

GOTOP

計算機結構(下冊)

計量接近

COMPUTER ARCHITECTURE
A
QUANTITATIVE APPROACH



JOHN L HENNESSY
&
DAVID A PATTERSON

賴飛熊 譯

正式獲得 Morgan Kaufmann 出版社授權中文地區獨家發行

計算機結構

下冊

計量接近

JOHN L HENNESSY & DAVID A PATTERSON

賴飛鵬 譯



碁峯資訊股份有限公司 印行

碁峯資訊股份有限公司已聘任本律師為常年
法律顧問，如有侵害其著作權、信用名譽權
利及其他一切法益者，本律師當依法保障之



國立中央圖書館出版品預行編目資料

林 天 平 律師

計算機結構：計 接近／John L, Hennessy
David A, Patterson著；賴飛熊譯，--
初版，--臺北市；碁峯資訊，1991

【民80】 冊； 公分

譯自：Computer architecture

參考書目：面

含索引

ISBN 957-641-087-8 (一套：平裝)

1. 電腦－設計

471.52

80001072

ACE002

編 著 賴飛熊譯
發行人 廖文良
發行所 碁峯資訊股份有限公司
台北市南港路三段50巷20號2樓
電話／(02)788-2408
傳真／(02)788-1031
印刷所 建發彩色印刷有限公司
台北縣中和市中山路二段366巷13號
版 次 1991年7月初版
劃 撥 帳戶／碁峯資訊股份有限公司
帳號／14244383
定 價 NT\$320 HK\$ US\$

版權所有 翻印必究



出版登記證：局版台業字第4869號

目次

第七章 向量處理器

7.1 為什麼要向量機器？	7-2
7.2 基本的向量結構	7-5
7.3 兩個真實世界的主題：向量長度及步距	7-24
7.4 向量處理效能的簡單模型	7-32
7.5 向量機器的編譯器技術	7-36
7.6 提高向量處理的效能	7-44
7.7 綜合：評估向量處理器的效能	7-53
7.8 謬誤與陷阱	7-64
7.9 結論	7-67
7.10 歷史展望及參考文獻	7-68
習題	7-74

第八章 記憶體——階層設計

8.1 簡介：區域性定律	8-2
8.2 記憶體階層的一般性定律	8-3
8.3 快取記憶體	8-9

8.4	主記憶體	8-38
8.5	虛擬記憶體	8-49
8.6	虛擬記憶體的保護及範例	8-59
8.7	幾個基於程式動作的最佳化	8-75
8.8	高等主題——改善快取記憶體的效能	8-83
8.9	綜合：VAX-11/780 的記憶體階層	8-118
8.10	謬誤與陷阱	8-128
8.11	結論	8-133
8.12	歷史展望與參考文獻	8-135
	習題	8-143

第九章 輸入 / 輸出

9.1	簡介	9-2
9.2	預測系統效能	9-5
9.3	I/O 效能的測量	9-16
9.4	I/O 裝置的種類	9-26
9.5	匯排流 —— 連接 I/O 裝置與 CPU/ 記憶體	9-54
9.6	到 CPU 的介面	9-62
9.7	到作業系統的介面	9-67
9.8	設計一個 I/O 系統	9-73
9.9	綜合——IBM 3990 儲存子系統	9-86

9.10 謬誤與陷阱	9-98
9.11 結論	9-105
9.12 歷史展望和參考文獻	9-106
習題	9-111

第十章 未來的方向

10-1 簡介	10-2
10-2 電腦的 Flynn 式分類	10-2
10-3 SIMD 電腦——單一指令流、多重資料流	10-3
10-4 SIMD 電腦——多重指令流、多重資料流	10-6
10-5 通往黃金國之路	10-10
10-6 特別用途的處理器	10-15
10-7 編譯器未來的方向	10-17
10-8 綜合：Sequent Symmetry 多重處理機	10-19
10-9 謬誤與陷阱	10-25
10-10 結論——計算機結構的演進和革命	10-27
10-11 歷史展望和參考文獻	10-29
習題	10-35

附錄 A 計算機算術

A.1	簡介	A-2
A.2	整數算術的基本技術	A-3
A.3	浮點數	A-18
A.4	浮點數的加法	A-24
A.5	浮點數的乘法	A-31
A.6	除法與餘數	A-35
A.7	準確度與例外處理	A-43
A.8	整數加法之增速	A-48
A.9	整數乘法與除法之增速	A-59
A.10	綜合	A-78
A.11	謬誤與陷阱	A-84
A.12	歷史展望及參考文獻	A-85
	習題	A-91

附錄 B 完整指令集表

B.1	VAX 使用者指令集	B-2
B.2	System / 360 指令集	B-7
B.3	8086 指令集	B-11

附錄 C 詳細指令集測量

C.1 VAX 詳細測量	C-2
C.2 360 詳細測量	C-4
C.3 Intel 詳細測量	C-5
C.4 DLX 詳細指令集測量	C-6

附錄 D 時間測量對頻率測量

D.1 在 VAX-11/780 的時間分布	D-2
D.2 在 IBM 370/168 的時間分布	D-4
D.3 在 IMV PC 8086 的時間分布	D-7
D.4 在 DAX 親戚的時間分布	D-9

附錄 E RISC 結構的鳥瞰

E.1 簡介	E-2
E.2 定址模式和指令格式	E-3
E.3 指令：DLX 子集	E-7
E.4 指令：DLX 的一般性擴充	E-14

E.5 MIPS 獨特的指令	E-19
E.6 SPARC 獨特的指令	E-23
E.7 M88000 獨特的指令	E-27
E.8 i860 獨特的指令	E-30
E.9 結論	E-36
E.10 參考文獻	E-38

我當然不是在發明向量機器。今天的世界上，就我所知有三種向量機器。他們各自以 Illiac - IV，(CDC) Star machine，以及 TI (ASC) machine 為代表。這三種都是先進的機器而作為一部先驅電腦常有的問題就是犯錯誤。而我卻不會，也永遠不想成為先驅人士。往往當你看清楚先驅人士們所犯的錯誤的時候，你就能在後繼者的地位上創造最好的。

Seymour Cray, Public Lecture at Lawrence Laboratories

CRAY-I 概論 (1976)

第七章

向量處理器

- 7.1 為什麼要向量機器？
- 7.2 基本的向量結構
- 7.3 兩個真實世界的主題：向量長度及步距
- 7.4 向量處理效能的簡單模型
- 7.5 向量機器的編譯器技術
- 7.6 提高向量處理的效能
- 7.7 綜合：評估向量處理器的效能
- 7.8 謬誤與陷阱
- 7.9 結論
- 7.10 歷史展望及參考文獻

習題

7.1 爲什麼要向量機器？

前一章我們詳細地研究管線處理以及觀察管線排程，在一個時計週期發出多個指令，還有更深的管線處理，可以使一個處理器產生兩倍的效能。然而管線在效能上的改進仍有一些限制。這些限制由兩個主要因素所造成：

- 時計週期時間—增加管線的深度可以減少時計週期間，然而比較深的管線會增加管線內資料的相依性而造成了較高的 CPI。從某些觀點上來看，管線深度的任何增加都會產生相對的 CPI 增加。正如 6.10 節中我們所看到的，一個很深的管線會使一個處理機變
- 指令擷取及解碼的速率—這個限制，有時又叫做 Flynn 瓶頸（根據 Flynn [1966]），阻止每一個時計週期內擷取及產生好幾個指令。我們所看到的管線式機器平均每個時計週期所發出的指令數目都比一少。

這兩個因為比較深的管線及發出多指令所生的限制可以從時計速率和 CPI 的觀點來看：去替一個 n 倍深的管線做排程的工作就和去替一個能在每個時計週期發出 n 個指令的機器做排程一樣的困難。

高速，而有管線的機器在大量的科學和工程應用上是特別有用的。高速管線式電腦通常會使用一個快取記憶體來防止記憶體存取指令產生太長的延遲時間。然而，大型的、執行起來很久的、科學性的程式往往有一個相當大而且活躍的資料集，這些資料集存取時往往只有很低的區域性，這也使得記憶體階層造成了不良的執行效能。結果就是造成快取記憶體效能的減退。假如我們能夠決定記憶體存取的型式以及在管線中有效地處理這些存取，這個問題可能可以克服。將來可能編譯技術可以幫助我們定位這個問題。

向量機器提供以向量（數字的線性陣列）方式處理的高階運算。典型的向量運算就像把兩個 64 項的浮點向量相加而得到一個 64 項的向量結果。向量指令相當於一個迴圈每次反覆運算 64 個元素之一的結果，更新指標，跳回開始的地方。

向量運算有解決上述大多數問題的重要性質：

- 每一個結果的計算和前一個結果的計算無關，這樣子允許一個很深的管線而不產生資料危障。基本上，避免資料危障是由編譯器和使用向量指令的程式設計師決定。
- 每一個向量指令決定相當多的工作——它和執行整個迴圈的能力相當。因此，指令頻寬需求降低了，而 Flynn 瓶頸也獲得緩和。

- 存取記憶體的向量指令有一個已知的存取型式。假如向量的元素都相鄰，那麼從一組高度交錯的記憶體儲藏處擷取一個向量的工作將會很有效率。對存取快取記憶體來說算是延遲很久的主記憶體存取啓始時間可以改善，因為一次的存取就替整個向量做準備而非一個字元。因此，主記憶體延遲的代價對整個向量只有一次，而非每個向量的字元都有一次。
- 由於整個迴圈被一個行為事先決定的向量指令所取代，因此由迴圈分支所產生的控制危障已不復存在。

從這幾點理由來看，向量運算可以在相同數量的資料的情況下執行得比一連串純量運算快。因假如應用範圍可以經常使用他們的話，設計師們很容易就會決定使用向量單元。

如同上面所說的，向量機器替一個向量的每一個元素做管線處理。而這樣子的管線處理不只包含算術運算（加減乘除…），而且包含記憶體的存取和有效位址的計算。除此之外，大部分高級的向量機器允許多重的向量指令同時執行，在不同元素的運算中產生平行性。這一章中，我們著重在利用管線處理和指令重疊增加效能的向量機器。第十章，我們將討論可以平行地在許多元素運算的平行機器而不是管線型的。

7.2 基本的向量結構

向量機器通常包括一個普通，有管線的純量單元再加上一個向量單元。向量單元裡的所有功能單元都有幾個時計週期的延遲時間。如此允許較短的時計週期時間以及和執行起來長時間且深度管線化而不會產生危障的向量運算相容。大多數的向量機器允許處理浮點數、整數及邏輯資料的向量，而我們現在只著重在浮點數上。純量單元基本上和第六章所討論的管線式 CPU 沒有差別。

兩種主要型式的向量架構是：向量—暫存器型向量機器和記憶體—記憶體型向量機器。在向量—暫存器型機器方面，所有的向量運算—除了載入和儲存之外—都是在向量暫存器上運作。這些機器是載入儲存架構的向量版。在 1990 年代所有上市的主要向量機器都採取向量—暫存器架構；其中包括 Cray Research 的機器 (CRAY-1, CRAY-2, X-MP 和 Y-MP)，日本的超級電腦 (NEC SX/2, Fujitsu VP2000 和 Hitachi S820)，還在迷你式超級電腦 (mini-supercomputer) (Convex C-1 和 C-2)。在記憶體—記憶體向量機器方面，所有的運算都是記憶體到記憶體的。早期的向量機器就是這一型的，比方說 CDC 的機器。從現在開始，我們集中注意力在向量—暫存器式的架構，在本章的結尾 (7.8 節)，我們將簡短地回來談一談記憶體—記憶體架構，探討為什麼他們不像向量—暫存器架構一樣成功。

我們將從一台包含 Figure 7.1 中所見的基本組成元件的向量一暫存器機器開始。這台機器概略的以 CRAY-1 為基礎，是這章中大部分討論的基礎。我們叫它 DLXV；它的整數部分叫 DLX，向量部分是 DLX 邏輯上向量式的擴充版。這一節其他的部分將觀察 DLXV 的基本架構和其它機器的關連。

DLXV 中指令集結構的主要組成元件是：

- 向量暫存器——每一個向量暫存器是一個儲存固定長度向量的地方。DLXV 有八個向量暫存器，每一個暫存器儲存 64 個雙倍字元。每一個暫存器儲存器必須至少有二個讀取埠和一個寫入埠。這樣子允許在對不同的向量暫存器運算的向量指令有比較高程度的重疊。(CRAY-1 設計者使用只有一個埠的暫存器檔案，但使用了一些聰明的實作技巧)。