



(19)中華民國智慧財產局

(12)發明說明書公告本

(11)證書號數：TW I548994 B

(45)公告日：中華民國 105 (2016) 年 09 月 11 日

(21)申請案號：101117850

(22)申請日：中華民國 101 (2012) 年 05 月 18 日

(51)Int. Cl. : G06F13/14 (2006.01)

G06F13/38 (2006.01)

G06F9/06 (2006.01)

(30)優先權：2011/05/20 美國

61/488,683

(71)申請人：軟體機器公司 (美國) SOFT MACHINES, INC. (US)

美國

(72)發明人：艾伯戴爾拉 摩翰麥德 ABDALLAH, MOHAMMAD (US)

(74)代理人：李宗德

(56)參考文獻：

TW 200602981A

CN 1848095A

US 7213248B2

US 7925869B2

審查人員：郭彥鋒

申請專利範圍項數：15 項 圖式數：14 共 44 頁

(54)名稱

以複數個引擎支援指令序列的執行之互連結構

AN INTERCONNECT STRUCTURE TO SUPPORT THE EXECUTION OF INSTRUCTION SEQUENCES BY A PLURALITY OF ENGINES

(57)摘要

一種通用互連系統。通用互連系統包括具有用以支援多個編碼序列之執行的資料的複數個資源以及用以實施多個編碼序列之執行的複數個引擎。複數個資源用戶係於複數個引擎之每一個內。通用互連系統係耦合至複數個資源用戶且耦合至複數個資源，以致能資料存取及多個編碼序列之執行，其中資源用戶經由通用互連結構之每一循環利用而存取資源。

A global interconnect system. The global interconnect system includes a plurality of resources having data for supporting the execution of multiple code sequences and a plurality of engines for implementing the execution of the multiple code sequences. A plurality of resource consumers are within each of the plurality of engines. A global interconnect structure is coupled to the plurality of resource consumers and coupled to the plurality of resources to enable data access and execution of the multiple code sequences, wherein the resource consumers access the resources through a per cycle utilization of the global interconnect structure.

指定代表圖：

符號簡單說明：

10 . . . 通用前端抓
取及排程器

11-14 . . . 引擎

20-23 . . . 編碼序列

30 . . . 通用互連結
構

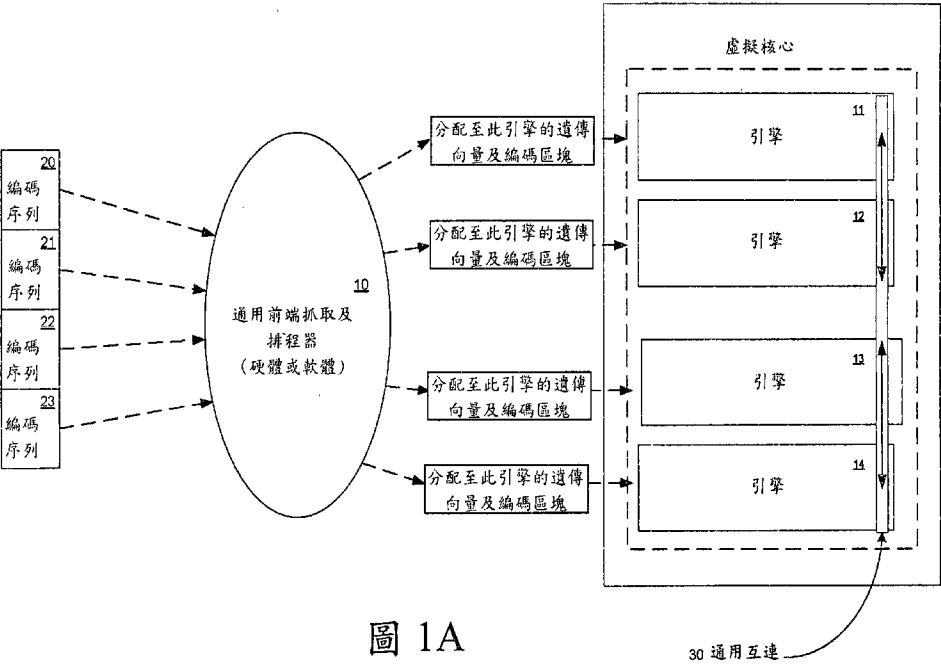


圖 1A

發明專利說明書

(本說明書格式、順序，請勿任意更動，※記號部分請勿填寫)

※ 申請案號：101117850

※ 申請日：101 年 5 月 18 日

※IPC 分類：

G06F 13/14 (2006.01)
13/38 (2006.01)
9/06 (2006.01)

一、發明名稱：(中文/英文)

以複數個引擎支援指令序列的執行之互連結構

AN INTERCONNECT STRUCTURE TO SUPPORT THE
EXECUTION OF INSTRUCTION SEQUENCES BY A
PLURALITY OF ENGINES

二、中文發明摘要：

一種通用互連系統。通用互連系統包括具有用以支援多個編碼序列之執行的資料的複數個資源以及用以實施多個編碼序列之執行的複數個引擎。複數個資源用戶係於複數個引擎之每一個內。通用互連系統係耦合至複數個資源用戶且耦合至複數個資源，以致能資料存取及多個編碼序列之執行，其中資源用戶經由通用互連結構之每一循環利用而存取資源。

三、英文發明摘要：

A global interconnect system. The global interconnect system includes a plurality of resources having data for supporting the execution of multiple code sequences and a plurality of engines for implementing the execution of the multiple code sequences. A plurality of resource consumers are within each of the plurality of engines. A global interconnect structure is coupled to the plurality of resource consumers and coupled to the plurality of

resources to enable data access and execution of the multiple code sequences, wherein the resource consumers access the resources through a per cycle utilization of the global interconnect structure.

四、指定代表圖：

(一)本案指定代表圖為：圖 1A。

(二)本代表圖之元件符號簡單說明：

10	通用前端抓取及排程器
11-14	引擎
20-23	編碼序列
30	通用互連結構

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：無。

六、發明說明：

本申請案係主張共同審查中且共同讓與之美國專利臨時申請案案號 61/488, 683 的優先權，其專利名稱為「以複數個引擎支援指令序列的執行之互連結構 (AN INTERCONNECT STRUCTURE TO SUPPORT THE EXECUTION OF INSTRUCTION SEQUENCES BY A PLURALITY OF ENGINES)」，由 Mohammad A. Abdallah 於 2011 年 5 月 20 日所申請，其整體內容係併入本文。

【相關美國專利申請案】

本申請案係關於共同審查中且共同讓與之美國專利公開號 2010/0161948，其專利名稱為「用以處理在支援各種上下文切換模式及虛擬化方案之多線程架構中之複雜指令格式的裝置及方法 (APPARATUS AND METHOD FOR PROCESSING COMPLEX INSTRUCTION FORMATS IN A MULTITHREADED ARCHITECTURE SUPPORTING VARIOUS CONTEXT SWITCH MODES AND VIRTUALIZATION SCHEMES)」，由 Mohammad A. Abdallah 於 2007 年 11 月 14 日所申請，其整體內容係併入本文。

本申請案係關於共同審查中且共同讓與之美國專利公開號 2009/0113170 其專利名稱為「用以處理指定相依操作中之平行之指令矩陣的裝置及方法 (APPARATUS AND METHOD FOR PROCESSING AN INSTRUCTION MATRIX SPECIFYING PARALLEL IN DEPENDENT OPERATIONS)」，由 Mohammad A. Abdallah 於 2007 年 4 月 12 日所申請，其整體內容係併入本文。

【發明所屬之技術領域】

本發明一般係關於數位電腦系統，特別是關於用以選擇包含一指令序列之指令的系統及方法。

【先前技術】

需要處理器來處理相關或完全無關的多樣任務。此類處理器的內部狀態通常由暫存器組成，其可保持在程式執行的每一特定瞬間之不同數值。在程式執行的每一瞬間，內部狀態影像係稱作處理器的架構狀態。

當編碼執行係切換以執行另一功能(例如另一線程、程序或程式)，則需儲存機器/處理器的狀態，使得新功能可利用內部暫存器來建立其新狀態。一旦新功能終止，則可拋棄其狀態，且先前上下文的狀態將恢復並重新開始執行。此一切換程序係稱作上下文切換，且通常包含數十或數百循環，特別是具有使用大量暫存器(如 64、128、256)及/或無序執行的現代架構。

在線程感知硬體架構中，標準情況為硬體針對有限數量的硬體支援線程支援多個上下文狀態。在此情況中，硬體針對每一支援線程複製所有架構狀態元件。這消除了當執行一新的線程時上下文切換的需要。然而，這仍具有多個缺點，即針對硬體中所支援之每一額外線程而複製所有架構狀態元件(即暫存器)之面積、功率、及複雜度。此外，若軟體線程的數量超過所明確支援之硬體線程的數量，則仍必須執行

上下文切換。

這已成為普遍的，如同需要大量線程之精細粒度基礎(fine granularity basis)上所需之平行化。硬體線程感知架構與複製的上下文狀態硬體儲存無助於非線程軟體編碼，且僅降低線程化軟體之上下文切換的數量。然而，那些線程通常係針對粗粒平行化(coarse grain parallelism)而建構，且導致沉重的軟體花費以初始化及同步化，捨棄了細粒平行化(fine grain parallelism)，例如功能呼叫及迴圈平行執行，而無有效的線程初始化/自動產生。這類所描述的花費係伴隨這類編碼之自動平行化的困難，其針對非明確/簡易平行化/線程軟體編碼使用最新式的編譯器或使用者平行化技術。

【發明內容】

在一具體實施例中，本發明係實施為一通用互連系統。通用互連系統包括具有用以支援多個編碼序列之執行的資料的複數個資源以及用以實施多個編碼序列之執行的複數個引擎。複數個資源用戶係於複數個引擎之每一者內。通用互連系統係耦合至複數個資源用戶且耦合至複數個資源，以致能資料存取及多個編碼序列之執行，其中資源用戶經由通用互連結構之每一循環利用而存取資源。

前述為發明內容，因此必然包含簡化、概括且省略細節。因此，熟此技藝者將了解，發明內容僅為說明性而無意圖做任何方式的限制。本發明的其他態樣、發明特徵及優點係僅由申請專利範圍所定義，且在以下所提出之非限制性詳細說明中將變得明顯。

【實施方式】

雖然本發明已相關於具體實施例而描述，本發明並不意欲限制於在本文中所提出之特定形式。相反地，本發明係意圖涵蓋可合理地包括於由後附申請專利範圍所定義之本發明範疇內之替代、修改及等效物。

在以下的詳細描述中，提出了許多具體的細節，像是特定的方法順序、結構、元件及連接。然而，應了解到，可不需利用這些及其他具體細節來實行本發明的具體實施例。在其他情況下，省略或沒有特別地詳細描述習知結構、元件、或連接，以避免不必要地模糊了說明內容。

說明書中所提到的「一個具體實施例(one embodiment)」或「一具體實施例(an embodiment)」係意指，關聯於一具體實施例而描述的特定特徵、結構或特性係包括於本發明之至少一個具體實施例中的一具體實施例。在本說明書中許多地方所出現的「在一個具體實施例中」一詞並不需全指相同的具體實施例，也不需是與其他具體實施例互相排斥的單獨或另外的具體實施例。此外，所描述的許多特徵可能會在某些具體實施例中呈現而沒有在其他具體實施例中呈現。類似地，所描述的許多需求可能為某些具體實施例的需求，但不為其他具體實施例的需求。

詳細說明的某些部分將於下文中以電腦記憶體內之資料位元上操作的程序、步驟、邏輯塊、處理、及其他符號表示。這些描述及表示為熟習資料處理技藝者所使用的手段，

以最有效地將其工作的實質內容傳達給其他熟此技藝者。程序、電腦執行步驟、邏輯塊、處理等等於此處且一般而言係設想為導致所需結果之自相一致順序的步驟或指令。這些步驟為需要實體量的實體操作。通常情況下，但非必須，這些量的形式為電腦可讀儲存媒體的電性或磁性訊號，且能夠被儲存、轉移、結合、比較、或操作於電腦系統中。已證實有時(主要為了平常使用)將此等訊號稱作位元、值、元件、符號、字元、術語、數字或其類似者係便利的。

然而，應注意，所有這些或類似用語與適當實體量有關，且僅為應用到這些量的便利符號。於下討論中除非有特別指明，不然應知本發明中使用例如「處理(processing)」、「存取(accessing)」、「寫入(writing)」、「儲存(storing)」、「複製(replicating)」、或類似者等用語之討論，係指電腦系統或類似電子計算裝置的動作及程序，其將電腦系統暫存器及記憶體及其他電腦可讀媒體內表示為實體(電子)量的資料操控且轉換成在電腦系統記憶體或暫存器或其他此類資訊儲存、傳輸、或顯示裝置內類似地表示為實體量的其他資料。

本發明具體實施例利用一前端排程器、複數個區段暫存器檔案或單一暫存器檔案、一記憶體次系統以實施用於多核心處理器之多核心的片段位址空間。在一具體實施例中，分段藉由允許額外的虛擬核心(如軟核心)以協力地執行包含一或多個線程的指令序列而致能微處理器效能的尺度化(scaling)。分段階層在每一快取階層(如 L1 快取、L2 快取)為相同。分段快取使用位址位元將位址空間劃分為片段，其中位址位元係使用使得片段係由在快取線邊界之上以及頁邊界之下的位元所識別。每一片段係組態以利用一多埠庫結

構供儲存。本發明具體實施例將在以下圖 1A 及 1B 中作進一步地描述。

圖 1A 顯示根據本發明一具體實施例之處理器的概要圖。如圖 1A 所繪示，處理器包括一通用前端抓取及排程器 10 及複數個可劃分引擎 11 至 14。

圖 1A 顯示通用前端產生編碼區塊及遺傳向量以支援編碼序列於其個別可劃分引擎上之執行的方式的概要。每一編碼序列 20 至 23 可屬於相同邏輯核心/線程或屬於不同的邏輯核心/線程，其取決於特定的虛擬核心執行模式。通用前端抓取及排程器將處理編碼序列 20 至 23 以產生編碼區塊及遺傳向量。這些編碼區塊及遺傳向量係分配至特定的可劃分引擎 11 至 14，如圖所示。

引擎根據一所選模式而執行虛擬核心。一引擎包括一區段、一片段、及一些執行單元。在引擎內的資源可用以執行具有多重模式的虛擬核心。當配置為虛擬核心模式，可實施一軟核心或許多軟核心以支援一邏輯核心/線程。在圖 1A 的具體實施例中，根據所選模式，虛擬核心可支援一邏輯核心/線程或四個邏輯核心/線程。在虛擬核心支援四個邏輯核心/線程的具體實施例中，每一虛擬核心的資源係散佈於每一可劃分引擎。在虛擬核心支援一邏輯核心/線程的具體實施例中，所有引擎的資源係專屬於該核心/線程。引擎係劃分使得每一引擎提供包含每一虛擬核心之資源的一子集。換言之，虛擬核心將包含引擎 11 至 14 之每一者之資源的一子集。引擎 11 至 14 之每一者之資源之間的通訊係由一通用互連結構 30 所提供，以幫助此程序。或者，引擎 11 至 14 可

用以實施引擎 11 至 14 之資源係專用以支援一專屬核心/線程之執行的一實體模式。在此方式中，由引擎所實施的軟核心包含具有散佈至每一引擎之資源的虛擬核心。虛擬核心執行模式將在以下圖式中進行進一步的描述。

應注意，在傳統的核心實施中，在一核心/引擎內的資源係單獨地分配至一邏輯線程/核心。相反地，在本發明具體實施例中，可劃分任何引擎/核心的資源(其與其他引擎/核心劃分協力)以實例化分配至一邏輯線程/核心的一虛擬核心。本發明具體實施例也可實施那些相同引擎可被劃分以支援許多專屬核心/線程或許多動態分配的核心/線程之多個虛擬執行模式、以及所有引擎的所有資源支援單一核心/線程之執行的組態。以下將進一步描述某些代表性具體實施例。在本發明的其他具體實施例中，本發明的技術可直接地應用至傳統的多核心實施，以致能多核心共享資源及互連的有效率競爭、保留及分配。類似地，本發明應用於單一核心或計算引擎內，以致能在核心內之任何共享資源或互連(即埠、匯流排、執行單元、快取、結構)的有效率競爭、保留及分配。

舉例來說，圖 1A、圖 1B、及圖 5 所示的具體實施例可由一典型的多核心設計所取代，其不具有通用前端或遺傳向量，而是具有實例化多核心或多線程(其具有對資源(如快取、共享互連(如網孔或網格)或共享多方向匯流排)的存取)的引擎。在這類具體實施例中，本發明仍可直接地應用以允許有效率的資源及互連競爭、保留、及分配。類似地，本發明具體實施例可應用至每一核心或引擎，以競爭、保留、及分配資源或互連。

圖 1B 顯示根據本發明一具體實施例之可劃分引擎及其構件的概要圖，其包括針對一多核心處理器的區段排程器及暫存器檔案、通用互連、及片段記憶體次系統。如圖 1B 所繪示，顯示了四個片段 101 至 104。分段階層在每一快取階層(如 L1 快取、L2 快取、及載入儲存緩衝器)為相同。資料可經由記憶體通用互連 110a 在每一 L1 快取、每一 L2 快取、及每一載入儲存緩衝器之間交換。

記憶體通用互連包含一路由矩陣，其允許複數個核心(如位址計算及執行單元 121-124)存取資料，資料可儲存於片段快取階層(如 L1 快取、載入儲存緩衝器及 L2 快取)中的任一點。圖 1B 也繪示每一片段 101 至 104 可由位址計算及執行單元 121 至 124 經由記憶體通用互連 110a 而存取的方式。

執行通用互連 110b 類似地包含一路由矩陣，允許複數個核心(如位址計算及執行單元 121 至 124)存取可能儲存於任何區段暫存器檔案的資料。因此，核心具有經由記憶體通用互連 110a 或執行通用互連 110b 之對儲存於任何片段中之資料及對儲存於任何區段中之資料的存取。

圖 1B 更顯示一通用前端抓取及排程器 150，其具有對整體機器的觀察且其管理暫存器檔案區段及分段記憶體次系統的使用。位址產生包含針對片段定義的基礎。通用前端抓取及排程器係藉由分配指令序列至每一區段的劃分排程器而作用。共同劃分排程器接著配送在位址計算及執行單元 121 至 124 上用以執行的那些指令序列。

此外，應注意，圖 1A 所示的可劃分引擎可以階層方式嵌套。在此一具體實施例中，第一等級可劃分引擎係包括一本地前端抓取及排程器及與其連接之多個次級可劃分引擎。

圖 2 顯示根據本發明一具體實施例之概要圖，其繪示在圖 1A 及 1B 的討論中所描述之互連 30 的額外特徵及複數個本地互連 40 至 42。圖 2 結構描述互連結構的一編排模式。圖 2 顯示連接至對應之複數個用戶的複數個資源。資源為可劃分引擎之每一者的資料儲存資源(如暫存器檔案、載入儲存緩衝器、L1 快取及 L2 快取)。用戶為可劃分引擎之每一者的執行單元及位址計算單元。圖 2 更顯示複數個編排器 21-23。

如上述，引擎 11 至 14 之每一者的資源之間的通訊係由一互連結構所提供。舉例來說，在圖 2 的具體實施例中，互連結構 30 為一專屬點對點匯流排。在圖 2 的具體實施例中，有六個跨越每一引擎之資源的匯流排。只有一個用戶/資源對可在每一循環利用六個匯流排的其中之一。用戶/資源對係經由圖 10 的 OR-AND 及臨界偵測邏輯而彼此競爭以使用六個匯流排。然而，可使用保留加法器及臨界限制或程序而達成針對一共享的多點匯流排組態之相同編排，如將進一步描述於圖 9 的討論中。

編排器 21 至 23 包含指示資源的路由至用戶的控制實體。舉例來說，在一具體實施例中，編排器可為一線程排程器，其安排預備好供執行的資源經由互連轉移至用戶。編排器(如線程排程器)識別正確的資源、保留必要的匯流排、並造成該資源轉移至一所選的用戶。在此方式中，編排器監視

指令的準備就緒並選擇將用以執行指令的執行單元。此資訊係用以編排橫越互連之資源轉移至所選的執行單元(如所選的用戶)，其藉由使用圖 9 或圖 10 所描述之保留及分配邏輯而競爭在互連的請求。在此方式中，用戶本身的執行單元係視作需要由編排器使用如針對互連所描述之類似資源保留及分配方法而競爭的資源。在執行單元中係藉由競爭請求而保留及分配，請求係來自使用圖 9 或圖 10 之保留及分配邏輯之任一的所有編排器。

互連包含一路由矩陣，其允許複數個資源用戶(在此情況中為複數個核心(如位址計算及執行單元 121 至 124))存取資源(在此情況中為資料，其可儲存於片段快取階層(如 L1 快取、載入儲存緩衝器及 L2 快取)中的任一點)。核心可類似地存取可能儲存於任何區段暫存器檔案的資料。因此，核心具有經由互連結構 30 之對儲存於任何片段中之資料以及對儲存於任何區段中之資料的存取。在一具體實施例中，互連結構包含兩個結構：記憶體互連 110a 及執行互連 110b，如前文中在圖 1B 的討論中所顯示及描述。

圖 2 亦顯示複數個本地互連 40 至 42。本地互連 40 至 42 包含一路由矩陣，其允許來自鄰近可劃分引擎的資源用戶快速地存取緊鄰之可劃分引擎的資源。舉例來說，一核心可使用本地互連 40 以快速地存取鄰近可劃分引擎的資源(如暫存器檔案、載入儲存緩衝器等)。

因此，互連結構本身包含必須由每一可劃分引擎之每一核心所共享的一資源。互連結構 30 及本地互連結構 40 至 42 係實施允許來自任何可劃分引擎之核心存取任何其他可

劃分引擎之資源的一互連結構。此互連結構包含傳輸線，其在互連結構的情況中係跨越積體電路裝置的所有可劃分引擎，且在本地互連結構的情況中係跨越在積體電路裝置的引擎之間。

本發明具體實施例係執行一非集中存取程序供使用互連及本地互連。通用匯流排及本地匯流排的有限數量包含必須有效率地由編排器共享的資源。此外，非集中存取程序係由編排器使用以有效率地共享有限數量的埠，其提供對每一可劃分引擎之資源的讀取/寫入存取。在一具體實施例中，非集中存取程序係由編排器保留進入到所需資源的一匯流排(如本地互連匯流排或互連匯流排)及一埠而實施。舉例來說，編排器 21 需保留一互連及一埠以供用戶 1 存取資源 3，而編排器 22 需保留一互連及埠以供用戶存取資源 2。

圖 3 顯示根據本發明一具體實施例之包含執行對一競爭資源之有效率存取的一資源保留機制的構件。如圖 3 所示，三個保留加法器 301 至 303 係顯示為耦合至臨界限制器 311 至 313，其控制對四個埠之每一者對三個資源之每一者的存取。每一加法器輸出總和(若無刪除的話)亦作為針對每一存取的埠選擇器，使得成功的每一請求可使用由在該請求加法器之輸出的總和所指示之埠數目。應注意，如圖 3 中所示，每一個所繪示之加法器的總和亦針對非刪除的對應請求指派埠數目。

應注意，此埠分配及保留問題可類似圖 7 之匯流排區段分配表格而描述，因此其實施邏輯也可類似於圖 9，其中在此情況中的每一區段係反映一暫存器檔案區段而非一匯流

排區段。以與此情況相同的類推，試圖存取多個暫存器檔案區段的指令僅可在其可保留所有其暫存器區段請求時成功，且若刪除針對該指令之任一暫存器區段存取時將失敗，類似於圖 7 中匯流排區段的描述。

本發明具體實施例執行非集中存取程序供使用互連及本地互連。請求、存取及控制可由多個非集中抓取器、傳送器、編排器、或代理器針對共享互連、資源或用戶而初始化。那些非集中請求、存取及控制係競爭共享資源，其係根據那些共享資源的拓撲及結構使用本發明所述之方法及邏輯實施的變化。舉例來說，引擎及其讀取/寫入埠的資源需由核心有效率地共享。此外，通用匯流排及本地匯流排的有限數量包含需被有效率共享的資源。在圖 3 的具體實施例中，非集中存取程序係經由保留加法器及臨界限制器而實施。在一具體實施例中，在每一競爭資源，保留加法器樹狀結構及臨界限制器控制對該競爭資源的存取。如本文中所使用，競爭資源一詞係指載入儲存緩衝器、記憶體/快取片段、暫存器檔案區段或 L2 快取、通用匯流排保留、或本地匯流排保留的讀取寫入埠。

保留加法器及臨界限制器控制對每一競爭資源的存取。如上述，為存取一資源，核心需保留必要的匯流排及保留必要的埠。在每一循環期間，編排器試圖保留執行其暫停指令所需的資源。舉例來說，針對安排圖 3 所示之指令 I1 的編排器，該編排器將設定一旗標或一位元於其所需資源的保留加法器中。在此情況中，位元係設定於暫存器檔案 1 及暫存器檔案 3 中。其他編排器將類似地設定位元於其所需資源的保留加法器中。舉例來說，針對指令 I2 之一不同的編

排器係針對暫存器檔案 2 設定兩個位元。當編排器請求其所需資源，保留加法器係總和請求直到其達到臨界限制器。在圖 4 的具體實施例中，對每一資源有四個埠。因此，保留加法器將從保留請求接受旗標直到四個埠皆保留。沒有其他旗標將被接受。

編排器將不接收確認以執行其指令，除非其所有需執行指令的旗標皆被設定。因此，若針對必要匯流排的旗標被設定且針對必要讀取寫入埠的旗標被設定，則編排器將接收確認以執行指令。若針對任何旗標接收到一刪除信號，則刪除針對該編排器之請求的所有旗標，且請求係佇列直到下一循環。

在此方式中，每一編排器在一循環週期基礎上針對資源彼此競爭。刪除的請求係佇列且在下一循環中給定優先性。這確保一特定核心未對大量的循環鎖定資源存取。應注意，在建議實施中的資源係自動地指派資源，例如若請求成功獲得一資源(如其未由加法器及臨界邏輯所刪除)，則對應該請求的加法器總和輸出表示指派給該請求的資源數目，因此完成資源指派而不需編排器的任何進一步參與。此保留及分配加法器及臨界限制器在分散式方式中公平地平衡對競爭資源的存取(例如不需要請求器/編排器主動地參與任何集中的仲裁)。每一遠端編排器傳送其請求至共享資源，成功的那些資源將被自動地授予資源/匯流排。

圖 4 顯示根據本發明一具體實施例之互連及進入至記憶體片段之埠。如圖 4 所繪示，每一記憶體片段係顯示為具有四個讀取寫入埠，其提供對載入儲存緩衝器、L1 快取、

及 L2 快取的讀取/寫入存取。載入儲存緩衝器包括複數個項目且 L1 快取包含複數個路徑。

如上述，本發明具體實施例執行非集中存取程序供使用互連及本地互連。通用匯流排及本地匯流排的有限數量包含必須由核心有效率共享的資源。因此，保留加法器及臨界限制器控制對每一競爭資源的存取，在此情況中為到每一片段的埠。如上述，為存取資源，核心需要保留必要的匯流排並保留必要的埠。

圖 5 顯示根據本發明一具體實施例之互連及進入至記憶體區段之埠。如圖 5 所示，每一區段係顯示為具有 4 個讀取寫入埠，其提供對運算元/結果緩衝器、線程暫存器檔案、及共同劃分或排程器的讀取/寫入存取。圖 5 的具體實施例係顯示為包括一共同劃分或排程器於每一區段中。在此具體實施例中，共同劃分排程器係組態以與圖 1B 所示之通用前端抓取及排程器協力地作用。

用以使用互連及本地互連的非集中存取程序係利用保留加法器及臨界限制器控制對每一競爭資源的存取，在此情況中為到每一區段的埠。如上述，為存取資源，核心需要保留必要的匯流排並保留必要的埠。

圖 6 顯示繪示根據本發明一具體實施例之區段互連 601 的一圖式。如圖 6 所示，互連 601 係顯示為連接資源 1-4 至用戶 1-4。互連 601 亦顯示為包含區段 1、2、及 3。

圖 6 顯示抓取模式互連結構的一範例。在圖 6 的具體實

施例中，沒有編排器。在此具體實施例中，當用戶試圖抓取必要的資源以支援消耗(如執行單元)時，資源由用戶競爭。用戶傳送必要的抓取請求至保留加法器及臨界限制器。

互連結構包含複數個通用區段匯流排。本地互連結構包含複數個本地連接的引擎至引擎匯流排。因此，為平衡在效能及製造兩者中的成本，具有有限數量的通用匯流排及有限數量的本地匯流排。在圖 6 的具體實施例中，顯示了四個通用區段匯流排。

在一具體實施例中，通用匯流排可分區段為 3 個部分。分區段允許根據通用存取的距離調整通用匯流排的整體長度。舉例來說，由用戶 1 對資源 4 的存取係橫跨整體匯流排，因此未分區段。然而，由用戶 1 對資源 3 的存取未橫跨整體匯流排，因此通用匯流排可在資源 3 及資源 4 之間分區段。

在圖 6 的具體實施例中，互連 601 係顯示為具有 4 個匯流排。分區段可透過例如三狀態緩衝器而實施。分區段導致匯流排的更快且更省電的傳輸特性。在圖 6 的具體實施例中，每一匯流排包括一方向性三狀態緩衝器(如緩衝器 602)及雙向三狀態緩衝器(如緩衝器 603)。在圖 6 中，雙向三狀態緩衝器係以陰影表示。緩衝器係致能互連被分段以改善其信號傳輸特性。這些區段也包含必須由資源用戶競爭及分配的資源。此程序係描述於以下的圖 7。

圖 7 顯示一表格，其描述根據本發明一具體實施例之針對互連 601 之區段之請求所競爭及分配的方式。圖 7 表格的左手側顯示當請求於循環內被接收時如何排序。在此情況

中，顯示了八個請求。當來自一資源用戶的請求想要保留一區段，該用戶設定一於所請求區段的保留表格中。舉例來說，針對請求 1，用戶 1 想要保留區段 1 及區段 2 以存取資源 3。因此，用戶 1 設定一旗標或一位元於區段 1 及區段 2 的請求行中，而區段 3 的行係保持為零。在此方式中，請求係加入於行內。請求係分配直到其超過通用匯流排的數量，在此情況中為 4 個。當請求超過通用匯流排的數量，其將刪除。這由因為超過限制而已刪除之請求數目 6 及請求數目 7 所顯示。

圖 8 顯示一表格，其描述根據本發明一具體實施例之處理針對點對點匯流排之請求的方式。與圖 7 的表格相反，圖 8 的表格顯示只有一用戶及只有一資源如何使用點對點匯流排(如圖 2 所示的互連)。請求來自想要經由點對點匯流排路由資源的多個編排器。在此情況中，點對點匯流排顯示可能的用戶資源對之數目(如從左至右的六行)以及從頂至底之請求 1-8 的數目。因為只有一資源用戶對可在任何給定時間使用一匯流排，在所有旗標因超過限制而刪除前，行可僅具有一請求旗標。因此，在每一行中，允許第一請求，而所有後來的請求係因超過限制而刪除。由於有六個通用點對點匯流排，有六個行可在每一循環中容納六個不同的請求。

圖 9 顯示根據本發明一具體實施例之執行圖 7 之表格之功能的範例邏輯實施的圖式。如上述，圖 7 的表格描述根據本發明具體實施例之競爭及分配針對互連 601 之區段之請求的方式。特別地，圖 9 顯示用以分配相關於來自圖 7 表格之匯流排區段 2 之行的邏輯。

圖 9 的具體實施例顯示複數個平行加法器 901-905。若超過限制則刪除兩個請求。如上述，有 4 個匯流排可用以實施區段 2。可處理並允許前四個請求，因為即使他們都被標誌旗標(藉由以邏輯一標示請求)，其將不會超過限制。需要檢查剩下的請求是否將超過限制。這由平行加法器 901-905 所完成。在前三列之後的每一加法器係加入其本身及所有先前的列，並核對限制。若加法器超過限制，則刪除請求，如所示。加法器總和輸出也決定哪一特定的匯流排區段係分配至每一請求。在圖 9 的具體實施例中，這是藉由匯流排區段數目，如所示。

圖 10 顯示根據本發明一具體實施例之處理針對點對點匯流排之請求之方式之功能的範例邏輯實施的圖式。圖 8 的表格顯示只有一用戶及只有一資源如何可使用點對點匯流排。特別地，圖 10 顯示用以分配相關於來自圖 8 表格之匯流排行 2-4 之行的邏輯。

圖 10 的具體實施例顯示耦合至 AND 閘的複數個多輸入 OR 閘，如所示。如前述，一用戶以及僅一資源可使用一點對點匯流排。因為只有一資源/用戶對可在任何給定時間使用一匯流排，在所有後續請求因超過限制而刪除前，行可僅具有一請求旗標。因此，在每一行中，允許第一請求，而所有後續請求因為超過限制而刪除。在圖 10 的具體實施例中，行的每一列係經由 OR 操作而與行的所有先前列邏輯地結合，且接著經由 AND 操作而與其本身邏輯地結合。因此，若任何先前列保留行，則刪除所有後續請求，如所示。

圖 11 顯示根據本發明一具體實施例之互連 1101 的圖

式。互連 1101 包含五個共享互連結構，其由每一傳送器及每一接收器所共享。

圖 11 的具體實施例顯示傳送模式互連結構的一範例。舉例來說，傳送器包含引擎的執行單元。接收器包含引擎的記憶體片段及暫存器區段。在此模式中，傳送器發佈必要的請求至保留加法器及臨界限制器以保留用以執行其轉移的資源。這些資源包括進入接收器的匯流排及互連 1101 的複數個共享匯流排。

圖 12 顯示一表格，其描述根據本發明一具體實施例之圖 11 之傳送器模式互連結構作用的方式。表格顯示從所有傳送器所接收之請求。表格的右手側顯示互連分配。由於互連 1101 包含五個共享匯流排，前五個請求將被允許，且任何進一步的請求因超過限制而刪除。因此，請求 1、請求 3、請求 4、請求 5 及請求 6 係被允許。然而，請求 7 因已超過限制而刪除。

圖 13 顯示根據本發明一具體實施例之範例邏輯實施的圖式，其執行處理針對共享匯流排互連結構之請求之方式的功能。

圖 13 顯示互連匯流排的分配如何由加法器 901-905 所處理。此邏輯實施圖 12 的表格。當接收請求，設定對應的旗標。加法器加入其個別旗標與所有先前的旗標。旗標將與其匯流排數目一起由加法器所允許，只要其未超過限制，在此情況中為五個。如上述，超過限制的任何請求將刪除。

應注意，互連的傳送器模式及抓取模式可使用共同互連結構及共同競爭機制而被同時地支援。這由圖 13 之圖式對圖 9 之圖式的相似性所顯示。

應注意，對本發明之不同模式的通訊(傳送器、抓取、編排器等)及不同的互連拓撲(點對點匯流排、多重匯流排、及區段匯流排等)的描述不應解釋為適用於本發明的唯一通訊模式或唯一互連拓撲。相反地，熟此技藝者可輕易地混合及匹配本發明不同的競爭、保留及分配技術與任何通訊模式或匯流排拓撲。

更應注意，所描述之本發明具體實施例係呈現沿著資源的互連。這應理解為一般化的描述，其表示顯示更廣泛的可能性以實施本發明，但應注意本發明所使用之互連的意義並不限於在不同核心或計算引擎之間的互連或是暫存器檔案或記憶體片段之間的互連，而是也指承載請求至資源的控制互連以及從結構承載資料的實體互連(亦即，暫存器檔案埠、記憶體埠、陣列解碼器匯流排等)。舉例來說，此廣泛的含意係描述於圖 3，其顯示只有當埠從每一暫存器檔案出來的互連。

圖 14 顯示根據本發明一具體實施例之範例微處理器管線 1400 的圖式。微處理器管線 1400 包含一抓取模組 1401，其執行程序之功能以識別及擷取包含一執行之指令，如前述。在圖 14 的具體實施例中，抓取模組係接著一解碼模組 1402、分配模組 1403、配送模組 1404、執行模組 1405、及引退模組 1406。應注意，微處理器管線 1400 只是管線的一範例，其執行上述之本發明具體實施例的功能。熟此技藝者

將了解到其他微處理器管線可實施為包含上述解碼模組之功能。

為解釋目的，前文之描述係指特定具體實施例，其無意為詳盡或限制本發明。在符合上述教示下可能有許多修改及變化。具體實施例係選擇及描述以最佳地解釋本發明的原理及其實際應用，以致能其他熟此技藝者最佳地使用本發明及其具有可適合於特定使用之各種修改的各種具體實施例。

【圖式簡單說明】

本發明係經由範例而非經由限制的方式而描述，在所附隨圖式之各圖中，類似的元件符號係指類似的元件。

圖 1A 顯示通用前端產生編碼區塊及遺傳向量以支援編碼序列於其個別引擎上之執行的方式的概要；

圖 1B 顯示根據本發明一具體實施例之引擎及其構件的概要圖，其包括針對一多核心處理器的區段排程器及暫存器檔案、互連、及片段記憶體次系統；

圖 2 顯示根據本發明一具體實施例之概要圖，其繪示在圖 1A 及 1B 的討論中所描述之互連的額外特徵及複數個本地互連；

圖 3 顯示根據本發明一具體實施例之包含執行對一競爭資源之有效率存取的一資源保留機制的構件；

圖 4 顯示根據本發明一具體實施例之互連及進入至記憶體片段之埠；

圖 5 顯示根據本發明一具體實施例之互連及進入至記憶體區段之埠；

圖 6 顯示繪示根據本發明一具體實施例之區段互連的

一圖式；

圖 7 顯示一表格，其描述根據本發明一具體實施例之針對互連之區段之請求所競爭及分配的方式；

圖 8 顯示一表格，其描述根據本發明一具體實施例之處理針對點對點匯流排之請求的方式；

圖 9 顯示根據本發明一具體實施例之執行圖 7 之表格之功能的範例邏輯實施的圖式；

圖 10 顯示根據本發明一具體實施例之處理針對點對點匯流排之請求之方式之功能的範例邏輯實施的圖式；

圖 11 顯示根據本發明一具體實施例之互連的圖式；

圖 12 顯示一表格，其描述根據本發明一具體實施例之圖 11 之傳送器模式互連結構作用的方式；

圖 13 顯示根據本發明一具體實施例之範例邏輯實施的圖式，其執行處理針對共享匯流排互連結構之請求之方式的功能；以及

圖 14 顯示根據本發明一具體實施例之範例微處理器管線的圖式。

【主要元件符號說明】

10	通用前端抓取及排程器
11-14	引擎
20-23	編碼序列、編排器
30	通用互連結構
40-42	本地互連
101-104	片段
110a	記憶體通用互連

110b	執行通用互連
121-124	位址計算及執行單元
150	通用前端抓取及排程器
301-303	保留加法器
311-313	臨界限制器
601	互連
602	緩衝器
603	緩衝器
901-905	加法器
1101	互連
1400	微處理器管線
1401	抓取模組
1402	解碼模組
1403	分配模組
1404	配送模組
1405	執行模組
1406	引退模組

七、申請專利範圍：

1. 一種具有耦合至一電腦可讀記憶體之一微處理器的電腦系統，其中該微處理器包含：

複數個資源，具有用以支援多個編碼序列之執行的資料；

複數個可劃分引擎，用以實施該多個編碼序列之執行；

複數個資源用戶於該複數個可劃分引擎之每一個內；
一互連結構，用以耦合該複數個資源用戶與該複數個資源以存取該資料及執行該多個編碼序列，其中該複數個資源用戶經由該互連結構之一循環週期利用而存取該複數個資源，且其中該複數個資源用戶包含該複數個可劃分引擎的執行單元，其中該互連結構包含一路由矩陣的傳輸線，其允許該複數個資源用戶存取來自該複數個資源中任意儲存位置的資料，且允許該複數個資源用戶存取緊鄰之可劃分引擎的資源；以及

一加法器結構，總和用於對該複數個資源之存取的請求以及根據該加法器結構輸出總和而特別地針對每一成功的請求指派一獨特埠。

2. 如請求項 1 所述之電腦系統，其中該資源包含記憶體片段。

3. 如請求項 1 所述之電腦系統，其中該資源包含暫存器檔案區段。

4. 如請求項 1 所述之電腦系統，其中該互連結構包含複數個點對點匯流排，其中該資源用戶經由該循環

週期利用存取該資源。

5. 如請求項 1 所述之電腦系統，其中該互連結構包含複數個區段匯流排，其中該資源用戶經由該循環週期利用存取該資源。

6. 如請求項 1 所述之電腦系統，其中該互連結構包含一記憶體互連結構及一執行互連結構。

7. 如請求項 1 所述之電腦系統，其中該系統更包括複數個本地互連結構，其致能鄰近引擎直接地從鄰近資源存取資料。

8. 如請求項 1 所述之電腦系統，其中該互連結構包含一抓取模式互連結構。

9. 如請求項 1 所述之電腦系統，其中該互連結構包含一傳送模式互連結構。

10. 如請求項 1 所述之電腦系統，其中該互連結構包含一編排模式互連結構。

11. 如請求項 1 所述之電腦系統，其中：

該加法器結構更總和用於對該複數個資源之存取的請求以及根據該加法器結構輸出總和而特別地針對每一成功的請求指派一獨特匯流排。

12. 如請求項 1 所述之電腦系統，其中：

該加法器結構更總和用於對該複數個資源之存取的請求以及根據該加法器結構輸出總和而特別地針對每一成功的請求指派一獨特匯流排及該獨特埠以在一循環週期基礎上仲裁及分配複數個資源。

13. 如請求項 1 所述之電腦系統，其中：

該加法器結構更根據一抓取模式、或一傳送模式、或一編排模式而作用，以及總和用於對該複數個資源之存取的請求且根據該加法器結構輸出總和而特別地針對每一成功的請求指派一獨特匯流排及該獨特埠以在一循環週期基礎上仲裁及分配複數個資源，且根據該抓取模式、該傳送模式、或該編排模式。

14. 一種具有耦合至一電腦可讀記憶體之一微處理器的電腦系統，其中該微處理器包含：

複數個資源，具有用以支援多個編碼序列之執行的資料；

複數個可劃分引擎，用以實施該多個編碼序列之執行；

複數個資源用戶於該複數個可劃分引擎之每一個內；
一互連結構，用以耦合該複數個資源用戶與該複數個資源以存取該資料及執行該多個編碼序列，其中該複數個資源用戶經由該互連結構之一循環週期利用而存取該複數個資源，且其中該複數個資源用戶包含該複數個可劃分引擎的執行單元，其中該互連結構包含一路由矩陣的傳輸線，其允許該複數個資源用戶存取來自該複數個資源中任意儲存位置的資料，且允許該複數個資源用戶存取緊鄰之可劃分引擎的資源；以及

一加法器結構，總和用於對該複數個資源之存取的請求以及根據該加法器結構輸出總和而特別地針對每一成功的請求指派一獨特匯流排及一獨特埠以在一循環週期基礎上仲裁及分配複數個資源。

15. 一種具有耦合至一電腦可讀記憶體之一微處理器的電腦系統，其中該微處理器包含：

複數個資源，具有用以支援多個編碼序列之執行的資料；

複數個可劃分引擎，用以實施該多個編碼序列之執行；

複數個資源用戶於該複數個可劃分引擎之每一個內；
一互連結構，用以耦合該複數個資源用戶與該複數個資源以存取該資料及執行該多個編碼序列，其中該複數個資源用戶經由該互連結構之一循環週期利用而存取該複數個資源，且其中該複數個資源用戶包含該複數個可劃分引擎的執行單元，其中該互連結構包含一路由矩陣的傳輸線，其允許該複數個資源用戶存取來自該複數個資源中任意儲存位置的資料，且允許該複數個資源用戶存取緊鄰之可劃分引擎的資源；以及

一加法器結構，根據一抓取模式、或一傳送模式、或一編排模式而作用，以及總和用於對該複數個資源之存取的請求且根據該加法器結構輸出總和而特別地針對每一成功的請求指派一獨特匯流排及一獨特埠以在一循環週期基礎上仲裁及分配複數個資源，且根據該抓取模式、該傳送模式、或該編排模式。

八、圖式：

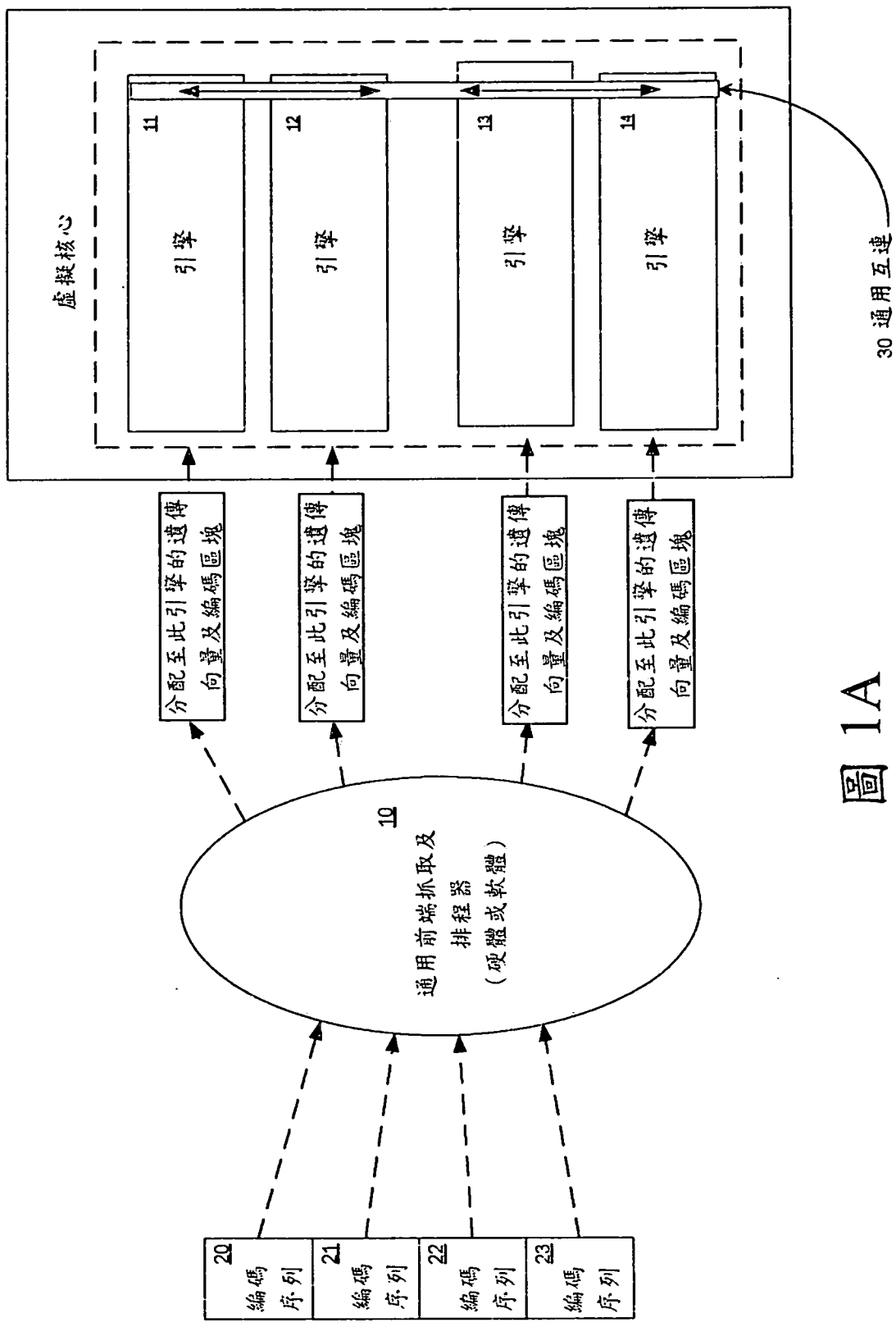


圖 1A

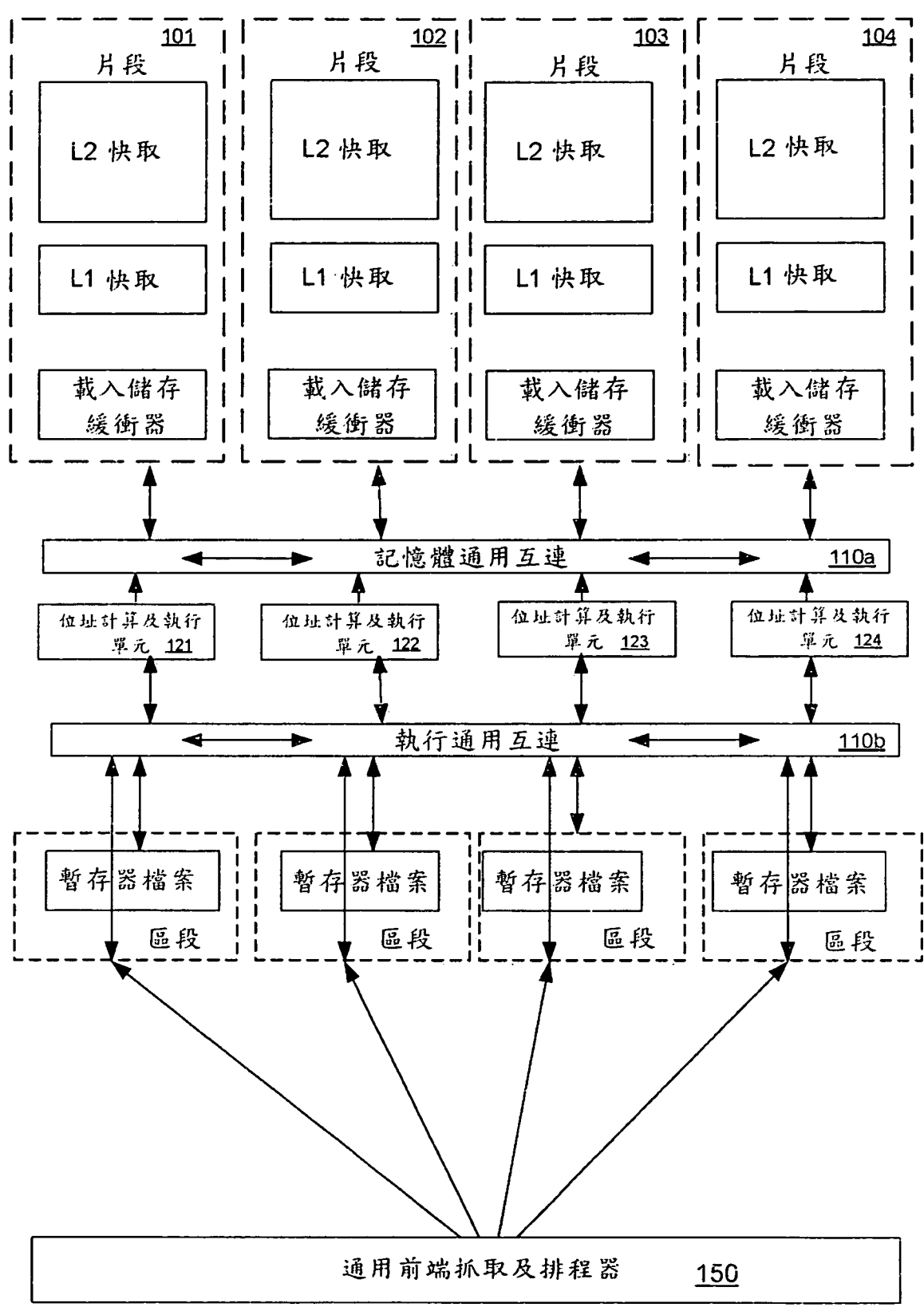


圖 1B

編排模式

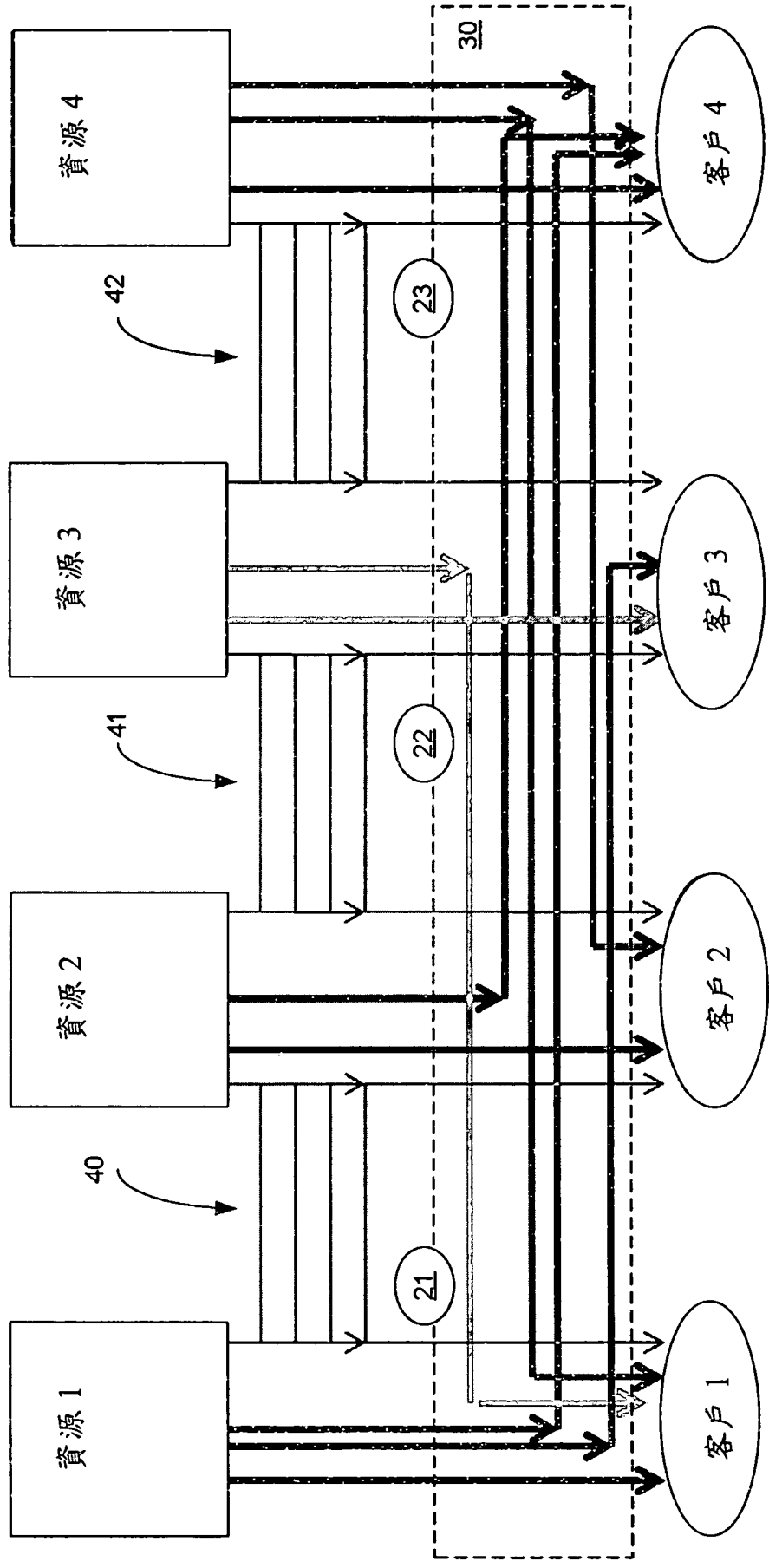


圖 2

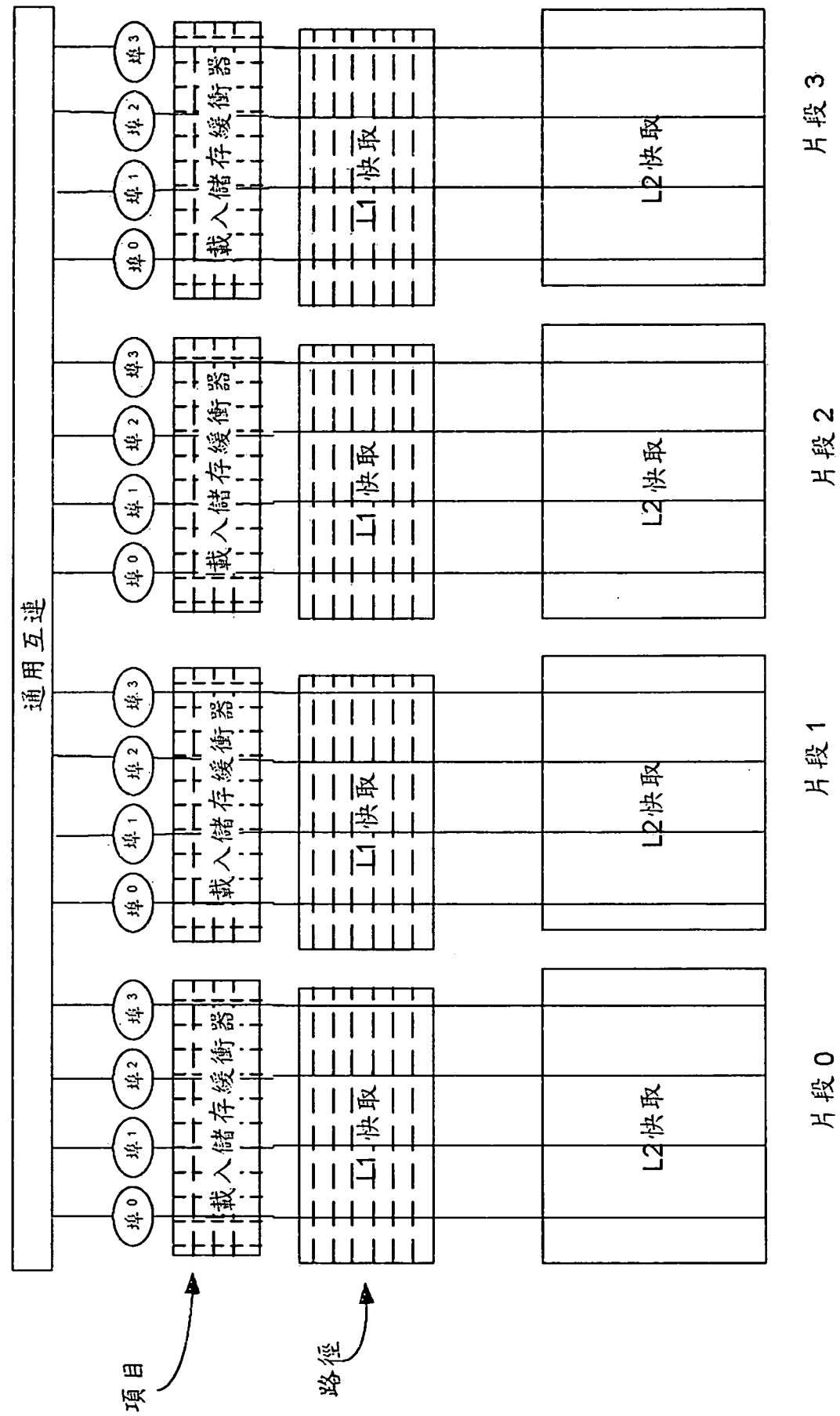


圖 4

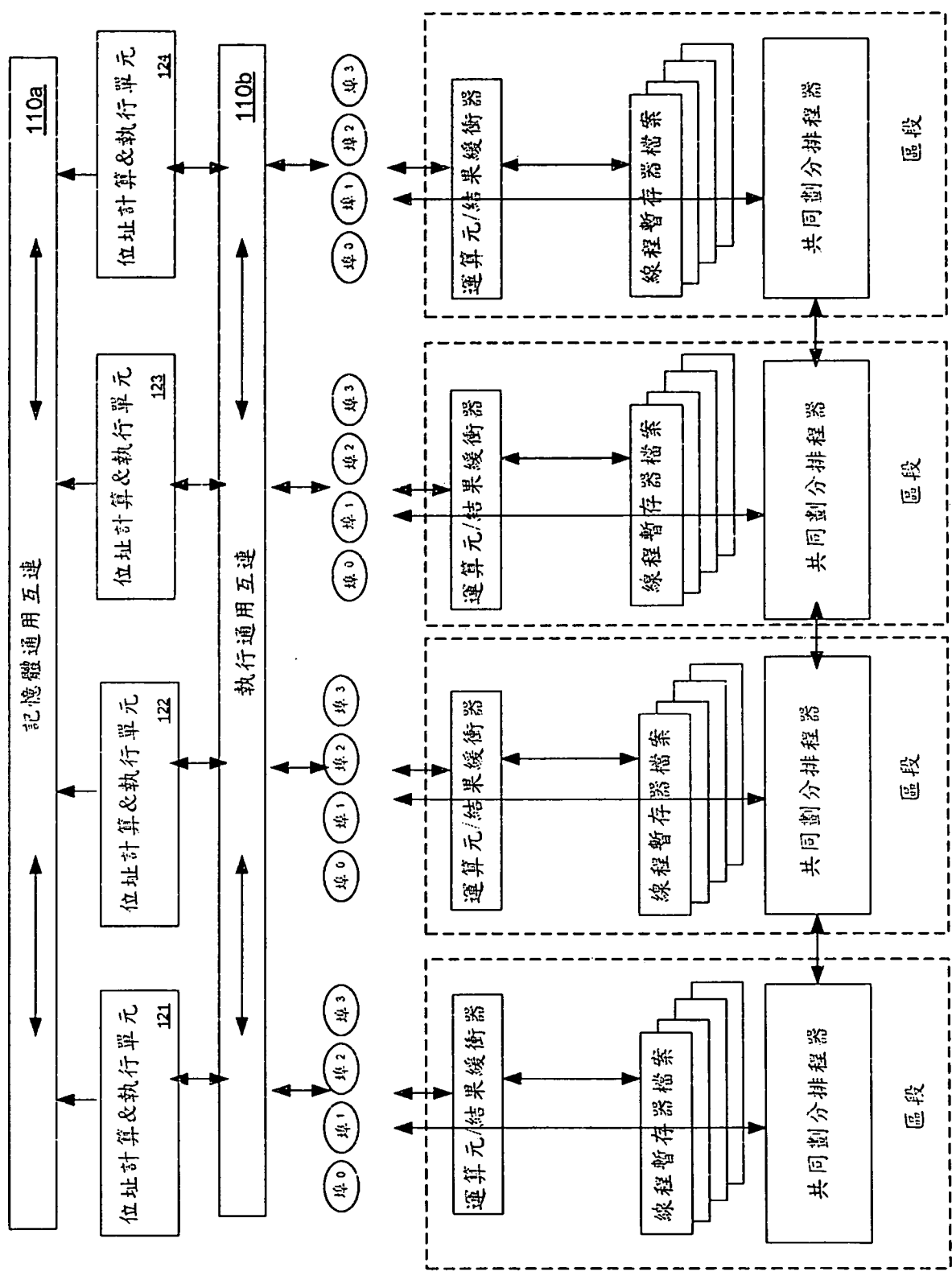


圖 5

抓取模式

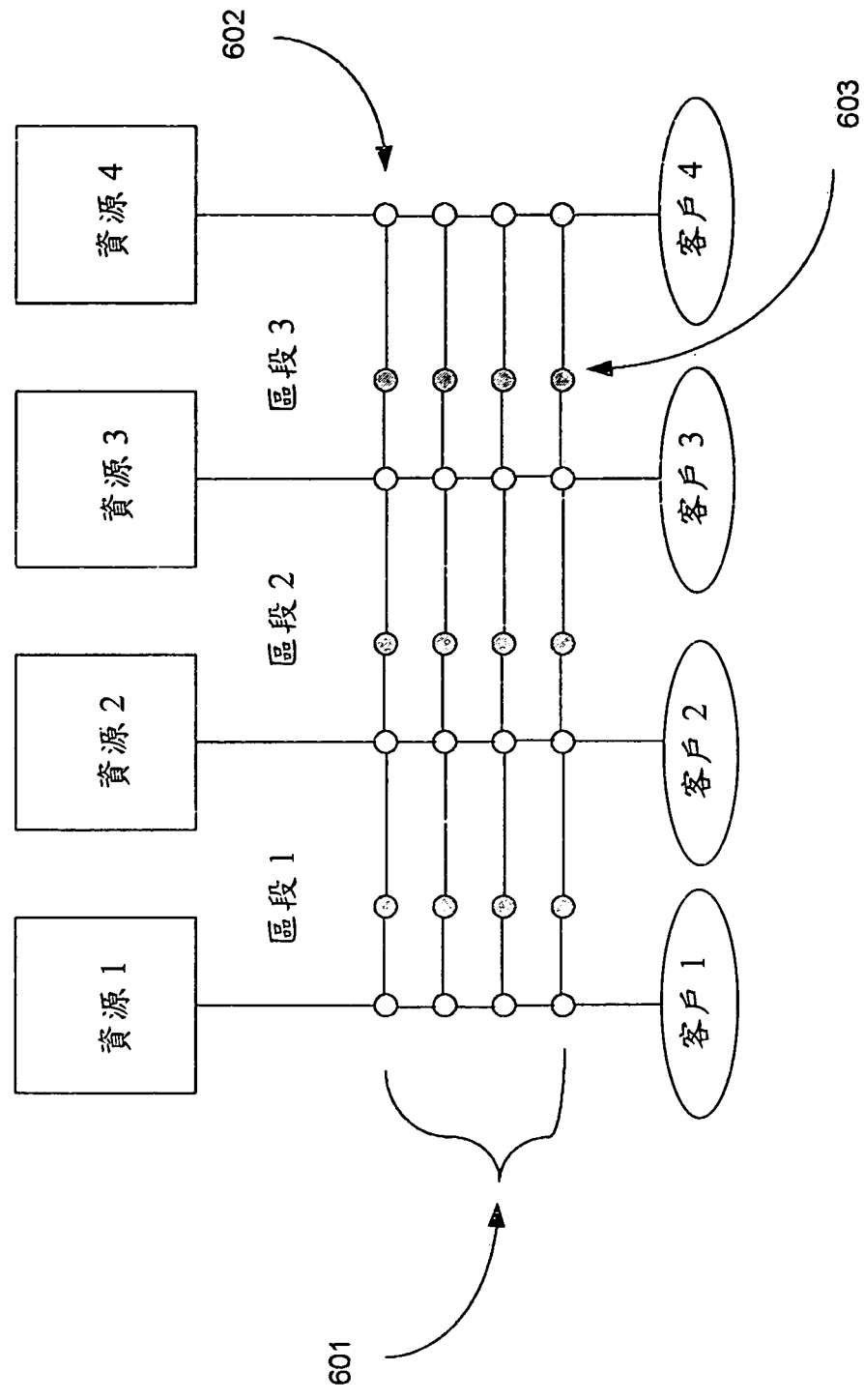


圖 6

	區段 1	區段 2	區段 3
請求 1	1	1	0
請求 2	0	0	1
請求 3	0	1	0
請求 4	0	1	1
請求 5	1	1	0
請求 6(刪除)	1	1(超過限制)	0
請求 7(刪除)	1	1(超過限制)	1
請求 8	0	0	1

圖 7

僅一對使用點對點匯流排

來自想要經由點對點
匯流排而路由之多個
編排器的請求

	1-3	3-1	4-1	1-4	2-4	4-2
請求 1	0	1	0	0	0	0
請求 2	0	0	1	0	0	1
請求 3	0	0	0	0	1	0
請求 4(刪除)	0	0	0	0	1(超過限制)	0
請求 5	1	0	0	0	0	0
請求 6(刪除)	0	1(超過限制)	0	1	0	0
請求 7(刪除)	1(超過限制)	1(超過限制)	0	0	0	0
請求 8	0	0	0	0	0	0

圖 8

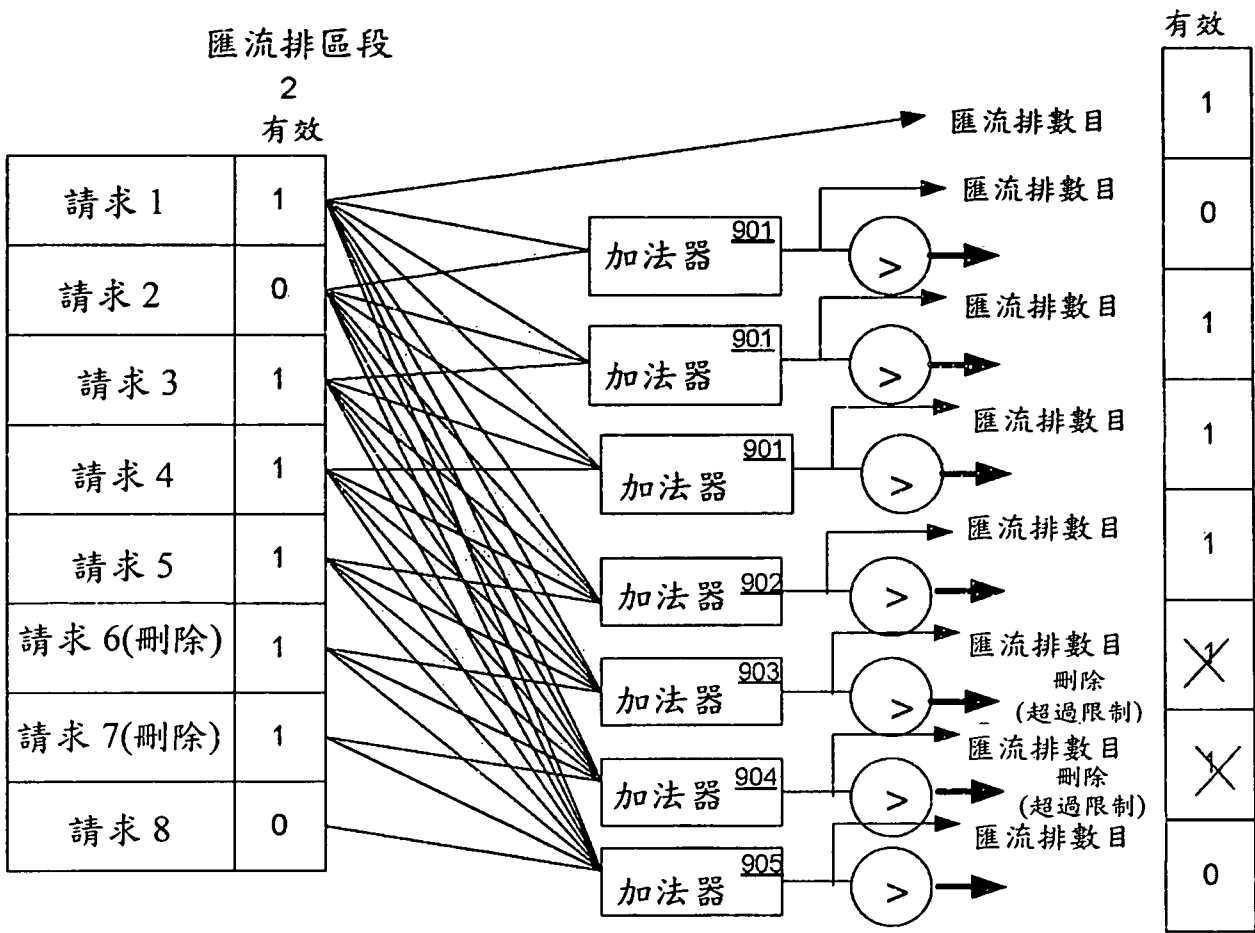


圖 9

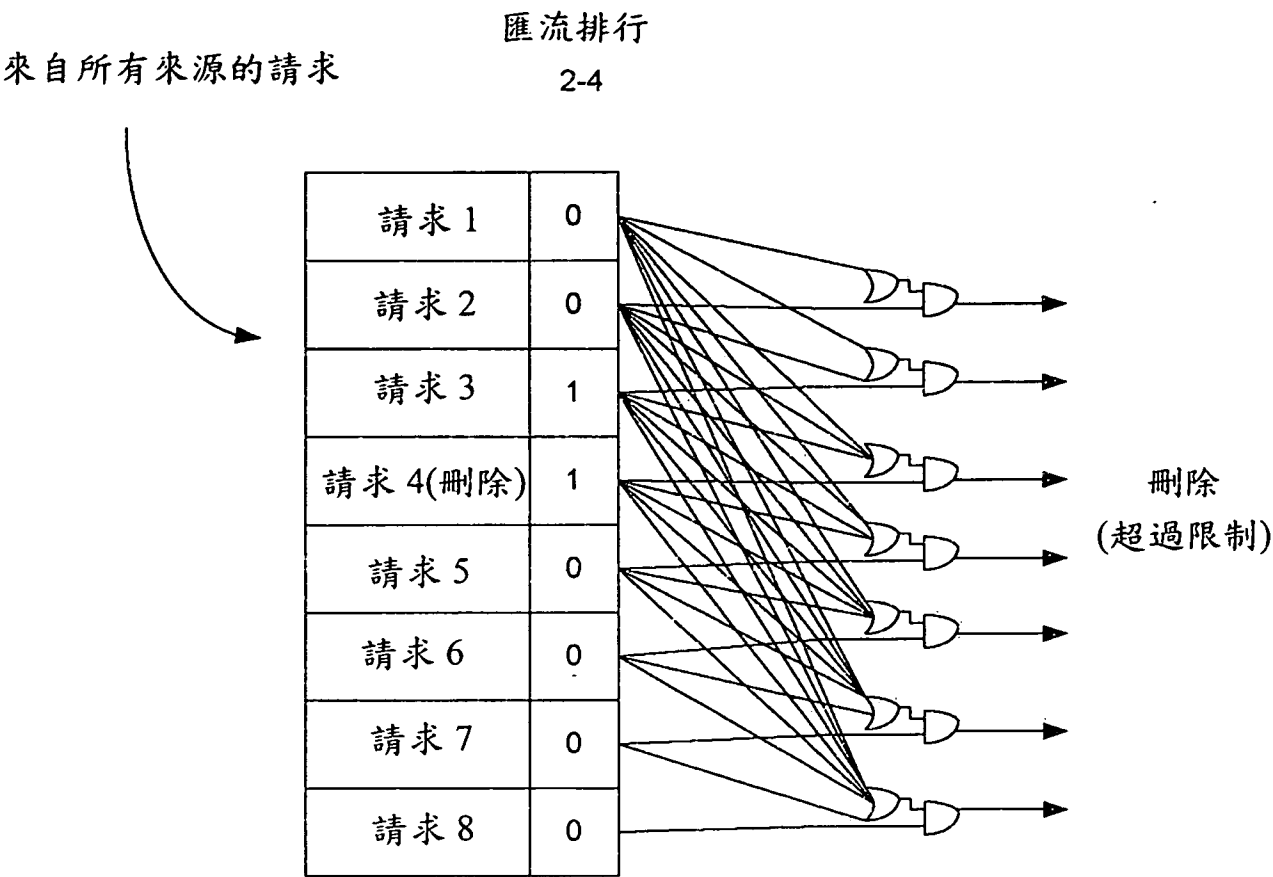
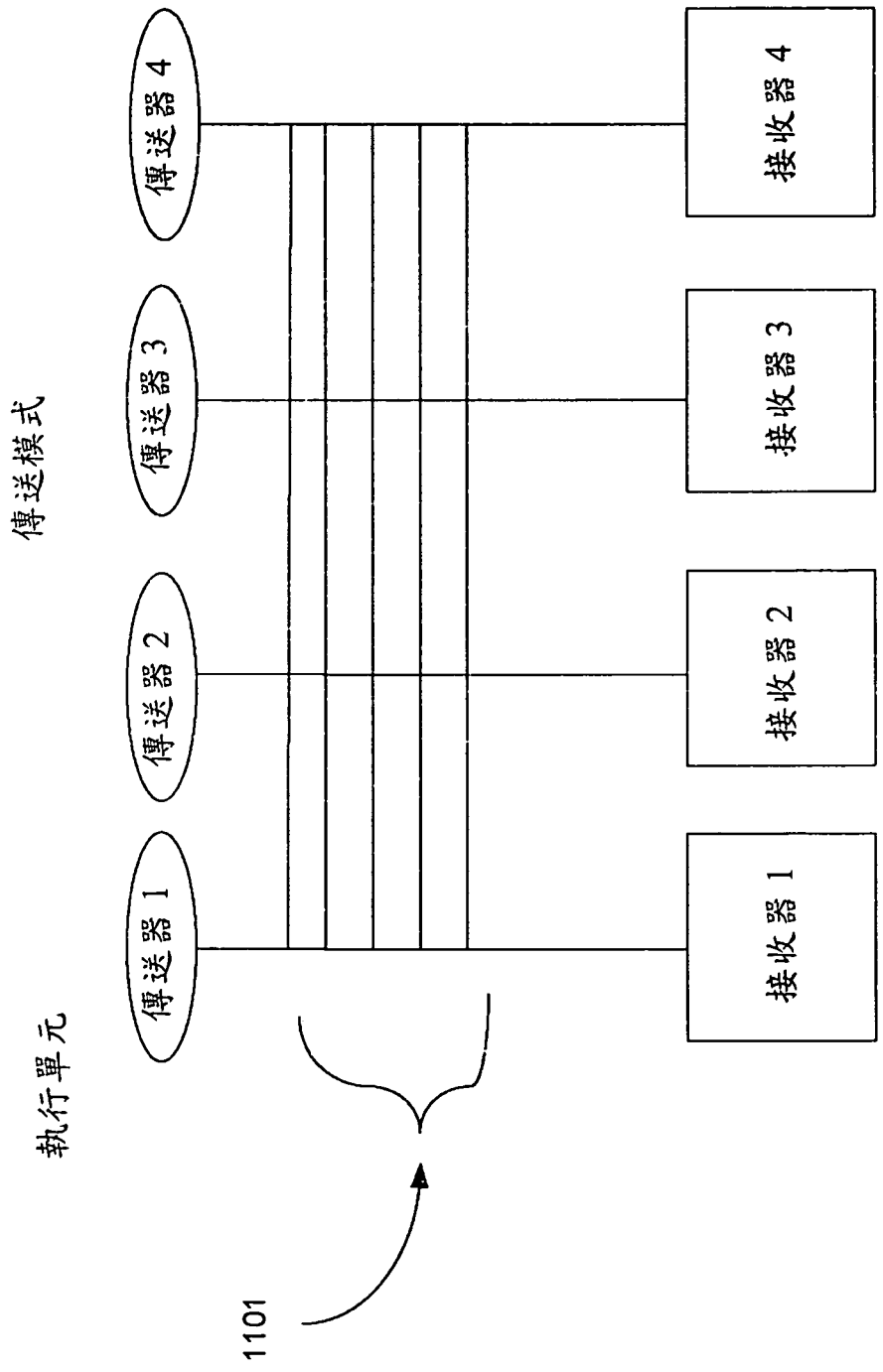


圖 10

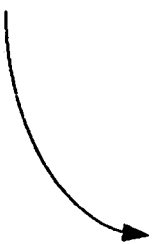


記憶體片段, 區段

需注意，傳送模式及抓取模式可使用相同互連及競爭機制而同時地被支援

圖 11

來自所有傳送器的請求



	互連分配
請求 1	1
請求 2	0
請求 3	1
請求 4	1
請求 5	1
請求 6	1
請求 7(刪除)	1(超過限制)
請求 8	0

圖 12

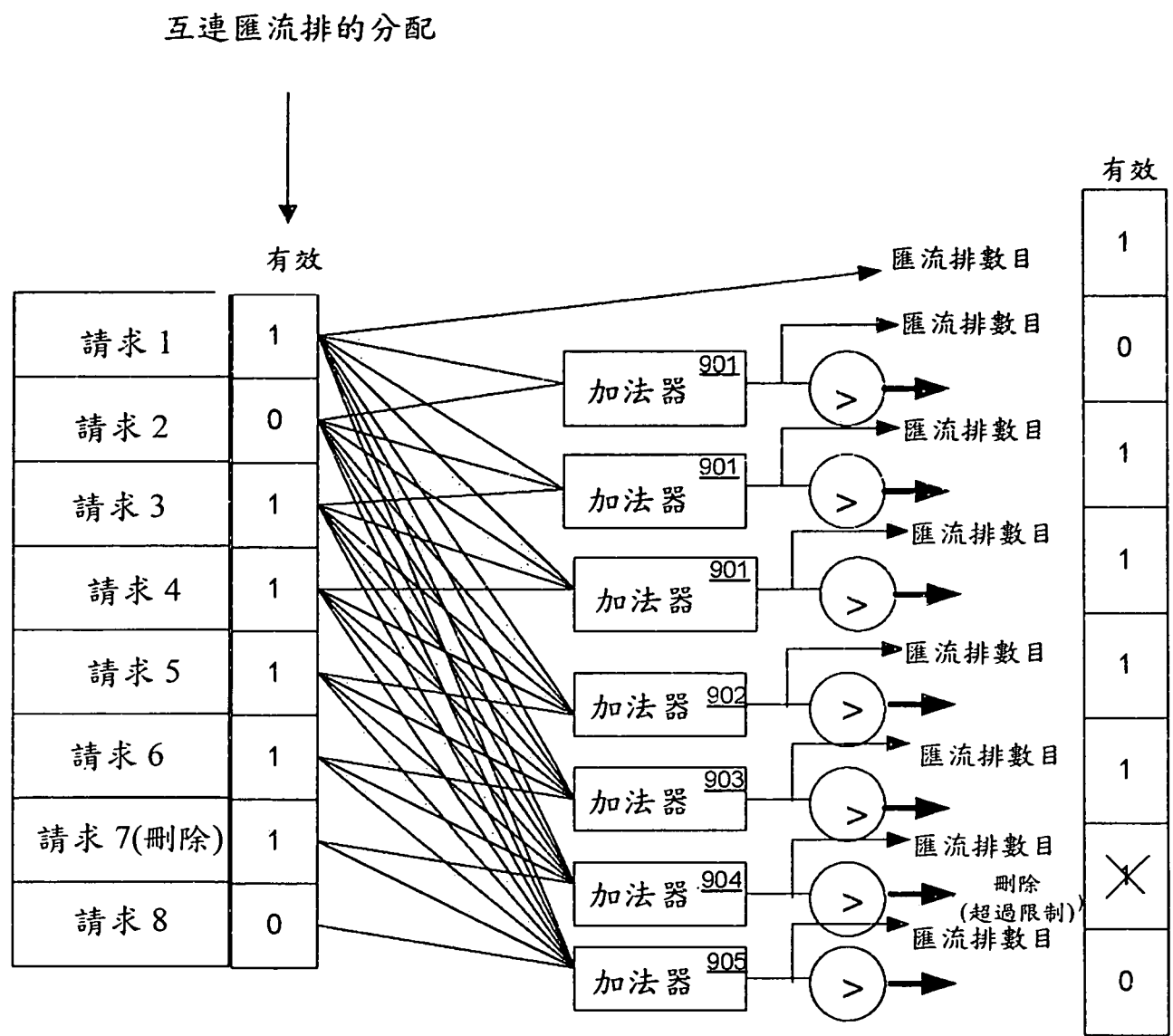


圖 13

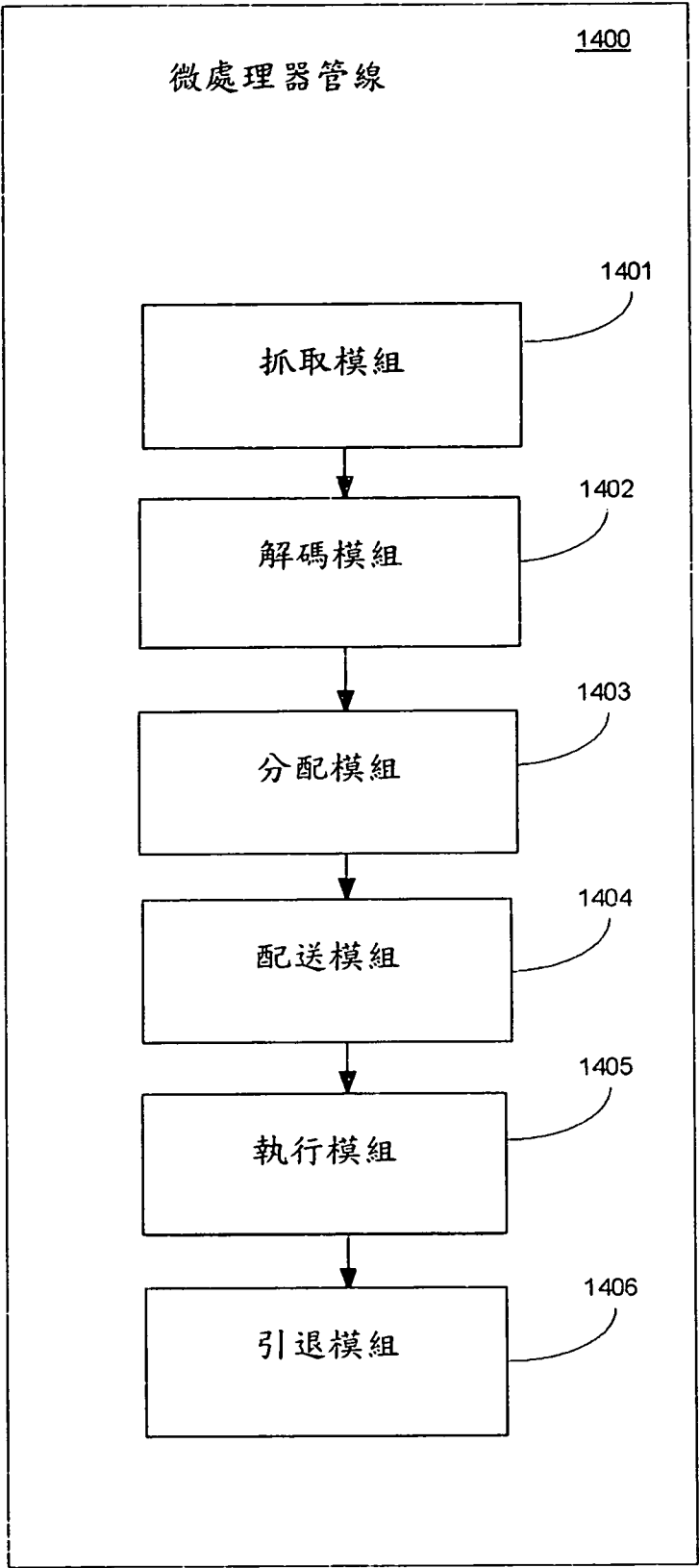


圖 14