



(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 I473013 B

(45)公告日：中華民國 104 (2015) 年 02 月 11 日

(21)申請案號：101101125

(22)申請日：中華民國 101 (2012) 年 01 月 11 日

(51)Int. Cl. : G06F9/28 (2006.01)

G06F12/02 (2006.01)

G06F15/167 (2006.01)

(30)優先權：2011/02/11 美國

13/025,446

(71)申請人：微軟公司(美國) MICROSOFT CORPORATION (US)

美國

(72)發明人：布萊克理查約翰 BLACK, RICHARD JOHN (GB)；哈里斯提姆西 HARRIS, TIMOTHY (GB)；麥克羅伊羅斯坎馬倫 MCILROY, ROSS CAMERON (GB)；施特勞斯卡林 STRAUSS, KARIN (US)

(74)代理人：蔡坤財；李世章

(56)參考文獻：

TW 389858

TW 567413

TW I283348

US 7320056B2

US 2006/0080513A1

US 2006/0259705A1

US 2008/0091884A1

審查人員：劉季涵

申請專利範圍項數：20 項 圖式數：13 共 55 頁

(54)名稱

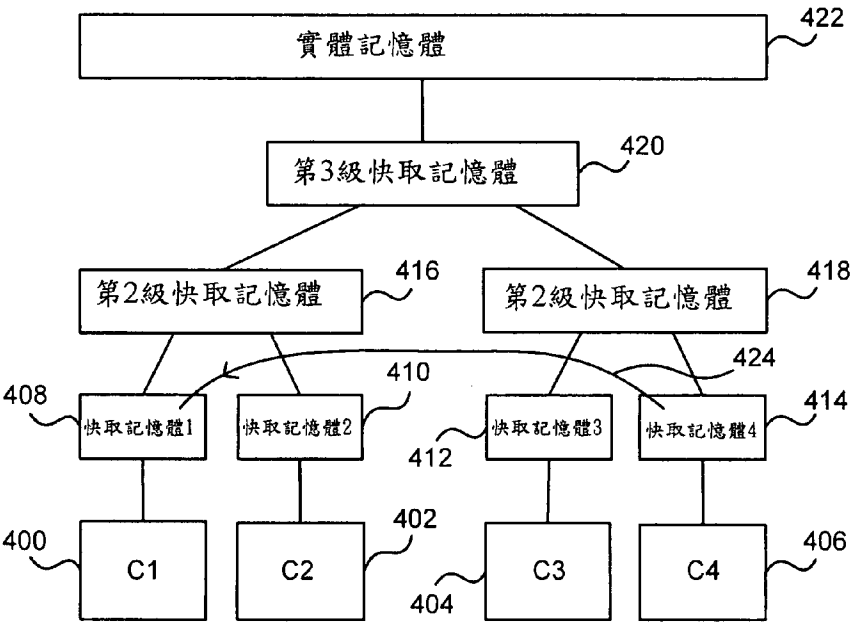
多核心電腦內的遠距核心操作

REMOTE CORE OPERATIONS IN A MULTI-CORE COMPUTER

(57)摘要

說明一種具有共享實體記憶體的多核心處理器。在一種具體實施例中，發送核心發送記憶體寫入要求至目的核心，以使要求可在目的核心上作動，如同要求係源自目的核心。在一種範例中，在共享實體記憶體中配置資料結構，並映射資料結構以可被發送核心與目的核心存取。在一種範例中，共享資料結構作為發送核心與目的核心之間的訊息通道，以使用記憶體寫入要求裝載資料。在一種具體實施例中，使用共享實體記憶體以致能通知機制，以藉由更新通知資料結構來向目的核心通知事件。在一種範例中，通知機制於目的核心觸發通知程序，以向接收程序告知通知。

A multi-core processor with a shared physical memory is described. In an embodiment a sending core sends a memory write request to a destination core so that the request may be acted upon by the destination core as if it originated from the destination core. In an example, a data structure is configured in the shared physical memory and mapped to be accessible to the sending and destination cores. In an example, the shared data structure is used as a message channel between the sending and destination cores to carry data using the memory write request. In an embodiment a notification mechanism is enabled using the shared physical memory in order to notify the destination core of events by updating a notification data structure. In an example, the notification mechanism triggers a notification process at the destination core to inform a receiving process of a notification.



- 400 . . . 核心
- 402 . . . 核心
- 404 . . . 核心
- 406 . . . 核心
- 408 . . . 第 1 級快取記憶體
- 410 . . . 第 1 級快取記憶體
- 412 . . . 第 1 級快取記憶體
- 414 . . . 第 1 級快取記憶體
- 416 . . . 第 2 級快取記憶體
- 418 . . . 第 2 級快取記憶體
- 420 . . . 第 3 級快取記憶體
- 422 . . . 實體記憶體

第 4 圖

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；惟已有申請案號者請填寫)

※ 申請案號：101101125

※ 申請日期：101 年 1 月 11 日

※IPC 分類：

G06F 9/38 2006.01
G06F 12/02 2006.01
G06F 15/16 2006.01

一、發明名稱：(中文/英文)

多核心電腦內的遠距核心操作/REMOTE CORE
OPERATIONS IN A MULTI-CORE COMPUTER

二、中文發明摘要：

說明一種具有共享實體記憶體的多核心處理器。在一種具體實施例中，發送核心發送記憶體寫入要求至目的核心，以使要求可在目的核心上作動，如同要求係源自目的核心。在一種範例中，在共享實體記憶體中配置資料結構，並映射資料結構以可被發送核心與目的核心存取。在一種範例中，共享資料結構作為發送核心與目的核心之間的訊息通道，以使用記憶體寫入要求裝載資料。在一種具體實施例中，使用共享實體記憶體以致能通知機制，以藉由更新通知資料結構來向目的核心通知事件。在一種範例中，通知機制於目的核心觸發通知程序，以向接收程序告知通知。

三、英文發明摘要：

A multi-core processor with a shared physical memory is described. In an embodiment a sending core sends a memory write request to a destination core so that the request may be acted upon by the destination core as if it originated from the destination core. In an example, a data structure is configured in the shared physical memory

and mapped to be accessible to the sending and destination cores. In an example, the shared data structure is used as a message channel between the sending and destination cores to carry data using the memory write request. In an embodiment a notification mechanism is enabled using the shared physical memory in order to notify the destination core of events by updating a notification data structure. In an example, the notification mechanism triggers a notification process at the destination core to inform a receiving process of a notification.

四、指定代表圖：

(一) 本案指定代表圖為：第 (4) 圖。

(二) 本代表圖之元件符號簡單說明：

400	核 心	402	核 心
404	核 心	406	核 心
408	第 1 級 快 取 記 憶 體	410	第 1 級 快 取 記 憶 體
412	第 1 級 快 取 記 憶 體	414	第 1 級 快 取 記 憶 體
416	第 2 級 快 取 記 憶 體	418	第 2 級 快 取 記 憶 體
420	第 3 級 快 取 記 憶 體	422	實 體 記 憶 體

五、本案若有化學式時，請揭示最能顯示發明特性的化學式：

無

六、發明說明：

【發明所屬之技術領域】

本發明係關於多核心電腦內的遠距核心操作。

【先前技術】

多核心處理器逐漸普及，並被使用於許多應用領域以增益效能，在使用多核心處理器的情況中軟體程序被平行化，使得軟體程序同時運行於多重核心上。每一核心可被想成為執行讀取並執行指令之處理器的部分；例如，每一核心可為在單一運算系統內的中央處理單元(CPU)。一般而言，單核心處理器一次僅能夠處理一個指令序列。多核心處理器具有兩個或更多個獨立核心，獨立核心可被提供於單一實體裝置（諸如晶片或晶片封裝）中。多重核心可（或可不）具有對共享實體記憶體體的存取，雖然本發明係關注具有共享實體記憶體的多核心處理器。

一個或多個共享實體記憶體可用以賦能多重核心之間的通訊，並可為隨機存取記憶體(RAM)或其他適合的記憶體方塊。通常提供階層快取記憶體以加速核心對記憶體的記憶體存取。核心、快取記憶體與記憶體通常藉由使用內連線裝置來與彼此連接。然而，以此方式使用共享實體記憶體產生了技術障礙。例如，在各種快取記憶體之間的同調性(coherency)通常使用快取記憶體同調性

協定來維持。然而，快取記憶體同調性機制在運算上是昂貴的，且使用昂貴的內連線資源。

下文所描述的具體實施例不限於解決使用共享實體記憶體的已知多核心處理器的任何(或所有)缺點的實施例。

【發明內容】

下文呈現所揭示發明之簡化概要，以使讀者能夠基本地瞭解。此概要不為所揭示發明之廣泛的概要，且不識別本發明的關鍵/重要元件，或描繪本發明的範圍。此發明內容的唯一目的為以簡化形式呈現本文所揭示的一些概念選擇，作為在其後呈現之更加詳盡描述的前言。

說明具有共享實體記憶體的多核心處理器。在一種具體實施例中，發送核心發送記憶體寫入要求至目的核心，以使要求可在目的核心上作動，如同要求係源自目的核心。在一種範例中，在共享實體記憶體中配置資料結構，並將資料結構映射為可由發送核心與目的核心存取。在一種範例中，共享資料結構作為發送核心與目的核心之間的訊息通道，以使用記憶體寫入要求來裝載資料。在一種具體實施例中，使用共享實體記憶體來致能通知機制，以藉由更新通知資料結構來通知事件的目的核心。在一種範例中，通知機制於目的核心觸發通知程序，以向接收程序告知通知。

連同附加圖式參照於下文詳細描述，將可更輕易認知並可更佳瞭解許多伴隨的特徵。

【實施方式】

下文與附加圖式一同提供詳細描述，意欲作為對當前範例的描述，且不意欲代表可建造或使用當前範例的唯一形式。描述記載了功能範例，以及建造與操作功能範例的步驟序列。然而，可由不同的範例來達成相同或等效的功能與序列。

雖然當前範例在本文中被描述並圖示說明為實施於具有共享實體記憶體，且不具有快取記憶體同調性的多核心電腦系統中，所描述的系統是被提供為範例且不為限制。如將由在本發明領域中具有通常知識者所認知，當前範例適合應用於各種不同類型的多核心電腦系統，包含具有快取記憶體同調性的多核心電腦系統。

第 1 圖為兩個多核心電腦 101、104 的示意圖，其中一個具有共享記憶體而另一個不具有共享記憶體。第一多核心電腦 101 包含兩個節點 100、102，每一節點具有核心 106，核心 106 具有快取記憶體 108。每一節點具有自身的記憶體 110，且節點如圖示說明使用網路介面 112 來通訊。相反的，本文所說明的具體實施例使用具有共享實體記憶體的多核心電腦，諸如圖示說明於第 1 圖中的電腦 104。電腦 104 具有兩個核心 114，每一核心具有

快取記憶體 116。核心共享單一實體記憶體 118，單一實體記憶體 118 可為隨機存取記憶體(RAM)或其他適合的記憶體類型。核心可為不同類型，例如，一些可被設計為用於特定工作(諸如影像處理)，而其他可被設計為用於一般用途運算。

由於各種理由，將資料寫入與寫出實體記憶體 118 是耗費時間的。例如，每一核心 114 可被提供為位於與實體記憶體 118 實體上獨立之處的晶片。為了賦能較快的資料存取，每一核心可具有相關的快取記憶體 116。雖然在第 1 圖中圖示每個核心僅具有一個快取記憶體，實際上可存在為階層結構的複數個快取記憶體，以提升可用的快取記憶體容量。使用快取記憶體能夠使資料存取加速，雖然此亦意謂在一些情況中，存取自快取記憶體的資料可能不是最新的。為了能夠使用最新的資料，先前係使用快取記憶體同調性機制。快取記憶體同調性機制可能需要大量的內連線資源。在一些情況中快取記憶體同調性機制可使用多至 30% 的多核心電腦功率與內連線資源。

第 2 圖為多核心電腦的示例性核心的示意圖。為了清楚圖示，未圖示出核心的許多傳統部件。示例性核心包含一個或更多個暫存器 202，暫存器 202 包含複數個通知控制暫存器 204，提供通知控制暫存器 204 以賦能通知程序的實施，如將於下文參照第 11 圖與第 12 圖更詳細描述的。示例性核心亦包含延伸轉譯旁路緩衝器

(translation lookaside buffer; TLB) 206，轉譯旁路緩衝器 206 經延伸以能夠實施識別複數個遠距核心的哪一者為發送程序之目的核心的程序。可選地，提供通知白名單(notification white list; NWL) 208，以在通知被保護的情況中用於實施通知程序。此將於本文件後文更詳細描述。可選地，提供線程轉譯緩衝器(thread translation buffer; TTB) 210，且亦為可選地，提供通知處理單元 212。第 3 圖為多核心電腦的另一示例性核心 300 的示意圖。為了清楚圖示，未圖示出核心的許多傳統部件。核心包含暫存器 302，暫存器 302 以相同於上文對於第 2 圖所說明的方式包含通知控制暫存器 304。提供轉譯旁路緩衝器 306 以及個別的通訊轉譯緩衝器(communication translation buffer; CTB) 308，使用通訊轉譯緩衝器 308 以能夠實施識別複數個遠距核心的哪一者為發送程序之目的核心的程序。亦可如上文參照第 2 圖說明般提供通知白名單 310。可選地提供線程轉譯緩衝器(TTB) 312，並亦可選地提供通知處理單元 314。

第 4 圖為具有階層快取記憶體與共享實體記憶體的多核心電腦示意圖。在此範例中有四個核心 400、402、404、406 (雖然此不為必要的)以及階層快取記憶體結構。可提供兩個或更多個核心。每一個核心具有第 1 級快取記憶體。例如，核心 1 400 具有第 1 級快取記憶體 408，核心 2 具有第 1 級快取記憶體 410，核心 3 具有第 1 級快取記憶體 412 而核心 4 具有第 1 級快取記憶體

414。提供兩個第 2 級快取記憶體 416、418；在此範例中，第 2 級快取記憶體之每一者由兩個第 1 級快取記憶體共享。亦提供由第 2 級快取記憶體 416、418 共享的第 3 級快取記憶體 420。第 3 級快取記憶體 420 與實體記憶體 422 通訊。第 4 圖圖示說明的階層快取記憶體結構僅為範例，並可使用具有不同的每一級快取記憶體數量與不同的快取記憶體共享設置的其他快取記憶體級結構。每一核心實體上最靠近每一核心的第 1 級快取記憶體，且因為晶片上的空間需求，此限制了第 1 級快取記憶體的實體大小。較高級的快取記憶體離核心更遠，且實體上可為較大。隨著離核心的距離增加，快取記憶體與核心之間通訊所需的時間也隨之增加。核心、快取記憶體與記憶體係由內連線來連接，且內連線用以在他們之間移動資料。使用快取記憶體同調性機制的多核心電腦，在階層快取記憶體結構分支之間的內連線上發送資料與元資料，以提供最新的資料至核心程序。發送資料與元資料越過階層快取記憶體結構的內連線耗費時間，且複雜並使用密集的資源。

在一種具體實施例中，不對多核心電腦提供快取記憶體同調性機制。例如，第 4 圖中的核心 4 406 可存取記憶體位置(其值存在於快取記憶體 414 中)，而核心 1 可存取相同的記憶體位置(其值亦存在於快取記憶體 408 中)。若核心 4 寫入記憶體位置，則因為不具有快取記憶體同調性，核心 1 無法確定在讀取核心 1 的快取記憶體

408 時是否能看見新的值。藉由使用程序以在核心之間發送記憶體寫入要求(如現參照第 5 圖而說明)，確保了最新的資料，而無需快取記憶體同調性。例如，第 4 圖中的箭號 424 以高層級圖示說明此程序。若運行於核心 1 400 的程序從運行於核心 4 406 的程序擷取輸入，則核心 4 可經配置以發送記憶體寫入要求至核心 1 400 的第 1 級快取記憶體。此係由箭號 424 圖示說明。記憶體寫入要求被接收於快取記憶體 1 408，且可被解譯為直接從核心 1 400 接收的記憶體寫入。以此方式，來自運行於核心 4 之程序的資料，被以簡單並有效的方式提供至核心 1。可使用記憶體寫入要求，以於共享實體記憶體之資料結構中有效率地建造訊息通道。達成了效能益處。例如，提供快速路徑訊息發送作業，其在發送核心與目的核心之間以單一下層訊息操作。此減少了個別通訊的成本。此亦在實施中維持本地性(locality)，以使發送核心對目的核心的發送不會涉及一些其他核心。

在另一種具體實施例中，在諸如圖示說明於第 4 圖中的多核心電腦處提供快取記憶體同調性機制。在此情況中，可如上文所述，使用相同的在核心之間發送記憶體寫入要求的程序。在此情況中，發送記憶體寫入要求的程序能夠使快取記憶體同調性機制的負擔減少，此並提供了效能益處。

參照第 4 圖所說明的範例使用單一共享實體記憶體。然而，在核心之間發送記憶體寫入要求的程序，以及其

他本文說明的程序，亦可應用至使用多於一個共享實體記憶體之架構。例如，每插槽具有一記憶體的不均勻記憶體架構(non-uniform memory architecture; NUMA)。

第 5 圖為在多核心電腦中的核心之間發送記憶體寫入要求、接收並執行彼指令之方法的流程圖。此方法包含位於發送核心與位於目的核心的步驟。

於實體記憶體配置資料結構，實體記憶體係由多核心電腦中的兩個或更多個核心共享。將資料結構關聯於發送核心與目的核心(其皆為多核心電腦中的核心)(500)。資料結構可為任何供軟體在發送核心與目的核心之間建造訊息通道的便利方法。關於此配置程序的更多細節係於下文參照第 6 圖以說明。

使用內連線從發送核心發送記憶體寫入要求至目的核心(504)。以此方式，對資料結構的更新，可被發送如從發送核心至目的核心的記憶體寫入要求。可於發送核心實行，以實施發送記憶體寫入要求的功能步驟範例，係於下文參照第 7 圖以說明。

在目的核心接收記憶體寫入要求(506)。可於目的核心實行，以讀取先前由記憶體寫入要求更新之值的功能步驟範例，係於下文參照第 9 圖以說明。

在目的核心執行記憶體寫入要求，如同記憶體寫入要求係來自目的核心(508)。換言之，從目的核心執行記憶體寫入要求。因為記憶體寫入要求係從目的核心寫入，對於目的核心而言，更新的可見性是同調的。在一

些範例中，此係由將值寫入目的核心的本地快取記憶體（第 1 級快取記憶體），而使值可見於運行於目的核心的程序以完成。其他範例將於本文件後文參照第 10 圖來說明。

可選地，目的核心可指示目的核心已完成處理被發送至目的核心的值。在彼情況下，接收核心執行「資料完成(data done)」指令，以指示值不再被需要（510）。可從接收核心的本地快取記憶體刪除值，而無需將值寫出至共享實體記憶體（512）。

第 6 圖說明更多配置訊息通道的細節。於發送核心與目的核心，作業系統內核(kernel)將實體記憶體共用區域映射入在彼等核心上運行的程序的位址空間。例如，在訊息通道欲用於兩個程序之間時，實體記憶體共用區域被映射入程序的虛擬位址空間。實體記憶體共用區域包含作為第 5 圖方法之部份而形成的資料結構。資料結構可為各種形式。例如，資料結構可為環形緩衝器、記憶體方塊、向量、佇列或堆積(heap)或其他資料結構。作業系統資料結構隨後被更新以配置訊息通道（例如）以指示於發送核心，寫入資料結構的任何資料係欲發送至目的核心。作業系統亦可配置為（例如）於發送核心資料結構可被寫入但不被讀取，且於目的核心資料結構可被讀取但不被寫入。使用通道發送訊息的作業，以及實施通知的作業，隨後可使用非特權碼來實行，亦稱為使用者模式。以此方式，在使用者模式中可實行一般情況

作業。例如，此允許訊息被發送自使用者模式，並傳遞至目的核心，而無需在目的核心上發生內核模式中斷。

可使用第 6 圖之程序建立許多訊息通道，且訊息通道用以發送記憶體寫入要求與通知(如下文所說明)。以此方式，程序對於大量的通訊通道而言是有效率的。例如，允許系統程序與系統中的所有其他程序通訊。

現在說明提供了如何實施第 5 圖之發送程序與接收程序之功能範例的虛擬碼作業。實際上，實施可調適已知的指令以執行該等功能。例如，藉由在已知指令之前附加前置碼，或標記虛擬或實體記憶體的特定分頁，以指示在他們上面的作業必須以實施本文所說明之功能性的方式來執行。處理器狀態旗標亦可用以指示記憶體作業欲被執行以實施本文所說明的功能性。

一種實施可要求由該等虛擬碼作業存取的記憶體位置，不被同時由可持續於多核心處理器上之其他程序之普通記憶體讀取與寫入來存取。此種同時存取可被視為一種競爭情況(race condition)形式。

參照第 7 圖以說明虛擬碼 `data_send` 作業 700。此可用以實施將記憶體寫入要求從發送核心發送至目的核心的程序(見第 5 圖步驟 504)。此作業可使用兩個引數：`v_addr` 與 `val`。`v_addr` 為在發送核心虛擬記憶體中資料結構的代表位址，`val` 為欲由記憶體寫入要求寫入的值。

發送核心可以各種方式獲取「`val`」。現在給定範例：一種實施可從處理器暫存器組的內容取得「`val`」。一種替

代性實施可從可由發送核心存取之記憶體的內容取得「val」，諸如從發送核心快取記憶體之觀點之「v_addr」的當前內容。一種實施可對 v_addr 與 val 施加限制條件，諸如 v_addr 必須對準快取記憶體線起始處，或者 val 的大小必須為完整的快取記憶體線，或者 val 的大小與 v_addr 的對準使寫入的進行不需要在較大位址範圍上的讀取-修改-寫入。

於發送核心，程序由對 v_addr 執行位址轉譯與保護查核，以獲取資料結構在實體記憶體中的相關實體位址 p_addr，而可實施 data_send 作業(702)。此係由使用轉譯旁路緩衝器來達成。發送核心程序對目的核心查閱 v_addr(704)。對於此查閱的更多細節於下文參照第 8 圖以說明。發送核心程序隨後在訊息通道上發送 remote_write(p_addr, val) 訊息給目的核心。

發送核心程序對目的核心查閱 v_addr (見第 8 圖方塊 800)。此查閱可由各種方式來達成。在一種範例中，在每一核心使用緩衝器以快取控制元資料(802)。此緩衝器可被稱為通訊轉譯緩衝器(CTB)。控制元資料可包含對於目的核心或目的快取記憶體的識別符。或者，控制元資料可包含對於目的軟體線程的識別符，且使用個別機制以將軟體線程識別符轉譯成對於目的核心或目的快取記憶體的識別符(使用線程轉譯緩衝器 TTB 為一種執行此第二映射的方法)。CTB 查閱可與轉譯旁路緩衝器查閱平行處理(806)。若未於 CTB 中找到項目(808)，則可產

生例外情況且切換執行至作業系統內核(810)。隨後，作業系統內核能夠使用由軟體管理的資料結構，來尋找適當的控制元資料，並將適當的控制元資料安裝進 CTB。隨後，失效的指令可被重新執行(816)。在另一範例中，若未於 CTB 中找到項目(808)，則可由硬體從由軟體維持的資料結構填滿 CTB (812)。在另一範例中，使用延伸轉譯旁路緩衝器(extended TLB)(804)。例如，TLB 被延伸以包含控制元資料。在此情況中第 7 圖的步驟 702 與 704，被整合為 TLB 查閱的部分。

藉由配置 CTB 及/或 TLB，系統軟體能夠控制哪些程序可通訊。達成經保護程序間通訊，表示訊息可被發送於不同的程序之間，而非僅操作於相同程序的線程之間。此使程序能夠用以實施系統功能，如於多內核(multi-kernel)或微內核(micro-kernel)作業系統中。

參照第 9 圖以說明虛擬碼 data_receive 作業 900。此可用以實施讀取先前於目的核心由記憶體寫入要求更新之資料的程序(見第 5 圖步驟 506 與 508)。data_receive 作業 900 使用一個引數 v_addr，v_addr 為在接收核心之虛擬位址空間中資料結構的代表的位址。程序於目的核心對 v_addr 執行位址轉譯與保護查核，以獲取共享記憶體(其實施訊息通道)中資料結構的實體位址 p_addr(902)。程序從本地快取記憶體中傳回 p_addr 的內容，或將 p_addr 的內容擷取入本地快取記憶體(904)。

參照第 9 圖以說明虛擬碼 data_done 作業 906。此可用

以實施表明資料不再為程序所需要的程序(見第 5 圖步驟 510 與 512)。Data_done 使用一個引數 v_addr。對 v_addr 執行位址與轉譯查核以獲取實體位址 p_addr(908)。p_addr 的內容在目的核心的本地快取記憶體中被標示為乾淨的(clean)。p_addr 的內容隨後不需被寫出至記憶體。

參照第 10 圖說明虛擬碼 remote_write 作業 1000。此可用以實施於目的核心執行記憶體寫入要求的程序(見第 5 圖步驟 508)。remote_write 使用兩個引數:p_addr 與欲由所接收之記憶體寫入要求寫入的值。若 p_addr 存在於目的核心本地之快取記憶體中，則被快取的值被更新至 val (1002)。若 p_addr 不存在於目的核心本地之快取記憶體中(1004)，則進行以下步驟之一者:將寫入前送至快取記憶體階層的下一級，作為對 p_addr 的寫入(1006)；或在於目的核心本地之快取記憶體中分配空間(可能藉由收回一些其他線)，藉以將 p_addr 插入於目的核心本地的快取記憶體(1008)。一種實施需要策略以在該兩個步驟之間選擇；在該兩個步驟之間選擇，涉及隨著資料抵達而消耗目的核心上的快取記憶體空間(而不知道在被從快取記憶體收回之前資料是否將會被讀取)，相對於使目的核心發生快取記憶體遺漏(cache miss)(若資料未儲存於快取記憶體中但隨後被存取)之間的權衡。現給定兩種策略範例。「預設前送」策略為若經快取項目存在則更新經快取項目，否則將更新前送至記憶

體。若位置被期待為 `remote_write` 要求的目標，則軟體可使用傳統的預擷取或讀取指令，以將位置帶入快取記憶體。「預設緩衝」策略為，若目的線程當前正在運行中則將更新緩衝於快取記憶體中，而若目的線程不在運行中則將更新前送至記憶體。此需要：(i)附加目的線程 ID 至 `remote_write` 要求的內容，以及(ii)附加當前線程 ID 狀態暫存器至核心。「預設緩衝」的變形為限制由 `remote_write` 要求對快取記憶體子集使用的快取記憶體空間；例如，在四路集合關聯快取記憶體(4-way set associate cache)中，`remote_write` 要求可被限制為將未經緩衝資料寫入其中一路。

現在說明使用如建立於第 5 圖中之共享記憶體中資料結構的通知程序。可對各種不同的目的使用通知，例如，以指示執行廢料收集(garbage collection)、或程序要退出、或給定訊息通道需要注意。在另一種範例中，通知可用以喚醒(wake up)睡眠核心或將核心電源切斷，作為能源節省程序的部分。可連同本文說明之發送記憶體寫入要求至遠距核心的程序來使用此通知程序，且因此可稱為「遠距通知」程序。用於通知的記憶體區域係與普通記憶體區分，並與訊息傳輸(message-passing)記憶體區域區分。用以實施訊息通道的技術，係同樣用以傳遞通知，發送「`remote_notify`」要求而非「`remote_write`」要求。

通知資料結構係配置於多核心電腦的共享實體記憶

體。將通知資料結構告知參與通知方案的每一核心(1100)。此係由類似於上文參照第 6 圖說明的方式，達成於配置階段中。在一種範例中，通知資料結構保持共享位元向量(shared bit-vector)，每一位元用以指示不同的條件。快取記憶體線大小向量可為足夠的；若需要更多位元，則可使用多重向量來建造階層。亦可使用其他類型的資料結構。可於每一核心使用專屬通知快取記憶體，以保持通知資料結構的內容(而非使用正常資料快取記憶體項目)。此可被完成以避免修改為效能關鍵的資料快取記憶體，但此不為必要的。

於發送核心，獲取關於發生了一組事件中的哪些事件的資訊(1102)。此資訊被濃縮成小型、低層級的表示。例如，資訊可包含於其上發送訊息的訊息通道列表，而小型、低層級表示可包含位元向量，每一位元關聯於一組訊息通道。

由發送核心可選地查核通知白名單(1104)。此使經保護通知方案能夠被實施，其將於下文更詳細說明。在一種實施中，通知白名單包含(addr, val)對序列，每一對指示核心可有效執行通知位址「addr」，只要 remote_notify 要求裝載的值僅包含設定於「val」中的位元。

發送核心使用如參照第 5 圖說明以建立的訊息通道，來從發送核心發送記憶體通知要求至目的核心(1106)。記憶體通知要求係寫入代表發生了事件組中的哪些事件的值。記憶體通知要求被接收於目的核心(1108)，並執

行如記憶體通知要求係源自目的核心(1110)，且隨之由邏輯 OR 運算更新通知資料結構。邏輯 OR 運算係於由記憶體通知要求指定的值與通知資料結構中的已存值之間進行。此以關於發生了哪些事件的資訊，更新通知資料結構。若通知資料結構的位址係快取於目的核心，則更新發生於快取記憶體(1112)。亦可使用其他邏輯運算，包含但不限於邏輯 AND、XOR。

若通知資料結構的位址未被快取於目的核心，則更新被前送至記憶體(1114)。注意到，記憶體通知要求被視為不同於記憶體寫入要求。此係因為記憶體通知要求需要使用邏輯 OR 運算將裝載於要求的值(「val」)與記憶體的已存內容結合，而非單純地不經處理而前送至記憶體。可能使用各種機制來執行此工作。給定兩種範例。第一，核心可將「val」替換成二進位值，其中所有位元被設為 1。此值可被前送至記憶體以代替「val」。此可造成假的通知被表明，但不會使真的通知被遺漏。第二，核心可包含通知處理單元，通知處理單元負責將「val」與已存記憶體內容結合。通知處理單元存取機器的記憶體系統、從記憶體讀取當前值、將當前值與「val」結合並寫入結果值。

在此範例中，發送核心能夠通知目的核心發生了事件組中的哪些事件。此係由更新通知資料結構來達成，目的核心能夠觀看通知資料結構。若目的核心忙碌且不觀看通知資料結構，則目的核心可遺漏事件。在另一種具

體實施例中，目的核心被輕推(nudge)或主動地被強迫，以在接收到通知時呼叫通知功能。

為了實施「輕推」通知方案，可於目的核心使用通知暫存器。在一種範例中，提供四個通知暫存器。該等暫存器經配置以保持觀看旗標、遮罩、被觀看區域以及通知目標功能(1200)。遮罩指示通知機制是否需執行「輕推」作業。遮罩與被觀看區域指示哪些特定通知會使「輕推」發生。通知目標功能為在事件發生時被主動呼叫，以輕推目的核心的任何功能。

參照第 12 圖，在從發送核心接收到通知時，目的核心檢查目的核心的觀看旗標(1202)。若旗標被設定，且若通知目的位址係於被觀看區域中，且若通知的值與遮罩的邏輯 AND 為非零，則執行輕推。為了執行輕推，核心清除觀看旗標(1204)，基元性地(atomically)讀取並清除被觀看區域的內容，並呼叫目標通知功能且以被觀看區域的先前內容作為參數。觀看旗標隨後被重置(1206)，且程序回到等待通知(1202)。

於目的核心的線程可與許多不同的訊息發送者互動，且藉由使用遠距通知程序，可能避免需要目的核心線程輪詢(poll)大量的訊息通道。每一輪詢操作將需要存取主記憶體，或需要存取保持訊息通道的標頭的本地快取記憶體線。

通知程序亦可實施於快取記憶體同調性硬體上。在此情況中，被觀看位址為一個或多個記憶體位置，且使用

同調性訊息以觸發通知。目的核心於共享或互斥模式中保持包含被觀看區域的線。在目的核心失去線時，目的核心在補償延遲(back-off delay)之後再次擷取線，並估量是否需要傳遞通知。(此可由延伸正常快取記憶體，或由個別的特殊情況單一項目快取記憶體來完成)。概言之，在一些具體實施例中，可使用快取記憶體同調性機制以內連線複數個核心。於目的核心，在告知對通知資料結構之快取記憶體同調性無效時，可從目的核心發送對通知資料結構的擷取。

如上文所述，可選地可使用通知白名單以實施具有保護的通知方案，藉此發送者能夠僅設定特定位置中的特定位元組。此可用於發送者與目的為個別的程序之情況(而非在相同程序中的線程)，或使用多重相異的發送者程序以發送通知至相同目標程序的情況。在此兩種情況中，可能防止發送者能夠任意地設定通知。例如，作業系統設置：發送者不具使用者模式存取而映射通知位址(通知資料結構的位址)。每一核心保持新的「通知白名單(NWL)」，NWL 包含相關於發送者的軟狀態(soft-state)，NWL 列出此發送者被准許發送至的(位址，遮罩)對。白名單一開始為空白，且在線程切換時被清除。在 `notify_send` 操作上，對 NWL 搜尋目標位址，且通知值被與可允許遮罩比較。若存在遺漏，或若被通知的值不被包含在可允許遮罩內，則發送者發生例外情況。

如同 CTB，發送者上的作業系統決定通知是否被允

許，且若被允許，則附加項目至 NWL。以此方式，CTB 與 NWL 被保持為分離，因為 CTB 處理聚合區域(諸如分頁)，而 NWL 處理個別位址。可提供通用的指令來以單一步驟清除 CTB 與 NWL 兩者。

在另一種具體實施例中，由儲存經准許通知位元於記憶體(對於通知發送者為唯讀)之作業系統實施 NWL，而 CTB 或 extended-TLB 對彼分頁儲存目的核心與通知記憶體的實體位址(除了 NWL 記憶體的實體位址以外)。在發送核心試圖將通知寫入分頁時，對應的 NWL 快取記憶體線首先被擷取、邏輯 AND 運算被執行而所產生的通知位元隨後作為 `remote_notify_request` 行進至目的核心。

可選地，一種實施可將虛擬位址置入 `remote_write` 與 `remote_notify` 要求(靠攏實體位址或代替實體位址)。在「以接收者轉譯」實施中，`remote_write` 與 `remote_notify` 要求僅裝載虛擬位址。轉譯與保護查核係於目的核心在 `remote_write` 程序(1002)或 `remote_notify` 程序(1110)之前執行。此種實施可省略 `data_send` 與 `notify_send` 操作中的位址轉譯與保護查核。此種實施可限制 `remote_write` 與 `remote_notify` 操作為僅發生於具有相同對共享資料結構之虛擬記憶體映射的程序之間。

可選地，目的核心可對發送核心指示，目的核心已完成處理要求(諸如 `remote_write` 要求或 `remote_notify` 要求)。目的核心 C1 可發送回應訊息至發送核心 C2。在接

收到回應訊息時，核心 C2 可確信要求訊息中指定的記憶體更新，係可為運行於核心 C1 之程序所可見。在一種具體實施例中，自動發送回應給所有要求。在另一種具體實施例中，核心 C2 指示是否需要回應。核心 C2 可使用回應以由在發出隨後要求之前，需要對於較先要求的回應，來提供 `remote_write` 與 `remote_notify` 要求系列之間的排序 (ordering)。核心 C2 亦可使用回應以經由 `remote_fence` 操作 (於下文說明)，提供 `remote_write` 與 `remote_notify` 要求與其他程序之間的排序。

本文所說明的遠距寫入程序與遠距通知程序可與作業系統功能整合，諸如機器中線程的排程、線程的搶奪 (pre-emption)、線程的遷移等等。此對多程式機器而言為有益的，其中核心可隨著時間在數個程序之間多工。現在說明關於此的更多細節。

發送者 T1 的搶奪或遷移

在 T1 搶奪時，程序在切換至在不同保護域中的線程之前清除 CTB 與 NWL。CTB 與 NWL 保持可在下次 T1 運行時重新建造的軟狀態。

在遷移時，可能確保訊息不被重新排序。例如，假設 T1 從核心 C-Old 遷移至 C-New，且彼 T1 係與核心 C2 上的 T2 通訊。實施可提供：T2 在完成處理接收自 C-Old 上的 T1 的訊息之前，不處理接收自 C-New 上的 T1 的訊息。在一種示例性實施中，可充足以延緩重新排程不同

核心上的線程，直到已經過了實施特定的延遲(例如，可被緩衝於實施內的最大訊息數量，與處理每一訊息可需要的最大時間的乘積)。此約束可為可被接受的，因為處理 `memory_write_request` 與 `memory_notify_request` 訊息，不涉及軟體處理器的執行。此延遲可經由新的「`remote_fence`」操作來對軟體暴露，「`remote_fence`」操作(i)延遲直到所有訊息被保證會被處理，或(ii)與呼叫者 CTB 中的每一遠距核心通訊，並在遠距核心已完成處理來自發送者的所有訊息時接收回應。在替代性實施中，使用回應訊息以實施「`remote_fence`」操作：「`remote_fence`」操作延遲直到已接收到來自所有已發送之要求的回應。

接收者 T2 的搶奪或遷移

注意到在 T2 搶奪時不需要特別的工作；`remote_write` 與 `remote_notify` 訊息可繼續抵達，且由上次運行 T2 的核心處理(更新主記憶體而非快取記憶體)。在 T2 已阻擋等待來自 T1 的資料時此係特別容易發生，且因此可期望允許此前送，以持續一些時間，並期待 T2 將回復到相同核心上(而非向運行 T1 的核心告知 T2 已被搶奪)。

若 T2 從 C-Old 遷移至 C-New，則可能設置如訊息不為遺失。若訊息被從 C-Old 前送至主記憶體，同時 C-New 具有本地快取記憶體中相同位置的非同調性複製品，則訊息可為遺失。因此，接收者可設置為任何發送者都覺

察到接收者的新位置。現在說明根據是否使用 TTB(線程轉譯緩衝器)的兩種替代性具體實施例。

沒有 TTB 的遷移：此問題係類似於改變虛擬至實體分頁轉譯(實際上，CTB 以與 TLB 快取關於實體記憶體中虛擬分頁位置的資訊相同的方式，來快取關於線程位置的資訊)。可使用類似於 TLB 終止操作(shoot-down)的作法來支援線程遷移。在於 C-New 上回復接收者 T2 之前，作業系統實行以下程序：

作業系統識別可能正運行發送至 T2 的線程之核心組(例如，從被維持以供 CTB 遺漏處理器使用的資訊，或從在設置 T1 與 T2 之間共享記憶體時被維持的資料結構)。

作業系統發送處理器間中斷(inter-processor interrupt; IPI)至該等核心。

IPI 處理器清除對於與 T2 共享之分頁的 CTB 項目，並執行「remote_fence」以確保恰於 IPI 之前發送的訊息，在任何訊息於 IPI 之後發送之前被接收到。

有 TTB 的遷移：TTB 提供一程度的間接性於線程 ID(例如 T2)與核心 ID(例如 C-New)之間，且因此遷移需要改變 TTB 中(而非 CTB 項目中)的映射。替代性 TTB 實施可提供 TTB 之間的硬體同調性。

在一種替代性實施中，可由將 TLB、CTB 或 TTB 關聯於記憶體中(TLB、CTB 或 TTB 為其快取記憶體)作業系統配置資料位址，並對彼配置資料結構從正執行遷移之

核心發送 `remote_write` 要求至發送核心，來執行被需要於發送核心的 TLB、CTB 或 TTB 更新。因此，TLB、CTB 或 TTB 將被以立即為發送核心所可見，而無需發送中斷的方式來更新。

於發送者 T1 改變記憶體配置

作業系統可重新配置記憶體於發送者 T1，以使發送者不再具有對實體分頁的寫入存取。例如，作業系統可移除 T1 對實體分頁的寫入准許，或者作業系統可改變虛擬至實體位址轉譯，以使給定虛擬分頁映射至不同的實體分頁。在完成此種重新配置之前，作業系統可使用「`remote_fence`」以確保先前由發送者 T1 發出的寫入，已在發生重新配置之前完成。

線程排程

在多程式系統中，對於運行於一個核心上的作業系統而言，知道何時指派給核心的給定線程變成可運行是有用的。可延伸通知機制，以藉由允許向 T2 發送通知而隱含地對作業系統表明 T2 已準備好運行，來提供此資訊。

以此延伸，除了上文所說明的使用者模式機制以外，每一核心具有內核模式通知機制。此外，每一線程具有作業系統指派線程 ID，且此 ID 被包含於發送至線程的「`remote_notify`」訊息中。若「`remote_notify`」訊息抵

達除了正運行之線程以外的線程，則發生內核模式通知。此發生如同在由目標線程 ID 索引的位元位置上，相關於由目標核心上作業系統提供的位址。(例如，作業系統可指示，內核模式通知必須被傳遞於位址 0xf0000000，而對線程 ID 256 的通知可由設定位元於位址 0xf0000020 來指示。)如使用者模式通知，所產生的值被與遮罩比較；若值與遮罩的邏輯 AND 為非零，則傳遞通知。經由中斷來傳遞內核模式通知，通常使作業系統重新排程被通知的線程(立即或遲些)。

注意到，使用位元映射與遮罩，允許內核選擇不接收對於給定線程的通知(例如，因為已經被通知)，並允許多重通知累積於硬體中(例如，若在處理給定通知的同時進一步的通知抵達)。

本文所說明的程序亦可被設置為允許使用排程之訊息傳輸的高效率整合。例如，藉由允許內核模式排程者識別接收程序何時具有可用於排程者的訊息，且允許使用者模式軟體識別各種訊息通道中何者具有可用的訊息(或相反的，具有可用空間以供新訊息被發送至可用空間)。

本文所說明的遠距寫入程序與遠距通知程序可提供以下所有者或一者或多者:執行使用者模式中的一般情況作業、經保護程序間通訊、與作業系統功能整合、對於快速路徑訊息發送作業的能力、對於大量通訊通道的作業以及使用排程之訊息傳輸的高效率整合。然而，本文

所說明的程序不必要提供所有該等功能。

本文所說明的遠距寫入程序與遠距通知程序可提供經保護程序間通訊，表示可於不同程序之間發送訊息，而非僅於操作於相同程序中的線程之間。此為有益的，例如在程序用以如於多內核或微內核作業系統中實施系統功能時。系統軟體可控制哪些程序可通訊。

本文所說明的遠距寫入程序與遠距通知程序可提供與作業系統功能的整合。例如，藉由允許訊息被發送至已被搶奪的接收者，並允許線程被遷移於機器中的核心之間，或為電源管理將整個機器中斷/回復。此對多程式機器而言為有益的，其中核心可隨著時間在數個程序之間多工。

本文所說明的遠距寫入程序與遠距通知程序可提供對大量通訊通道的高效功能性。例如，允許系統程序與系統中的所有其他程序通訊。(訊息通道係由實體記憶體中的儲存來支援，實體記憶體中的儲存在機器中相對地是豐足的，而非由專屬訊息傳送緩衝器記憶體中的空間(可為稀有的)來支援。)

第 13 圖圖示說明示例性多核心以運算為基礎之裝置 1300 的各種部件，裝置 1300 可被實施為任何運算及/或電子裝置的形式，且本文所說明的遠距寫入與遠距通知程序的具體實施例可實施於裝置 1300 中。

以運算為基礎之裝置 1300 亦包含一個或多個處理核心 1302，處理核心 1302 可為微處理器、控制器或任何

其他適合以處理可由運算執行之用於一般用途運算的指令，以及實施本文所說明之遠距寫入與遠距通知程序的處理器類型。核心 1302 可為不同的類型(諸如影像處理器與一般用途處理器的混合)，或可為相同類型。例如，核心可如上文參照第 2 圖或第 3 圖說明者。在一些範例中，例如使用晶片上系統架構的情況，處理器 1302 可包含一個或多個固定功能方塊(亦稱為特定目的核心)，固定功能方塊於硬體中(而非軟體或韌體)實施本文所說明的方法的部分。可提供包含作業系統 1308 的平台軟體，或任何其他適合的平台軟體於以運算為基礎之裝置，以使應用程式軟體 1310 能夠在裝置上執行。

核心 1302 係與內連線 1304 通訊。核心 1302 經由內連線 1304 共享實體記憶體 1306。

可使用可由以運算為基礎之裝置 1300 存取的任何電腦可讀取媒體來提供電腦可執行指令。電腦可讀取媒體可包含(例如)電腦儲存媒體，諸如記憶體 1306 與通訊媒體。電腦儲存媒體(諸如記憶體 1306)包含以任何用於儲存資訊的方法或科技實施的揮發性與非揮發性、可移除與不可移除式媒體，資訊諸如電腦可讀取指令、資料結構、程式模組或其他資料。電腦儲存媒體包含(但不限於)RAM、ROM、EPROM、EEPROM、快閃記憶體、相位改變記憶體、憶阻器或其他記憶體科技、CD-ROM、數位多功能光碟(DVD)或其他光學儲存器、磁匣、磁帶、磁碟儲存器或其他磁性儲存裝置，或可用於儲存資訊以

供運算裝置存取的任何其他非傳輸媒體。相對地，通訊媒體可實施電腦可讀取指令、資料結構、程式模組或其他資料於經調變資料訊號中，諸如載波或其他運輸機制。如本文所界定，電腦儲存媒體不包含通訊媒體。雖然電腦儲存媒體(記憶體 1306)被圖示為位於以運算為基礎之裝置 1300 中，將認知到，儲存可為分散式或位於遠距並經由網路或其他通訊鏈結(例如使用通訊界面 1312)來存取。

以運算為基礎之裝置 1300 亦包含輸入/輸出控制器 1314，輸入/輸出控制器 1314 經設置以輸出顯示資訊至顯示裝置 1316，顯示裝置 1316 可與以運算為基礎之裝置 1300 分離，或整合至以運算為基礎之裝置 1300。顯示資訊可提供圖形使用者介面。輸入/輸出控制器 1314 亦經設置以接收並處理來自一個或多個裝置的輸入，諸如使用者輸入裝置 1318(例如滑鼠或鍵盤)。可使用此使用者輸入以設定配置參數，以及使用用於一般用途運算的電腦。在一種具體實施例中，若顯示裝置 1316 為觸控顯示裝置，則顯示裝置 1316 亦可作動如使用者輸入裝置 1318。輸入/輸出控制器 1314 亦可輸出資料至除了顯示裝置以外的裝置，例如連接於本地的印刷裝置。

在本文中，名詞「電腦」被使用以代表任何具有處理能力而可執行指令的裝置。在本發明領域中具有通常知識者將理解，此種處理能力被併入許多不同的裝置，且因此名詞「電腦」包含個人電腦、伺服器、行動電話、

個人數位助理與許多其他的裝置。

可由在有形儲存媒體上，為機器可讀取形式的軟體，來執行本文所說明的方法，例如為包含電腦程式碼構件的電腦程式形式，電腦程式碼構件經調適以在程式運行於電腦上時，執行本文所說明之任何方法的所有步驟，且其中電腦程式可被實施在電腦可讀取媒體上。有形(或非暫態性)儲存媒體的範例，包含碟片、拇指碟與記憶體等等，且不包含傳輸訊號。軟體可適合執行於平行處理器或序列處理器上，以使方法步驟可由任何適合的次序實行或同時實行。

此確認了軟體可為有價值的、可被個別交易的商品。意為包含在控制「啞(dumb)」或標準硬體或運行於其上的軟體。亦意為包含「描述」或界定硬體配置的軟體，諸如硬體描述語言(hardware description language; HDL)軟體，如被用以設計矽晶片或用以配置通用可程式晶片，以實行所需的功能。

在本發明領域中具有通常知識者將瞭解到，用以儲存程式指令的儲存裝置，可被分散於網路上。例如，遠距電腦可儲存被說明為軟體的範例程序。本地電腦或終端電腦可存取遠距電腦，並下載軟體的全部或部分，以運行程式。或者，本地電腦可如所需下載數件軟體，或分別於本地終端與遠距電腦(或電腦網路)執行一些軟體指令。在本發明領域中具有通常知識者亦將理解，藉由使用已為在本發明領域中具有通常知識者所熟知的習知技

術，可由專屬電路(諸如，DSP、可程式邏輯陣列與類似者)實行所有或部分的軟體指令。

如將顯然於在本發明領域中具有通常知識者，在本文中給定的任何範圍或裝置值可被延伸或改變，而不失去所尋求的效果。

雖然已由特定於結構特徵及/或方法步驟的語言說明了發明標的，需瞭解界定於附加申請專利範圍中的發明標的不須被限制於上文所說明的特定特徵或步驟。相反的，上文所說明的特定特徵或步驟被揭示為用以實施申請專利範圍的範例格式。

將瞭解到，上文所說明的益處與優點可關於一種具體實施例，或可關於數種具體實施例。具體實施例不限於解決任何或所有所述及之問題的具體實施例，或具有任何或所有所述及之益處與優點的具體實施例。將進一步瞭解到，參照於「一個」物件，代表彼等物件的一個或多個。

本文所說明之方法步驟可由任何適合的次序來實行，或可在適當時同時執行。此外，可從彼等方法之任一者中刪除個別方塊，而不脫離本文所說明之發明標的的精神與範圍。上文所說明之任何範例的態樣，可與所說明之任何其他範例的態樣結合，以形成進一步的範例，而不失去所尋求的效果。

在本文中使用的名詞「包含」，表示包含所識別的方法方塊或元件，但此等方塊或元件不包含排他性的列表，

且方法或設備可包含額外的方塊或元件。

將瞭解到，上文給定之對於較佳具體實施例的說明，係僅為示例，且在本發明領域中具有通常知識者可進行各種修改。上文說明、範例與資料，對結構以及本發明示例性具體實施例的使用，提供了完整的敘述。雖然已於上文以某種程度的特定性，或參照一個或多個個別具體實施例說明了本發明的各種具體實施例，在本發明領域中具有通常知識者可對所揭示之具體實施例進行多種改變，而不脫離本發明的精神或範圍。

【圖式簡單說明】

由參照附加圖式來閱讀上文實施方式，將可更佳地瞭解本說明書，在附加圖式中：

第 1 圖為兩個多核心電腦的示意圖，其中一個多核心電腦具有共享記憶體，而另一個不具有共享記憶體；

第 2 圖為多核心電腦之示例性核心的示意圖；

第 3 圖為多核心電腦之另一種示例性核心的示意圖；

第 4 圖為具有階層快取記憶體與共享實體記憶體之多核心電腦的示意圖；

第 5 圖為在多核心電腦的核心之間發送記憶體寫入要求、接收彼指令與執行彼指令之方法的流程圖；

第 6 圖為使用第 5 圖方法之配置程序的更多細節的流程圖；

第 7 圖為發送記憶體寫入要求之方法的流程圖；

第 8 圖為識別目的核心之方法的流程圖；

第 9 圖為讀取先前由記憶體寫入要求更新之資料的方法與在從另一個核心接收的資料不再被需要時的方法的流程圖；

第 10 圖為從遠距核心接收記憶體寫入要求之方法的流程圖；

第 11 圖為通知程序的流程圖；

第 12 圖為用以於接收核心實施具有輕推之通知程序的程序的流程圖；

第 13 圖圖示說明示例性多核心以運算為基礎之裝置，其中可實施說明於本文中的遠距記憶體寫入與通知程序。

在附加圖式中使用類似的元件編號以代表類似的元件。

【主要元件符號說明】

100	節點	101	多核心電腦
102	節點	104	多核心電腦
106	核心	108	快取記憶體
110	記憶體	112	網路介面
114	核心	116	快取記憶體
118	隨機存取記憶體	200	核心

202	暫存器	204	通知控制暫存器
206	延伸轉譯旁路緩衝器	208	通知白名單
210	線程轉譯緩衝器	212	通知處理單元
300	核心	302	暫存器
304	通知控制暫存器	306	轉譯旁路緩衝器
308	通訊轉譯緩衝器	310	通知白名單
312	線程轉譯緩衝器	314	通知處理單元
400	核心	402	核心
404	核心	406	核心
408	第 1 級快取記憶體	410	第 1 級快取記憶體
412	第 1 級快取記憶體	414	第 1 級快取記憶體
416	第 2 級快取記憶體	418	第 2 級快取記憶體
420	第 3 級快取記憶體	422	實體記憶體
500-512	步驟	600-602	步驟
700	data_send 作業	702-706	步驟
800-816	步驟	900	data_receive 作業
902-904	步驟	906	data_done 作業
908-910	步驟	1000-1008	步驟
1100-1114	步驟	1200-1206	步驟
1300	裝置	1302	處理核心

1304	內 連 線	1306	實 體 記 憶 體
1308	作 業 系 統	1310	應 用 程 式 軟 體
1312	通 訊 界 面	1314	輸 入 / 輸 出 控 制 器
1316	顯 示 裝 置	1318	使 用 者 輸 入 裝 置
1320	網 路		

七、申請專利範圍：

1. 一種位於一多核心處理器的方法，該多核心處理器具有複數個核心與至少一個共享實體記憶體，該方法包含以下步驟：

發送步驟，從作為一發送核心之該等核心的一第一核心，對一目的核心發送對於該共享實體記憶體之一部分的一更新，作為一記憶體寫入要求；

接收步驟，於該目的核心接收該記憶體寫入要求；

執行步驟，於該目的核心對一快取記憶體階層執行該記憶體寫入要求，如同該記憶體寫入要求係源自該目的核心。

2. 如請求項 1 所述之方法，包含以下步驟：在該共享實體記憶體中產生一資料結構，由至少該發送核心寫入並由作為該目的核心的一單一核心讀取。
3. 如請求項 2 所述之方法，包含以下步驟：於該發送核心與該目的核心使用作業系統內核，以將對於該資料結構之一實體記憶體共用區域映射入對該等發送核心與目的核心之每一者而言為本地的虛擬位址空間；以及更新作業系統資料結構，以配置對在一發送核心上的該虛擬位址的一寫入為由對該目的核心的該記憶體寫入要求來實施。

4. 如請求項 1 所述之方法，其中係由該發送核心使用控制元資料來識別該目的核心，該控制元資料在一配置程序期間被建立，且該控制元資料被使用一緩衝器被快取於該發送核心，該緩衝器係選自下列之任意者：被延伸以併入該控制元資料的一轉譯旁看緩衝器；以及儲存該控制元資料的一專屬通訊轉譯緩衝器。
5. 如請求項 1 所述之方法，其中該複數個核心係使用不具有一快取記憶體同調性機制的一階層快取記憶體結構來內連線。
6. 如請求項 1 所述之方法，其中該複數個核心係使用具有一快取記憶體同調性機制的一階層快取記憶體結構來內連線。
7. 如請求項 1 所述之方法，包含以下步驟：於該目的核心執行一要求，該要求指示由該記憶體寫入要求寫入的一值不再被需要，以及在彼值被寫入該共享記憶體之前從該目的核心的一快取記憶體清除彼值。
8. 如請求項 1 所述之方法，其中於該目的核心對一快取記憶體階層執行該記憶體寫入要求的該執行步驟，係被實行而使由該記憶體寫入要求寫入於該目的核心之該快

取記憶體階層的一值，係為運行於該目的核心的程序所可見。

9. 一種位於一多核心處理器的方法，該多核心處理器具有複數個核心，該複數個核心共享至少一個實體記憶體，該方法包含以下步驟：

配置步驟，於該共享實體記憶體配置一通知資料結構，並於該複數個核心之每一者的一虛擬位址空間建立彼通知資料結構的一代表，每一虛擬位址空間為於一核心之至少該共享實體記憶體的部分的一映射；

獲取步驟，於作為一發送核心之該等核心之一者，獲取關於發生了一組事件中的哪些事件的資訊；

通知步驟，向一目的核心通知該資訊，該目的核心為該複數個核心之一者，藉由：

更新該通知資料結構或彼通知資料結構的該代表於該目的核心虛擬位址空間。

10. 如請求項 9 所述之方法，包含以下步驟：藉由從該發送核心發送一記憶體通知要求至該目的核心，以寫入代表於該目的核心發生了該組事件中的哪些事件的一值，來更新該通知資料結構或彼通知資料結構的該代表於該目的核心虛擬位址空間，如同彼通知資料結構更新係源自該目的核心。

11. 如請求項 10 所述之方法，進一步包含以下步驟：使用一邏輯 OR 運算來實行該更新。
12. 如請求項 9 所述之方法，其中該通知資料結構包含一位元向量。
13. 如請求項 12 所述之方法，其中關於發生了一組事件中的哪些事件的資訊，包含關於訊息通道的資訊，且其中該位元向量之每一位元係關聯於該等訊息通道之一者或多者。
14. 如請求項 9 所述之方法，進一步包含以下步驟：設置該目的核心為在接收到該記憶體通知要求時呼叫一通知功能。
15. 如請求項 14 所述之方法，包含以下步驟：於每一核心使用至少四個通知控制暫存器，其中一個通知控制暫存器保持一觀看旗標、一個通知控制暫存器保持代表關於發生了該複數個事件中的哪些事件的一遮罩、一個通知控制暫存器保持一被觀看區域而一個通知控制暫存器保持一通知目標功能，該被觀看區域為該通知資料結構的該虛擬位址空間。

16. 如請求項 10 所述之方法，進一步包含以下步驟：僅在特定條件下允許該發送核心發送該記憶體通知要求至該目的核心。

17. 如請求項 10 所述之方法，進一步包含以下步驟：於每一核心提供包含位址遮罩對的一通知白名單，且僅於該目的核心的該位址與欲由該記憶體通知要求寫入的該值係存在於該通知白名單中做為一位址遮罩對時，允許該發送核心發送該記憶體通知要求至該目的核心。

18. 如請求項 9 所述之方法，包含以下步驟：使用一快取記憶體同調性機制以內連線該複數個核心；以及於該目的核心告知對於該通知資料結構的一快取記憶體同調性失效，並從該目的核心發送對該通知資料結構之該共享記憶體的一擷取。

19. 一種多核心處理器，包含：

複數個核心，該複數個核心使用一階層快取記憶體結構來內連線；

至少一個共享實體記憶體，該至少一個共享實體記憶體使用該階層快取記憶體結構與該複數個核心通訊；

位於該共享實體記憶體的一資料結構；

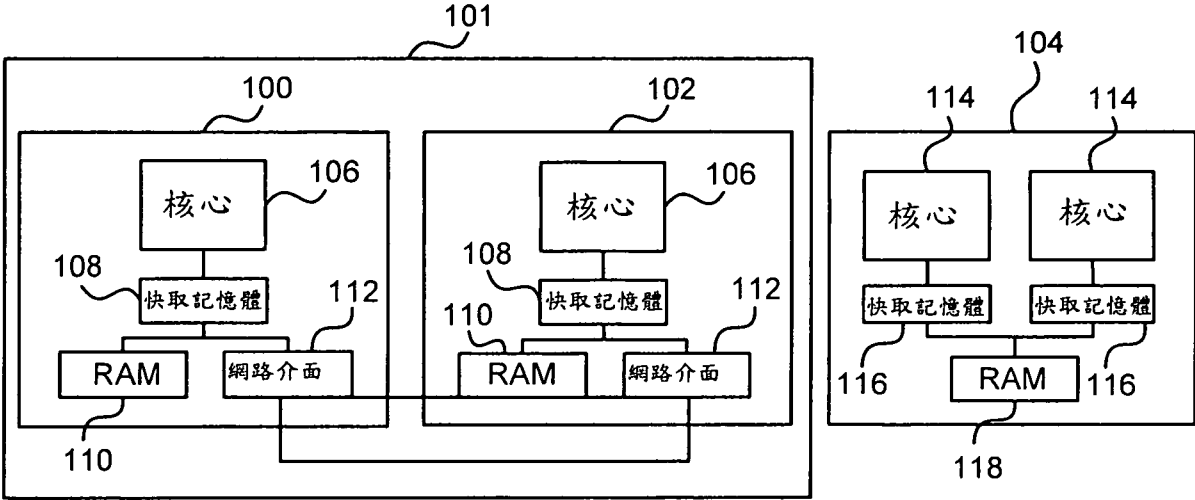
作為一發送核心的該等核心之一者，經設置以發送一記憶體寫入要求至作為一目的核心的該等核心之一

者，以於該共享實體記憶體更新該資料結構，該資料結構係被映射入位於對該等發送核心與目的核心之每一者而言為本地之該快取記憶體結構之部分的虛擬位址空間；

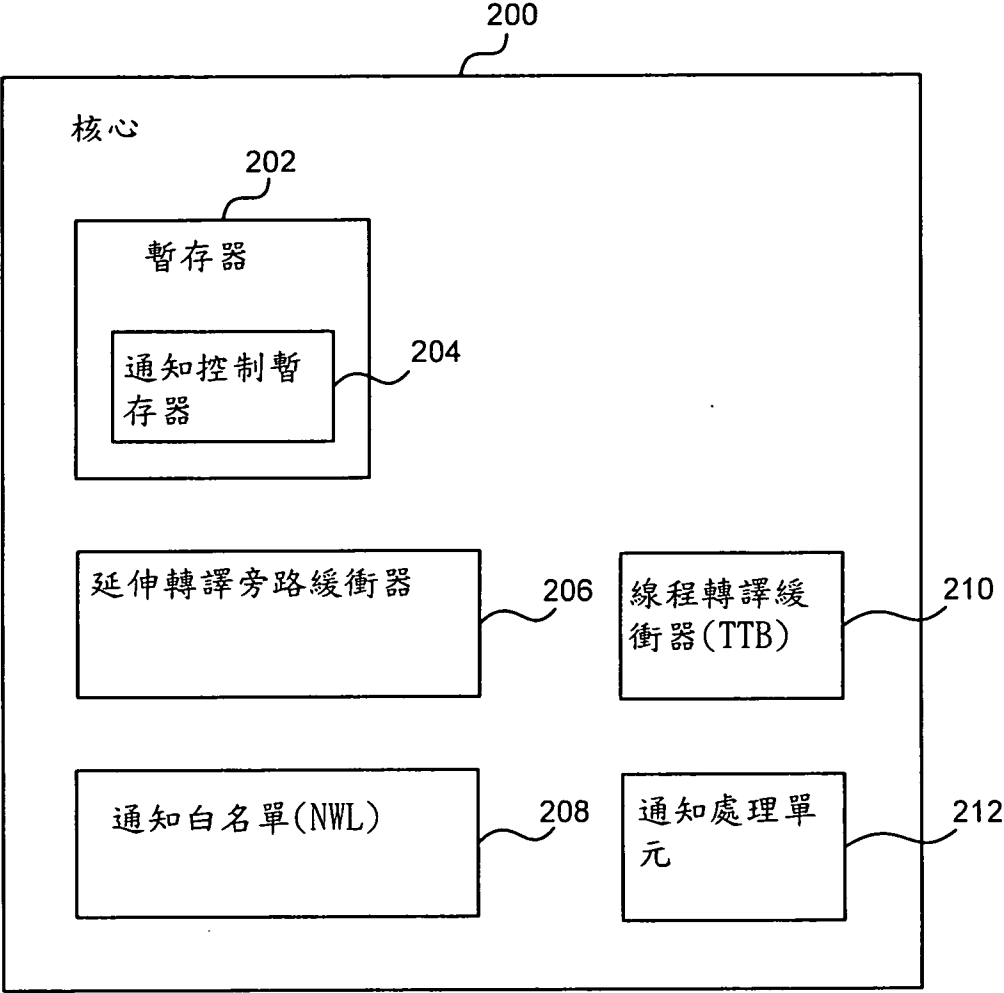
該目的核心經設置以接收該記憶體寫入要求並執行該記憶體寫入要求，如同該記憶體寫入要求係源自該目的核心，並使由該記憶體寫入要求寫入的一值，以可見於運行於該目的核心之程序的方式，被儲存在該目的核心的該虛擬位址空間中。

20. 如請求項 19 所述之多核心處理器，其中每一核心包含一緩衝器，該緩衝器經設置以儲存控制元資料，該控制元資料用以使一發送核心能夠識別一目的核心，且其中該緩衝器為一延伸轉譯旁路緩衝器與一通訊轉譯緩衝器之任意者，該通訊轉譯緩衝器僅保持用以使一發送核心能夠識別一目的核心的控制元資料。

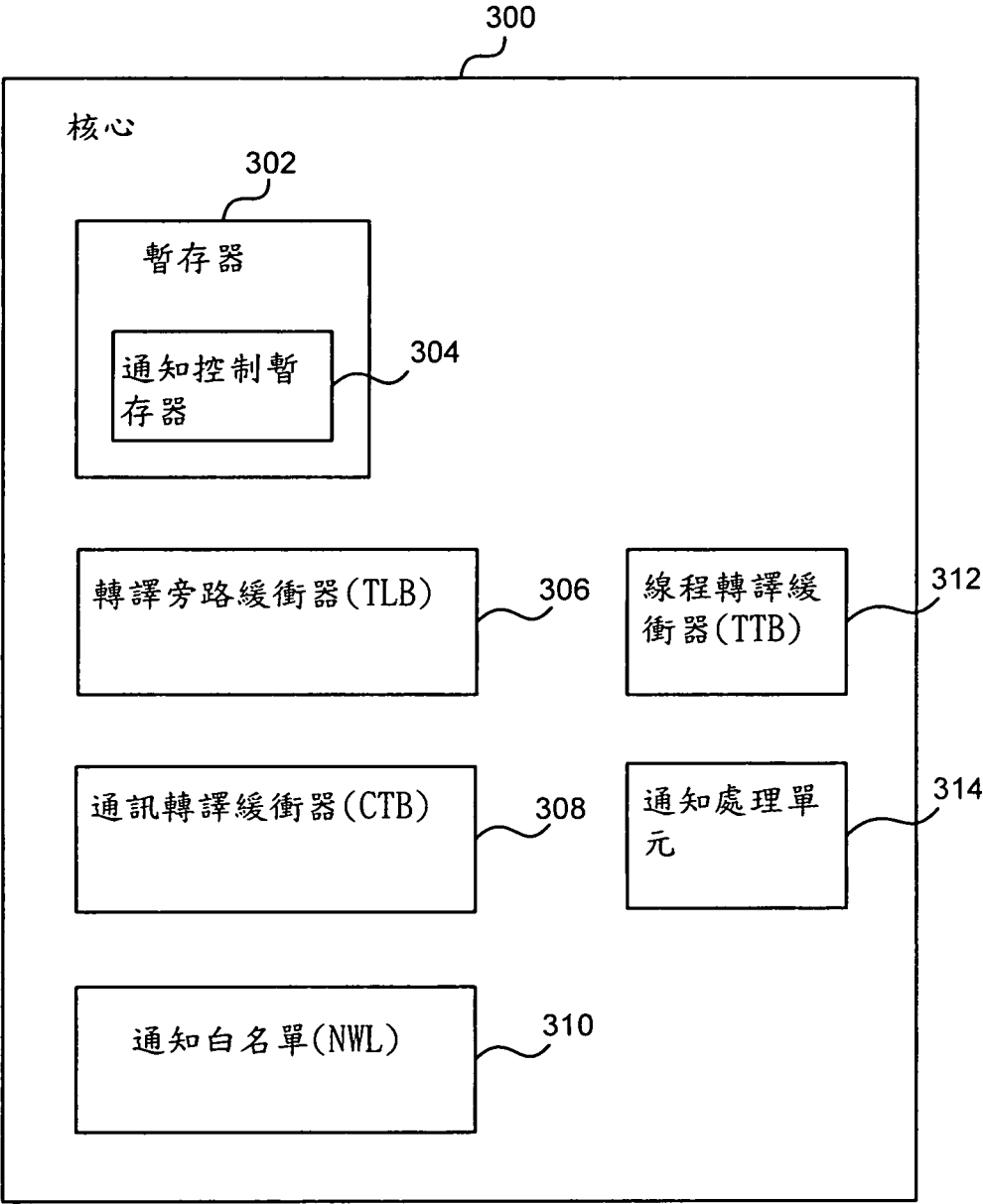
八、圖式：



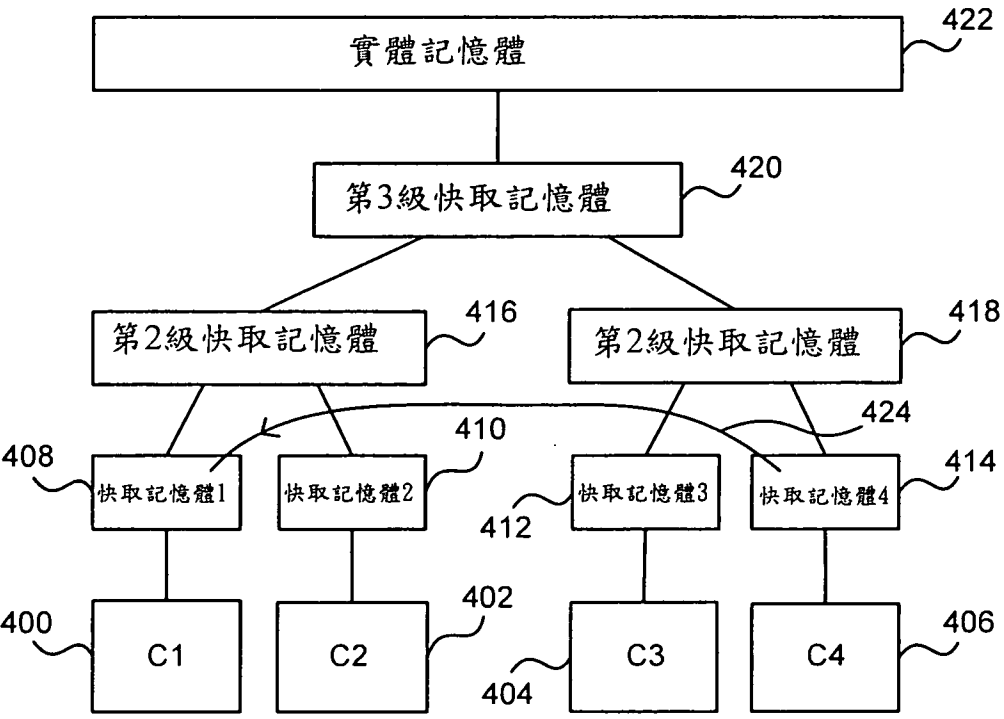
第 1 圖



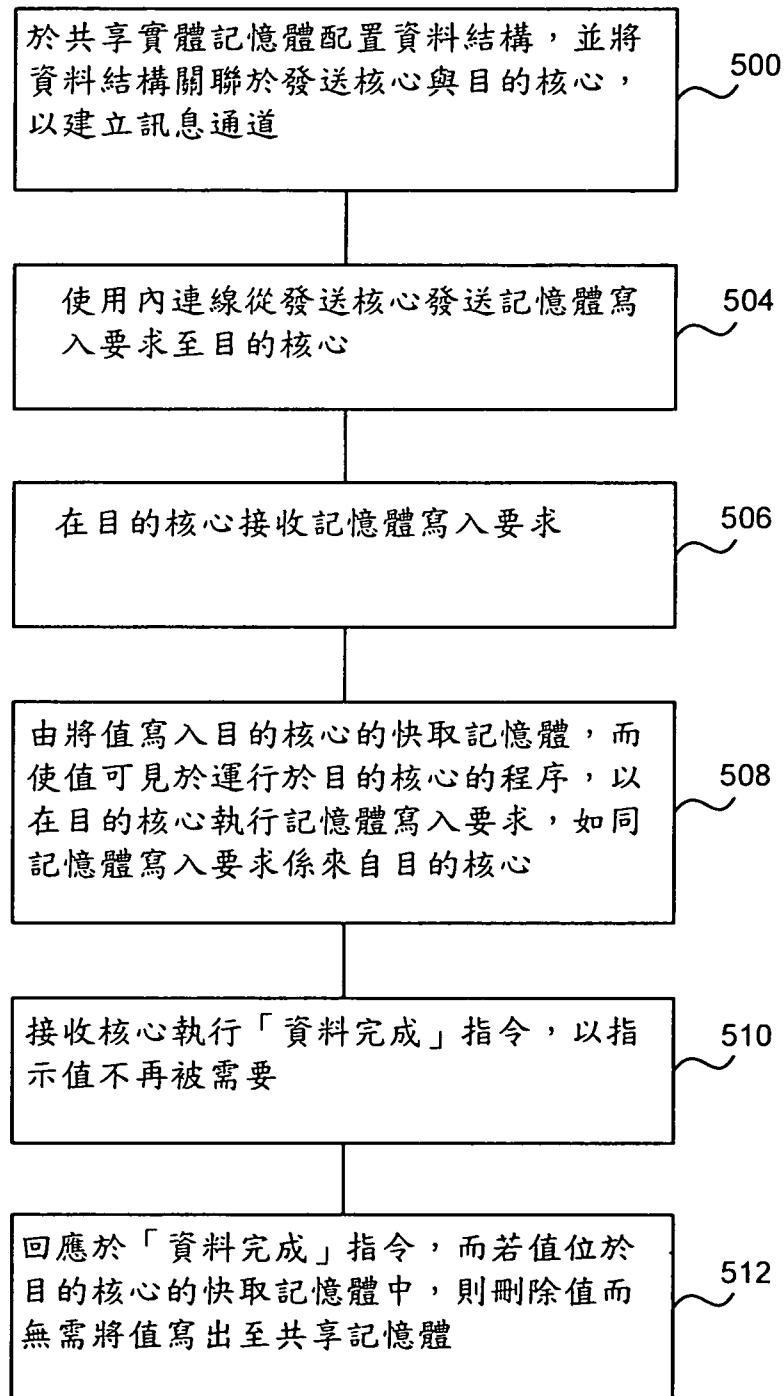
第 2 圖



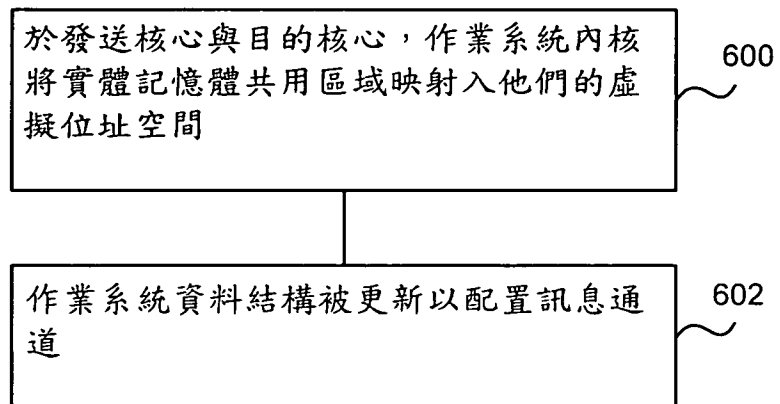
第 3 圖



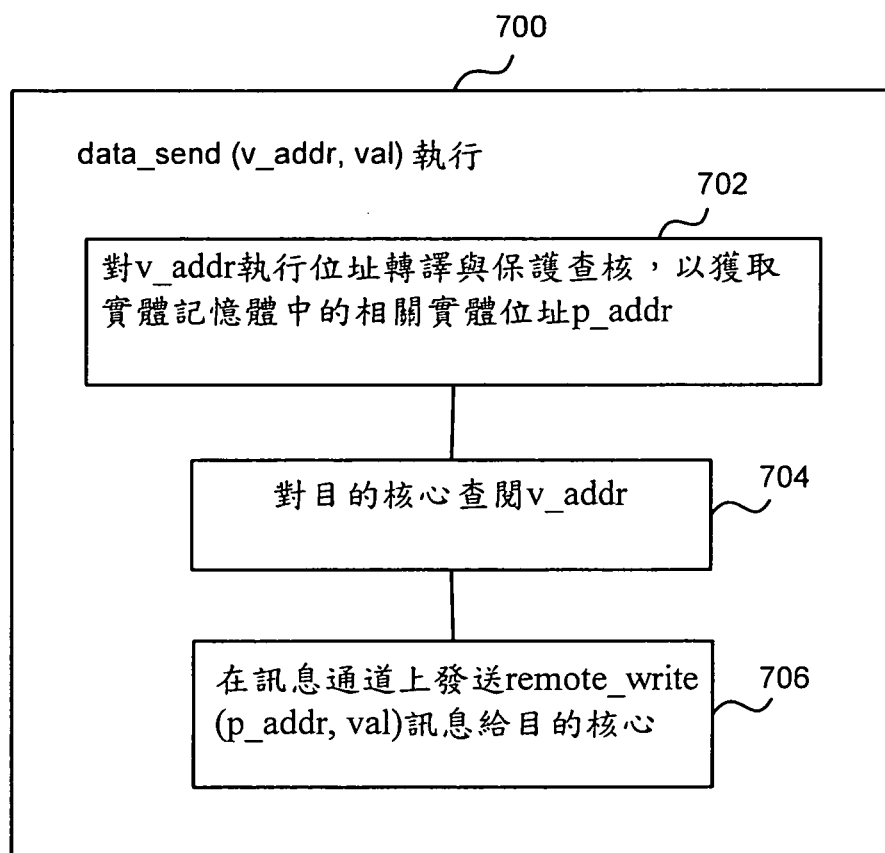
第 4 圖



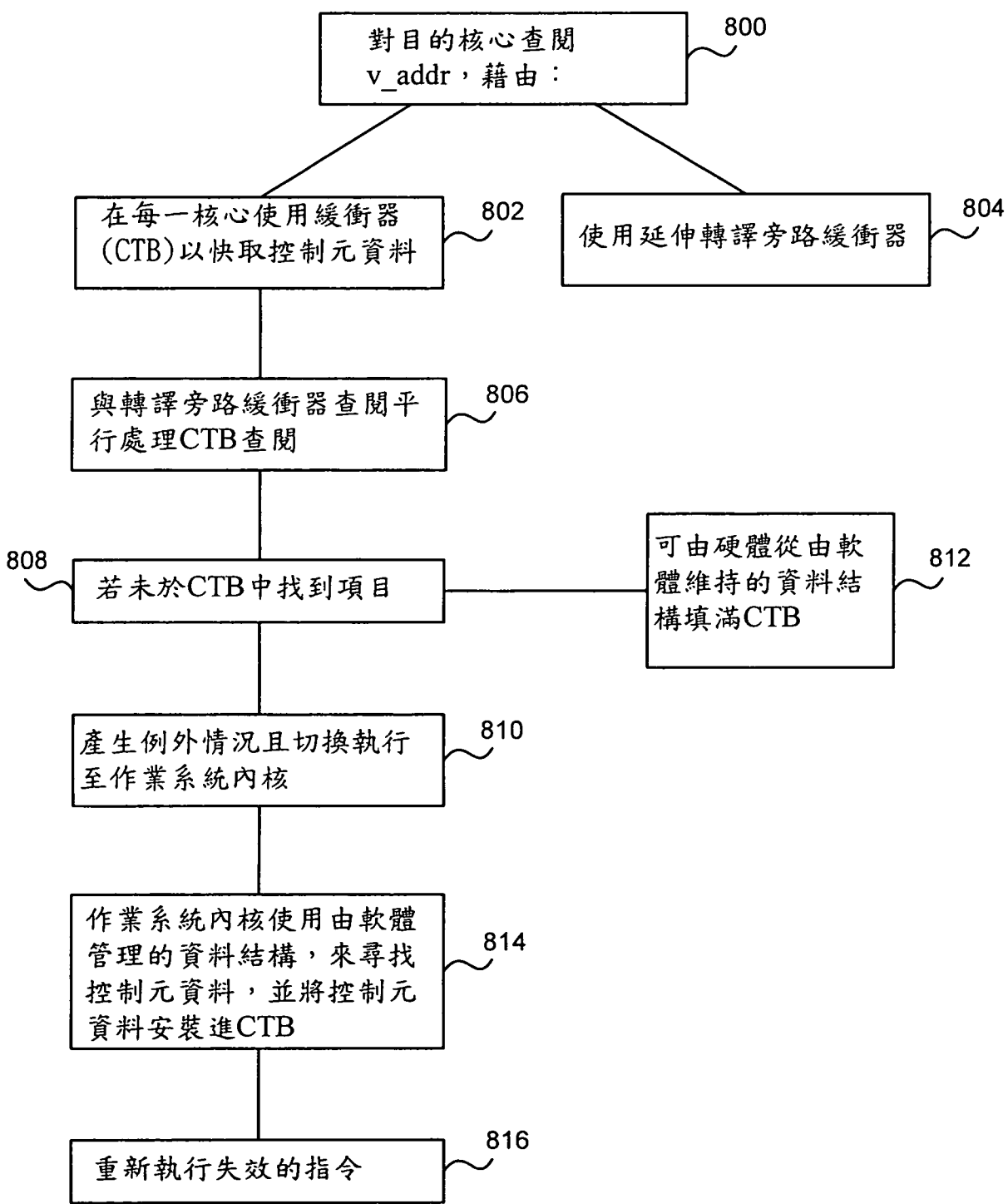
第 5 圖



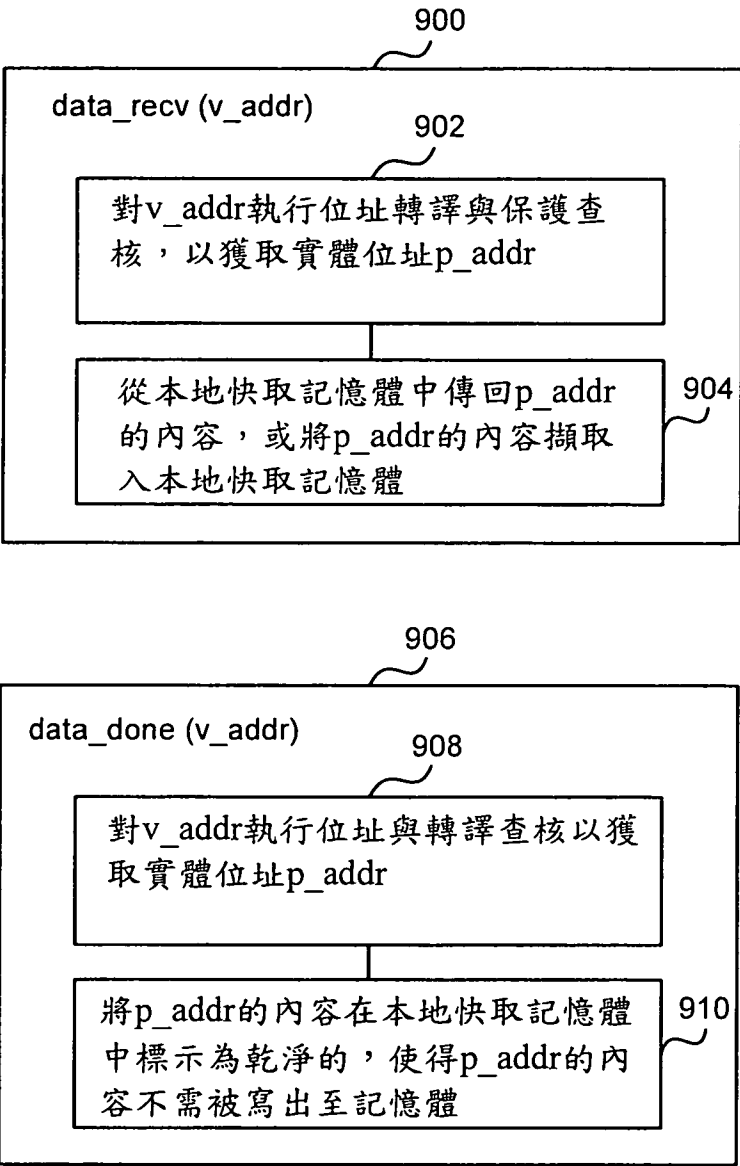
第 6 圖



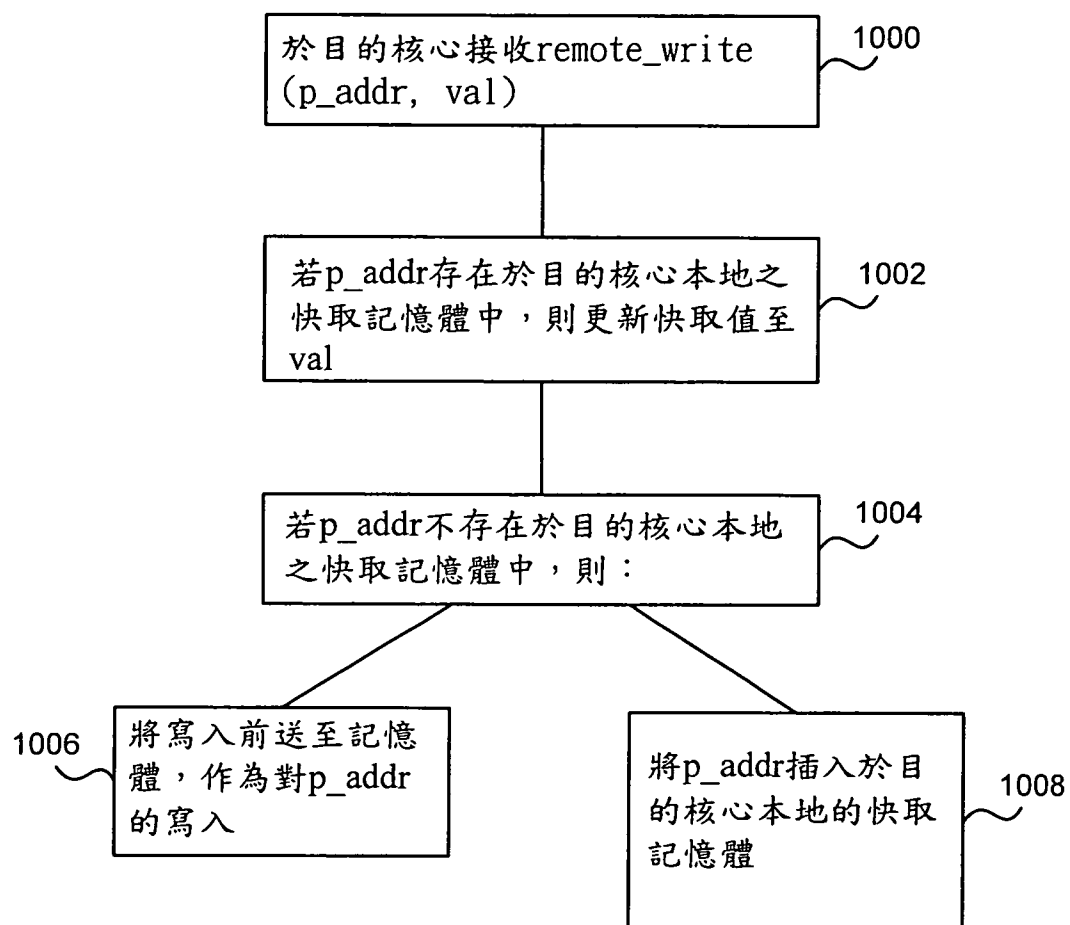
第 7 圖



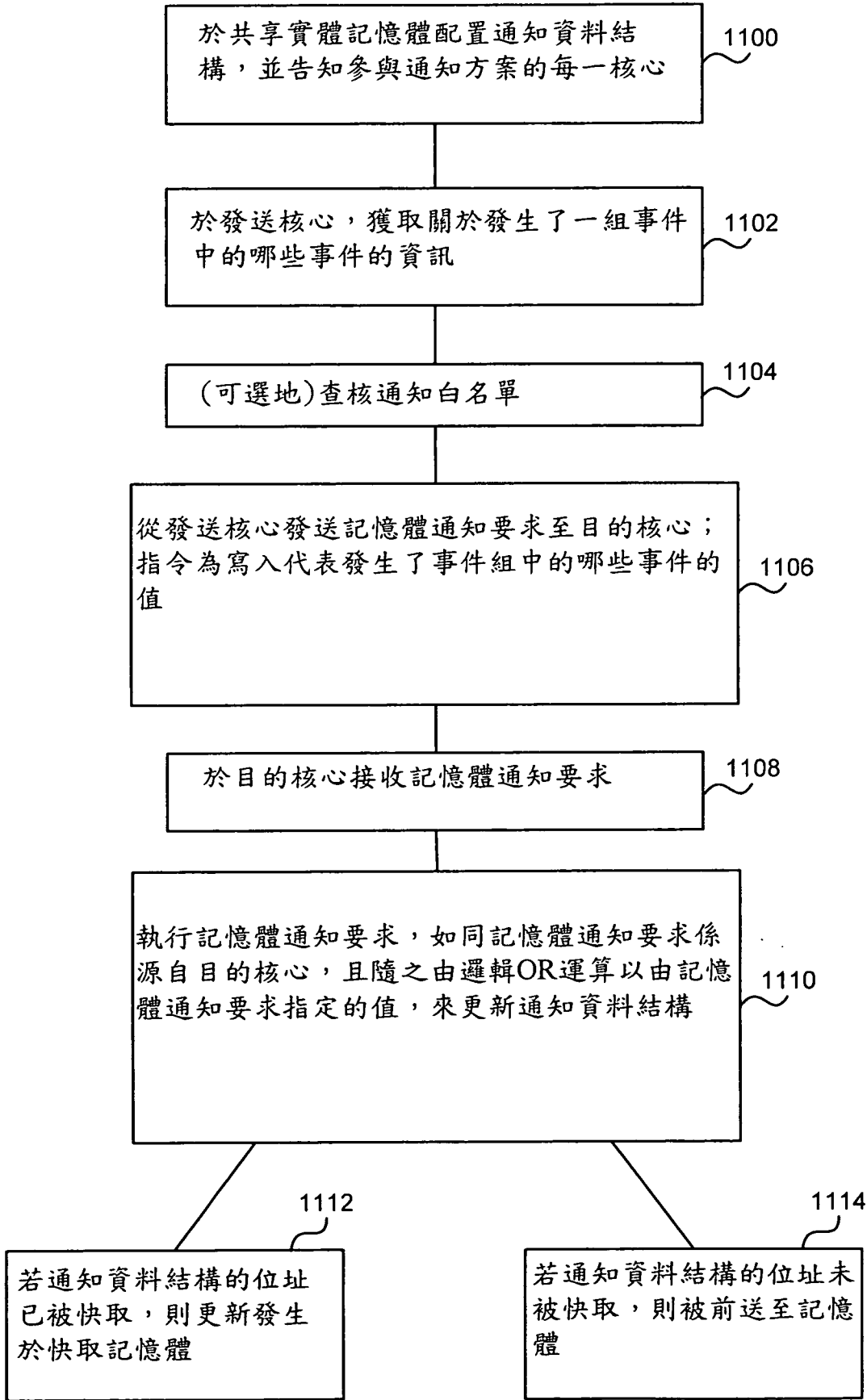
第 8 圖



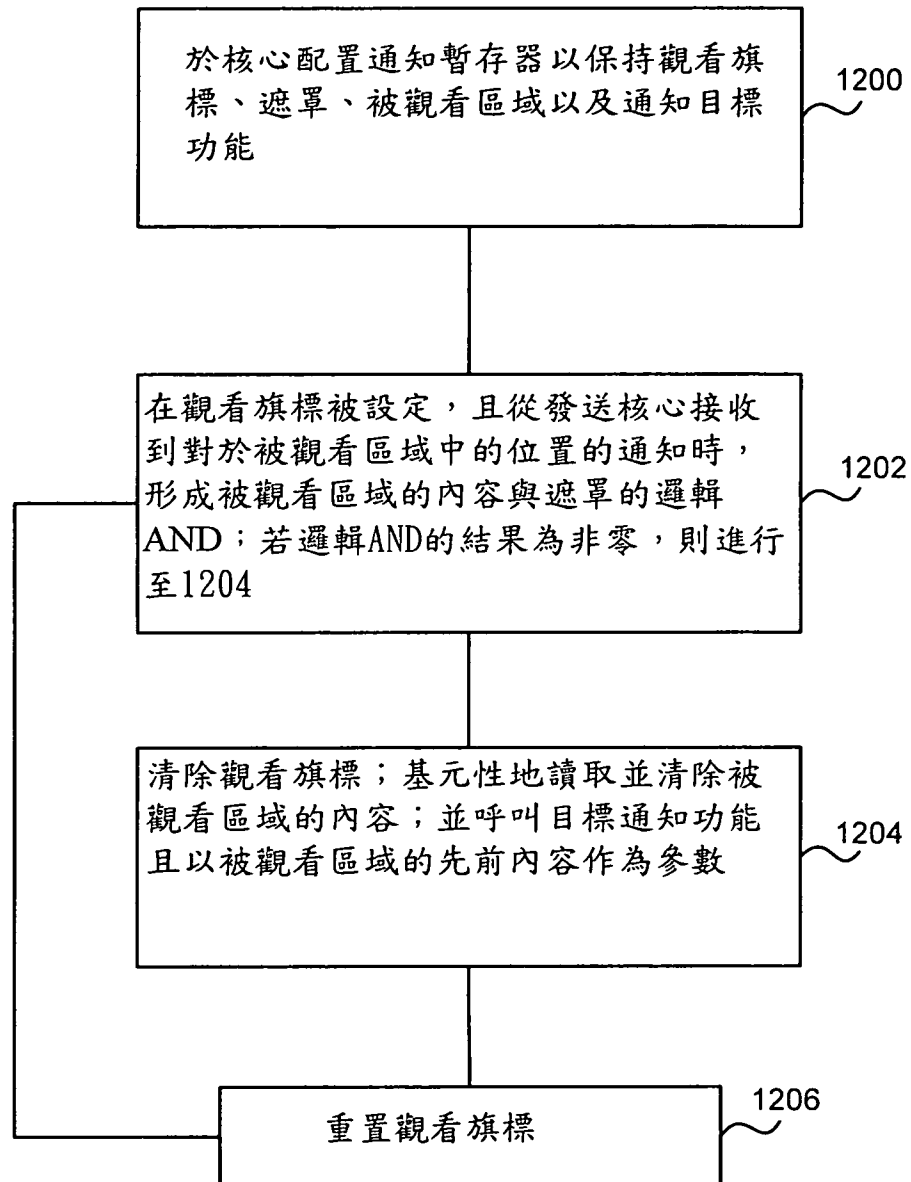
第 9 圖



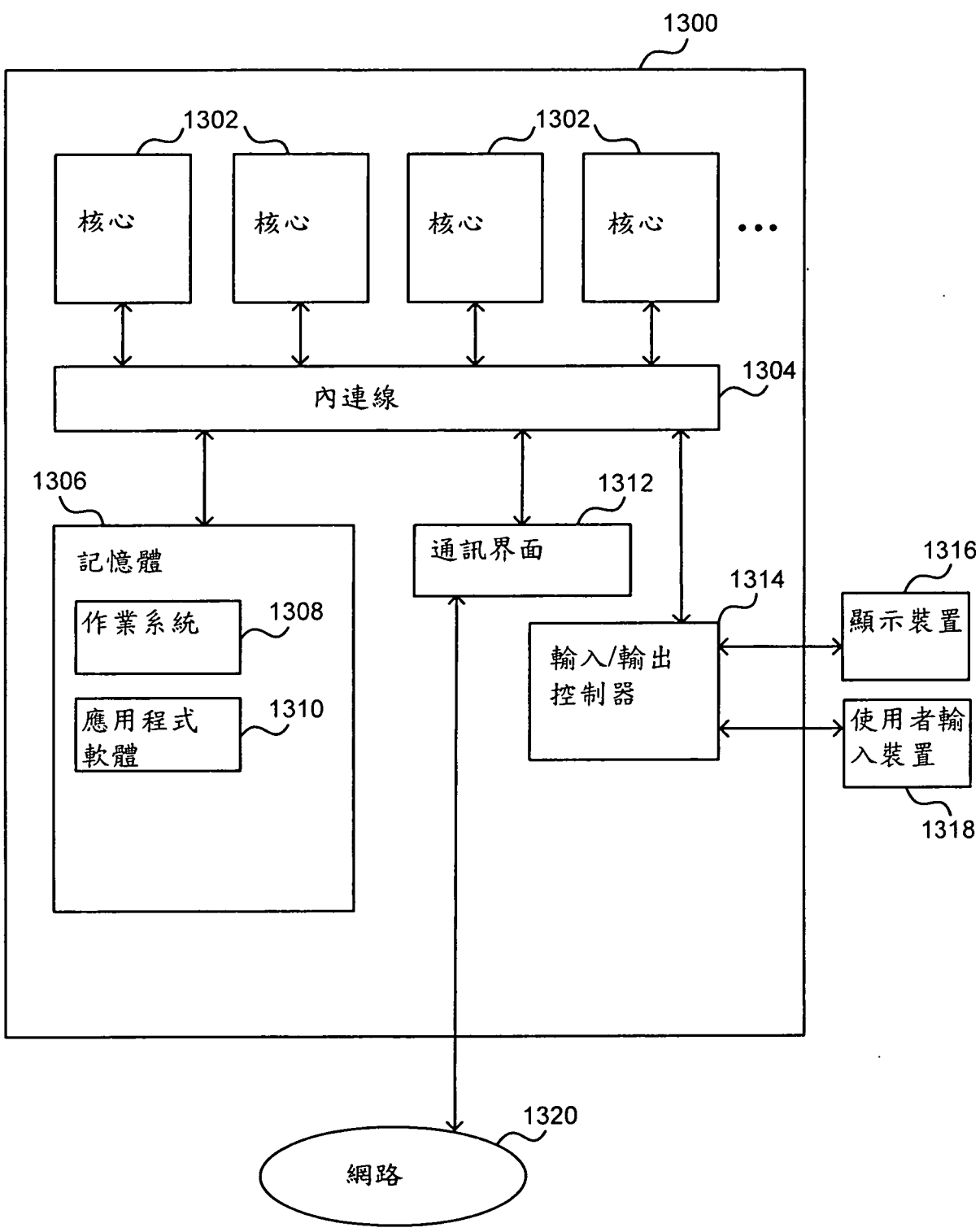
第 10 圖



第 11 圖



第 12 圖



第 13 圖