去控制執行。

中央處理單元有三個主要部分，第一是暫存器群(Registers)，其次是算術與邏輯運算單元(Arithmetic and Logic Operation Unit, ALU)，以及控制單元(Control Unit, C.U)。

暫存器群主要是讀取儲存系統內指令和資料，做臨時儲存的地方；或者為運算結果的臨時儲存處，而最後會將結果值回寫入儲存系統。

算術邏輯運算單元，其主要的功能是依指令的要求，計算暫存器內的資料；其內部分別有算術、邏輯、移位與比較等電路。而控制單元是計算機的控制中心，它的主要目的有：1.使所有裝置的動作同步，2.完成指令的整個執行要求。簡單的說，中央處理單元為了要執行每個指令的動作，所以必須控制儲存系統與輸出入系統，其資料的傳送與讀取，並切實的執行每個指令的運算。