



(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 I509519 B

(45)公告日：中華民國 104 (2015) 年 11 月 21 日

(21)申請案號：102117869

(22)申請日：中華民國 102 (2013) 年 05 月 21 日

(51)Int. Cl. : G06F9/50 (2006.01)

(30)優先權：2012/05/21 美國 13/476,791

(71)申請人：輝達公司 (美國) NVIDIA CORPORATION (US)
美國

(72)發明人：費特曼 邁克爾 FETTERMAN, MICHAEL (US)；加德雷 詩利緒 GADRE, SHIRISH (US)；埃德蒙森 約翰 H EDMONDSON, JOHN H. (US)；帕蘭杰珮 奧姆卡爾 PARANJAPE, OMKAR (IN)；拉金德蘭 安亞納 RAJENDRAN, ANJANA (IN)；希爾 艾利克 萊爾 HILL, ERIC LYELL (US)；薩爾瓦內斯恩 拉杰史華蘭 SELVANESAN, RAJESHWARAN (IN)；麥卡弗 查爾斯 MCCARVER, CHARLES (US)；米切爾 凱文 MITCHELL, KEVIN (US)；海因里希 史蒂芬 詹姆斯 HEINRICH, STEVEN JAMES (US)

(74)代理人：蔡濱陽

(56)參考文獻：

US	005974523A	US	2004/0153763A1
US	2004/0216112A1	US	2005/0091656A1

審查人員：張發祥

申請專利範圍項數：9 項 圖式數：11 共 68 頁

(54)名稱

維持公平和秩序的資源管理子系統

RESOURCE MANAGEMENT SUBSYSTEM THAT MAINTAINS FAIRNESS AND ORDER

(57)摘要

本發明的一個具體實施例公佈一種最佳方式，以在關於重播操作的共用資源存取要求排程方面維持公平與秩序。尤其是，一串流多重處理器(SM，streaming multiprocessor)包含一全順序佇列(TOQ，total order queue)，設置成排定一或多個執行循環之上的該存取要求。當所需共用資源已經分配給該要求時，讓存取要求的進度往前。在多個存取要求都需要該相同共用資源之處，優先權會給予該較舊的存取要求。存取要求可進入睡眠狀態，等待特定共用資源變成可用。允許較舊存取要求從較年輕資源要求當中偷取資源，如此可避免僵持情況。所揭示技術的一項優點為：較新的存取要求不會重複阻擋較舊的共用資源存取要求之前進進度。

One embodiment of the present disclosure sets forth an effective way to maintain fairness and order in the scheduling of common resource access requests related to replay operations. Specifically, a streaming multiprocessor (SM) includes a total order queue (TOQ) configured to schedule the access requests over one or more execution cycles. Access requests are allowed to make forward progress when needed common resources have been allocated to the request. Where multiple access requests require the same common resource, priority is given to the older access request. Access requests may be placed in a sleep state pending availability of certain common resources. Deadlock may be avoided by allowing an older access request to steal resources from a younger resource request. One advantage of the disclosed technique is that older

410 . . . 串流多重處理器分割區

420 . . . 位址產生單元

432 . . . 全順序佇列項目緩衝區

455 . . . 仲裁單元

460 . . . 第一層快取記憶體

500 . . . 全順序佇列項目老化單元

510 . . . 要求狀態分配單元

520 . . . 資源狀態分配單元

530 . . . 全順序佇列多工器

540 . . . 全順序佇列項目單元

550 . . . 保留項目計數器

560 . . . 優先權選擇器

588 . . . 信號路徑

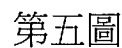
590 . . . 信號路徑

580 . . . 信號路徑

582 . . . 信號路徑

584 . . . 信號路徑

586 . . . 信號路徑



發明摘要

※ 申請案號：102117869

※ 申請日：102.5.21

※IPC 分類：G06F 9/50 (2006.01)

【發明名稱】(中文/英文)

維持公平和秩序的資源管理子系統

RESOURCE MANAGEMENT SUBSYSTEM THAT MAINTAINS FAIRNESS
AND ORDER

【中文】

本發明的一個具體實施例公佈一種最佳方式，以在關於重播操作的共用資源存取要求排程方面維持公平與秩序。尤其是，一串流多重處理器(SM, streaming multiprocessor)包含一全順序佇列(TOQ, total order queue)，設置成排定一或多個執行循環之上的該存取要求。當所需共用資源已經分配給該要求時，讓存取要求的進度往前。在多個存取要求都需要該相同共用資源之處，優先權會給予該較舊的存取要求。存取要求可進入睡眠狀態，等待特定共用資源變成可用。允許較舊存取要求從較年輕資源要求當中偷取資源，如此可避免僵持情況。所揭示技術的一項優點為：較新的存取要求不會重複阻擋較舊的共用資源存取要求之前進進度。

【英文】

One embodiment of the present disclosure sets forth an effective way to maintain fairness and order in the scheduling of common resource access requests related to replay operations. Specifically, a streaming multiprocessor (SM) includes a total order queue (TOQ) configured to schedule the access

requests over one or more execution cycles. Access requests are allowed to make forward progress when needed common resources have been allocated to the request. Where multiple access requests require the same common resource, priority is given to the older access request. Access requests may be placed in a sleep state pending availability of certain common resources. Deadlock may be avoided by allowing an older access request to steal resources from a younger resource request. One advantage of the disclosed technique is that older common resource access requests are not repeatedly blocked from making forward progress by newer access requests.

【代表圖】

【本案指定代表圖】：第（五）圖。

【本代表圖之符號簡單說明】：

410	串流多重處理器分割區	540	全順序佇列項目單元
420	位址產生單元	550	保留項目計數器
432	全順序佇列項目緩衝區	560	優先權選擇器
455	仲裁單元	588	信號路徑
460	第一層快取記憶體	590	信號路徑
500	全順序佇列項目老化單 元	580	信號路徑
510	要求狀態分配單元	582	信號路徑
520	資源狀態分配單元	584	信號路徑
530	全順序佇列多工器	586	信號路徑

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：

無

發明專利說明書

(本說明書格式、順序，請勿任意更動)

【發明名稱】(中文/英文)

維持公平和秩序的資源管理子系統/RESOURCE MANAGEMENT
SUBSYSTEM THAT MAINTAINS FAIRNESS AND ORDER

【技術領域】

【001】 本發明一般係關於電腦架構，尤其係關於維持公平與秩序的資源管理子系統。

【背景技術】

【002】 並行處理系統內的共用實踐為設計一種同時執行一些執行緒的處理器。每一執行緒都可在處理器內的一分離執行管線內執行。執行處理並行時，不同的執行緒可要求存取一共用資源，例如一共享的記憶體，如此可導致該等執行緒遭遇資源衝突。例如：多個執行緒可執行一共用資源存取操作，例如一記憶體載入指令，其中該等多個執行緒無法同時存取該資源。在這種情況下，允許至少一個執行緒存取該共用資源，並且完成該共用資源存取操作。該管線傳輸該共用資源的副本，例如某些執行緒所存取的該等快取線之一者，並且這些執行緒可完成該共用資源存取操作。不過，並不允許存取該相同共用資源的其他執行緒完成該共用資源存取操作並且維持無作用。如此某些執行緒可完成該共用資源存取操作，而其他執行緒則否。在缺乏裝置來處理多執行緒存取一共用資源之下，操作無法成功完成。

【003】 服務多執行緒存取一共用資源的一種方式為，將該資源分配給一或更多執行緒，直到該共用資源不再有額外可用存取埠為止。存取相

同資源的剩餘執行緒都處於待命狀態，等待該共用資源可用。當該共用資源具有一或更多可用的存取埠時，則將該共用資源分配給一或更多服務的待命要求，直到所有存取埠都已再次分配為止。該處理持續到對於該共用資源的所有要求都已經服務為止。

【004】 此方式的一項缺點為新要求會持續來到，並且消耗該共用資源的存取埠，而還在等待相同共用資源的待命中的要求尚未被服務。當新要求在待命要求之前得以存取一共用資源時，這種現象稱為「優先權反轉」。此方式的另一項缺點為當一共用資源變成可用時，可選擇服務待命中的要求。不過由於管線延遲，在該待命中要求完成該共用資源存取操作之前，另一個較新的要求可能已經消耗該共用資源。仍舊是此方式的另一項缺點為一要求可能需要存取兩個共用資源 A 和 B，其中 A 可用但是 B 不可用。該要求會處於待命狀態，直到 B 變成可用。當 B 變成可用時，則排定完成該待命中的要求。不過由於管線延遲，在該待命中要求可完成該共用資源存取操作之前，另一個較新的要求可能已經消耗該 A。當較新存取要求在較舊現有存取要求之前繼續分配資源，則該等較舊存取要求的進度可能無法往前，如此無法完成執行。

【005】 針對前面所例示，業界內需要一種更有效的方式來服務執行緒存取一共用資源。

【發明內容】

【006】 本發明的一個具體實施例揭示一種用於排定共用資源存取要求的電腦實施方法。該方法包含：接收對於一共用資源的一第一要求；決定該共用資源無法分配給要求；以及在回應上延遲服務該第一要求。該

方法另包含：在延遲服務之後，接收該共用資源的一第二要求；以及在接收該第二要求之後，決定該共用資源變成可分配給要求。該方法另包含在讓該共用資源可用於該第二要求之前，讓該共用資源可用於該第一要求。

【007】 本發明的其他具體實施例包含但不受限於一子系統，其包括一串流多重處理器(SM，streaming multiprocessor)，設置成實施該等揭示方法的一或更多態樣，以及包括一計算裝置，設置成實施該等揭示方法的一或更多態樣。

【008】 所揭示技術的一項優點為：較新的存取要求不會重複阻擋較舊的共用資源存取要求之前進進度。利用將該共用資源分配至一現有較舊存取要求優先於該新存取要求，可避免優先權反轉。

【圖式簡單說明】

【009】 所以，可以詳細瞭解本發明上述特徵之方式中，本發明的一更為特定的說明簡述如上，其可藉由參照到具體實施例來進行，其中一些例示於所附圖式中。但應注意所附圖式僅例示本發明的典型具體實施例，因此其並非要做為本發明之範圍的限制，本發明自可包含其它同等有效的具體實施例。

【010】 第一圖為例示設置來實施本發明一或多個態樣的電腦系統方塊圖；

【011】 第二圖為根據本發明的一個具體實施例，用於第一圖中該電腦系統的一並行處理子系統之方塊圖；

【012】 第三 A 圖為根據本發明的一個具體實施例，第二圖中前端的方塊圖；

【013】 第三 B 圖為根據本發明的一個具體實施例，第二圖中並行處理單元之一者內一般處理叢集的方塊圖；

【014】 第三 C 圖為根據本發明的一個具體實施例，第三 B 圖中該串流多重處理器一部分的方塊圖；

【015】 第四圖根據本發明的一個具體實施例，例示第三 C 圖中該 SM 內設置來服務等待中共用資源存取操作的一子系統；

【016】 第五圖根據本發明的一個具體實施例，例示第四圖中該 TOQ 內的一 TOQ 項目老化單元；

【017】 第六 A 圖和第六 B 圖根據本發明的一個具體實施例，揭示用於新增共用資源存取要求至一全順序佇列的方法步驟流程圖；

【018】 第七圖為根據本發明的一個具體實施例，用於更新一共用資源狀態的方法步驟流程圖；

【019】 第八圖為根據本發明的一個具體實施例，用於將一共用資源分配給與一共用資源存取要求相關聯的一 TOQ 項目之方法步驟流程圖；

【020】 第九圖為根據本發明的一個具體實施例，用於在一全順序佇列(TOQ)內選擇一共用資源存取要求來服務之方法步驟流程圖；

【021】 第十圖為根據本發明的一個具體實施例，用於將一 TOQ 項目緩衝區內 TOQ 項目老化的方法步驟流程圖；以及

【022】 第十一圖為根據本發明的一個具體實施例，用於將與一共用資源存取要求相關聯的一 TOQ 項目從一睡眠狀態轉換成一啟用狀態之方法步驟流程圖。

【實施方式】

【023】 在以下描述中，揭示許多特定細節以對本發明有更徹底之理解。但是，精通技術人士應該了解，在無一或多個該等特定細節之下還是可實施本發明。

系統概觀

【024】 第一圖為例示設置來實施本發明一或多個態樣的電腦系統 100 之方塊圖。電腦系統 100 包含一中央處理單元(CPU，central processing unit) 102，以及通過一互連路徑通訊的一系統記憶體 104，其中該路徑可包含一記憶體橋接器 105。例如可為北橋晶片的記憶體橋接器 105 透過匯流排或其他通訊路徑 106 (例如超傳輸(HyperTransport)連結)，連接至一 I/O (輸入/輸出)橋接器 107。例如可為南橋晶片的 I/O 橋接器 107 接收來自一或多個使用者輸入裝置 108 (例如鍵盤、滑鼠)的使用者輸入，並透過通訊路徑 106 和記憶體橋接器 105 將該輸入轉送至 CPU 102。一並行處理子系統 112 透過匯流排或第二通訊路徑 113 (例如周邊組件互連(PCI，Peripheral Component Interconnect) Express、加速圖形連接埠(Accelerated Graphics Port)或超傳輸連結)耦合至記憶體橋接器 105；在一個具體實施例內，並行處理子系統 112 為傳遞畫素給顯示裝置 110 (例如傳統陰極射線管或液晶監視器)的圖形子系統。系統磁碟 114 也連接至 I/O 橋接器 107。一開關 116 提供 I/O 橋接器 107 與其他組件，像是網路配接器 118 以及許多外接卡 120 和 121 之間的連接。其他組件(未明確顯示)，包含萬用序列匯流排(USB，universal serial bus)或其他連接埠連接、CD(compact disc)光碟機、數位影音光碟(DVD，digital video disc)機、影片記錄裝置等等，也可連接至 I/O 橋接器 107。第一圖內顯示的許多通訊路徑，包含特地指名的通訊路徑 106 和 113，

都可使用任何合適的協定來實施，例如 PCI Express、AGP (加速圖形連接埠，Accelerated Graphics Port)、超傳輸或任何其他匯流排或點對點通訊協定，以及不同裝置之間的連接都可使用業界內已知的不同協定。

【025】 在一個具體實施例內，並行處理子系統 112 併入將圖形與視訊處理最佳化的電路，包含例如視訊輸出電路，並且構成一圖形處理單元 (GPU，graphics processing unit)。在另一具體實施例內，並行處理子系統 112 併入將一般用途處理最佳化的電路，同時保留底層計算架構，本文內有更詳細描述。尚且在另一具體實施例內，並行處理子系統 112 可在單一子系統內合併一或多個其他系統元件，例如結合記憶體橋接器 105、CPU 102 和 I/O 橋接器 107 來形成一晶片上系統 (SoC，system on chip)。

【026】 吾人將了解，本文中顯示的該系統為例示，所以可進行改變與修改。包含橋接器的數量與配置、CPU 102 的數量以及並行處理子系統 112 的數量這類連接拓撲可依照需求修改。例如在某些具體實施例內，系統記憶體 104 直接連接至 CPU 102，而不是透過橋接器，並且其他裝置透過記憶體橋接器 105 以及 CPU 102 與系統記憶體 104 通訊。在其他替代拓撲中，並行處理子系統 112 連接至 I/O 橋接器 107 或直接連接至 CPU 102，而不是連接至記憶體橋接器 105。仍舊在其他具體實施例內，I/O 橋接器 107 和記憶體橋接器 105 可整合成為單一晶片，替代現有的一或多個分散裝置。大型組件可包含二或多個 CPU 102 以及二或多個並行處理子系統 112。本文中顯示的該等特定組件為選擇性；例如可支援任何數量的外接卡或周邊裝置。在某些具體實施例內，省略開關 116，並且網路配接器 118 和外接卡 120、121 都直接連接至 I/O 橋接器 107。

【027】 第二圖例示根據本發明一個具體實施例的並行處理子系統 112。如所示，並行處理子系統 112 包含一或多個並行處理單元(PPU, parallel processing unit) 202，每一個都耦合至本機並行處理(PP, parallel processing) 記憶體 204。一般來說，並行處理子系統包含數量為 U 的 PPU，其中 $U \geq 1$ 。(本文中多個類似物體的實例都用參考號碼標示出該物體，並且括號標示所需的實例。)PPU 202 以及並行處理記憶體 204 可使用一或多個積體電路裝置來實施，例如可程式處理器、特殊應用積體電路(ASIC, application specific integrated circuit)或記憶體裝置，或以任何其他技術可行方式。

【028】 請再次參閱第一圖及第二圖，在一些具體實施例內，並行處理子系統 112 內的某些或全部 PPU 202 為具有彩現管線的圖形處理器，其可設置成執行許多操作，這些操作有關透過記憶體橋接器 105 和第二通訊路徑 113，從 CPU 102 及/或系統記憶體 104 供應的圖形資料產生畫素資料、與本機並行處理記憶體 204 (可用來當成圖形記憶體，包含例如傳統畫框緩衝區)互動來儲存與更新畫素資料、傳遞畫素資料給顯示裝置 110 等等。在某些具體實施例內，並行處理子系統 112 可包含操作當成圖形處理器的一或多個 PPU 202，以及用於一般用途計算的一或多個其他 PPU 202。該等 PPU 可一致或不同，並且每一 PPU 都可擁有一專屬的並行處理記憶體裝置或無專屬的並行處理記憶體裝置。並行處理子系統 112 內的一或多個 PPU 202 可輸出資料至顯示裝置 110，或並行處理子系統 112 內的每一 PPU 202 都可輸出資料至一或多個顯示裝置 110。

【029】 在操作上，CPU 102 為電腦系統 100 的主要處理器，控制與協調其他系統組件的操作。尤其是，CPU 102 發出指令控制 PPU 202 的操

作。在某些具體實施例內，CPU 102 將每一 PPU 202 的指令串流寫入至一資料結構(第一圖或第二圖內未明確顯示)，其可位於系統記憶體 104 內、並行處理記憶體 204 內或可存取 CPU 102 和 PPU 202 的其他儲存位置內。每一資料結構的指標已經寫入一推送緩衝區，開始在該資料結構內處理該指令串流。PPU 202 讀取來自一或多個推送緩衝區的指令串流，然後關於 CPU 102 的操作非同步執行指令。利用應用程式透過裝置驅動程式 103 來控制不同推送緩衝區的排程，指定每一推送緩衝區的執行優先順序。

● **【030】** 此時請回頭參閱第二圖以及第一圖，每一 PPU 202 都包含一 I/O (輸入/輸出)單元 205，其透過連接至記憶體橋接器 105 (或在一個替代具體實施例內，直接至 CPU 102)的通訊路徑 113，與電腦系統 100 的剩餘組件通訊。PPU 202 與電腦系統 100 剩餘組件的連接也可改變。在某些具體實施例內，並行處理子系統 112 實施成為可插入電腦系統 100 內擴充槽的外接卡。在其他具體實施例內，PPU 202 可與例如記憶體橋接器 105 或 I/O 橋接器 107 這類匯流排橋接器整合在單一晶片上。仍舊在其他具體實施例內，PPU 202 的某些或全部元件可與 CPU 102 整合在單一晶片上。

● **【031】** 在一個具體實施例內，通訊路徑 113 為一 PCI Express 連結，其中分配專屬通道給每一 PPU 202，如業界內所熟知。在此也可使用其他通訊路徑。一 I/O 單元 205 產生在通訊路徑 113 上傳輸的封包(或其他信號)，也從通訊路徑 113 上接收所有傳入封包(或其他信號)，將該等傳入封包導引至 PPU 202 的適當組件。例如：有關處理任務的指令可導引至主介面 206，而有關記憶體操作的指令(例如讀取或寫入並行處理記憶體 204)可導引至記憶體橫桿單元 210。主介面 206 讀取每一推送緩衝區，並將該推送緩衝區內

儲存的該指令串流輸出至一前端 212。

【032】 每一 PPU 202 都有利地實施一高並行處理架構。如細節內所示，PPU 202(0)包含一處理叢集陣列 230，其包含數量為 C 的一般處理叢集(GPC，general processing cluster) 208，其中 $C \geq 1$ 。每一 GPC 208 都可同時執行大量的(例如數百或數千)執行緒，其中每一執行緒都是一程式的實例。在許多應用當中，不同的 GPC 208 可分配用於處理不同種類的程式，或用於執行不同種類的計算。GPC 208 的分配根據針對每一種程式或計算所賦予的工作負擔而改變。

【033】 GPC 208 從任務/工作單元 207 內的工作分配單元當中接收要執行的處理任務，該工作分配單元接收指標來處理任務，其編碼為任務中繼資料(TMD，task metadata)並儲存在記憶體內。至 TMD 的該指標包含在指令串流內，其儲存當成一推送緩衝區並由前端單元 212 從主介面 206 接收。可編碼為 TMD 的處理任務包含要處理的資料索引，以及定義如何處理該資料的狀態參數與指令(例如要執行哪個程式)。任務/工作單元 207 從前端 212 接收任務，並且確定在每一 TMD 指定的處理開始之前已經將 GPC 208 設置成有效狀態。一優先順序可指定給每一 TMD，用來排定該處理任務的執行時間。處理任務也可從處理叢集陣列 230 接收。選擇性，該 TMD 可包含一參數，其控制該 TMD 加入處理任務清單(或處理任務指標清單)的頭部或尾部，藉此提供優先順序之上的另一控制等級。

【034】 記憶體介面 214 包含數量為 D 的分割單元 215，這些單元每一個都直接耦合至一部分並行處理記憶體 204，其中 $D \geq 1$ 。如所示，分割單元 215 的數量一般等於動態隨機存取記憶體(DRAM，dynamic random

access memory) 220 的數量。在其他具體實施例內，分割單元 215 的數量可不等於記憶體裝置的數量。精通技術人士將了解，DRAM 220 可用其他合適的儲存裝置取代，並且可為一般傳統設計，因此省略其詳細說明。像是畫框緩衝區或紋理地圖這類彩現目標可通過 DRAM 220 儲存，允許分割單元 215 並行寫入每一彩現目標的部分，以有效使用並行處理記憶體 204 的可用頻寬。

【035】 任一 GPC 208 都可處理寫入並行處理記憶體 204 內任一 DRAM 220 的資料。橫桿單元 210 設置成將每一 GPC 208 的輸出繞送至任意分割單元 215 的輸入或至用於進一步處理的其他 GPC 208。GPC 208 透過橫桿單元 210 與記憶體介面 214 通訊，來讀取或寫入許多外部記憶體裝置。在一個具體實施例內，橫桿單元 210 具有一連接至記憶體介面 214 來與 I/O 單元 205 通訊，以及一連接至本機並行處理記憶體 204，藉此讓不同 GPC 208 內的處理核心與系統記憶體 104 或不在 PPU 202 本機上的其他記憶體通訊。在第二圖所示的具體實施例中，橫桿單元 210 直接連接 I/O 單元 205。橫桿單元 210 可使用虛擬通道，以分隔 GPC 208 與分割單元 215 之間的流量串流。

【036】 GPC 208 同樣可程式編輯來執行有關廣泛應用的處理任務，包含但不受限於線性與非線性資料傳輸、視訊及/或音訊資料篩選、模型化運算(例如套用實體規則來決定位置、速度以及其他物體屬性)、影像彩現運算(例如曲線細分著色、影點著色、幾何著色及/或畫素著色程式)等等。PPU 202 可從系統記憶體 104 及/或本機並行處理記憶體 204 將資料傳輸進入內部(晶片上)記憶體、處理該資料並將結果資料寫回系統記憶體 104 及/

或本機並行處理記憶體 204，其中這種資料可由其他系統組件存取，包含 CPU 102 或另一並行處理子系統 112。

【037】 一 PPU 202 可提供任何數量的本機並行處理記憶體 204，包含非本機記憶體，並且可在任何情況下使用本機記憶體和系統記憶體。例如：PPU 202 可為統一記憶體架構(UMA，unified memory architecture)具體實施例內的圖形處理器。在這種具體實施例內，提供一些或無專屬圖形(並行處理)記憶體，並且 PPU 202 完全或幾乎完全使用系統記憶體。在 UMA 具體實施例內，PPU 202 可整合至橋接器晶片或處理器晶片，或提供當成分散式晶片，具有高速連結(例如 PCI Express)透過橋接器晶片或其他通訊方式將 PPU 202 連接至系統記憶體。

【038】 如上述，任何數量的 PPU 202 都可包含在一並行處理子系統 112 內。例如：單一外接卡上可提供多個 PPU 202，或多張外接卡可連接至通訊路徑 113，或一或多個 PPU 202 可整合到一橋接器晶片上。多 PPU 系統內的 PPU 202 可彼此一致或不同。例如：不同的 PPU 202 可具有不同數量的處理核心、不同數量的本機並行處理記憶體等等。當存在多個 PPU 202 時，這些 PPU 可並行操作，以比單一 PPU 202 還要高產量的方式來處理資料。合併一或多個 PPU 202 的系統可在許多設置與外型因素之下實施，包含桌上型、膝上型或手持式個人電腦、伺服器、工作站、遊戲機、嵌入式系統等等。

多重並行任務排程

【039】 多重處理任務可在 GPC 208 上同時執行，並且一處理任務可在執行期間產生一或多個「子代」處理任務。任務/工作單元 207 接收該

等任務，並且動態排定要由 GPC 208 執行的處理任務以及子代處理任務。

【040】 第三 A 圖為根據本發明的一個具體實施例，第二圖中任務/工作單元 207 的方塊圖。任務/工作單元 207 包含一任務管理單元 300 以及工作分配單元 340。任務管理單元 300 根據執行優先順序等級來組織要排程的任務。針對每一優先順序等級，任務管理單元 300 儲存指標清單到對應至排程器表 321 內任務的 TMD 322，其中該清單可用連結清單來實施。TMD 322 可儲存在 PP 記憶體 204 或系統記憶體 104 內。任務管理單元 300 接受任務並且將該等任務儲存在排程器表 321 內的速率與任務管理單元 300 排定任務來執行的速率無關，因此任務管理單元 300 可在排定該等任務之前集中許多任務。然後根據優先順序資訊或使用其他技術，例如輪轉排程，來排定集中的任務。

【041】 工作分配單元 340 包含一任務表 345，其中有插槽讓 TMD 322 針對執行的任務所佔用。任務管理單元 300 可排定任務在任務表 345 內有空插槽時執行。當無空插槽時，未佔用插槽的較高優先順序任務會驅逐佔用插槽的較低優先順序任務。當一任務遭到驅逐時，該任務會停止，並且若該任務尚未執行完畢，則該任務的指標會加入要排定的任務指標清單中，如此稍後會恢復執行該任務。當在一任務執行期間產生子代處理任務時，將該子代任務的指標加入要排定的任務指標清單內。由在處理叢集陣列 230 內執行的 TMD 322 可產生一子代任務。

【042】 與從前端 212 的任務/工作單元 207 所接收的任務不同，子代任務從處理叢集陣列 230 接收。子代任務並未插入推送緩衝區或傳輸至該前端。當子代任務已經產生或該子代任務的資料已經儲存在記憶體內

時，並不會通知 CPU 102。透過推送緩衝區提供的該等任務與子代任務間之另一差異在於，透過推送緩衝區提供的該等任務由該應用程式定義，而該等子代任務則在該等任務執行期間動態產生。

任務處理概觀

【043】 第三 B 圖為根據本發明的一個具體實施例，第二圖中 PPU 202 之一者內 GPC 208 的方塊圖。每一 GPC 208 都可設置成同時執行大量執行緒，其中「執行緒」一詞代表在特定輸入資料集上執行的特定程式之實例。在某些具體實施例內，單一指令、多重資料(SIMD, single-instruction, multiple-data)指令發行技術用於支援大量執行緒的並行執行，而不用提供多個獨立指令單元。在其他具體實施例內，單一指令、多重執行緒(SIMT, single-instruction, multiple-thread)技術用於支援大量一般同步執行緒的並行執行，使用共用指令單元，其設置成發出指令至每一 GPC 208 內的處理引擎集。不同於一 SIMD 執行區域，其中所有處理引擎一般都執行一致的指令，SIMT 執行允許不同執行緒更迅速遵循分散的執行路徑通過一已知執行緒程式。業界內精通技術人士將了解，SIMD 處理區域代表一 SIMT 處理區域的功能子集。

【044】 透過將處理任務分配至串流多重處理器(SM, streaming multiprocessor) 310 的管線管理員 305 可有利地控制 GPC 208 的操作。管線管理員 305 也可設置成利用指定 SM 310 所輸出已處理資料的目的地，控制一工作分配橫桿 330。

【045】 在一個具體實施例內，每一 GPC 208 都包含數量為 M 的 SM 310，其中 $M \geq 1$ ，每一 SM 310 都設置成處理一或多個執行緒群組。另外，

每一 SM 310 有利地包含可管路化的一致功能執行單元集(例如執行單元與載入儲存單元，在第三 C 圖內顯示為執行單元 302 以及 LSU 303)，允許完成先前指令之前發出新指令，如先前技術所知。可提供功能執行單元的任何組合。在一個具體實施例內，該等功能單元支援許多種運算，包含整數與浮點演算(例如加法與乘法)、比較運算、布林運算(AND、OR、XOR)、位元位移和許多代數函數的計算(例如平面插值、三角函數、指數以及對數函數等等)；並且該相同功能單元硬體可用來執行不同運算。

● **【046】** 這一系列指令傳輸至構成一執行緒的特定 GPC 208，如本文先前所定義，並且通過 SM 310 內並行處理引擎(未顯示)的特定數量同時執行的執行緒之集合在此稱為「經線」或「執行緒群組」。如本文所使用，「執行緒群組」代表在不同輸入資料上同時執行相同程式的執行緒群組，其中該群組的一個執行緒指派給 SM 310 內不同的處理引擎。一執行緒群組可包含數量比 SM 310 內處理引擎數量還要少的執行緒，在此案例中，某些處理引擎會在循環期間處理該執行緒群組時閒置。執行緒群組也可包含數量比 SM 310 內處理引擎數量還要多的執行緒，在此案例中，將在連續時脈循環上進行處理。因為每一 SM 310 都可同時支援最多 G 個執行緒群組，所以在任何已知時間上 GPC 208 內都可執行最多 $G*M$ 個執行緒群組。

● **【047】** 此外，在 SM 310 內可同時啟用複數個相關執行緒群組(在不同執行階段內)。此執行緒群組的集合稱為「合作執行緒陣列」(CTA，cooperative thread array)或「執行緒陣列」。特定 CTA 的大小等於 $m*k$ ，其中 k 為執行緒群組內同時執行的執行緒數量，通常為 SM 310 內並行處理引擎數量的整數倍數，並且 m 為 SM 310 內同時啟用的執行緒群組數量。CTA

的大小一般由程式設計師以及該 CTA 可用的硬體資源數量，例如記憶體或暫存器，來決定。

【048】 每一 SM 310 都包含第一層(L1)快取(如第三 C 圖所示)，或使用 SM 310 之外對應 L1 快取內的空間，其用於執行負載與儲存操作。每一 SM 310 也要存取所有 GPC 208 之間共享的第二層(L2)快取，並且可用於在執行緒之間傳輸資料。最後，SM 310 也要存取至晶片外「全域」記憶體，其可包含例如並行處理記憶體 204 及/或系統記憶體 104。吾人了解，PPU 202 之外的任何記憶體都可用來當成全域記憶體。此外，第一點五層(L1.5)快取 335 可包含在 GPC 208 內，設置成接收並固定由 SM 310 要求透過記憶體介面 214 從記憶體擷取的資料，包含指令、統一資料以及常數資料，並且將該要求的資料提供給 SM 310。具有 GPC 208 內多個 SM 310 的具體實施例得利於共享共用指令和在 L1.5 快取 335 內快取的資料。

【049】 每一 GPC 208 都可包含一記憶體管理單元(MMU，memory management unit) 328，其設置成將虛擬位址映射至實體位址。在其他具體實施例內，MMU 328 可位於記憶體介面 214 之內。MMU 328 包含一組頁面表記錄(PTE，page table entry)，用於將一拼貼以及選擇性一快取線索引的虛擬位址映射至實體位址。MMU 328 可包含能夠位於多處理器 SM 310 或 L1 快取或 GPC 208 內的位址轉譯後備緩衝區(TLB，translation lookaside buffer)或快取。該實體位址經過處理來分配本機存取的表面資料，以便讓有效要求交錯於分割單元 215 之間。該快取線索引可用於決定一快取線的要求是否命中或未命中。

【050】 在圖形與計算應用當中，GPC 208 可經過設置，如此每一

SM 310 都耦合至一紋理單元 315，用於執行紋理映射操作，例如決定紋理樣本位置、讀取紋理資料以及篩選該紋理資料。紋理資料從內部紋理 L1 快取(未顯示)讀取，或在某些具體實施例內從 SM 310 內的該 L1 快取讀取，並且依照需求從所有 GPC 208 之間共享的 L2 快取、並行處理記憶體 204 或系統記憶體 104 擷取。每一 SM 310 都輸出處理過的任務至工作分配橫桿 330，以便將該處理過的任務提供至另一 GPC 208 供進一步處理，或透過橫桿單元 210 將該處理過的任務儲存在 L2 快取、並行處理記憶體 204 或系統記憶體 104 內。preROP (預先光柵運算) 325 設置成從 SM 310 接收資料、將資料引導至分割單元 215 內的 ROP 單元，並且執行顏色混合、組織畫素顏色資料以及執行位址轉譯的最佳化。

【051】 吾人將了解，本文中顯示的該核心架構為例示，所以可進行改變與修改。任何數量的處理單元，例如 SM 310 或紋理單元 315、preROP 325 都可包含在 GPC 208 內。進一步如第二圖內所示，PPU 202 可包含任意數量功能彼此類似的 GPC 208，如此執行行為並不取決於接收特定處理任務的 GPC 208。進一步，每一 GPC 208 都得利於與其他 GPC 208 無關的操作，運用分離並分散的處理單元、L1 快取，來執行一或多個應用程式的任務。

【052】 精通此技術人士將了解，第一圖、第二圖、第三 A 圖和第三 B 圖內描述的架構並非用於限制本發明範疇，本說明書內的技術可在任何正確設置的處理單元上實施，在不背離本發明範疇之下包含但不受限於一或多個 CPU、一或多個多核心 CPU、一或多個 PPU 202、一或多個 GPC 208、一或多個圖形或特殊用途處理單元等等。

【053】 在本發明的具體實施例內，吾人想要使用一計算系統的 PPU 202 或其他處理器，運用執行緒陣列來執行一般用途計算。該執行緒陣列內的每一執行緒都可指派一個獨一的執行緒識別碼(「執行緒 ID」)，其可在該執行緒執行期間存取該執行緒。該執行緒 ID 可定義為一維度或多維度數值，控制該執行緒處理行為的許多態樣。例如：一執行緒 ID 可用於決定設定一執行緒的哪個輸入資料部分要處理及/或決定設定一執行緒的哪個輸出資料部分要產生或寫入。

【054】 每個執行緒指令的序列可包含至少一個指令，其定義該代表性執行緒與該執行緒陣列的一或多個其他執行緒之間的合作行為。例如：每一執行緒指令的順序可包含將在該順序內特定點上該代表執行緒運算執行中斷，直到一或多個其他執行緒到達該特定點上為止之指令、讓該代表執行緒將一或多個其他執行緒可存取的資料儲存至一共享記憶體內之指令、讓該代表執行緒根據其執行緒 ID 基本上讀取與更新儲存在一共享記憶體內一或多個該等其他執行緒已經存取過的資料之指令等等。該 CTA 程式也可包含一指令，計算該共享記憶體內所要讀取資料的位址，其中該位址為執行緒 ID 的函數。利用定義合適的功能並且提供同步技術，資料可利用 CAT 的一個執行緒寫入共享記憶體內的一已知位置，並且以可預測方式用相同 CTA 的不同執行緒從該位置當中讀取。因此，支援在執行緒之間共享的任何所要資料圖案，並且一 CTA 內的任何執行緒都可與相同 CTA 內任何其他執行緒共享資料。在 CTA 執行緒之間共享的資料內容(若有的話)由該 CTA 程式決定；如此吾人了解在使用 CTA 的特定應用當中，根據該 CTA 程式，CTA 的執行緒彼此之間會或不會確實共享資料，並且在本文中可同

時使用「CTA」與「執行緒陣列」等詞。

【055】 第三 C 圖為根據本發明的一個具體實施例，第三 B 圖中 SM 310 的方塊圖。SM 310 包含一指令 L1 快取 370，其設置成透過 L1.5 快取 335 接收指令與常數。經線排程器與指令單元 312 接收來自指令 L1 快取 370 的指令與常數，並且根據該等指令與常數控制本機暫存檔 304 以及 SM 310 功能單元。SM 310 功能單元包含 N 個執行(執行或處理)單元 302 以及 P 個載入儲存單元(LSU，load-store unit) 303。

● 【056】 SM 310 提供具備不同存取階層的晶片上(內部)資料儲存。特殊暫存器(未顯示)可由 LSU 303 讀取但是無法寫入，並且可用於儲存定義每一執行緒「位置」的參數。在一個具體實施例內，特殊暫存器包含每一執行緒(或 SM 310 內每一執行單元 302)一個暫存器，其儲存一執行緒 ID；每一執行緒 ID 暫存器都只能由個別一個執行單元 302 存取。特殊暫存器也可包含額外暫存器，可由執行 TMD 322 (或由所有 LSU 303)所呈現相同處理任務的所有執行緒讀取，其儲存一 CTA 識別碼、該 CTA 維度、該 CTA 所屬網格的維度(或若 TMD 322 編碼一佇列任務而非一網格任務時的佇列位置)以及指派 CTA 的 TMD 322 之識別碼。

● 【057】 若 TMD 322 為一網格 TMD，則執行 TMD 322 會導致啟動並執行固定數量的 CTA，來處理佇列 525 內所儲存的固定數量資料。CTA 的數量依照網格寬度、高度與深度的乘積來指定。該固定數量的資料可儲存在 TMD 322 內，或 TMD 322 可儲存將由 CTA 處理的資料之指標。TMD 322 也儲存該 CTA 所執行程式的開始位址。

● 【058】 若 TMD 322 為佇列 TMD，然後使用 TMD 322 的佇列功能，

表示要處理的資料量並不需要固定。佇列記錄儲存指派給 TMD 322 由 CTA 處理的資料。該等佇列記錄也呈現執行緒執行期間由另一 TMD 322 產生的子代任務，藉此提供巢狀並行。一般來說，執行緒的執行或包含該執行緒的 CTA 會中止，直到子代任務執行完成為止。該佇列可儲存在 TMD 322 內，或與 TMD 322 分開，在此案例中 TMD 322 儲存至該佇列的佇列指標。有利的是，由該子代任務產生的資料可寫入該佇列，同時 TMD 322 代表已經執行的該子代任務。該佇列可實施為一圓形佇列，如此資料總量並不受限於該佇列的大小。

【059】 屬於一網格的 CTA 具有暗示的網格寬度、高度和深度參數，指示該網格內個別 CTA 的位置。在初始化期間會寫入特殊暫存器，以回應透過前端 212 從裝置驅動程式 103 接收的命令，並且在一處理任務執行期間不會改變。前端 212 排程執行的每一處理任務。每一 CTA 都關聯於一特定 TMD 322，以便同時執行一或多個任務。此外，單一 GPC 208 可同時執行多個任務。

【060】 一參數記憶體(未顯示)儲存可由相同 CTA (或任何 LSU 303) 內的任何執行緒讀取但無法寫入的執行時間參數(常數)。在一個具體實施例內，裝置驅動程式 103 在導引 SM 310 開始執行使用這些參數的任務之前，提供參數給該參數記憶體。任何 CTA 內的任何執行緒(或 SM 310 內的任何執行單元 302)都可透過記憶體介面 214 存取全域記憶體。全域記憶體的一部分可儲存在 L1 快取 320 內。

【061】 每一執行緒都使用本機暫存檔 304 當成伸展空間；每一暫存器都分配給一個執行緒專用，並且任何本機暫存檔 304 內的資料都只能由

分配給該暫存器的該執行緒存取。本機暫存檔 304 可實施為實體上或邏輯上區分成 P 個通路的暫存檔，每一通路都具有某些數量的記錄(在此每一記錄都可儲存例如 32 位元字)。一個通路指派給該 N 個執行單元 302 以及 P 個載入儲存單元 LSU 303 的每一個，並且不同通路內的對應記錄可填入執行相同程式的不同執行緒之資料，來幫助 SIMD 執行。通路的不同部分可分配給該 G 個同時執行緒群組中不同的執行緒，如此本機暫存檔 304 內的一已知記錄只能由特定執行緒存取。在一個具體實施例內，本機暫存檔 304 內的特定記錄保留用於儲存執行緒識別碼，實施該等特殊暫存器之一者。此外，一統一 L1 快取 375 儲存 N 個執行單元 302 以及 P 個載入儲存單元 LSU 303 的每一通路之統一或常數值。

【062】 共享記憶體 306 可由單一 CTA 內的執行緒存取；換言之，共享記憶體 306 內任何位置都可由相同 CTA 內的任何執行緒(或 SM 310 內任何處理引擎)來存取。共享記憶體 306 可實施為具有互連的一共享暫存檔或共享晶片上快取記憶體，允許任何處理引擎從該共享記憶體內任何位置讀取或寫入。在其他具體實施例內，共享的狀態空間可映射至晶片外記憶體的每一 CTA 區域，並且快取在 L1 快取 320 內。該參數記憶體可實施為該相同共享暫存檔或實施共享記憶體 306 的共享快取記憶體內的一指定區段，或實施為 LSU 303 具有唯讀存取的一個別共享暫存檔或晶片上快取記憶體。在一個具體實施例內，實施該參數記憶體的區域也用於儲存該 CTA ID 和任務 ID，以及 CTA 和網格尺寸或佇列位置，實施該特殊暫存器的部分。SM 310 內每一 LSU 303 都耦合至統一位址映射單元 352，將提供用於載入與儲存統一記憶體空間內所指定指令的一位址轉換至每一分散記憶體空間

內的一位址。因此，利用指定該統一記憶體空間內的一位址，可使用一指令來存取任何該本機、共享或全域記憶體空間。

【063】 每一 SM 310 內的 L1 快取 320 都可用於快取私用每一執行緒本機資料，並且也快取每一應用程式全域資料。在某些具體實施例內，該每一 CTA 共享資料可快取在 L1 快取 320 內。LSU 303 可透過一記憶體與快取互連 380 耦合至共享記憶體 306 以及 L1 快取 320。

維持公平與秩序的全順序佇列

【064】 第四圖根據本發明的一個具體實施例，例示第三 C 圖中該 SM 310 內設置來服務等待中共用資源存取操作的一子系統 400。如所示，子系統 400 包含 SM 分割區 410、位址產生單元(AGU, address generation unit) 420、一位址多工器 440、一寫入資料多工器 450、一仲裁單元 455、一 L1 快取記憶體 460、一讀取資料解多工器 465 以及一全順序佇列(TOQ, total order queue) 430。

【065】 該等 SM 分割區位於 SM 310 的該經線排程器與指令單元 312 之內。該等 SM 分割區排定並且派遣共用資源存取要求至 LSU 303。如所示，SM 分割區 410 包含一排程器 412、一派遣單元 414 以及一重播指令緩衝區(RIB, replay instruction buffer) 416。排程器 412 接收來自指令 L1 快取 370 的新指令，並且排定該等指令由 SM 310 執行。排程器 412 提交排定的指令給派遣單元 414。

【066】 派遣單元 414 接收來自排程器 412 的排程指令，並且將該等指令轉送至 SM 310 內的該適當的功能單元來執行。在一指令包含一共用資源存取要求之處，例如載入或儲存資料至記憶體內的一要求，該派遣單元

將該等指令轉送至一 LSU 303。在派遣這一指令至 LSU 303 之前，該派遣單元決定 RIB 416 內是否有可用的項目來分配給該指令。一 RIB 項目已建立用於任何共用資源存取要求，其在該要求執行期間可能需要至少一個重播操作。若有一可用的 RIB 項目，則該派遣單元儲存與 RIB 416 內該指令有關的資訊，並且將該指令派遣至 LSU 303。若無可用的 RIB 項目，則在一 RIB 項目變成可用之前，派遣單元 414 停止指向 LSU 303 的待命指令串流。在這種情況下，仍舊允許完成執行已派遣指令的重播操作。一旦一指令已經執行完成，則撤回該對應 RIB 項目，並且該撤回的 RIB 項目可讓派遣單元 414 分配一新指令。如底下進一步描述，派遣單元 414 接收重播操作要求，並且要求從 TOQ 430 撤回 RIB 項目。此外，派遣單元 414 將寫入資料傳送至寫入資料多工器 450，並且從讀取資料解多工器 465 接收讀取資料。

【067】 RIB 416 儲存有關內含一共用資源存取要求的指令之資訊，這種要求可需要一或更多重播操作，以便完成執行。在 LSU 303 的一個執行循環內可完成一共用資源存取要求。不過在某些條件下，該存取要求無法完成執行，並且建立一或更多額外執行循環，稱為重播操作，以便完成該存取要求。這些條件包含但不受限於：存取要求跨越比 LSU 303 同時存取還要多的快取線、存取一結構內比該結構同時服務還要多的位置、存取目前由先前要求消耗的一結構或存取優先順序低於其他待命要求順序的要求。有關每一這種存取要求的資訊都儲存在一相關聯的 RIB 項目內。在後續重播操作期間已經取得該 RIB 項目並傳遞至 LSU 303，幫助該重播操作的執行。該 RIB 項目可包含但不受限於該指令、該操作所需的常數值

以及該操作所需的暫存區範圍。吾人了解，可用 RIB 項目的最佳數量係以因素數為準，包含但不受限於需要一重播操作的條件發生頻率、針對一存取要求建立的重播操作之平均數量以及派遣單元 414 內的潛時。RIB 416 將命令與位址傳送至 AGU 420。

【068】 AGU 420 位於 LSU 303 之內，每一 AGU 420 都與一對應 SM 分割區 410 相關聯。例如：AGU 420(0)與 SM 分割區 410(0)相關聯，並且 AGU 420(N)與 SM 分割區 410(N)相關聯。AGU 420 接收沿著信號路徑 470 來自 RIB 416 的命令與位址。在回應上，AGU 420 沿著信號路徑 472 將 AGU 要求狀態傳送至 TOQ 430。接著 TOQ 430 讀取該 AGU 要求狀態，以便決定該當前存取要求是否需要額外重播操作，或是否要撤回對應至該當前存取要求的該 RIB 項目。此外，AGU 420 沿著信號路徑 474 將位址資訊傳送至位址多工器 440。

【069】 位址多工器 440 從對應至該當前存取要求的 AGU 420 選擇一位址，並且將該選取的位址傳送至該 L1 快取 320。每一 AGU 420 都產生一位址給該當前存取要求，如上述。位址多工器 440 沿著信號路徑 474 接收來自每一 AGU 420 的位址。位址多工器 440 也接收一控制信號(未顯示)，識別選擇哪一個 AGU 420 用於該當前存取要求。若該控制信號識別 AGU 420(0)，則位址多工器 440 將 AGU 420(0)產生的該位址傳送至 L1 快取 320。若該控制信號識別 AGU 420(N)，則位址多工器 440 將 AGU 420(N)產生的該位址傳送至 L1 快取 320。

【070】 當該共用資源存取要求為將資料儲存至該共用資源的一要求時，寫入資料多工器 450 選擇來自派遣單元 414 並對應至該當前存取要

求的寫入資料，並且將該選取的寫入資料傳送至 L1 快取 320。當該當前存取要求為一儲存操作時，SM 分割區 410 內的每一派遣單元都產生寫入資料給該當前存取要求。寫入資料多工器 450 沿著信號路徑 476 接收來自每一派遣單元 414 的寫入資料。寫入資料多工器 450 也接收一控制信號(未顯示)，識別選擇哪一個 SM 分割區 410 用於該當前儲存存取要求。若該控制信號識別 SM 分割區 410(0)，則寫入資料多工器 450 將 SM 分割區 410(0)產生的該寫入資料傳送至 L1 快取 320。若該控制信號識別 SM 分割區 410(N)，則寫入資料多工器 450 將 SM 分割區 410(N)產生的該寫入資料傳送至 L1 快取 320。

【071】 仲裁單元 455 接收來自 SM 分割區 410 以及其他用戶端的共用資源存取要求，並且將一選取的存取要求傳送給該共用資源進行服務，例如 L1 快取記憶體 460。該仲裁接收位址多工器 440 選擇的該位址，以及來自其他用戶端要求存取至該共用資源的位址。仲裁單元 455 將與該選取存取要求相關聯的該位址繞送至 L1 快取記憶體 460。類似地，針對將資料儲存在記憶體內的存取要求，仲裁單元 455 將與該選取存取要求相關聯的該寫入資料繞送至 L1 快取記憶體 460。此外，仲裁單元 455 沿著信號路徑 478 將仲裁狀態資訊傳送至 TOQ 430。

【072】 位於 L1 快取 320 內的 L1 快取記憶體 460 為可存取至 SM 分割區 410 和其他外部用戶端的一共用資源範例。L1 快取記憶體 460 包含可存取至共用資源存取要求的記憶體位置，以便儲存或載入資料。L1 快取記憶體 460 接收來自仲裁單元 455 的位址與控制信號。L1 快取記憶體 460 接收與來自仲裁單元 455 的儲存存取要求相關聯之寫入資料，並且將與載

入存取要求相關聯的讀取資料傳送至讀取資料解多工器 465。如底下進一步描述，L1 快取記憶體 460 沿著信號路徑 482 將與 L1 快取要求相關聯的狀態，以及沿著路徑 484 將資源狀態傳送至 TOQ 430。L1 快取記憶體 460 沿著信號路徑 486 接收來自 TOQ 430 的保留計數。

【073】 當該共用資源存取要求為從該共用資源載入資料的一要求，讀取資料多工器 465 將來自 L1 快取記憶體 460 的讀取資料傳送至要求該載入存取要求的用戶端。該要求單元可為 SM 分割區 410 或另一個 L1 快取用戶端。讀取資料多工器 465 接收一控制信號(未顯示)，識別選擇哪一個 SM 分割區 410 用於該當前載入存取要求。若該控制信號識別 SM 分割區 410(0)，則讀取資料多工器 465 將 L1 快取記憶體 460 產生的該讀取資料傳送至 SM 分割區 410(0)。若該控制信號識別 SM 分割區 410(N)，則讀取資料多工器 465 將 L1 快取記憶體 460 產生的該讀取資料沿著信號路徑 480 傳送至 SM 分割區 410(N)。

【074】 如結合第九圖內的進一步描述，TOQ 430 設置成根據並未受阻等待一共用資源的該最舊存取要求，以公平並且避免優先權反轉的方式，選擇一重播操作來執行。如所示，TOQ 430 包含一 TOQ 項目緩衝區 432，設置成將對應至每一 RIB 項目的 TOQ 項目儲存在 SM 分割區 410 內。如結合第六 A 圖至第六 B 圖內進一步描述，當接收到一新存取要求並撤回該 TOQ 項目來完成該存取要求時，TOQ 430 在 TOQ 項目緩衝區 432 內建立一 TOQ 項目。TOQ 項目緩衝區 432 的大小係根據 SM 分割區 410 的數量以及每一 RIB 416 內 RIB 項目的數量。例如：在一 SM 310 包含「P」個 SM 分割區 410，並且每一 SM 分割區 410 都包含一內含「B」個 RIB 項目的一

RIB 416 之處，TOQ 項目緩衝區 432 應包含 $P \times B$ 個 RIB 項目。如此，每一 TOQ 項目都與 RIB 416 內一對應項目配對。

【075】 TOQ 項目緩衝區 432 內的每一 TOQ 項目都包含多個欄位，包含但不受限於該相關聯存取要求的年齡、起始該存取要求的 SM 分割區 410 或其他用戶端之識別、包含該重播操作資訊的該 RIB 之識別、與該存取要求相關聯的該共用資源以及該等資源的狀態、與該存取要求相關聯的喚醒條件。使用已知的技術可明確記錄一存取要求的年齡，例如透過一優先權法則、最近使用過或最近分配過方式。如結合第十圖的進一步描述，利用到達時間以及該共用資源存取要求在一儲存佇列，例如 TOQ 項目緩衝區 432 內的位置來暗示年齡，如結合第四圖至第五圖的描述。另外，利用到達時間以及與該共用資源存取要求相關聯的一重播操作在一儲存佇列，例如 TOQ 項目緩衝區 432 內的位置來暗示年齡。

【076】 在此可關於與該 TOQ 項目相關聯的該存取要求來描述一共用資源的狀態。如結合第八圖的進一步描述，利用更新一共用資源記錄在該對應 TOQ 項目內的狀態，該共用資源可分配給一存取要求。例如：一共用資源的狀態可指示該存取要求不需要該共用資源。在另一個範例中，該存取要求需要該共用資源，但是已經分配給該存取要求。在另一個範例中，該存取要求需要該共用資源，但是尚未分配給該存取要求。在另一個範例中，該共用資源可分成幾個部分，例如目前使用的快取線，成一集合時，代表一或更多快取線的群組，或一通道時，代表一已知集合內一特定快取線。仍舊在另一個範例中，該資源可受阻並且該集合或通道使用中。如結合第七圖的進一步描述，隨著該共用資源的狀態情況改變，可定期更新該

TOQ 項目的該共用資源狀態欄位。

【077】 在所需資源尚未分配、已鎖定或使用中之處，TOQ 430 決定該資源短缺是短期或過度狀態亦或是長期狀態。具備短期或過度資源衝突的存取要求則負責立即執行或在有限延遲之後執行。具備長期或不確定資源短缺的存取要求可進入睡眠狀態。如結合第十一圖的進一步描述，TOQ 430 可在發生特定喚醒條件時喚醒睡眠中的存取要求。對應至睡眠狀態中一存取要求的該 TOQ 項目可指定一或更多這些喚醒條件，例如：該資源可用於分配、該資源已經解除鎖定或快取型資源的所需集合或通道不再使用。

【078】 正在睡眠的 TOQ 項目正在等待一或多個喚醒條件。符合喚醒條件時，TOQ 430 可選擇喚醒單一睡眠中 TOQ 項目，或廣播一喚醒操作給多個睡眠項目。例如：在該資源為設置成一次傳第一個資料值的一先進先出(FIFO, first-in first-out)記憶體之處，TOQ 430 應該喚醒等待該 FIFO 可用的單一 TOQ 項目。在另一個範例中，TOQ 430 應該廣播一喚醒操作給等待相同快取線填入資料的多個 TOQ 項目。在這種情況下，該快取線為比能夠滿足進入睡眠而等待該快取線可用的所有 TOQ 項目還要多的一資源。在某些情況下，一 TOQ 項目可存在而不需要喚醒條件。這種 TOQ 項目有資格選擇用於立即重播操作，並且適合造成該共用資源衝突為過度或短期性質之情況。

【079】 在一個具體實施例內，於每一執行循環期間，TOQ 430 選擇一或多個重播操作的最舊合適的存取要求。一合適的存取要求為已經分配所有必須資源並且已經滿足所有喚醒條件之要求。該睡眠狀態內的存取要求並不適合選取。TOQ 430 根據針對該項目所發現的該全重播操作衝

擊、要求事先排定重播操作的次數以及要求隨選重播操作的次數。一衝擊包括與一特定共用資源存取要求相關聯的一或多個重播操作。

【080】 當一共用資源存取要求原本就派遣用於執行時，並且假設該要求並未由 SM 310 或仲裁單元 455 預先清空，該重播操作計數器(未顯示)設定為零，並且根據接收自 AGU 420 和 L1 快取記憶體 460 的該要求狀態，將該全衝擊發現計數器(未顯示)初始化。然後，若該存取要求因為該預先排程重播操作而未完成，則 TOQ 430 累積與隨選重播操作相關聯的衝擊。

● 【081】 吾人應該了解，TOQ 430 容易受到僵持或活鎖情況的影響。一僵持發生，例如若兩個存取要求中的每一個都需要兩個共用資源，每一存取要求都分配給兩個資源之一者，並且每一存取要求都在等待其他資源變成可用的睡眠情況下。在這種情況下，沒有一個存取要求可獲得該額外資源，並且該等存取要求會永久留在睡眠狀態下。在另一個範例中，當一需求的資源變成可用時，一睡眠存取要求應該進入清醒狀態，並且該資源應該指派給一第二存取要求。不過，該第二存取要求應該等待另一個資源，● 因此應該回到睡眠狀態。等待該共用資源的其他存取要求應該同樣進入清醒狀態，分配給與該共用資源相關聯的一埠，並且回到睡眠狀態。此程序會繼續到該共用資源的所有埠都由睡眠要求消耗，並且該共用資源無法用於任何現有存取要求。

【082】 當在一或多個存取要求需要該相同共用資源，並且至該共用資源的所有存取埠都已經分配給其他要求時，則可特別宣布僵持情況。這種共用資源可稱為稀缺資源。在一個具體實施例內，當供應稀少時，TOQ 430 可具有一機制來管理資源。在一共用資源似乎是稀缺資源時，存取該稀缺

資源的優先權可給予現有較舊的 TOQ 項目以及任何相關聯預排程重播操作，而非與最近派遣存取要求相關聯的較年輕 TOQ 項目。在至少一個具體實施例內，若至少一個存取要求目前在等待該資源變成可用的睡眠狀態下，則可考慮一共用資源為稀缺資源。可變成稀缺的資源範例包含但不受限於一先進先出(FIFO)記憶體、一快取型記憶體內的特定集合與通道或快取型記憶體的特定集合內任何通道。一旦決定一共用資源為稀缺資源，則 TOQ 430 不會傳送廣播喚醒操作。當考慮一資源為稀缺時，TOQ 430 可使用單一喚醒操作，而非一廣播喚醒操作，將該資源指派給一單獨存取要求。TOQ 430 可使用該 TOQ 項目的年齡，在此該年齡可為 TOQ 項目緩衝區 432 內的該 TOQ 項目的位置，將該稀缺資源分配給一睡眠 TOQ 項目。一旦該指派的存取要求完成執行，則該稀缺資源變成可用，並且可指派給另一個存取要求。一旦一共用資源不再認定為稀缺，也就是無睡眠存取要求在等待該資源，則 TOQ 430 可恢復傳送與該資源相關聯的廣播喚醒操作。

【083】 在特定情況下，利用允許較舊 TOQ 項目所需但是目前已經分配給一較年輕 TOQ 項目的一共用資源，從該較年輕 TOQ 項目當中「偷」回，可解決僵持情況。若兩 TOQ 項目都需要多個資源，則兩項目都無法分配到所有需要的資源。在一個範例中，當許多情況隨著該存取要求的執行循環而演變，一存取要求所需的共用資源集合會隨時間推移。較舊的 TOQ 項目應需要兩個資源，識別為 A 和 B，其中只有 B 已經分配給該存取要求。較舊的 TOQ 項目應該進入等待資源 A 的睡眠狀態。雖然較舊的 TOQ 項目在該睡眠狀態下，較年輕的 TOQ 項目應需要共用資源 B 和 C，但是只分配到資源 C。該較年輕資源可進入等待資源 B 的睡眠狀態。一旦清醒，該較

舊 TOQ 項目應發現該 TOQ 項目此時需要當前已經分配給該較年輕 TOQ 項目的資源 C。該較舊 TOQ 項目與該較年輕 TOQ 項目都在等待已經分配給該其他 TOQ 項目的一共用資源之分配。一種解決這種僵持情況的方式為：允許該較舊 TOQ 項目從一較年輕 TOQ 項目「偷」該需要的共用資源。

【084】 在一個具體實施例內，TOQ 430 可運用虛擬偷竊，在此即使並非所有需要資源都已經分配，該較舊 TOQ 項目可選取用於一或更多重播操作。在該重播操作執行期間，該相關聯存取要求可消耗保留給較年輕項目的資源。該資源保留可不綁定至該較年輕 TOQ 項目，如此該較舊項目可收集所有需要的資源讓進度往前。一旦該較舊 TOQ 項目進度往前，則該相關聯的存取要求可釋放所需資源，讓該共用資源可分配給較年輕 TOQ 項目。

【085】 在另一個具體實施例內，TOQ 430 可運用實際偷竊，在此可從一特定 TOQ 項目偷取一特定資源，並分配給需要該資源的較舊 TOQ 項目。在這種情況下，該較年輕 TOQ 項目所需的資源之狀態可從已分配狀態改成未分配狀態，並且該等資源的狀態可轉換成一已分配狀態給該較舊 TOQ 項目。一旦該較舊 TOQ 項目的進度足夠，則該相關聯的要求釋放該已分配資源，讓該較年輕 TOQ 項目獲得所需資源。

【086】 在許多具體實施例內，發現一資源衝突時可立即觸發資源偷竊，或資源偷竊可在發現衝突之後延遲一固定或可變時間週期。資源偷竊可限制在最舊的 TOQ 項目，在此該最舊的 TOQ 項目偷竊所有需要、未分配的資源。在這種情況下，潛在消耗該最舊 TOQ 項目所需資源的所有其他存取要求都擱置到至少該最舊 TOQ 項目進度些微前進。

【087】 共用資源的許多設置都可考慮在本發明的範圍內。在一個具體實施例內，TOQ 430 管理的某些存取要求可具有多個處理階段，在此每一階段可能需要的該等資源可變。在一個範例中，一存取要求包含應屬於不同位址空間的執行緒，在此一非一般記憶體階層的一快取等級部分滿足一個位址空間，並且由該一般記憶體階層內未含的一自洽式本機記憶體滿足另一個位址空間。在另一個具體實施例內，一存取要求可選擇相鄰記憶體位置內的一順序運算元，在此每一執行緒都可存取不同順序的運算元。

【088】 第五圖根據本發明的一個具體實施例，例示第四圖中該 TOQ 430 內的一 TOQ 項目老化單元 500。如所示，TOQ 項目老化單元 500 包含一要求狀態分配單元 510、一資源狀態分配單元 520、一 TOQ 項目緩衝區 432、一保留項目計數器 550 以及一優先權選擇器 560。

【089】 要求狀態分配單元 510 將有關共用資源存取要求的狀態分配給 TOQ 項目緩衝區 432。要求狀態分配單元 510 接收沿著信號路徑 472 來自 AGU 420 的位址產生單元要求狀態、沿著信號路徑 478 來自仲裁單元 455 的仲裁狀態以及沿著信號路徑 482 來自 L1 快取記憶體 460 的 L1 快取要求狀態。然後要求狀態分配單元 510 將來自這三個單元的該要求狀態資訊傳送至 TOQ 項目緩衝區 432 內的每一 TOQ 項目單元 540。接著該狀態資訊儲存在每一 TOQ 項目單元 540 之內的一適當位置內。

【090】 資源狀態分配單元 520 將有關共用資源的狀態分配給 TOQ 項目緩衝區 432。資源狀態分配單元 520 接收沿著信號路徑 484 來自 L1 快取記憶體 460 的 L1 快取資源狀態。然後資源狀態分配單元 520 將該資源狀態資訊傳送至 TOQ 項目緩衝區 432 內的每一 TOQ 項目單元 540。接著該狀

態資訊儲存在每一 TOQ 項目單元 540 之內的一適當位置內。

【091】 TOQ 項目緩衝區 432 包含一 TOQ 項目單元 540，用於每一 SM 分割區 410 之內的 RIB 416 中的每一項目。TOQ 項目緩衝區 432 建立與接收自圖式左側上 SM 分割區 410 的新存取要求相關聯之 TOQ 項目。TOQ 項目緩衝區 432 將 TOQ 項目單元 540 位移到右邊，當成該 TOQ 項目年齡。如所示，TOQ 項目緩衝區 432 包含 TOQ 項目單元 540 以及 TOQ 多工器 530。

【092】 TOQ 多工器 530 選擇 TOQ 項目單元 540 是否循環 TOQ 項目緩衝區 432 的該當前狀態，或將 TOQ 項目單元 540 位移至右邊當成接收自 SM 分割區 410 的新存取要求。要循環 TOQ 項目單元 540，TOQ 多工器 530 沿著標示為 580、584、588 的信號路徑選擇該循環路徑。要位移 TOQ 項目單元 540，TOQ 多工器 530 沿著標示為 582、586、590 的信號路徑選擇該位移路徑。

【093】 TOQ 項目單元 540 設置成儲存與接收自 SM 分割區的共用資源存取要求有關之資訊。TOQ 項目單元 540 隨著分別來自要求狀態分配單元 510 與資源狀態分配單元 520 的要求狀態以及資源狀態改變而更新。一開始 TOQ 項目單元 540 和相關聯的 TOQ 多工器 530 都設置成循環。TOQ 項目單元[0]的該當前狀態沿著信號路徑 580 循環、TOQ 項目單元[1]的該當前狀態沿著信號路徑 584 循環並且 TOQ 項目單元[2]的該當前狀態沿著信號路徑 588 循環。當從該 SM 分割區接收一新存取要求時，TOQ 項目單元[0]設置成接收並儲存沿著信號路徑 582 與該新存取要求相關聯的一 TOQ 項目。若 TOQ 項目單元[0]的該當前狀態指示一無效的 TOQ 項目，則循環剩餘的 TOQ 項目單元 540。不過，若 TOQ 項目單元[0]標示成一有效的 TOQ

項目，則該 TOQ 項目位移至下一個 TOQ 項目單元 540。在這種情況下，TOQ 項目單元[1]接收沿著信號路徑 586 儲存在 TOQ 項目單元[0]內的該 TOQ 項目。若 TOQ 項目單元[1]的該當前狀態指示一無效的 TOQ 項目，則循環剩餘的 TOQ 項目單元 540。不過，若 TOQ 項目單元[1]標示成一有效的 TOQ 項目，則該 TOQ 項目位移至下一個 TOQ 項目單元 540。在這種情況下，TOQ 項目單元[2]接收沿著信號路徑 590 儲存在 TOQ 項目單元[1]內的該 TOQ 項目。該處理持續到發現一無效的(空白) TOQ 項目單元 540 為止。

【094】 保留項目計數器 550 分析標記為有效的 TOQ 項目單元 540，並且決定至少一個有效 TOQ 項目所需每一共用資源之保留計數。一已知資源的保留計數等於需要該資源並標記為有效的 TOQ 項目單元 540 之數量。保留項目計數器 550 將該等保留計數傳送給每一該等對應共用資源，例如 L1 快取記憶體 460。

【095】 優先權選擇器 560 選擇要服務的一 TOQ 項目，其關聯於一已知 SM 分割區 410 內的一 RIB 416。一般來說，優先權選擇器 560 選擇最右邊 TOQ 項目單元 540 內標記為有效，並且已經保留所有需要資源來完成執行的該 TOQ 項目。請注意，該選取的 TOQ 項目並不需要儲存在最右邊並且標記為有效的 TOQ 項目單元 540 內。如此，在有效 TOQ 項目單元 540 的群組中間內可能出現空 TOQ 項目單元 540 或孔洞。這些孔洞可填入接收自 SM 分割區 410 的新存取要求。優先權選擇器 560 傳送 RIB 讀取要求，並且要求將 RIB 項目撤退至 SM 分割區 410。

【096】 請注意，可運用替代技術移動 TOQ 項目的區塊，而非使用上述的位移暫存器方式。此外，關聯於 TOQ 項目的靜態資訊，例如未直接

關於睡眠與喚醒條件的欄位，可儲存在 TOQ 項目單元 540 內，或例如靜態記憶體這類個別記憶體結構內。

【097】 第六 A 圖和第六 B 圖根據本發明的一個具體實施例，揭示用於新增共用資源存取要求至一全順序佇列 430 的方法步驟流程圖。雖然已經結合第一圖至第五圖的系統來描述該等方法步驟，精通技術人士將了解，設置來執行該等方法步驟(以任何順序)的任何系統都在本發明範疇內。

【098】 如所示，方法 600 從步驟 602 開始，在此 TOQ 430 接收與一共用資源相關聯的一存取要求。在步驟 604 上，TOQ 430 決定該存取要求是否為一重播操作要求。若該存取要求並非一重播操作要求，則該存取要求為一新要求，並且方法 600 前往步驟 606，在此 TOQ 430 遞增該存取要求所需的每一共用資源的保留計數。在步驟 608 上，TOQ 430 決定該存取要求所需的該共用資源是否可分配給多個要求。若該共用資源可分配，則方法 600 前往步驟 610，在此 TOQ 430 遞減該存取要求所需的每一共用資源的該保留計數。

【099】 在步驟 612 上，TOQ 430 決定該存取要求是否準備完成。若無剩餘工作要完成該要求，並且 L1 快取 320 已經接受該要求，則該存取要求準備完成。若該存取要求準備完成，則方法 600 前往步驟 614，在此 TOQ 430 完成該要求。然後存取對應至該 TOQ 項目的該 RIB 項目，以完成該要求。該 RIB 項目包含與該存取要求相關聯，用於完成一或多個重播操作的執行資訊。在步驟 616 上，TOQ 430 撤回與該存取要求相關聯的 TOQ 和 RIB 項目，若有的話。然後方法 600 終止。

【0100】 此時回到步驟 604，若該存取要求為一重播操作要求，則方

法 600 前往步驟 622，在此 TOQ 430 決定與該存取要求相關聯的該 TOQ 項目是否指出每一所需共用資源都可用或先前已經保留。若每一所需共用資源都可用或已經保留，則方法 600 返回步驟 610，如上述。在步驟 622 上，若至少一個所需共用資源不可用或未保留，則方法 600 前往步驟 624，在此 TOQ 430 決定是否可從另一個存取要求偷取不可用或先前未保留的該共用資源。若可偷取該共用資源，則方法 600 返回步驟 610，如上述。若無法偷取該共用資源，則方法 600 前往步驟 626，在此 TOQ 430 保留該存取要求所需的共用資源並維持可用。TOQ 430 用此資訊更新該 TOQ 項目。在步驟 628 上，TOQ 430 讓該 TOQ 項目進入睡眠。然後方法 600 終止。

【0101】 此時回到步驟 608，若該等存取要求所需的一或多個共用資源無法分配，則方法 600 前往步驟 630，在此 TOQ 430 將一 TOQ 項目分配給此存取要求。在這種情況下，會延遲服務該存取要求。在步驟 632 上，TOQ 430 決定該存取要求是否錯失任何長期資源。若該存取要求錯失任何長期資源，則方法 600 返回步驟 626，如上述。若該存取要求並未錯失任何長期資源，則方法 600 前往步驟 634，在此 TOQ 430 將與該存取要求相關聯的該 TOQ 項目標記為可重播。然後方法 600 終止。

【0102】 此時回到步驟 612，若該存取要求無法完成，則方法 600 前往步驟 618，在此 TOQ 430 完成該存取要求的一個衝擊。一衝擊包含與一特定共用資源存取要求相關聯的一或多個重播操作。然後存取對應至該 TOQ 項目的該 RIB 項目，以完成該要求。該 RIB 項目包含與該存取要求相關聯，用於完成該衝擊內含的一或多個重播操作之執行資訊。在步驟 620 上，TOQ 430 維持與該存取要求相關聯的 TOQ 和 RIB 項目，若有的話。然

後方法 600 終止。

【0103】 一旦已經建立一 TOQ 項目給一相關聯共用資源存取要求，則 TOQ 430 可更新該 TOQ 項目當成該 TOQ 項目變更所需共用資源的狀態。

【0104】 第七圖為根據本發明的一個具體實施例，用於更新一共用資源狀態的方法步驟流程圖。雖然已經結合第一圖至第五圖的系統來描述該等方法步驟，精通技術人士將了解，設置來執行該等方法步驟(以任何順序)的任何系統都在本發明範疇內。

● 【0105】 如所示，方法 700 從步驟 702 開始，在此 TOQ 430 接收一或多個 TOQ 項目所需之一共用資源的已更新狀態。在步驟 704 上，TOQ 430 決定該狀態更新是否表示該共用資源可用。若該共用資源不可用，則終止方法 700。

● 【0106】 不過，若該共用資源可用，則方法 700 前往步驟 706，在此 TOQ 430 計算最高編號 TOQ 項目的該 TOQ 索引，在此一較高編號 TOQ 項目代表比一較低編號 TOQ 項目還要舊的存取要求。在步驟 708 上，TOQ 430 決定該索引所參照的該 TOQ 項目是否為一有效 TOQ 項目。若該相關聯存取要求已經完成並且接著已經撤回該 TOQ 項目，則該 TOQ 項目無效。若該 TOQ 項目無效，則方法 700 前往步驟 710，在此 TOQ 430 遞減該索引。在步驟 712 上，TOQ 430 決定該索引是否大於或等於零。若該索引並未大於或等於零，則表示已經掃描整個 TOQ，並且方法 700 終止。

【0107】 在步驟 712 上，若該索引大於或等於零，則方法 700 返回步驟 708，如上述。在步驟 708 上，若該 TOQ 項目有效，則方法 700 前往步驟 714，在此 TOQ 430 決定該索引參照的該 TOQ 項目是否需要狀態已經改

變的共用資源。若該 TOQ 項目不需要該資源，則方法 700 返回步驟 710，如上述。不過若該 TOQ 項目需要該資源，則方法 700 前往步驟 716，在此 TOQ 430 將該資源標記為分配給與該索引相關聯的該 TOQ 項目。在步驟 718 上，TOQ 430 決定該共用資源是否仍舊可用。若該共用資源仍舊可用，則方法 700 返回步驟 710，如上述。在這種情況下，在該先前分配之後該共用資源仍舊可用，並且仍舊可分配給其他 TOQ 項目。若該共用資源不再可用，則終止方法 700。

【0108】 如上面關聯於第六 A 圖至第六 B 圖所述，TOQ 項目可進入一或多個所需共用資源的睡眠等待可用性。一旦該所需共用資源變成可用，則 TOQ 430 喚醒該 TOQ 項目，讓該 TOQ 項目進入啟用狀態，如此該相關聯的存取要求可繼續執行。

【0109】 第八圖為根據本發明的一個具體實施例，用於將一共用資源分配給與一共用資源存取要求相關聯的一 TOQ 項目之方法步驟流程圖。雖然已經結合第一圖至第五圖的系統來描述該等方法步驟，精通技術人士將了解，設置來執行該等方法步驟(以任何順序)的任何系統都在本發明範疇內。

【0110】 如所示，方法 800 開始於步驟 802，在此 TOQ 430 決定與一特定索引相關聯的該 TOQ 項目是否為一有效 TOQ 項目。若該相關聯存取要求已經完成並且接著已經撤回該 TOQ 項目，則該 TOQ 項目無效。若該 TOQ 項目無效，則終止方法 800。

【0111】 若該 TOQ 項目有效，則方法 800 前往步驟 804，在此 TOQ 430 接收該 TOQ 項目所需共用資源的一更新狀態。在步驟 806 上，TOQ 430

決定該共用資源的狀態是否等於「不需要」，在此「不需要」表示該 TOQ 項目不需要該共用資源。在這種情況下，不需要變更該 TOQ 項目。方法 800 前往步驟 816，在此 TOQ 430 決定該 TOQ 項目是否等待額外資源。若該 TOQ 項目不等待其他資源，則終止方法 800。若該 TOQ 項目等待其他資源，則方法 800 返回步驟 804，如上述。

● **【0112】** 此時回到步驟 806，若該共用資源的狀態不等於「不需要」，則方法 800 前往步驟 808，在此 TOQ 430 決定該共用資源的狀態是否等於「需要並已分配」。「需要並已分配」狀態表示該 TOQ 項目需要該共用資源，並且已經分配給該 TOQ 項目。在這種情況下，不需要變更該 TOQ 項目。方法 800 前往步驟 816，如上述。

● **【0113】** 此時回到步驟 808，若該共用資源的狀態不等於「需要並已分配」，則方法 800 前往步驟 810，在此 TOQ 430 決定該共用資源的狀態是否等於「集合/通道使用中並已鎖定」。「集合/通道使用中並已鎖定」狀態表示該 TOQ 項目已經進入睡眠，因為該 TOQ 項目需要該共用資源，但是該共用資源的該所需集合或通道已經使用，並且該資源已經鎖定。在這種情況下，方法 800 前往步驟 818，在此 TOQ 430 決定該資源目前是否未鎖定。若該資源鎖定，則方法 800 前往步驟 816，如上述。若該資源未鎖定，則方法 800 前往步驟 820，在此 TOQ 430 決定該共用資源的該集合與通道此時是否可用。若該集合與通道不可用，則方法 800 前往步驟 816，如上述。若該集合與通道可用，則方法 800 前往步驟 822，在此 TOQ 430 決定該資源是否可用。若該資源不可用，則方法 800 前往步驟 816，如上述。若該資源可用，則方法 800 前往步驟 824，在此 TOQ 430 將該資源的狀態設定為「需

要並已分配」，表示此時該共用資源已經分配給該索引所參照的該 TOQ 項目。然後方法 800 前往步驟 816，如上述。

【0114】 此時回到步驟 810，若該共用資源的狀態不等於「集合/通道使用中並已鎖定」，則方法 800 前往步驟 812，在此 TOQ 430 決定該共用資源的狀態是否等於「集合/通道使用中」。「集合/通道使用中」狀態表示該 TOQ 項目已經進入睡眠，因為該 TOQ 項目需要該共用資源，但是該共用資源的該所需集合或通道已經使用。在這種情況下，方法 800 前往步驟 820，如上述，決定該資源以及該集合和通道目前是否可用。

【0115】 此時回到步驟 812，若該共用資源的狀態不等於「集合/通道使用中」，則方法 800 前往步驟 814，在此 TOQ 430 決定該共用資源的狀態是否等於「需要但不可用」。「需要但不可用」狀態表示該 TOQ 項目已經進入睡眠，因為該 TOQ 需要該共用資源，但是該資源不可用。在這種情況下，方法 800 前往步驟 822，如上述，決定該資源目前是否可用。

【0116】 此時回到步驟 814，若該共用資源的狀態不等於「需要但不可用」，則方法 800 前往步驟 816，如上述。

【0117】 一旦一 TOQ 清醒並且啟用，則 TOQ 430 可選擇該 TOQ 用於一或多個重播操作，以便該相關聯的存取要求可恢復執行。

【0118】 第九圖為根據本發明的一個具體實施例，用於在一全順序佇列(TOQ) 430 內選擇一共用資源存取要求來服務之方法步驟流程圖。雖然已經結合第一圖至第五圖的系統來描述該等方法步驟，精通技術人士將了解，設置來執行該等方法步驟(以任何順序)的任何系統都在本發明範疇內。

【0119】 如所示，方法 900 從步驟 902 開始，在此 TOQ 430 起始一

索引，指向 SM 310 內的一第一分割區。在步驟 904 上，TOQ 430 計算該最高編號 TOQ 項目的該 TOQ 索引。在步驟 906 上，TOQ 430 決定該 TOQ 索引所參照的該 TOQ 項目是否為一有效 TOQ 項目；並且決定該分割區索引所參照的該分割區是否為一有效分割區。若該相關聯存取要求已經完成並且接著已經撤回該 TOQ 項目，則該 TOQ 項目無效。若已經識別一分割區來執行與該 TOQ 項目相關聯的該存取要求，則該分割區有效。若該 TOQ 項目或該分割區無效，則方法 900 前往步驟 908，在此 TOQ 430 遞減該索引。在步驟 910 上，TOQ 430 決定該索引是否大於或等於零。若該索引大於或等於零，則方法 900 返回步驟 908，如上述。

【0120】 若該索引並未大於或等於零，則表示已經掃描整個 TOQ，並且方法 900 前往步驟 916，在此 TOQ 430 遞增該分割區索引。在步驟 918 上，TOQ 430 決定該分割區索引是否小於分割區的編號 P。若該分割區索引小於 P，則方法 900 返回步驟 904，如上述。若該分割區索引不小於 P，則表示已經掃描整組 SM 分割區，並且方法 900 前往步驟 920，在此 TOQ 430 將任何發現的重播操作要求傳送至個別 SM 分割區。然後方法 900 終止。

【0121】 此時回到步驟 906，若該 TOQ 項目與分割區都有效，則方法 900 前往步驟 912，在此 TOQ 430 決定該 TOQ 項目是否都不需要每一共用資源，或需要該共用資源並且已經分配給該 TOQ 項目。若並非所有需要的資源都已經分配給該 TOQ 項目，則方法 900 返回步驟 908，如上述。若所有所需資源都已經分配給該 TOQ 項目，則方法 900 前往步驟 914，在此 TOQ 430 選擇該已識別分割區之內的一重播指令緩衝區來執行該存取要求。然後方法 900 前往步驟 916，如上述。

【0122】 第十圖為根據本發明的一個具體實施例，用於將一 TOQ 項目緩衝區 432 內的 TOQ 項目老化的方法步驟流程圖。雖然已經結合第一圖至第五圖的系統來描述該等方法步驟，精通技術人士將了解，設置來執行該等方法步驟(以任何順序)的任何系統都在本發明範疇內。

【0123】 如所示，方法 1000 從步驟 1002 開始，在此 TOQ 430 接收一新共用資源存取要求。在步驟 1004 上，TOQ 430 建立並儲存對應至 TOQ 項目緩衝區 432 內的該存取要求的一 TOQ 項目，在此該 TOQ 緩衝區內的项目都以暫時順序儲存。暫時順序可有任何技術上有用的方式來決定，包含但不受限於透過一優先權法則、最近使用過或最近分配方式、該 TOQ 項目緩衝區 432 內該共用資源存取要求的到達時間與位置，或由與該 TOQ 項目緩衝區 432 內該共用資源存取要求相關聯的一重播操作之到達時間與位置。對應至該新存取要求的該 TOQ 項目可佔用 TOQ 項目緩衝區 432 內對應至該緩衝區內最新位置的該位置。在 TOQ 項目緩衝區 432 內每一 TOQ 項目都位移一個位置，直到在 TOQ 項目緩衝區 432 內發現一個空位。該空位填入一現有 TOQ 項目。該空位可為 TOQ 項目緩衝區 432 內該最舊的有效 TOQ 項目。另外，該空位可為「泡泡」，在此較新的有效 TOQ 項目存在於該空位一側上，並且較舊的有效 TOQ 項目存在於該空位的另一側上。在此情況下，填滿該空位就是填滿 TOQ 項目緩衝區 432 之內的該「泡泡」。

【0124】 在步驟 1006 上，TOQ 430 決定該存取要求所需的一第一共用資源是否可分配。若該第一共用資源不可用，則該 TOQ 等待到該資源變成可用。在步驟 1006 上，若該第一共用資源可用，則方法 1000 前往步驟 1008，在此 TOQ 430 搜尋 TOQ 項目緩衝區 432，找出對應至等待該第一共

- 用資源的存取要求之 TOQ 項目子集。在步驟 1010 上，TOQ 430 識別 TOQ 項目子集內的該 TOQ 項目，在此所有所需共用資源，包含該第一共用資源，都可用。

● 【0125】 在步驟 1012 上，TOQ 430 讀取 TOQ 項目子集內該 TOQ 項目緩衝區的最舊項目，在此所有所需共用資源，包含該第一共用資源，都可用。此 TOQ 項目可或可不為 TOQ 項目緩衝區 432 內該最舊的有效 TOQ 項目。在步驟 1014 上，TOQ 430 導致一 RIB 項目，對應至步驟 1012 內的該 TOQ 項目，讀入該處理管線內，以便執行該存取要求的一或多個循環。在步驟 1016 上，TOQ 430 決定是否需要額外循環，以便完成該存取要求。若不需要額外循環，則方法 1000 前往步驟 1018，在此 TOQ 430 撤回對應至該存取要求的該 TOQ 項目。TOQ 430 也可導致撤回該對應 RIB 項目。若已撤回的該 TOQ 項目並非 TOQ 項目緩衝區 432 內之該最新有效要求，則可在 TOQ 項目緩衝區 432 內建立一「孔洞」或「泡泡」，如此有效的 TOQ 項目存在於該泡泡的兩側。然後方法 1000 終止。

● 【0126】 在步驟 1016 上若需要額外循環，則方法 1000 前往步驟 1020，在此 TOQ 430 維持對應至該存取要求的該 TOQ 項目。TOQ 430 也可導致維持該對應 RIB 項目。然後方法 1000 終止。

【0127】 第十一圖為根據本發明的一個具體實施例，用於將與一共用資源存取要求相關聯的一 TOQ 項目從一睡眠狀態轉換成一啟用狀態之方法步驟流程圖。雖然已經結合第一圖至第五圖的系統來描述該等方法步驟，精通技術人士將了解，設置來執行該等方法步驟(以任何順序)的任何系統都在本發明範疇內。

【0128】 如所示，方法 1100 從步驟 1102 開始，在此 TOQ 430 決定一共用資源現在可分配給存取要求。在步驟 1104 上，TOQ 430 決定 TOQ 項目緩衝區 432 內的一 TOQ 項目是否在睡眠狀態內，在此當該共用資源可用時，滿足該 TOQ 項目的喚醒條件。若無 TOQ 項目正在等待該共用資源，則終止方法 1100。

【0129】 在步驟 1104 上若一或多個 TOQ 項目正在等待該共用資源，則該方法前往步驟 1106，在此 TOQ 430 發出一喚醒操作。該喚醒操作可當成單一喚醒操作，發出至一特定 TOQ 項目。另外，該喚醒操作可廣播至多個等待該共用資源的 TOQ 項目。在步驟 1108 上，TOQ 430 決定在步驟 1106 內接收該喚醒的該 TOQ 項目是否正在等待仍舊不可用的其他共用資源。若該 TOQ 項目並未等待其他不可用的資源，則方法 1100 前往步驟 1110，在此 TOQ 430 允許執行該存取要求的一或多個循環。然後方法 1100 終止。

【0130】 在步驟 1108 上若該 TOQ 項目正在等待其他不可用的資源，則方法 1100 前往步驟 1112，在此 TOQ 430 決定該不可用的資源在短期或瞬間之內是否不可用。若該資源短期上不可用，則方法 1100 前往步驟 1110，如上述。若該資源長期不可用，則方法 1100 前往步驟 1114，在此 TOQ 430 決定是否從另一個 TOQ 項目偷取該不可用的資源。若無法偷取該不可用的資源，則方法前往步驟 1116，在此 TOQ 430 讓該 TOQ 項目回到睡眠狀態。然後方法 1100 終止。

【0131】 在步驟 1114 上若該不可用的資源已經遭竊，則方法 1100 前往步驟 1118，在此 TOQ 430 從其他 TOQ 項目偷取該不可用的資源。TOQ 430 可透過實際偷取或虛擬偷取來偷該資源，如上面結合第四圖所描述。在一

個範例中，只有在預定時間經過之後才能偷取該不可用的資源。在另一個範例中，只有該最舊的有效 TOQ 項目適合從其他 TOQ 項目偷取資源。然後方法 1100 前往步驟 1110，如上述。

【0132】 吾人將了解，本文中顯示的該架構僅為例示，所以可進行改變與修改。例如本說明書內描述的架構呈現在串流多重處理器 310 的載入儲存單元 303 之範疇內，但是也可運用在存取共用資源的任何處理單元內，包含但不限制於中央處理單元(CPU，central processing unit)、通用處理單元(GPU，general processing unit)或任何其他技術可行的計算環境內。仍舊在另一個範例中，每一 RIB 416 並不需要具有一致的項目數量，如此在 TOQ 430 上設置成追蹤每一 RIB 416 內的該等項目。在另一個範例中，該等技術描述於多重 SM 分割區 410 以及競爭一共用執行單元的其他用戶端之範圍內。不過，在本發明技術範圍內可支援其他組態，包含但不受限於單一用戶端存取多重執行單元，或多重用戶端存取多重執行單元。任何技術可行的方式都可服務該等許多用戶端，包含但不受限於旋轉時間片段順序、優先權型方式或動態載入。仍舊在另一個範例中，該揭示的 TOQ 430 描述於中央控制單元服務多重 SM 分割區 410 以及用於一共用執行單元的其他要求來源之範圍內。另外，可運用非中央控制，在此每一 SM 分割區 410 和用戶端可管理一個別子 TOQ，其中每一要求的年齡都相對於每一子 TOQ。在這種情況下，可利用一全體老化機制來選擇喚醒操作。

【0133】 本文內描述的該技術係以共用資源存取操作來說明，例如載入與儲存存取要求通過多重執行緒，其中該等執行緒透過分散的快取線存取記憶體位置。該等技術具有充分彈性，可運用於存在共用資源存取要求

的其他應用當中。在一個範例中，本文內描述的該等技術並不受限於執行緒的執行，而是可運用於透過存取共用資源的多功能處理單元來進行之任何操作。在另一範例中，一指令可分散通過一快取內快取線以外的一共用資源。這種資源可包含但不受限於快取標籤、快取資料、暫存列以及共享的記憶體。執行該共用資源存取要求的該等執行緒可存取該共用資源的不同態樣或部分，例如一快取記憶體內不同的集合或一快取集合內不同的通道。仍舊在另一範例中，LSU 303 可在任何已知操作期間擷取多個共用資源，例如多條快取線。仍舊在另一個範例中，一已知的共用資源同時可具有一個以上的存取埠，當所有可用的存取埠都由其他要求者消耗時則拒絕存取。

【0134】 精通技術人士將了解，本說明書中描述的該共用資源組態僅為例示，可進行改變與修改。例如：雖然本說明書內描述某些共用資源具有一集合與一通道，通常在一集合關聯快取內，不過在所描述技術的範圍內可運用其他共用資源組態，包含但不受限於任何組合之下的直接映射快取(通常每集合具有一個通道)、全關聯快取(通常具有一個集合)以及分段快取。在分段快取內，由一集合與一通道表示的一快取線應該進一步分成段落。在這種組態中，除了依照集合和通道追蹤以外，TOQ 430 應依照段落追蹤共用資源用量。在另一個範例中，可運用許多快取線置換政策，包含但不受限於最近已用(LRU, least recently used)以及最近已分配(LRA, least recently allocated)方式。

【0135】 總結來說，所揭示技術提供一種最佳方式，在關於重播操作的共用資源存取要求排程方面維持公平與秩序。尤其是，一串流多重處理

器(SM) 310 包含一全順序佇列(TOQ) 430，設置成針對在單一執行循環內未完成的存取要求，排定一或多個重播操作之上的該存取要求。若所有共用資源都已經分配給該存取要求或者可用，則允許該存取要求執行一或多個重播操作。若該存取要求等待至少一個長期資源，則 TOQ 430 將該存取要求放入睡眠狀態，等待該資源變成可用。當一需要的資源變成可用，則與該存取要求相關聯的該 TOQ 項目會進入清醒狀態。根據該存取要求的年齡以及共用資源的狀態，賦予存取要求優先權。當二或多個存取要求僵持在等待相同資源時，則該 TOQ 允許較舊的存取要求從該較年輕的資源要求偷取資源，如此每一存取要求都可完成執行。

【0136】 這樣的優點為較新的存取要求不會重複阻擋較舊的共用資源存取要求之前進進度。利用將該共用資源分配至一現有較舊存取要求優先於該新存取要求，可避免優先權反轉。因為該睡眠狀態內的存取要求不會主動完成共用資源，因此可更有效運用 SM 310 的資源。此外，允許較舊 TOQ 項目偷取分配給一較年輕項目的資源，如此與該較舊項目相關聯的該存取要求可以完成執行，接著與該較年輕項目相關聯的該存取要求可完成，如此避免僵持與活鎖。進一步，即使所需共用資源的集合在執行期間變更，一樣可維持排程存取要求內的公平與秩序。

【0137】 雖然上述指向本發明的具體實施例，不過可想出不背離本發明基本領域以及底下申請專利範圍領域的其他與進一步具體實施例。

【符號說明】

100 電腦系統

103 裝置驅動程式

102 中央處理單元

104 系統記憶體

105	記憶體橋接器	230	處理叢集陣列
106	通訊路徑	300	任務管理單元
107	輸入/輸出橋接器	302	執行單元
108	使用者輸入裝置	303	載入儲存單元
110	顯示裝置	304	本機暫存檔
112	並行處理子系統	305	管線管理員
113	第二通訊路徑	306	共享記憶體
114	系統磁碟	310	串流多重處理器
116	開關	312	經線排程器與指令單元
118	網路配接器	315	紋理單元
120-121	外接卡	320	第一層快取
202	並行處理單元	321	排程器表
204	本機並行處理記憶體	322	任務中繼資料
205	輸入/輸出單元	325	預先光柵運算
206	主介面	328	記憶體管理單元
207	任務/工作單元	330	工作分配橫桿
208	一般處理叢集	335	第一點五層快取
210	記憶體橫桿單元	340	工作分配單元
212	前端	345	任務表
214	記憶體介面	352	統一位址映射單元
215	分割單元	370	指令第一層快取
220	動態隨機存取記憶體	380	記憶體與快取互連

400	子系統	500	全順序佇列項目老化單元
410	串流多重處理器分割區	510	要求狀態分配單元
412	排程器	520	資源狀態分配單元
414	派遣單元	530	全順序佇列多工器
416	重播指令緩衝區	540	全順序佇列項目單元
420	位址產生單元	550	保留項目計數器
430	全順序佇列	560	優先權選擇器
432	全順序佇列項目緩衝區	580	信號路徑
440	位址多工器	582	信號路徑
450	寫入資料多工器	584	信號路徑
455	仲裁單元	586	信號路徑
460	第一層快取記憶體	588	信號路徑
465	讀取資料解多工器	590	信號路徑
470	信號路徑		
472	信號路徑		
474	信號路徑		
476	信號路徑		
478	信號路徑		
480	信號路徑		
482	信號路徑		
484	信號路徑		
486	信號路徑		

申請專利範圍

1. 一種用於管理要求共用資源之子系統，包括：

- 一處理管線；以及

- 一全順序佇列(TOQ)係耦接於該處理管線，其設置成執行以下步

驟：

- 接收對於一第一共用資源的一第一要求；

- 接收對於一第二共用資源的一第二要求；

- 決定該第一共用資源與該第二共用資源無法分配；

- 作為回應，延遲服務該第一要求與該第二要求；

- 導致在一重播緩衝區內建立一第一項目，其中該重播緩衝區內的該第一項目包含與該第一要求相關聯的執行資訊，用於與該第一要求相關聯的重播操作；

- 在延遲服務該第一要求之後，接收該第一共用資源的一第三要求；

- 在延遲服務該第二要求之後，接收該第二共用資源的一第四要求；

- 在接收該第三要求之後，決定該第一共用資源變成可分配；

- 在接收該第四要求之後，決定該第二共用資源變成可分配；

- 讓該第一共用資源可用於該第三要求之前，讓該第一共用資源可用於該第一要求；以及

- 讓該第二共用資源可用於該第四要求之前，讓該第二共用資源可用於該第二要求。

2. 如申請專利範圍第 1 項之子系統，其中該第一項目包括一欄位來追蹤該第一要求的年齡，其中該第一項目指出該第一要求關聯於至少該第一

共用資源。

3. 如申請專利範圍第 2 項之子系統，其中該 TOQ 另設置成執行以下步驟，在接收該第二要求時，搜尋該重播緩衝區是否有對應至與該第一共用資源相關聯的一要求之任何項目。
4. 如申請專利範圍第 3 項之子系統，其中該 TOQ 另設置成執行以下步驟，透過該第一項目識別該第一要求關聯於該第一共用資源，並且在回應上，決定應該在該第二要求之前服務該第一要求。
5. 如申請專利範圍第 2 項之子系統，其中該 TOQ 另設置成執行在該重播緩衝區內建立一第二項目給該第二要求，來追蹤該第二要求的年齡的該一步驟，其中該第二項目指出該第二要求關聯於至少該第一共用資源。
6. 如申請專利範圍第 5 項之子系統，其中該 TOQ 另設置成執行以下步驟，在決定該第一共用資源可用時，搜尋該重播緩衝區是否有對應至與該第一共用資源相關聯的一要求之任何項目。
7. 如申請專利範圍第 6 項之子系統，其中該 TOQ 另設置成執行以下步驟，透過該第一項目和該第二項目識別該第一要求和該第二要求關聯於該第一共用資源，並且該第一要求比該第二要求老舊，並且在回應上，決定應該在該第二要求之前服務該第一要求。
8. 如申請專利範圍第 2 項之子系統，其中該第一項目另包含指出該第一共用資源狀態的資訊，並且該第一共用資源狀態為，該第一要求需要該第一共用資源，但是並未分配給該第一要求，或該第一共用資源已鎖定並且無法用於該第一要求。

9. 一種計算裝置，其包括：

一處理器；以及

一全順序佇列(TOQ)係耦接於該處理器，其設置利用執行以下步

驟，管理關聯於該處理器的一處理管線內共用資源的要求：

接收對於一第一共用資源的一第一要求；

接收對於一第二共用資源的一第二要求；

決定該第一共用資源與該第二共用資源無法分配；

作為回應，延遲服務該第一要求與該第二要求；

導致在一重播緩衝區內建立一第一項目，其中該重播緩衝區內的該第一項目包含與該第一要求相關聯的執行資訊，用於與該第一要求相關聯的重播操作；

在延遲服務該第一要求之後，接收該第一共用資源的一第三要求；

在延遲服務該第二要求之後，接收該第二共用資源的一第四要求；

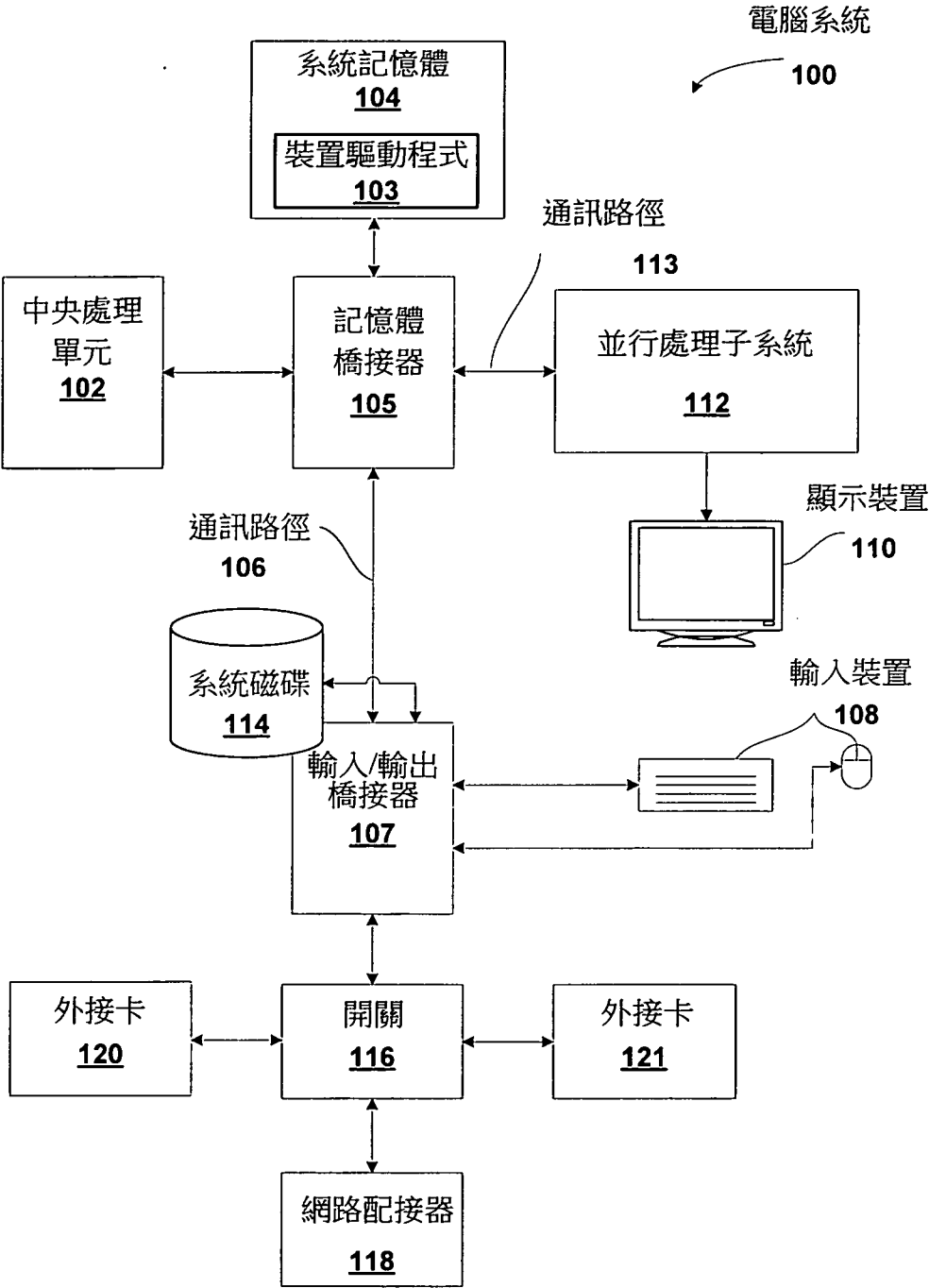
在接收該第三要求之後，決定該第一共用資源變成可分配；

在接收該第四要求之後，決定該第二共用資源變成可分配；

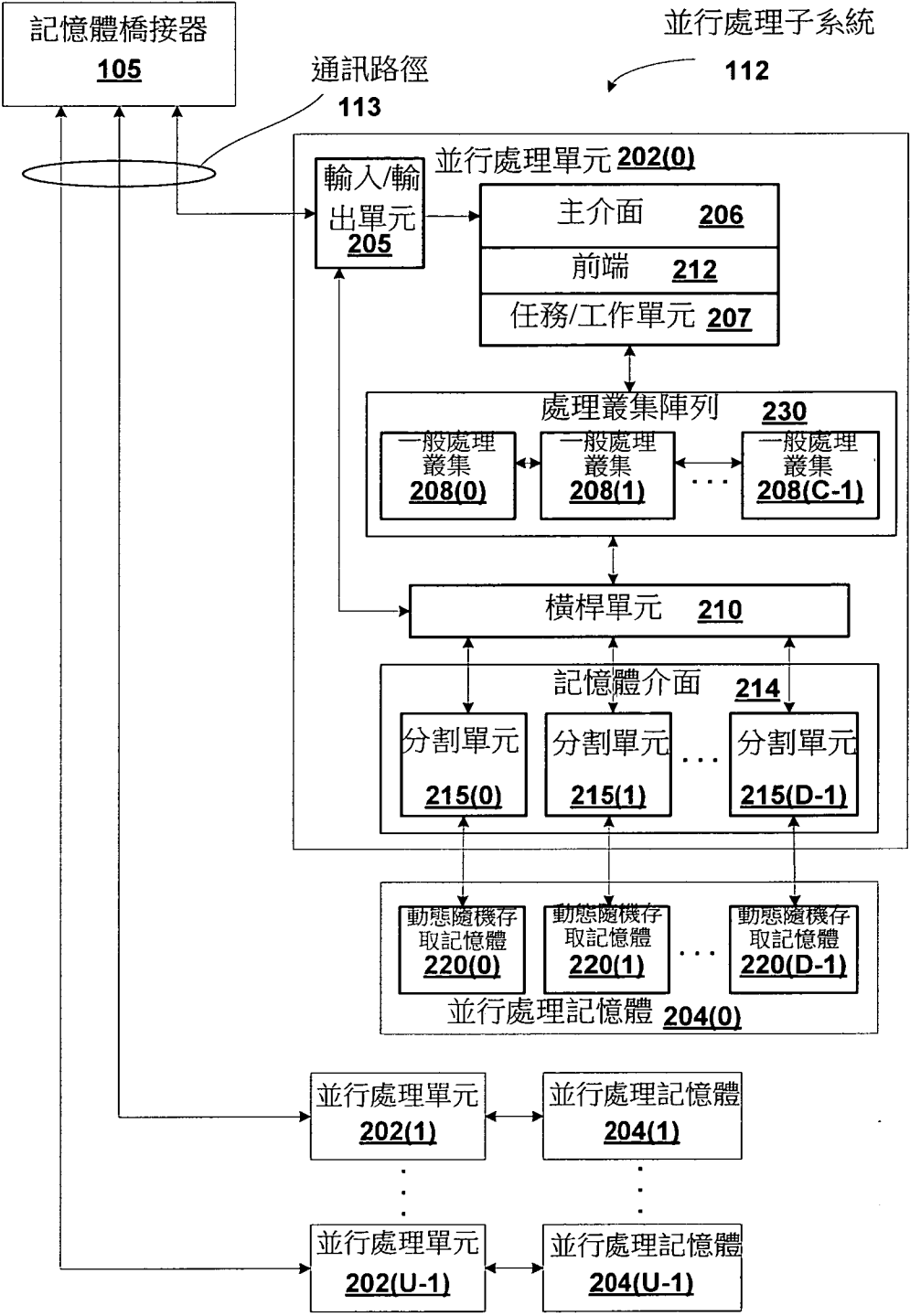
讓該第一共用資源可用於該第三要求之前，讓該第一共用資源可用於該第一要求；以及

讓該第二共用資源可用於該第四要求之前，讓該第二共用資源可用於該第二要求。

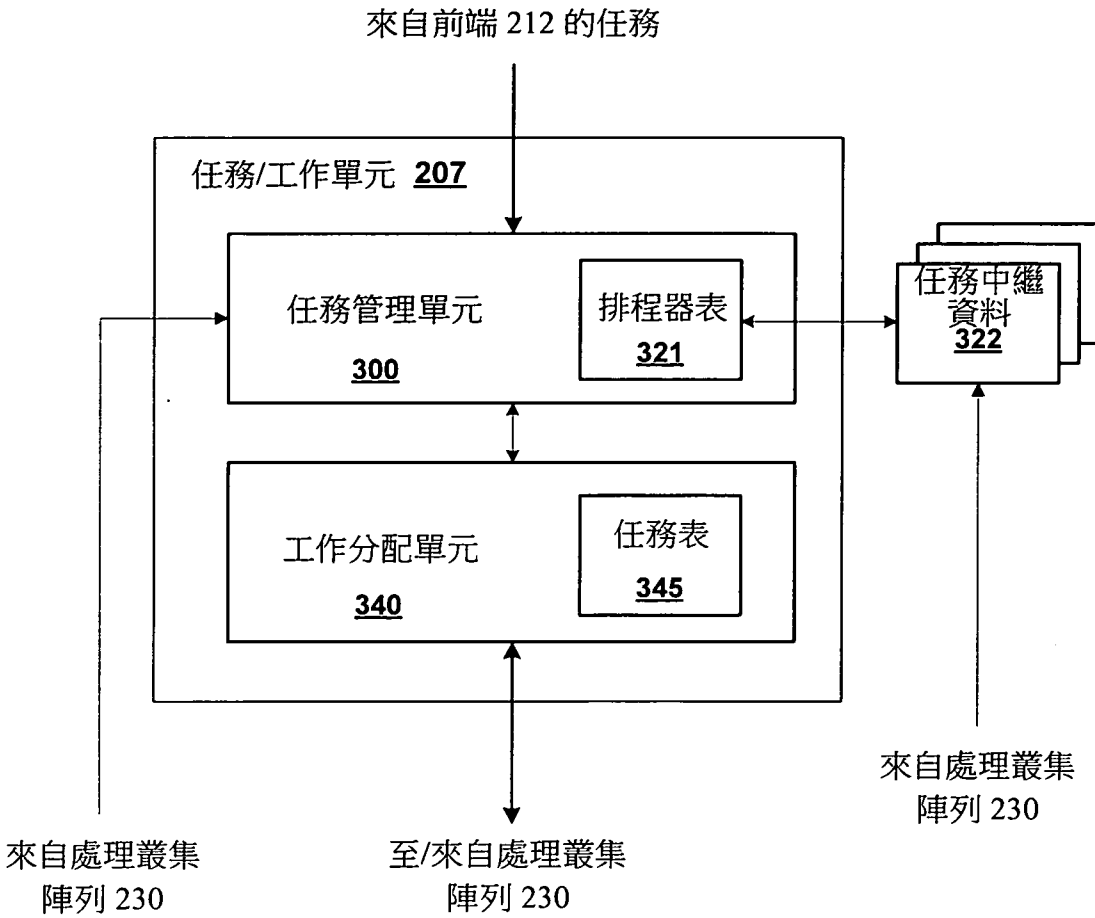
圖式



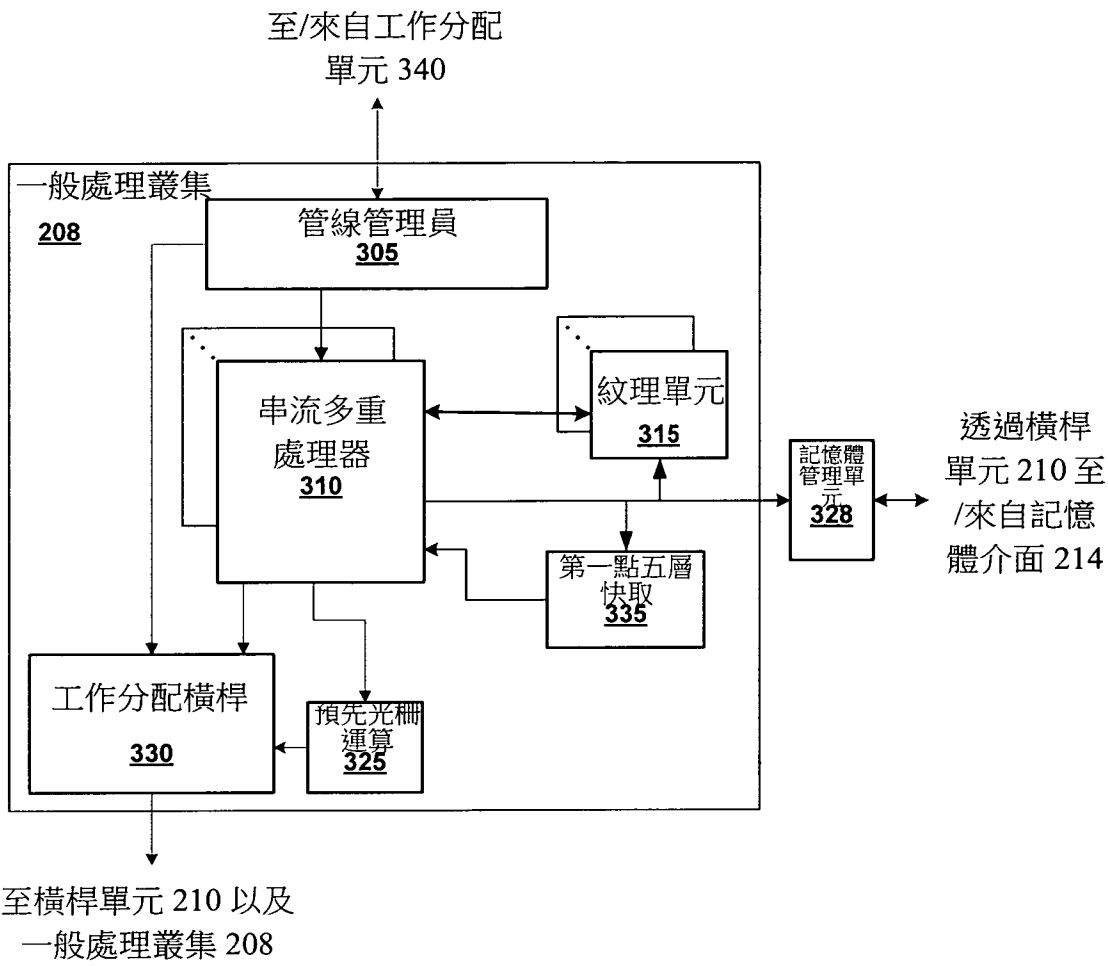
第一圖



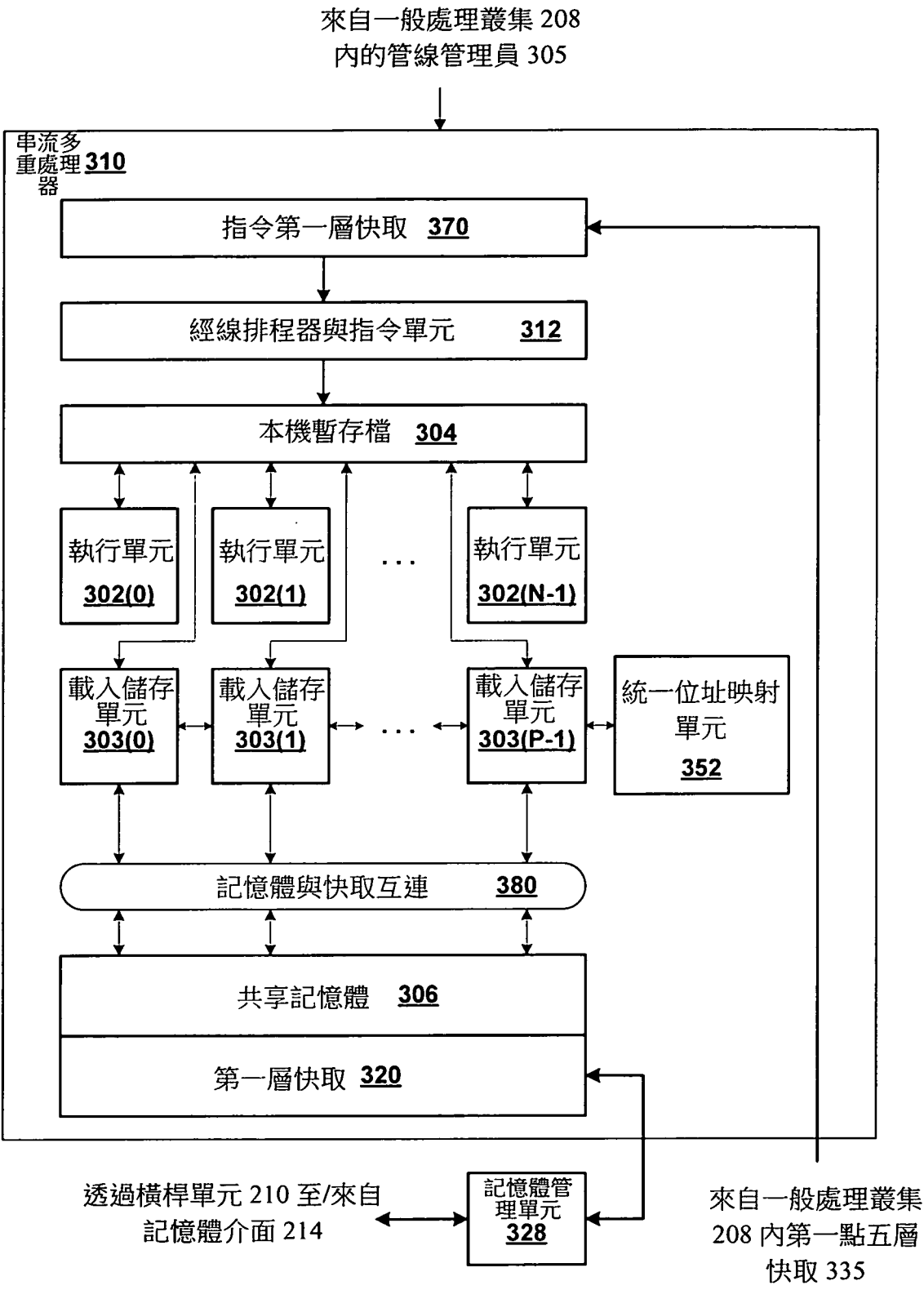
第二圖



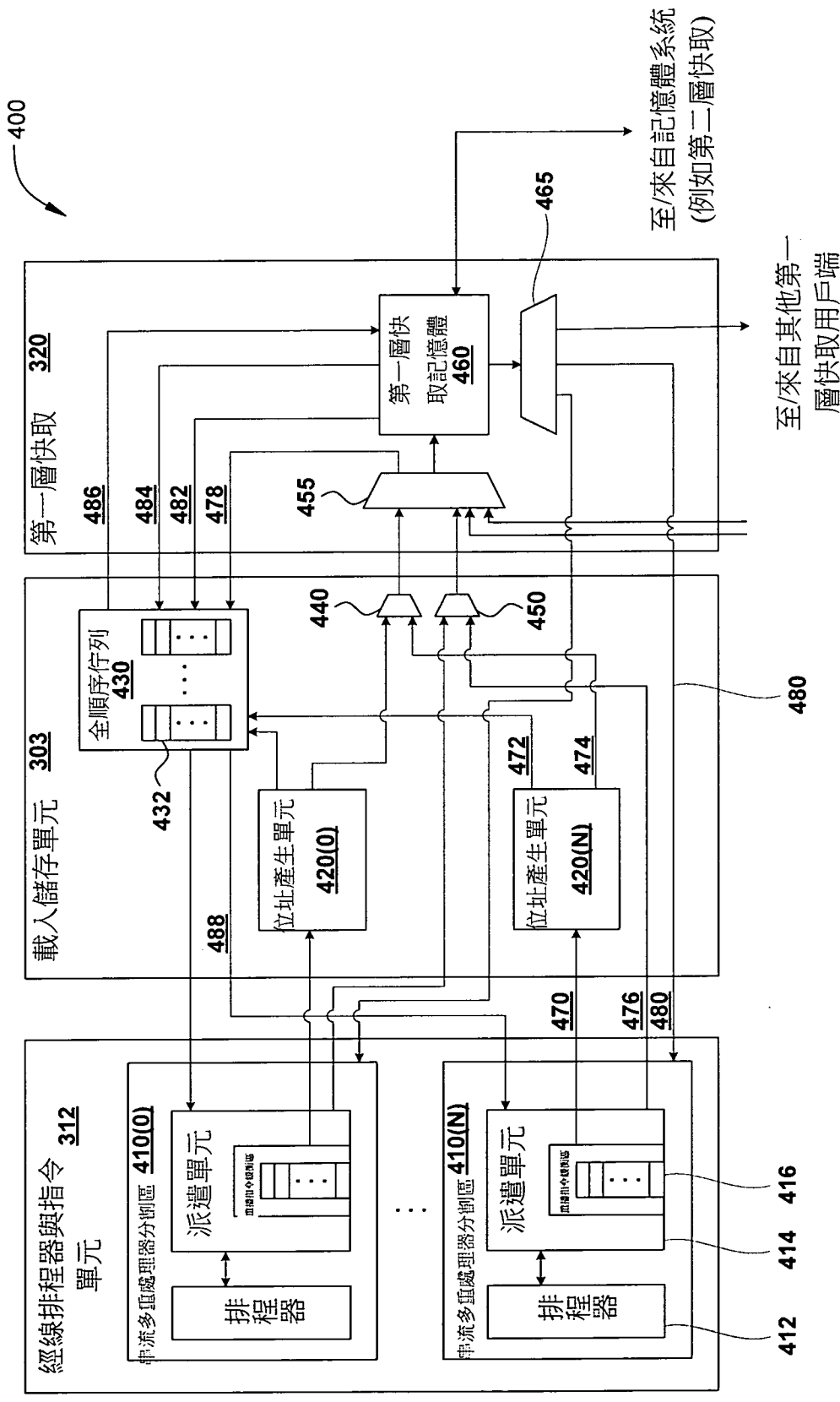
第三 A 圖



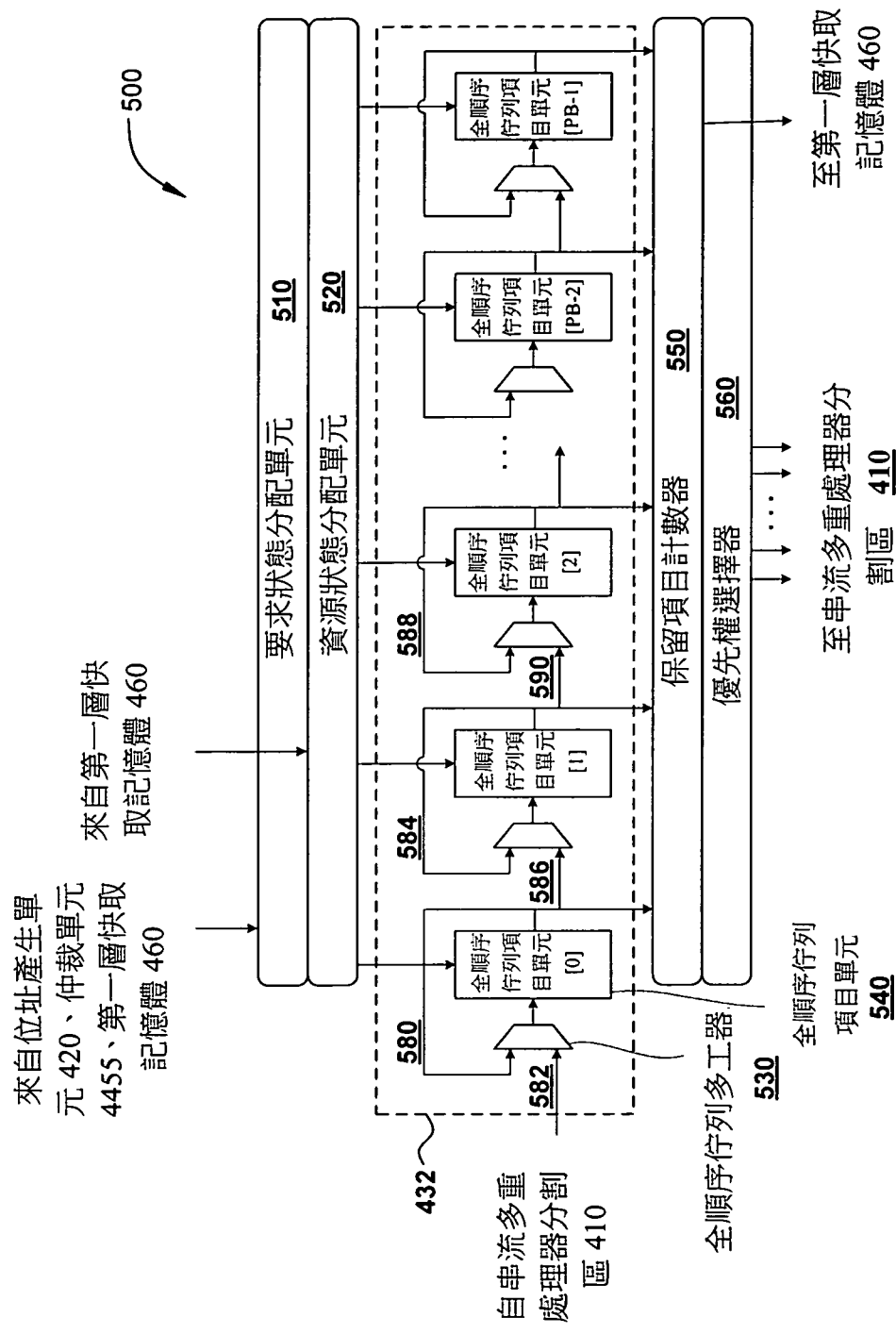
第三 B 圖



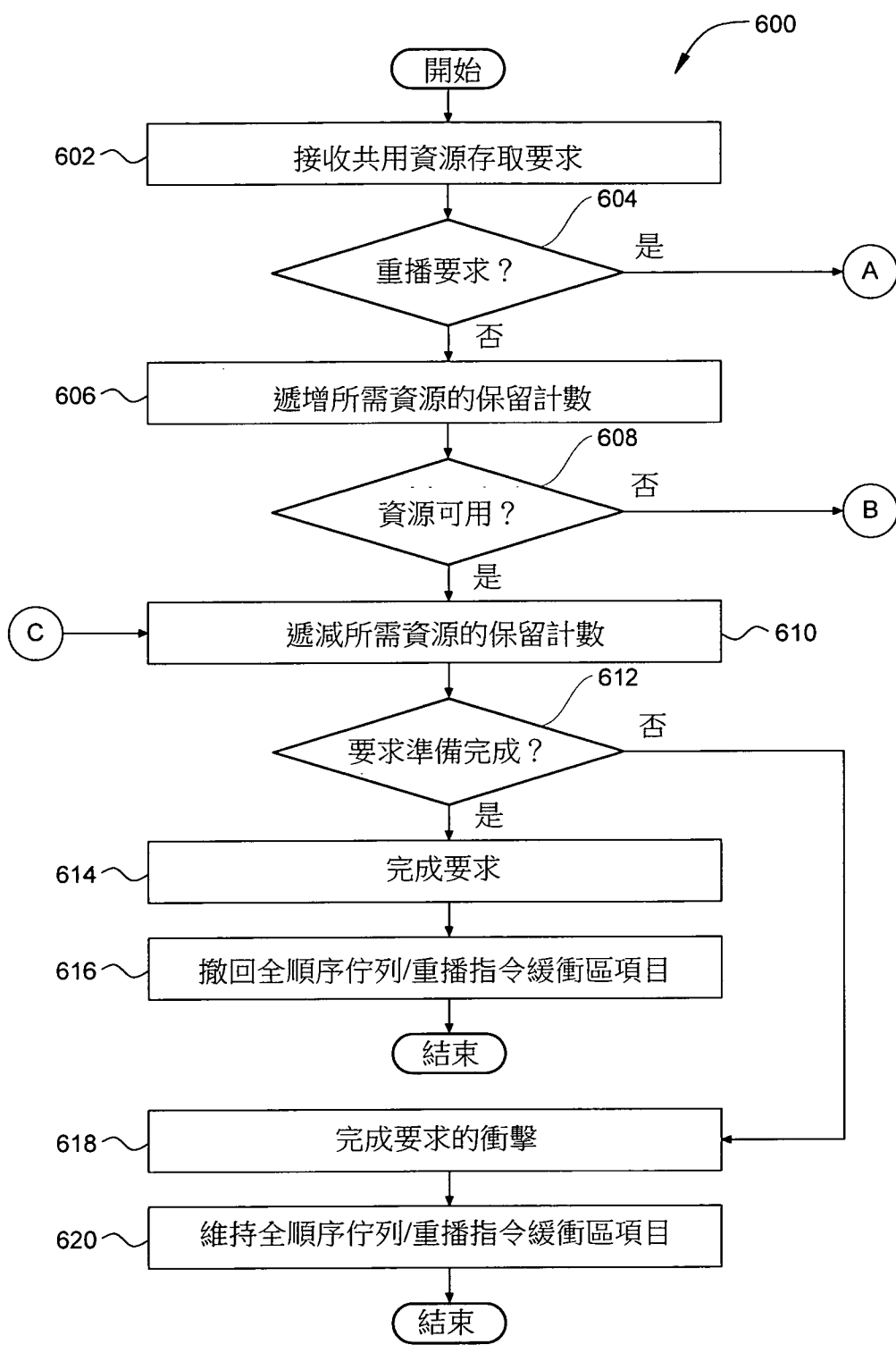
第三 C 圖



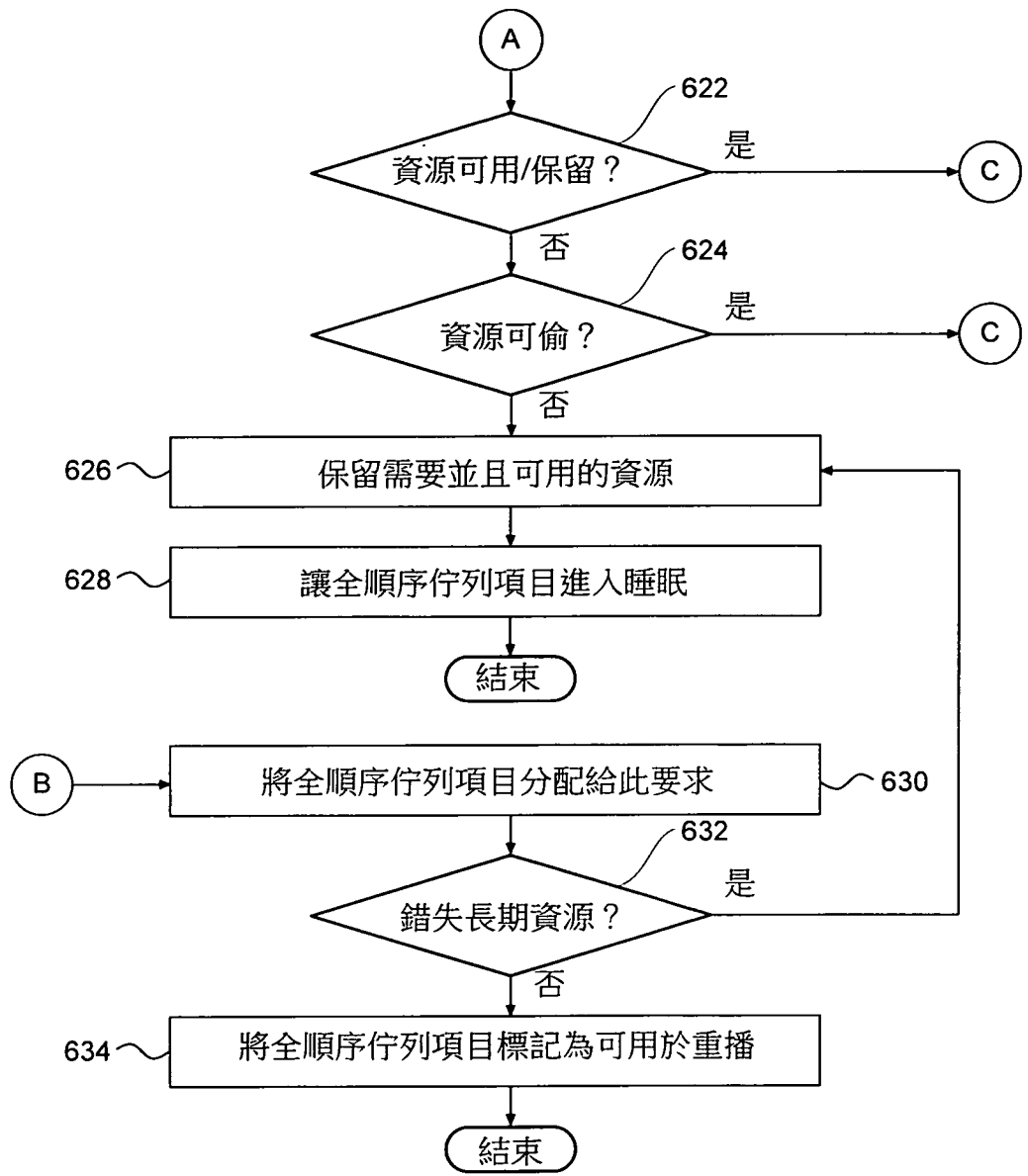
第四圖



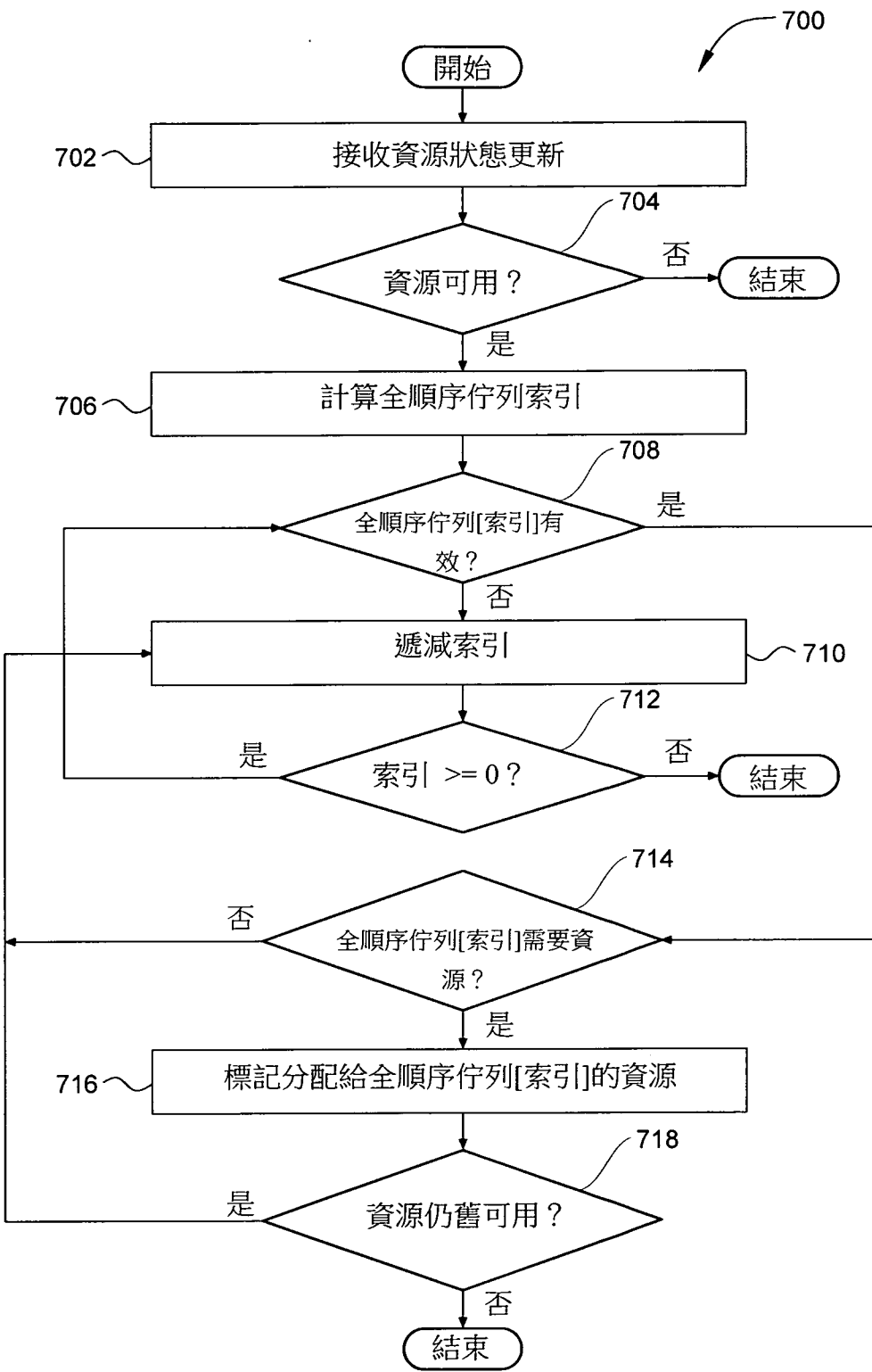
第五圖



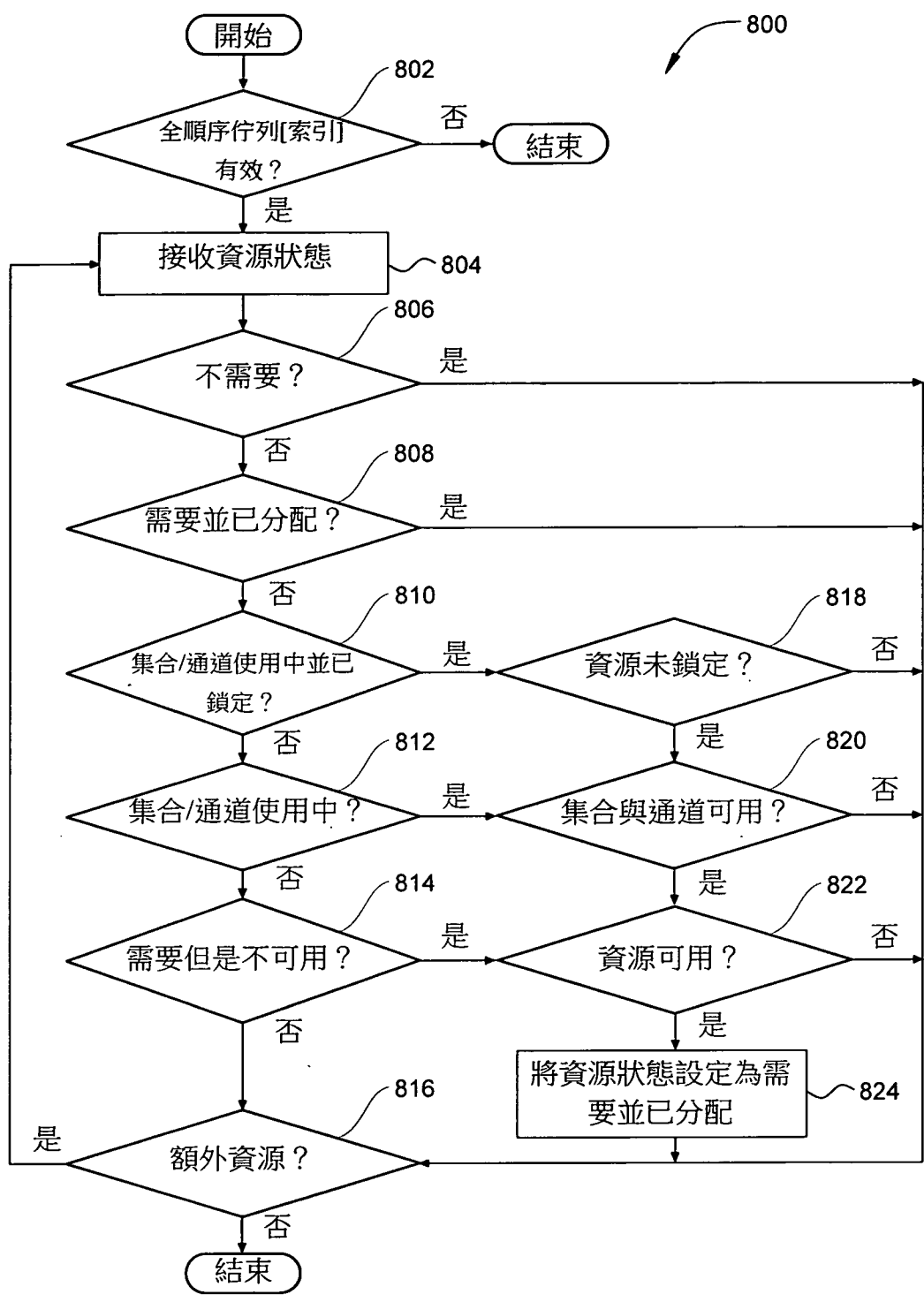
第六 A 圖



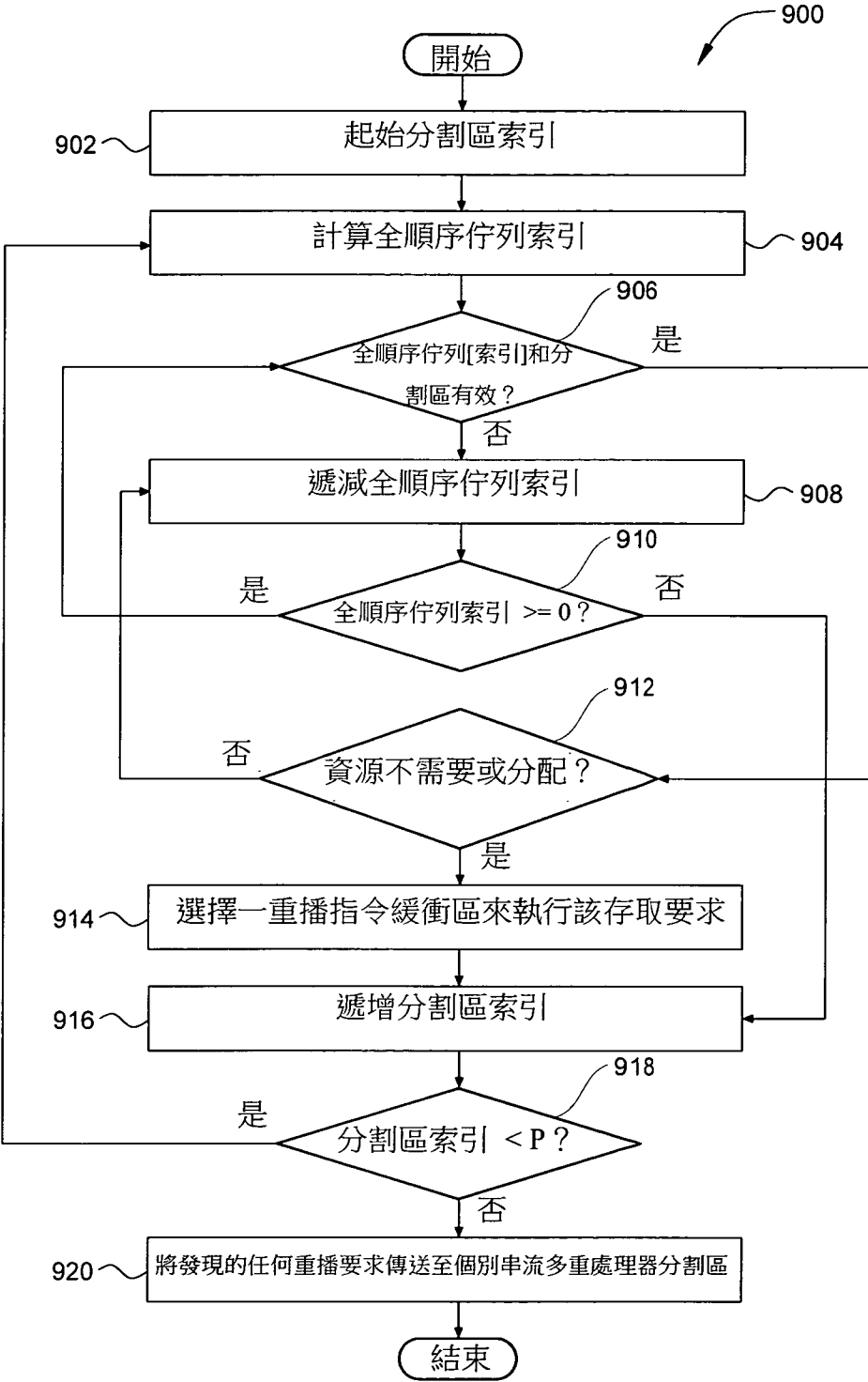
第六 B 圖



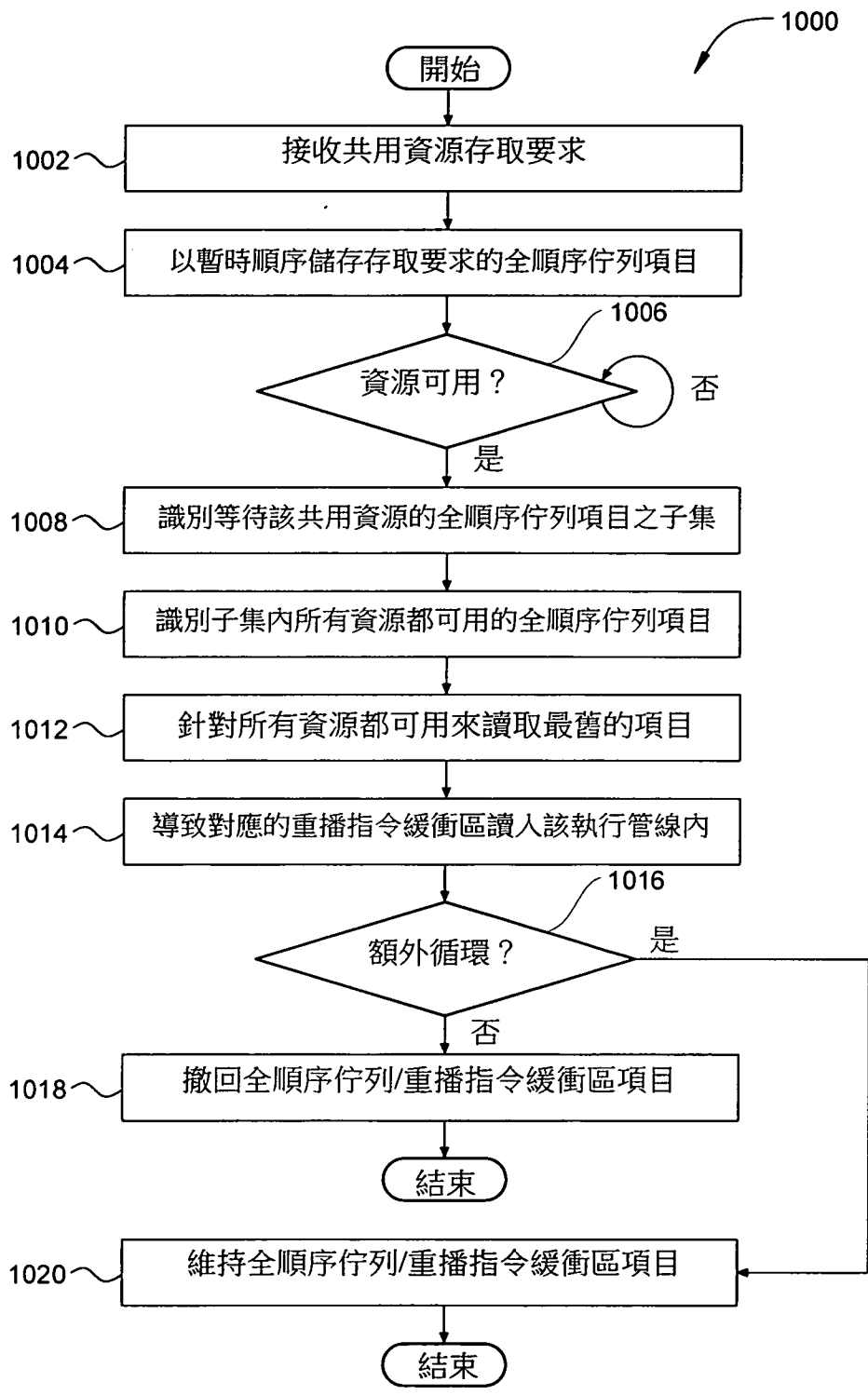
第七圖



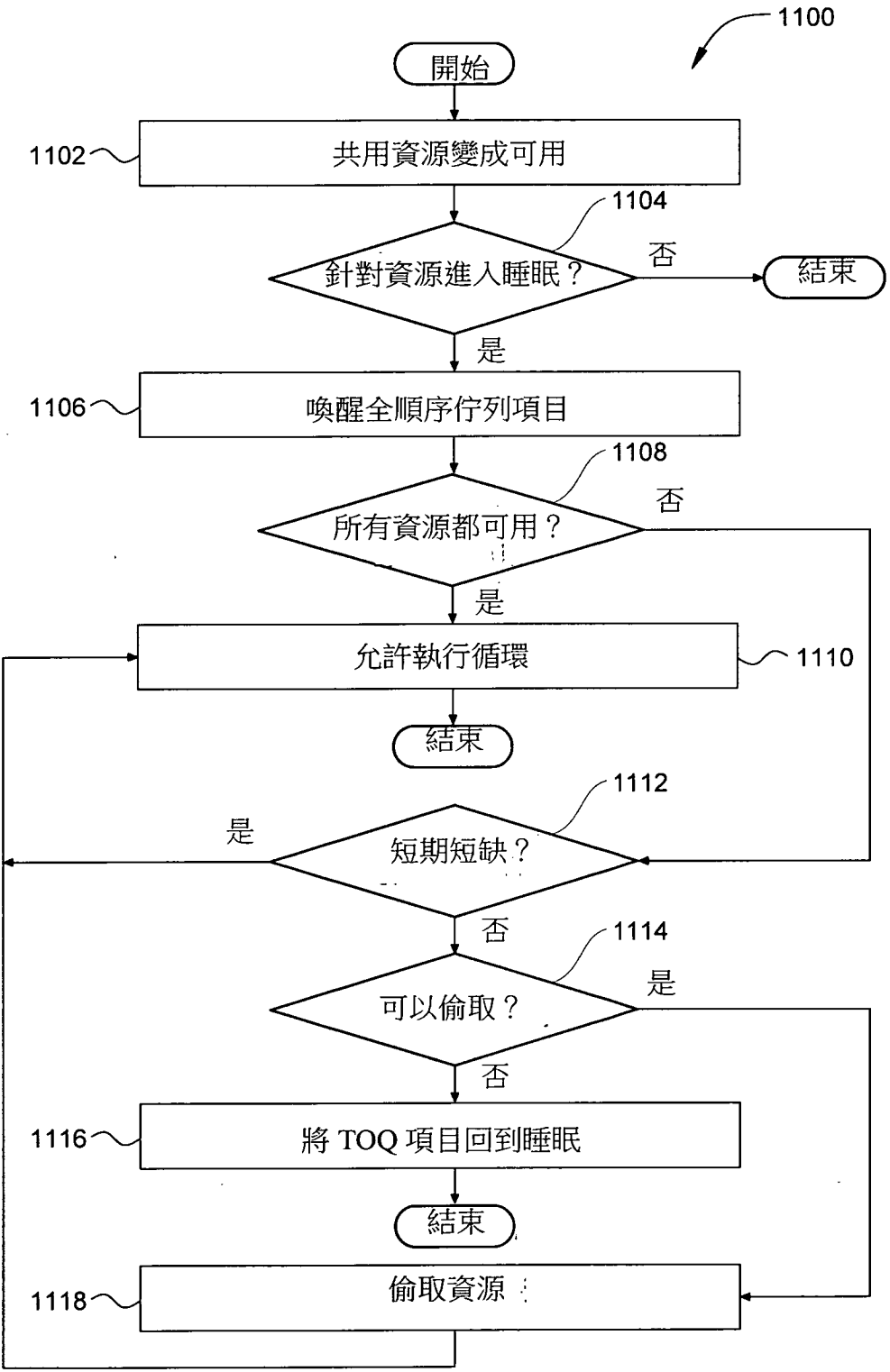
第八圖



第九圖



第十圖



第十一圖