

科學圖書大庫

微 處 理 機

技術、結構、應用

譯者 魏聰明

徐氏基金會出版

科學圖書大庫

微 處 理 機

技術、結構、應用

譯者 魏聰明

徐氏基金會出版

序

在近十年中，計算機系統的設計已經由分立元件發展到更複雜的積體電路（IC），就是將大量的邏輯閘濃縮在一小塊“晶片”（chip）上，因此產生出“晶片上的計算機”的觀念。

微處理機是一種可以執行算術與邏輯功能的LSI晶片，亦即是“晶片上的計算機”。本書對微處理機與微型電腦的技術，結構與應用均有深入淺出的介紹，各章的主要內容如下：

第一章：提出一些與微型電腦系統有關的基本計算機的構造。

第二章：計算機系統設計時，必須的準備工作，以及微型電腦的基本元件。

第三章：說明微處理機製造方面的一些半導體技術。

第四章：列出目前市面上已有的部份微處理機，並說明其結構，與特性。

第五章：以Intel 8080微處理機為實例，說明指令集、定時與同步，內部暫存器的操作。

第六章：涵蓋了系統定義，流程圖、寫碼、測試與除錯等工作。

第七章：微處理機發展系統的效用，設計與發展一套原型微電腦系統的重要工具。

第八章與第九章：提出一些微處理機的應用，其範圍從簡單的家庭用微型電腦，到精巧複雜的遠距通訊。

目 錄

序

第一章 微型電腦簡介

計算機結構.....	1
微處理機結構.....	14
微處理機軟體.....	18
迷你計算機與微型電腦.....	20
微處理機應用與影響.....	21

第二章 微型電腦系統之設計

系統設計之標準.....	29
微處理機系統的可行性.....	31
微處理機的選用.....	32
輔助硬體的設計.....	33
軟體的發展.....	34
硬體與軟體的組合.....	34
除錯與測試.....	35
發展維護設備.....	36
微型電腦元件.....	36

第三章 微處理機的技術

積體電路技術.....	50
-------------	----

平面矽處理技術.....	55
雙極性之處理.....	57
MOS處理.....	59
微處理機技術.....	66
技術的淘汰.....	70
技術的比較.....	72

第四章 微處理機綜觀

Intel 4004	74
Intel 4040	77
Intel 8008	81
Intel 8080	83
Intel 3002	84
Motorola M6800	90
National IMP-16	91
Fairchild F-8	92
Texas Instruments SBP-0400	98
General Instrument CP-1600	100
Signetics 2650	102

第五章 微處理機運算

8080 微處理機架構.....	106
指令集.....	110
定時與同步.....	126
電氣規定.....	132
界面.....	135

第六章 微處理機規劃

系統定義.....	138
功能流程圖.....	147

程式流程圖.....	150
寫碼.....	150
測試與除錯.....	153
第七章 微處理機發展系統	
使用發展系統的系統設計.....	157
軟體發展的協助程式.....	160
第八章 微處理機的應用	
低成本的通用計算機.....	163
銷售站終端機.....	167
汽車的應用.....	172
第九章 微處理機的高級應用	
遠距通訊.....	177
多處理機系統.....	197
解釋名詞	203
附錄	

第一章 微型電腦簡介

微型電腦 (Microcomputer) 乃是採用微處理機 (Microprocessor) 做為系統之中央處理單元 (Central Processing Unit) 的內儲程式計算機 (Stored-program Computer) 。在我們更加詳盡地描述微處理機與微型電腦之前，有必要先簡略地複習一下數位計算機的基本概念——有關硬體構造、軟體，以及作業系統。

一般而言，數位計算機 (Digital Computer) 主要包含四個基本單元：輸入／輸出，記憶體 (Memory) ，算術／邏輯單元，以及控制單元。所有的普及型計算機，自微型電腦至大型的多元處理系統 (Large-Scale Multiprocessing System) ，均能以此四種基本單元加以描述。有關這四項基本單元間的特殊設計與相互關聯，稱為機器之結構 (Architecture of the Machine) 。

就整體概念而言，硬體 (Hardware) 與軟體 (Software) 是計算機結構的兩大骨架。此項結構並描寫存在於靜態的硬體設計與動態的軟體處理之間的相互作用。雖然，計算機結構有許多種分類與型式，然而在本書中，僅要論及兩種最實際的結構，做為微型電腦之應用：

方紐曼 (Von Neuman) 結構

堆疊式 (Stack) 結構

等待詳細研討了這些結構之後，我們會指出如何用這些結構來完成微型電腦系統，然後再提出一些採用微處理技術的經濟價值。

計算機結構 (Computer Architecture)

對於組成通用型數位計算機的四個基本構架單元，是以圖 1-1 的方塊圖來解說。通用型計算機屬於資訊處理裝置；可以對提供的資訊執行

一序列的算術或者邏輯運算。這些資訊必須是經由輸入 / 輸出 (或者 I / O) 設備進出計算機。在計算機內部任何即將處理的資訊以及將執行編碼 (Encode) 程序的資訊, 均儲存於記憶器。至於確實的處理過程由數位邏輯電路來執行, 這些邏輯電路亦得受控制單元來完成時激 (Clocked) 與安排順序 (Sequenced)。而控制單元也關聯到輸入 / 輸出設備的動作, 算術 / 邏輯電路的記憶器。

計算機結構有不同的型式, 是因為這簡化的功能方塊結構 (Function Block Structure) 的變動。至於這些變動乃因增加數個記憶器, 或者增加算術 / 邏輯電路 (處理機), 或者是因修改了基本單元間的控制性能而引起的。某些更週密的結構可能將記憶器與處理機構細地區分成幾個階層, 並且又建立另一套嚴密的系統控制程序, 而循最佳方法去使用較缺乏的系統資源 (System Resource)。雖然這種結構可能 (或許是在將來某時刻) 借助處理機而完成的, 然而目前的微型電腦却用不着如此週密的設計。

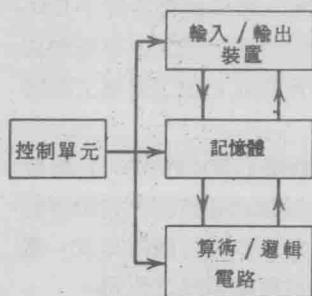


圖 1-1 數位計算機方塊圖

方紐曼結構

方紐曼結構被廣泛地用於數位計算機, 尤其是大部份微處理機結構的基礎。方紐曼結構規定了控制單元與記憶單元間相互作用的型式。控制單元是按照一序列的指令 (Instructions) 來執行工作, 這一序列的指令稱為程式 (Program), 它是儲存於記憶單元中。每一指令包含兩部份: 運算體 (Operator) 與運算元 (Operand) (圖 1-2 所示)

。運算體指示微處理機應該去執行那些指定的運算（ADD為算術運算，AND是邏輯運算，控制動作則如JUMP），而運算元則指示在記憶單元內有那些資料或位址將被用來運算。

方紐曼計算機須經由一序列指令的執行而達到運算的功能，這正是系統所要的運算。因為控制單元與記憶單元間有重要的關聯，所以必須提供特殊的硬體構造，（如圖1-3所示），用以增加資料與指令的轉送速度，這些硬體結構為：

指令暫存器（Instruction Register）

程式計數器（Program Counter）

累加器（Accumulator）

算術／邏輯單元（又稱ALU）

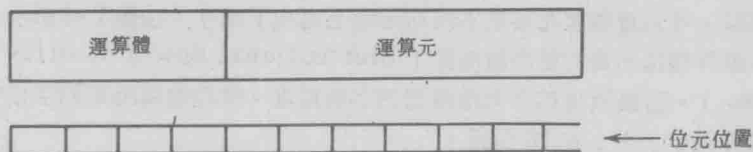


圖 1-2 指令字的格式

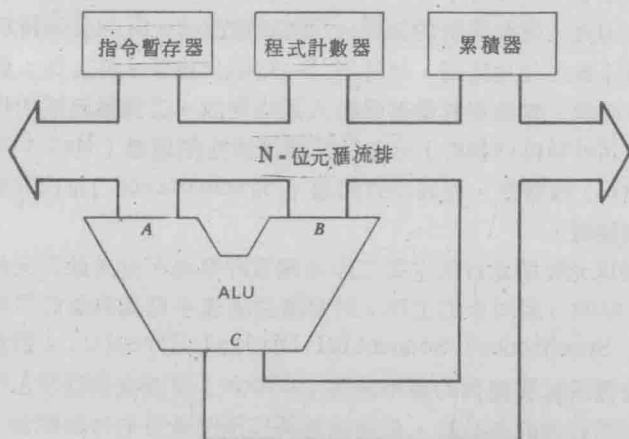


圖 1-3 雙向匯流排結構

指令暫存器，程式計數器、累加器，是如圖 1-1 所示的記憶單元的最基本元件。這些元件均為暫存器，儲存單體 (Storage Cells)，其位元容量恰等於機器的字長度 (Word Length)。指令暫存器記載著計算機即將執行的下一個指令。程式計數器所記載的是計算機下一個要執行的指令在記憶單元中的位址。累加器是記載即將被處理的資料，或者接受經過處理後的新資料。

ALU 負責對程式預先指定的資料執行算術或邏輯運算。ALU 可表為只有 A、B 兩個輸入，一個輸出 C。ALU 所要執行的算術與邏輯運算的資料，是自 A 與 B 輸入，然後將運算結果置於輸出 C。

這些硬體結構彼此被一個或更多的匯流排 (Bus) 所相連著。匯流排，顧名思義就是指許多接點之間用以互相連結的電路，而這些接點均可做為系統中的信號源。(與我們在日常生活中所見的巴士，BUS，很相似，可以讓乘客在某些不同的地點上車或下車)。由圖 1-3 所示，有一條 N 個位元寬的雙向匯流排 (Bidirectional Bus of N Bits Wide)，這裏所提的 N 是指機器的字組長度。微型電腦所用的字長度大部份是 4，8，或 16 位元。

經由匯流排可連通這些暫存器，ALU，記憶單元，以及 I/O 裝置。在一個單一匯流排 (Single-Bus) 結構中，每個裝置必須連結至匯流排，因此，便與其他的裝置分用這條匯流排，所以必須特別分配各裝置對匯流排的使用時間，通常在某一時刻只能做一件工作；將資訊傳送給另一裝置，或是從其他裝置輸入某些資訊。這種匯流排的時間多工 (Time Multiplexing) 分配是受匯流排控制硬體 (Bus Control Hardware) 的管制，而同步的問題 (Synchronized) 則由計算機的控制單元來操縱。

控制單元所要遂行的主要工作是掌管計算機所有其餘單元的監督 (Supervision) 與同步的工作。計算機系統幾乎都是典型的同步順序數位電路 (Synchronous Sequential Digital Circuit)。計算機的各單元都是靠系統所提供的標準時激 (Clock) 信號來做同步工作。每一個元件必需經過詳細設計，然後經過某些預定數目的時激脈衝 (Pulse)，才能產生順序的變化。

圖 1-4 是方紐曼計算機內基本元件的簡化表圖。其中的時激信號產生器 (Generator) 有二個輸出相位 ϕ_1 和 ϕ_2 。這種計算機最基本的作用元件便是 ALU，它在圖 1-3 中已經討論了。指令暫存器將下個指令提供指令解碼器，解碼器 (Decoder) 能夠解釋指令的位元圖型 (Bit-Pattern)，從而供給適切的指令指示 (Instruction Indication) 給機器週期控制 (Machine Cycle Control) 單元與 ALU。此外機器週期控制單元，亦要將這些指示翻譯成指令的型式，以便提供外部的指示給控制輸出單元去執行，同樣地，也要提供給記憶單元。所以，其他的系統部門才能與 ALU 的運算取得同步。記憶單元可以是讀 / 寫 (Read / Write) 記憶單元或者是機器的內部暫存器，這些記憶單元必須經由信號的指示才能提供資料去處理。為了使機器在作業中能夠暫停一下 (Halt) 或者中斷 (Interrupt) 一會，各種不同的控制輸入信號被用以供給機器週期的外部控制功能。

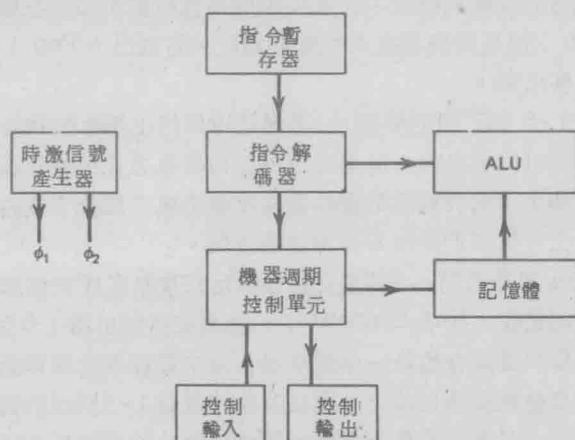


圖 1-4 方紐曼結構

堆疊結構 (Stack Architecture)

計算機結構可以配合使用者的程式語言來製造。這種結構有很多種變化的型式，其稱法為“可再組合的結構” (Reconfigurable Archi-

tructures)，或是“可編譯的結構”(Compilable Architectures)，類似這樣的結構方式在往後的微處理機應用方面，必定佔重要的領域。然而時至今日，這種結構型式中最主要的是“堆疊結構”，我們所以要如此稱之，乃是因為它在運用系統資源時，是採取“後進先出”(Last-In-First-Out)的堆疊方式來做為管制資料的方法。

堆疊結構遠在1960年代就已發展出來，用它所提供的所謂“堆疊”的方式，機器可以直接執行指令，而增加了計算的效果。然而，堆疊在實際上僅是一種暫時儲存的記憶單元，或者可以說是一種主要記憶單元的特殊部份，而它可用來連續地儲存資料。當有新的資訊要進入堆疊時，其他原已在堆疊內的那些資訊，就必須往下移動(Move Down)，因此，在任何時候，最新的資訊都是堆在堆疊的最頂端(Top)。一旦有些資訊要從堆疊中移出時，就要以相反的順序，將資訊往上擠出(Removed Up)。因此，最後放在堆疊頂端的資訊，便是最先移出。這種後進先出的堆疊的概念，我們詳細地描述於圖1-5，在圖中，所謂壓進(Push)便是將資訊送入堆疊的過程，而取出(Pop)便是將資訊自堆疊中移出來。

在圖1-5中所示的構造，只不過是用以指出堆疊作用如何產生，而並不是要表明計算機內，後進先出的這種堆疊方式的真正構造。換而言之，在功能上，記憶單元的資料是在堆疊中被“壓進”或被“取出”，但是實際上，真正的操作却有相當的差異。

圖1-6則是表明一個堆疊在記憶單元內實際形成的情形。其中有一項關鍵性的元件(Key Element)，就是堆疊指引器(Stack Pointer)，堆疊指引器本身也是一個暫存器，其中儲存著記憶單元位置的位址，那就是堆疊最頂端的位址。現在讓我們以圖1-6所示的例子來講解，在圖1-6左上方就是堆疊指引器，其中已存放的是位址051，因此，指的是在記憶單元內位置為051裏面所含的東西，現在051裏面的內容是 D_1 ， D_1 表示資料。當其他資料 D_2 要“壓進”堆疊時，只能將它放在位置050裏面，所以這時的堆疊指引器就變為可以讀出050位址，050表示此刻堆疊最頂端的位置，就如圖1-6下方所顯示。

同樣的，當資料要從堆疊中被取出時，只要讀出堆疊指引器內的位

址，便尋到記憶單元的位置，然後取出資料，並且記得把堆疊指引器所含的值加一。

堆疊指引器所指到的記憶單元，可能是計算機的主要記憶單元，或者是一種高速緩衝記憶單元。這二種裝置都適合用於微處理機系統。

雖然按照方紐曼結構而製作完成的計算機能夠具有堆疊構造，正如同很多微處理機系統一般，可增他們的計算效果，然而它們是“堆疊處

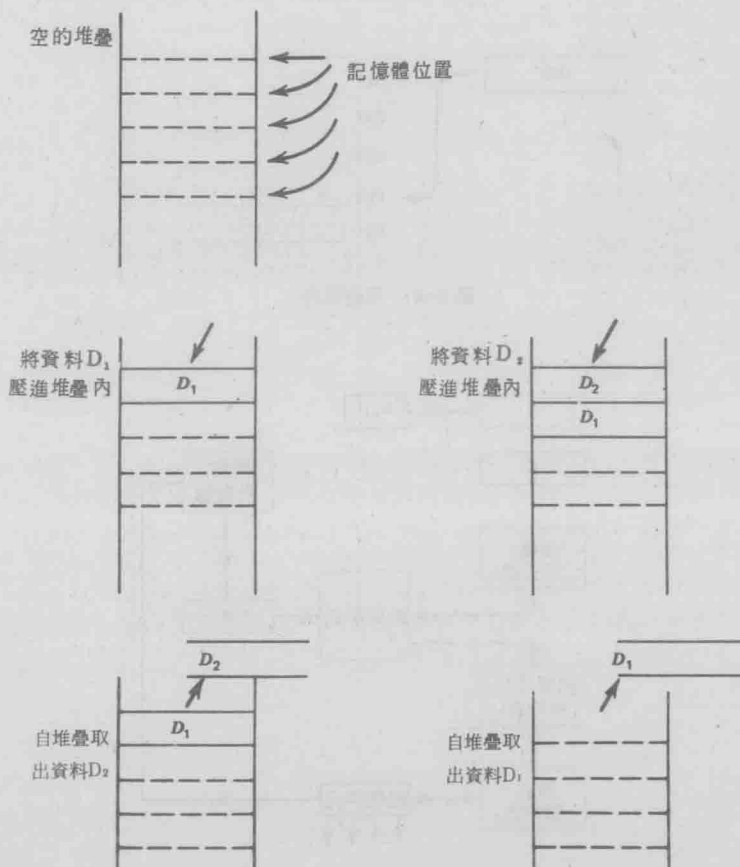


圖 1-5 壓進堆疊的操作

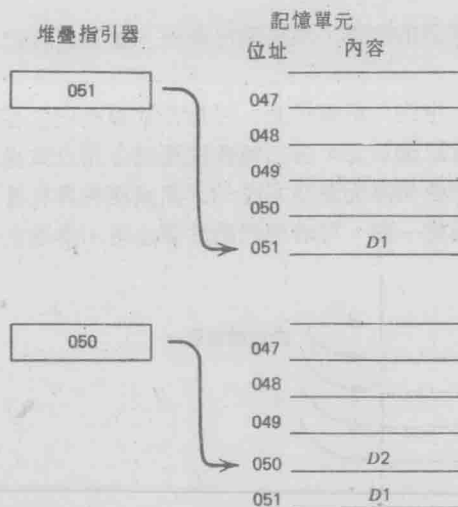


圖 1-6 堆疊完成

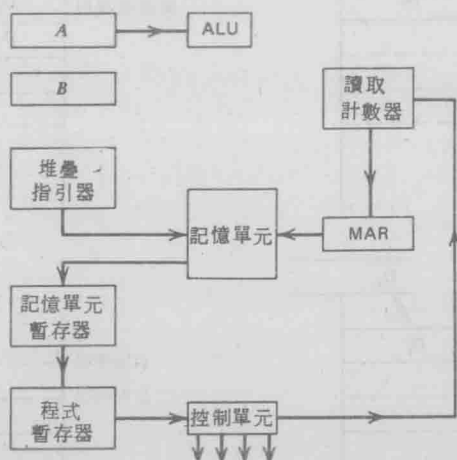


圖 1-7 堆疊處理機

理機”(Stack Processor)基本的運算結構,因此,可用於設計堆疊結構的計算機。

堆疊式微處理機 我們之所以應用堆疊式微處理機,是因為要更有效地執行高水準的語言編碼。因此,這種特殊的方法當然要應用特殊的格式或資料表示法,而且又要有特殊的指令編碼,用來給計算機執行。這種格式即是有名的“波力西”(Polish)表示式,此外尚有一些不同的稱法,如波力西前接法(Polish prefix),波力西後接法(Polish Postfix),以及反波力西表示式(Reverse Polish Notation)。

在堆疊機器裏,編譯程式需要對程式編碼實施文句法(Syntactical)與語義(Semantic)的分析,然後才能將程式表為一串機器運算順序。波力西後接表示式已經在Burroughs B-5500與B-6500型機器裏確實應用了,同時English Electric的KDF-9也應用完成。

由於已經有了實際的堆疊產品問世,所以拿B-6500來做為討論堆疊結構的例證是很值得的。堆疊式處理機的硬體結構表示於圖1-7,在堆疊式處理機裏有一個標準記憶器與記憶器位址暫存器(MAR),同樣地,在方紐曼機器內亦有這些相似的裝置。

爲了有別於方紐曼機器內的“指令”稱法,我們將堆疊式處理機內的資料與控制敘述引用為“語節”(Syllables)。目前共有三種基本的語節型態:

運算體語節(Operator Syllables)

宣用值語節(Value Call Syllables)

文字語節(Literal Syllables)

運算體語節所表示的是處理機所要執行的一些特定的算術或邏輯運算。宣用值語節則表示運算元被放置於記憶單元內的位置。最後文字語節才是真正的運算元。

堆疊式處理機所要執行的程式是由序列的語節組成的,這些語節又以連續的方式儲存在記憶單元的指定位置內。第一個程式語節的位址先放入讀取計數器(Fetch Counter),接著便由讀取計數器發出脈衝信號,通知MAR把指定的程式語節自記憶單元傳送出去。

當程式語節由記憶單元傳出來時，讀取計數器的數值被增加，而後把程式語節儲存在記憶單元的暫存器內。然後下一個步驟是把儲存在記憶單元暫存器內的程式語節轉移到程式暫存器內，接著便在程式暫存器內加以解碼了。語節內部必須要有二個位元用來顯示這個語節是運算體，或是宣用值，或是文字語節。等到程式暫存器內的語節被解釋後，就會有一序列的特定控制脈衝被送到系統的其他元件，因此可以使執行這些解釋語節的運算，能夠同步進行。

關於控制運算與堆疊機構的詳情，讓我們先來參考一個簡單的例子之後，就能更容易瞭解。現在我們要計算如下的一個算術式子：

$$X = (a + b) / (c - d)$$

其中 a, b, c, d 是一組已知的整數值。

第一步是將上述的標準型算術式子，表為沒有括號的波力西表示式。雖然一般形式的波力西表示式都可以使用，但在商業計算機方面，通常都採用波力西後接表示式。在後接表示式中，運算體（亦即加和減）被寫在一對運算元的右邊。因此，若用後接表示式，則可將上面的算術式改寫成如下的形式：

$$a \ b + \ c \ d - /$$

將一般式轉換成波力西後接表示式的規則如下：

1. 將式列 (String) 由左至右掃描一遍 (Scan)。
2. 記住運算元與其在式列中出現的次序。
3. 當遇到一個運算體時，須先做如下的動作：
 - a. 取出最後出現的兩個運算元。
 - b. 根據遇到的運算體的型式，對取出的兩個運算元執行運算。
 - c. 然後去掉這兩個已經運算過的運算元。
 - d. 記住在步驟 b 得出的結果，並將它視為最後的運算元。

另一種描述波力西表示式列的方法，是利用一種二元樹狀法 (Binary Tree) 來表示。這種二元樹狀結構，對於算術式列的執行，首先依照由左至右規則，然後按照分枝順序來運算。二元樹狀結構示於圖 1-8，這種結構形式，能更明顯地表示出運算體的優先順序。

這些運算均使用堆疊式微處理機來完成。這種處理機的硬體元件已

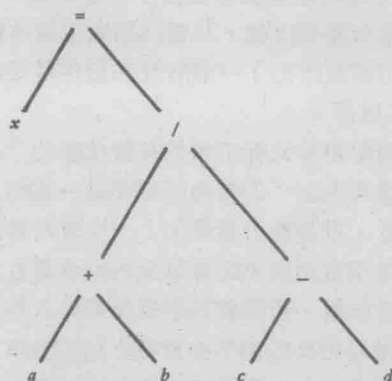


圖 1-8 二進位樹狀結構

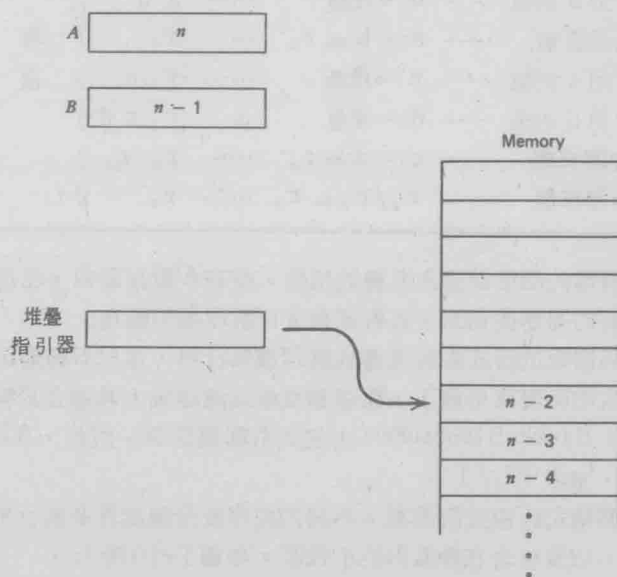


圖 1-9 處理機中的堆疊機構