

計算機組織與 RISC-V CPU 設計 · 第 1 章

計算機組織與結構導論

Organization and Architecture / 結構化描述方法 / 計算機的功能與結構

摘要

本章為整套教材的起點。我們首先釐清「計算機結構 (Architecture)」與「計算機組織 (Organization)」這兩個核心術語的差異, 接著以階層式系統的觀點, 說明如何有條理地描述一部由數十億電晶體構成的複雜機器。最後從功能與結構兩個視角拆解計算機: 資料處理、資料儲存、資料搬移與控制四大功能, 以及 CPU、記憶體、I/O 與系統互連四大結構元件。本章內容對應 Stallings 《Computer Organization and Architecture》第 1 章, 並補充現代 (RISC-V 時代) 的觀點。

目錄

1	結構與組織	2
1.1	兩個術語的定義	2
1.2	為什麼要區分	2
2	結構化的描述方法	2
2.1	階層式系統	2
3	計算機的功能	3
4	計算機的結構	4
4.1	頂層結構	4
4.2	CPU 的內部結構	4
5	為什麼要學計算機組織與結構	5
6	本教材的組織	5
7	本章重點回顧	6
8	複習問題	6

1 結構與組織

1.1 兩個術語的定義

在描述計算機系統時,常常需要區分「結構」與「組織」兩個層次:

定義 1.1 (計算機結構 Computer Architecture). 指程式設計師可見的系統屬性,亦即直接影響程式邏輯執行的屬性。包括: 指令集、各種資料型態的位元數 (例如以 32 位元表示整數)、輸入/輸出機制、記憶體定址技術等。

定義 1.2 (計算機組織 Computer Organization). 指實現結構規格的操作單元及其互連方式。組織的屬性對程式設計師是透明的,包括: 控制訊號、計算機與週邊的介面、所使用的記憶體技術等。

結構 vs. 組織的經典例子

「機器是否提供乘法指令」是一個結構層次的設計決策;「乘法指令是由專用的乘法單元實現,還是由重複使用加法器的機制實現」則是組織層次的決策。組織上的抉擇取決於乘法指令的預期使用頻率、兩種作法的相對速度,以及專用乘法單元的成本與晶片面積。

1.2 為什麼要區分

歷史上,這個區分具有重大的商業意義。許多計算機製造商提供同一結構、不同組織的整個產品家族,不同型號之間價格與效能不同,但同一套軟體可以在整個家族上執行:

- IBM System/370 結構於 1970 年推出,涵蓋一系列價格效能各異的機型。客戶可以先買便宜的機型,需求成長後升級到更快的機型,而不必放棄既有軟體。多年來 IBM 持續以改良的技術推出新機型,都維持同一結構,保護了客戶的軟體投資。這一結構的後裔一路延續至今日的 IBM zSeries 大型主機。
- 在微處理器世界,x86 結構 (Intel 8086 至今日的 Intel/AMD 處理器) 同樣展現了結構與組織分離的威力: 數十年來組織 (微結構) 已徹底翻新無數次,但結構保持向後相容。
- RISC-V 則是現代的極致案例: 它是一個開放、免授權費的指令集結構規格,任何人都可以自由設計自己的組織 (微結構) 來實現它——從幾千個邏輯閘的微控制器,到超純量亂序執行的高效能核心。這也是本教材後半部能「自己動手用 Verilog 造一顆 CPU」的根本原因。

重點觀念

結構 (Architecture) 可以存活數十年,組織 (Organization) 則隨著技術進步不斷改變。指令集結構是軟體與硬體之間的契約: 只要契約不變,兩邊都可以獨立演進。

2 結構化的描述方法

2.1 階層式系統

現代計算機含有數十億個基本電子元件,如何清楚地描述它? 關鍵在於認識大多數複雜系統 (包括計算機) 的階層 (hierarchical) 性質:

- 階層式系統是一組相互關聯的子系統, 每個子系統又可能在結構上是階層式的, 直到達到最低層次的基本子系統。
- 這種階層性質對設計者與描述者都至關重要: 每一層次的設計者只需關心該層次的結構與功能, 把下層視為可靠的黑盒子, 把上層視為使用情境, 因而大幅降低複雜度。
- 在每一層, 設計者關心兩件事:
 - 結構 (**Structure**): 各元件相互關聯的方式;
 - 功能 (**Function**): 作為結構一部分的每個個別元件的運作。

本教材採「由上而下 (top-down)」的描述方式: 先把計算機視為單一整體, 說明它的功能與外部介面, 再逐層向下拆解, 直到暫存器傳輸層 (RTL) 與邏輯閘——這也正是我們最終以 Verilog 描述 CPU 時所處的層次。

3 計算機的功能

計算機能執行的基本功能只有四種, 如圖1所示:

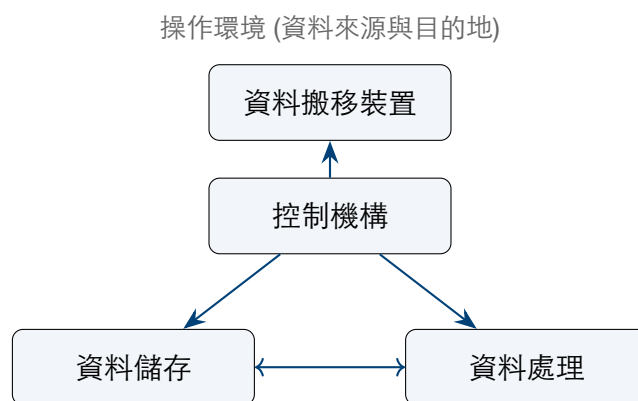


圖 1 計算機的功能觀 (依 Stallings 圖 1.2 重繪)

1. 資料處理 (**Data processing**): 資料可能有極多種形式, 處理需求的範圍也很廣, 但基本的處理方法其實只有少數幾類 (算術、邏輯、移位等)。
2. 資料儲存 (**Data storage**): 即使資料只是「流過」計算機, 機器也必須暫存正在處理的片段; 長期儲存功能則讓資料檔案可供日後檢索與更新。
3. 資料搬移 (**Data movement**): 計算機必須能在自身與外部世界之間搬移資料。與計算機直接相連的裝置稱為週邊 (**peripheral**), 此過程稱為輸入/輸出 (**I/O**); 當資料被搬移到更遠的裝置時, 則稱為資料通訊 (**data communications**)。
4. 控制 (**Control**): 上述三種功能必須受到控制。在計算機內部, 控制單元依據指令來管理計算機的資源、協調各功能部件的運作。

四種功能的組合情境

計算機可以作為資料搬移裝置 (例如將鍵盤輸入回顯到螢幕)、儲存裝置 (從記憶體讀出資料送到 I/O, 或反向)、以及對儲存中或往返於外界的資料進行處理。看似無所不能的應用軟體, 底層都是這四種功能的排列組合。

4 計算機的結構

4.1 頂層結構

計算機是一個與外部環境互動的實體。它的內部結構在最頂層可分為四大元件：

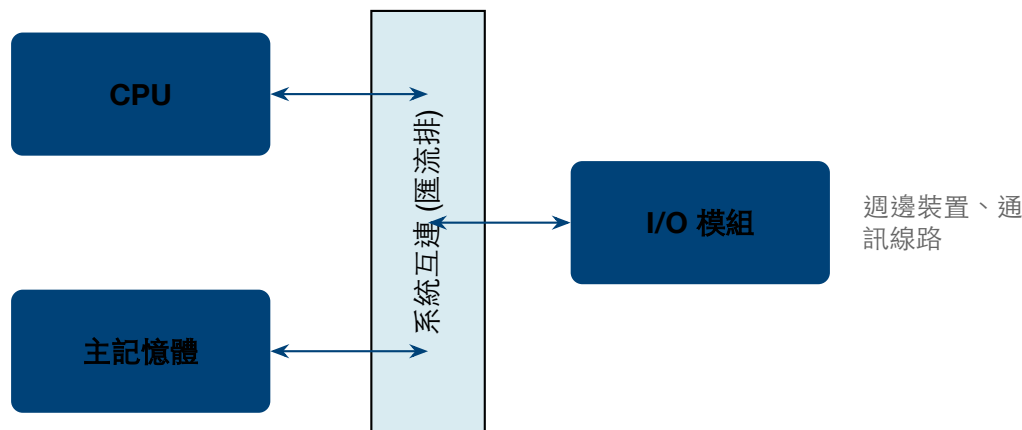


圖 2 計算機的頂層結構

1. 中央處理單元 (CPU): 控制計算機的運作並執行其資料處理功能; 常簡稱為處理器 (processor)。
2. 主記憶體 (Main memory): 儲存資料與程式。
3. I/O: 在計算機與其外部環境之間搬移資料。
4. 系統互連 (System interconnection): 提供 CPU、主記憶體與 I/O 之間通訊的機制, 傳統上是匯流排 (bus), 現代系統亦常採用交換式互連。

可能存在一個以上的上述元件; 傳統上只有一個 CPU, 但近年來多處理器/多核心已成為常態 (第 16 章)。

4.2 CPU 的內部結構

四大元件中最複雜、也最值得深入研究的是 CPU。其主要結構元件為：

- 控制單元 (Control Unit): 控制 CPU 的運作, 因而控制整部計算機 (第 15 章討論其實現, 包括微程式控制)。
- 算術邏輯單元 (ALU): 執行計算機的資料處理功能 (第 9 章計算機算術)。
- 暫存器 (Registers): 提供 CPU 內部的高速儲存 (第 12 章)。
- CPU 內部互連: 提供控制單元、ALU 與暫存器之間通訊的機制。

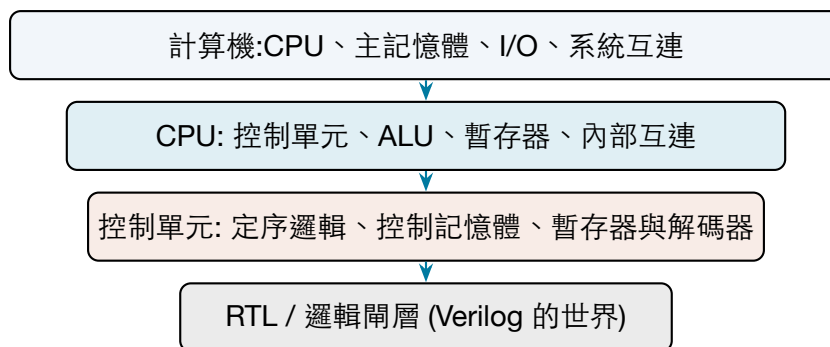


圖 3 由上而下逐層拆解: 本教材的路線圖

5 為什麼要學計算機組織與結構

1. 寫出高效能的軟體: 理解快取、管線、分支預測之後, 你會明白為何某些寫法快、某些寫法慢。例如以列優先 (row-major) 順序走訪二維陣列, 能充分利用快取的空間區域性, 可能比行優先快一個數量級 (第 4 章)。
2. 做出正確的系統選型: 效能不能只看時脈頻率。指令集、快取容量、記憶體頻寬、核心數與互連方式, 共同決定實際表現 (第 2 章)。
3. 理解軟硬體介面: 編譯器、作業系統、驅動程式的設計, 都建立在對指令集結構、中斷、虛擬記憶體等機制的理解上 (第 8、10、11 章)。
4. 設計硬體: 在 RISC-V 開放結構與 FPGA 普及的今天, 「自己設計一顆 CPU」已是大學部可完成的專題。本教材第三部分將以 Verilog 一步步實現 RV32I 處理器 (第 22–26 章)。

6 本教材的組織

部	章	內容
第一部	1	導論: 結構與組織、功能與結構的觀點
	2	計算機演進與效能設計: 歷史、摩爾定律、效能量測
	3–8	系統層: 匯流排互連、快取、內部/外部記憶體、I/O、作業系統支援
	9–16	處理器層: 計算機算術、指令集、管線、RISC、超純量、控制單元、平行處理
第二部	17	RISC-V 架構總覽與設計哲學
	18–21	RV32I 指令集詳解、組合語言程式設計、特權架構與 CSR
第三部	22	Verilog HDL 基礎
	23–26	單週期 RISC-V CPU 設計、五級管線設計、危障處理、驗證

注意

本教材以 Stallings 的經典教科書為骨架, 但把「範例機器」從書中的 Pentium 4 / PowerPC G4 更新為 **RISC-V**。理由有二: (1) RISC-V 是開放結構, 規格書可自由取得, 適合教學; (2) 第三部分的 Verilog 實作需要一個簡潔、正交的指令集, RV32I 僅 37 條基礎指令, 是理想的實作對象。

7 本章重點回顧

本章要點

- 結構是程式設計師可見的屬性 (指令集、資料型態、定址); 組織是實現結構的方式 (控制訊號、介面、記憶體技術)。
- 同一結構可以有許多不同組織的實現; 結構長壽, 組織常變。
- 計算機是階層式系統; 每層只需關心該層的結構與功能。
- 四大功能: 資料處理、資料儲存、資料搬移、控制。
- 四大結構元件: CPU、主記憶體、I/O、系統互連; CPU 又分為控制單元、ALU、暫存器與內部互連。

8 複習問題

1. 計算機結構與計算機組織的差異為何? 各舉兩個屬於該層次的設計決策。
2. 為什麼「提供乘法指令」是結構決策, 而「如何實現乘法」是組織決策?
3. 計算機的四種基本功能是什麼? 請針對「播放一段影片」這個應用, 說明四種功能各自扮演的角色。
4. 計算機頂層結構的四大元件為何? CPU 內部的四大結構元件為何?
5. 為什麼階層式的觀點對描述複雜系統至關重要?
6. 查閱 RISC-V 官方網站, 說明「開放指令集結構」與傳統商業結構 (x86、Arm) 在授權模式上的差異, 以及這對「同一結構、多種組織」現象的影響。

參考資料

- [1] W. Stallings, *Computer Organization and Architecture: Designing for Performance*, 6th ed., Prentice Hall, 2003, Chapter 1.
- [2] RISC-V International, *The RISC-V Instruction Set Manual, Volume I: Unprivileged Architecture*, version 20260120. <https://docs.riscv.org>
- [3] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed., Morgan Kaufmann, 2019.