



(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 I470435 B

(45)公告日：中華民國 104 (2015) 年 01 月 21 日

(21)申請案號：099124707

(22)申請日：中華民國 99 (2010) 年 07 月 27 日

(51)Int. Cl. : **G06F13/16 (2006.01)**

(30)優先權：2009/09/18 美國 12/562,477

(71)申請人：英特爾股份有限公司 (美國) INTEL CORPORATION (US)
美國(72)發明人：辛亞 高森 CHINYA, GAUTHAM N. (IN)；王航 WANG, HONG (US)；梅斯酷堤
迪派克 MATHAIKUTTY, DEEPAK A. (US)；柯林斯 傑米森 COLLINS,
JAMISON D. (US)；夏克曼 尹森 SCHUCHMAN, ETHAN (US)；赫德 詹姆斯
HELD, JAMES P. (US)；貝特 阿傑 BHATT, AJAY V. (US)；西斯 普雷沙特
SETHI, PRASHANT (IN)；華利 史帝芬 WHALLEY, STEPHEN F. (US)

(74)代理人：林志剛

(56)參考文獻：

TW	200951980A	CN	101187902A
US	5361340A	US	6766424B1
US	2004/0044872A1		

審查人員：許俊岳

申請專利範圍項數：30 項 圖式數：8 共 43 頁

(54)名稱

為本地與遠端實體記憶體間之共用虛擬記憶體提供硬體支援

PROVIDING HARDWARE SUPPORT FOR SHARED VIRTUAL MEMORY BETWEEN LOCAL AND
REMOTE PHYSICAL MEMORY

(57)摘要

在一個實施例中，本發明包括記憶體管理單元(MMU)，其具有用以儲存虛擬位址至實體位址轉譯的項目，其中，各項目包括用以表示該對應的項目之記憶體位置是否存在於本地或遠端的記憶體中之位置指示器。以此方式，共同虛擬記憶體空間可於該兩個記憶體之間被共用，該兩個記憶體可藉由一個或多個非同調鏈結而被分離。其他實施例被說明及主張。

In one embodiment, the present invention includes a memory management unit (MMU) having entries to store virtual address to physical address translations, where each entry includes a location indicator to indicate whether a memory location for the corresponding entry is present in a local or remote memory. In this way, a common virtual memory space can be shared between the two memories, which may be separated by one or more non-coherent links. Other embodiments are described and claimed.

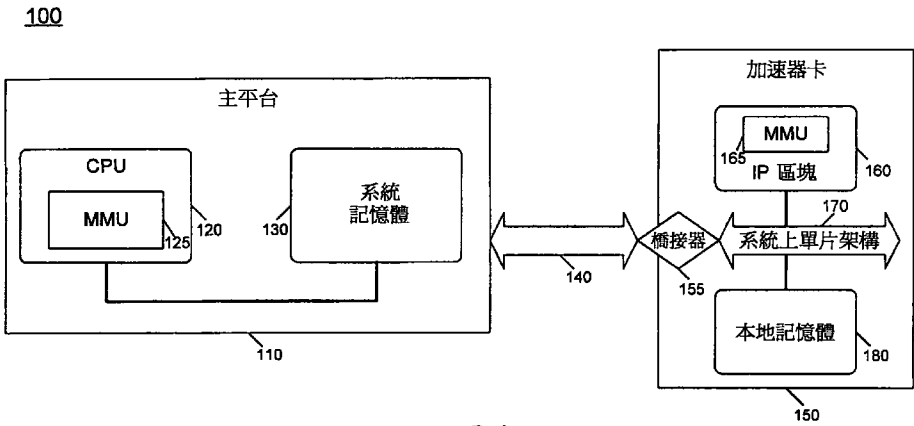
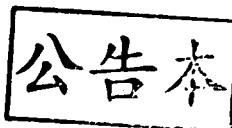


圖1

- 100 . . . 系統
- 110 . . . 主個人電腦 (PC)平台
- 120 . . . CPU
- 125 . . . 記憶體管理單元(MMU)
- 130 . . . 系統記憶體
- 140 . . . 非同調互連
- 150 . . . 加速器卡
- 155 . . . 橋接器
- 160 . . . 智慧財產權 (IP)區塊
- 165 . . . MMU
- 170 . . . 晶片 (SoC) 架構
- 180 . . . 本地記憶體



發明專利說明書

(本申請書格式、順序，請勿任意更動，※記號部分請勿填寫)

※申請案號：099124707

※申請日：099年07月27日

※IPC分類：

G06F13/16

2006.01

一、發明名稱：(中文／英文)

為本地與遠端實體記憶體間之共用虛擬記憶體提供硬體支援

Providing hardware support for shared virtual memory between local and remote physical memory

二、中文發明摘要：

在一個實施例中，本發明包括記憶體管理單元（MMU），其具有用以儲存虛擬位址至實體位址轉譯的項目，其中，各項目包括用以表示該對應的項目之記憶體位置是否存在於本地或遠端的記憶體中之位置指示器。以此方式，共同虛擬記憶體空間可於該兩個記憶體之間被共用，該兩個記憶體可藉由一個或多個非同調鏈結而被分離。其他實施例被說明及主張。

三、英文發明摘要：

In one embodiment, the present invention includes a memory management unit (MMU) having entries to store virtual address to physical address translations, where each entry includes a location indicator to indicate whether a memory location for the corresponding entry is present in a local or remote memory. In this way, a common virtual memory space can be shared between the two memories, which may be separated by one or more non-coherent links. Other embodiments are described and claimed.

四、指定代表圖：

(一) 本案指定代表圖為：第(1)圖。

(二) 本代表圖之元件符號簡單說明：

100：系統

110：主個人電腦(PC)平台

120：CPU

125：記憶體管理單元(MMU)

130：系統記憶體

140：非同調互連

150：加速器卡

155：橋接器

160：智慧財產權(IP)區塊

165：MMU

170：晶片(SoC)架構

180：本地記憶體

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：

【發明所屬之技術領域】

本發明係有關為本地與遠端實體記憶體間之共用虛擬記憶體提供硬體支援。

【先前技術】

隨著以處理器為基礎的系統進展，透過諸如快速週邊組件互連（PCIe™）而連接至系統之可用的可程式加速器依據以PCI Express™規格的基礎規格版本2.0（2007年1月17日所發表）（在下文中稱為PCIe™規格）或另一個此種協定為基礎的鏈結互連，其讓系統積分器能夠將更多個計算馬力包裝成系統。然而，挑戰存在於確保應用程式能夠透通地利用額外的計算馬力，而不會對以手動方式分離主處理器（例如，多核中央處理單元（CPU））與加速器之間的計算，及管理資料往返地移動之應用程式做重大的改變。習知上，僅藉由作業系統（OS）所管理的主系統記憶體被配置以供應用程式使用。透過週邊互連而被耦接的任何加速器之本地的實體記憶體被分開地管理。特別是，此種加速器上的本地記憶體並非可視做為藉由於主處理器上執行的OS可辨識之系統記憶體的部分。反而是，裝置驅動程式軟體負責詳盡地管理本地記憶體與遠端記憶體之間的資料移動。

藉由處理器來予以存取的實體記憶體作業系統來予以管理，而作業系統藉由將此實體記憶體的存取虛擬化，以

產生相鄰的大虛擬位址空間之錯覺。OS使用為虛擬記憶體管理的潛在（underlying）處理器支援，因為處理器讓此軟體能夠建立將虛擬分頁對映至實體分頁的對映表。處理器虛擬記憶體位址。每次記憶體存取需要被實施時，處理器藉由查看對映表來支援虛擬記憶體位址轉譯。頻繁存取的轉譯可藉由處理器來予以快取，以加速此程序。這些對映表（一般稱為分頁表）也包含屬性位元（諸如，讀取/寫入），及控制給定的虛擬分頁之存取的使用者/監督者優先權位元。雖然OS管理主機板上之可用的實體記憶體（系統記憶體），但是它並不管理或配置加速器的本地，及加速器上之可用的記憶體。因此，目前的解決方式產生如由程式設計者所視的共用記憶體模型，且根據記憶體保護機制來找出往返於不同記憶體之間的分頁之錯誤，及移動這些分頁。

【發明內容及實施方式】

實施例使用加速器（其藉由例如快速週邊組件互連（PCIe™）介面之介面而被互連至系統），藉由存取存在於加速器上的記憶體，及使用特別的載入/儲存交易來定址記憶體，而能使處理器（例如，插座上的中央處理單元（CPU））產生及管理完全共用的虛擬位址空間。定址遠端記憶體的能力直接讓如同由應用軟體所看見的有效計算容量能增加，且讓應用程式能無縫地共用資料，而不會使程式設計者明顯涉及往返移動資料。就其本身而論，記憶體

可被定址，而不必依靠記憶體保護，及找出虛擬位址存取上的錯誤，而將記憶體存取重新導向至自錯誤處理程式（`fault handler`）中完成。就其本身而論，現有的共用記憶體多核心處理可被延伸為包括加速器，其並不位於插座上，反而是透過週邊非同調鏈結而被連接。

相反地，諸如以叢集為基礎的系統之典型系統產生如同由程式設計者所看到的部分共用記憶體模型，且根據記憶體保護機制來找出往返於CPU與週邊裝置之間的分頁之錯誤，及移動這些分頁。再者，在以叢集為基礎的系統中，因為僅一部分的位址空間被共用，且程式設計者自共用區域配置，或明確指定此資料的哪個部分係設置於共用位址空間中，所以各節點執行作業系統（OS）堆疊（在它的頂部上，應用程式執行，且系統的此方面被顯露給程式設計者）的分離複製。此執行環境與完全共用記憶體執行環境不同，完全共用記憶體執行環境類似於單一共用記憶體多核心系統。

反而在不同的實施例中，插座上的處理器可定址遠端記憶體，其對於加速器而言是本地的，此加速器讓處理器能夠透通地使用遠端記憶體位址來存取共用資料。為了實現此，可提供架構上的延伸，讓虛擬記憶體管理系統能夠被擴展，使得可發出特別的載入/儲存交易，以定址遠端共用資料，且進一步能使系統將記憶體分頁移動到更接近其更頻繁地被存取之處，而不需明確的程式設計者涉及來這樣做。此外，記憶體管理延伸讓程式設計者能夠直接地

執行應用程式碼，而不必詳盡地表示必須共用位址空間，或程式設計者要管理一般共用資料區域的哪個部分。

依此而論，在快取同調CPU插座上之核心與加速器（包括多核心CPU）（其透過週邊互連而被互連至系統）之間的共用虛擬位址空間可被產生及管理。因此，在此互連的兩側上之CPU/加速器可存取共用虛擬分頁（其可實際上被設置於系統記憶體中，或位於穿過互連的加速器之本地的記憶體上），其可為或可不為快取同調的。

因此，即使加速器的本地記憶體為通過互連，且未透過同調架構（諸如，前側匯流排（FSB）或快速路徑互連（QPI）），可直接藉由CPU來存取，所以加速器上之本地的實體記憶體可用作為CPU的額外系統記憶體，且依次作為OS及應用程式的額外系統記憶體。

實施例可以許多不同的系統型式來予以實施。現在參考圖1，所顯示的是依據本發明的一個實施例之系統的方塊圖。如圖1中所顯示，系統100可為具有主個人電腦（PC）平台110的範例電腦系統，主個人電腦（PC）平台110係透過非同調互連140（其可例如為PCIe™鏈結）而被耦接至加速器卡150。如同所看到的，主平台110可包括CPU 120，及系統記憶體130，其在某些實施例中可為動態隨機存取記憶體（DRAM）。雖然為了簡化圖1中的圖示，僅以這些最少的組件來予以顯示，但是瞭解到的是，給定的平台可包括許多其他典型的組件，其包括輸入/輸出中心、晶片組組件、週邊裝置、大量儲存裝置、輸入/輸出裝置

等等。

如同圖1中所看到的，CPU 120可包括記憶體管理單元（MMU）125。MMU 125可使虛擬位址能對映至實體位址，且在某些實施例中，可包括一個或多個轉譯後備緩衝區（TLB）。如同將於底下進一步討論的，依據本發明的實施例之MMU的不同架構延伸能產生及使用在與平台110及加速器卡150相關聯的記憶體之間的共用虛擬記憶體。

仍然參考圖1，加速器卡150可包括智慧財產權（IP）區塊160，其可為任何型式的加速器，諸如，圖形處理單元、CPU、或任何其他此種裝置。如同所看到的，此IP區塊本身可包括MMU 165。為了能與平台110相通訊，橋接器155可存在，以便將依據用於互連140的協定出現之通訊轉換成與系統上晶片（SoC）架構170（其依序使IP區塊160耦接至本地記憶體180）上所使用的協定一致之協定。雖然以圖1的實施例中之此特定實施例來予以顯示，但是本發明的範圍在此方面並未限制。

實施例可實施反向代理執行（RPE），其增加CPU識別板上（例如，主機板）系統記憶體的外部之實體記憶體存取的能力。然後，對此類位置的存取可被轉換成隧穿過週邊架構而至加速器之一類別的存取。加速器依序服務來自其本地實體記憶體的存取。結合使用RPE與代理執行（其中，CPU可根據加速器的請求來協助記憶體存取的完成），具有分離的MMU（其透過（同調或非同調）架構而被耦接至多插座CPU）之任何加速器可建構用於實體記憶體

（同時包括系統記憶體及加速器本地記憶體）的共用虛擬位址空間。使用 RPE 與代理執行，實施例可讓建構用於習知對稱式多重處理（SMP）之以相同多執行緒的（multithreaded）共用虛擬記憶體為基礎之程式將執行緒傳遍 CPU，其遍及多個 CPU 插座，或遍及週邊 I/O 架構上的多個站。

伴隨架構機制的是，實施例也可包括韌體及系統軟體延伸部分，其讓插座與通過週邊互連的加速器（或 CPU）上之核心之間的控制及資料轉移能夠在不同的抽象層（自全部不知道的 OS 至知道的 OS 之範圍，各個具有最佳化的不同選擇）處透通地運作。

根據需求，資料可根據自 CPU 側以及加速器側的共用資料之存取方式而被共用。可與虛擬位址一起運作且支援位址轉譯的加速器可透通地執行與參考被保持完整的資料及碼相同之碼，如當參考加速器執行應用程式的部分時之碼或資料時，可使用相同的虛擬位址。包含碼或資料的實體分頁可被設置於加速器的本地，或可自系統記憶體中提取。因為應用程式不需管理資料移動來組成加速器上的計算，所以虛擬分頁可根據存取的頻率而自遠端位置移至本地位置，而不明確涉及做如此的應用軟體堆疊。

驅動程式軟體常常被交付使用主系統記憶體與遠端記憶體（其對於加速器是本地的）之間的直接記憶體存取（DMA）轉移來明確地移動大量資料之工作的任務。在習知的驅動程式模型中，於 CPU 上執行的應用程式，及管理加

速器的驅動程式典型上存在於兩個不同的虛擬位址空間中。因此，應用程式與驅動程式之間的資料通訊，及系統記憶體與加速器本地記憶體之間的資料轉移經常導致重大的附加耗費（overhead）。另外，此資料轉移典型上係藉由程式設計者所寫的應用程式碼來予以實施。例如，會需要程式設計者使用廠商特定的應用程式介面（API）集，以手動的方式，將資料自系統記憶體移至加速器記憶體。反而，因為整個應用程式碼及資料可被放置在共同的共用虛擬位址空間中，而不必藉由改變應用程式（例如，程式設計者對DMA操作的詳盡編排）來詳盡地移動資料，所以在不需DMA操作的詳盡管理之情況下，依據本發明的實施例之CPU與加速器的核心之間的共用虛擬位址空間之產生而使資料共用簡化。因此，雖然資料轉移仍然會藉由DMA，但其並非程式設計者所控制。換言之，在使用者層級的應用程式之執行期間，處理器可直接存取存在於遠端記憶體中的資料，而不會藉由程式設計者詳盡地程式化來結構及管理能資料存取的潛在架構。

為了建構CPU與加速器之間的共用位址空間，記憶體管理單元可根據使用來轉譯虛擬至實體位址的分頁表之內容，讓共用虛擬位址空間的載入/儲存存取能夠被傳送至遠端記憶體。

系統軟體支援可讓執行時期系統能夠透通且動態地移動虛擬分頁的位置，使得CPU與加速器之間之共同的共用虛擬位址空間可被產生，且此程式之執行時期工作集的局

部行為被利用來找出位於遠端的虛擬分頁（若存取不頻繁的話），或找出有關頻繁存取的分頁之位於本地的虛擬分頁。

在各種的實施例中，可提供延伸虛擬記憶體支援的不同機制。一種實施並不包括對現有的傳統分頁系統設計之任何OS改變，而其他的實施可將更多的資訊加入分頁表項目。這些機制涉及支援反向代理執行的類似架構機制，亦即，CPU用以識別及服務不被對映於系統記憶體上，反而被對映至通過週邊架構之加速器的本地之遠端實體記憶體的那些虛擬位址存取之能力。

為了支援RPE，CPU可識別給定的虛擬位址是否被對映至系統記憶體，或對映至通過週邊架構的遠端記憶體。若實體位址被對映至系統記憶體，則此存取可使用正常記憶體存取而於本地被處理，否則可指示RPE來處理此存取。在一個實施例中，RPE可使用專用的微碼流程來予以實施。RPE可藉由標記具有將藉由微碼處理程式來予以處理之特別錯誤條件的存取（例如，載入/儲存（LD/ST））而開始。此處理程式可將此存取轉換成通過週邊架構的讀取/寫入/直接記憶體讀取（READ/WRITE/DMA）交易，然而許多變化是可行的。為了簡化說明，假設週邊架構為PCIe™互連，且遠端實體記憶體之每一個個別的存取被轉換成非可快取的存取，且依序成被轉換成PCIe™資料交易，以使請求/資料通向PCIe™架構。此交易可壓縮原始的虛擬位址或實體位址。實施此存取的CPU執行緒在某些實

施例中，可使遠端存取的未決完成停滯（且可切換到另一執行緒）。當加速器接收到來自CPU的存取請求之PCIe™交易通知時，加速器中的定序器將此請求處理成為特別的中斷事件。定序器自此請求中取得此存取位址及存取型式。若此存取位址為虛擬位址，則定序器可透過本地的MMU，在本地實施此交易，以獲得實體位址。使用實體位址，加速器的定序器實施此儲存（若寫入交易），或獲得用於此載入的資料（若讀取交易）。定序器將回覆（例如，在載入的情況中）壓縮成PCIe™交易，且傳回至主根物件（root）（亦即，CPU）。CPU核心接收完成存取的PCIe™交易及狀態，且恢復連續的操作，其可能根據遠端存取的存取狀態而增加存取錯誤。

現在參考圖2，所顯示的是依據本發明的實施例之反向代理執行的流程圖。如圖2中所顯示，方法200可被使用來存取存在於加速器的本地記憶體（亦即，有關CPU的遠端記憶體）上之資料。如同圖2中所看到的，方法200可藉由接收記憶體存取請求（方塊210）而開始。此請求可於主CPU中被接收，其依序可將此請求傳送至MMU（例如，TLB），以判定此項目是否關於存在於此TLB中的虛擬位址（菱形220）。若否，可執行分頁遺失處理程式，以獲得及儲存此項目至此TLB中（方塊225）。

當此項目存在於此TLB中時，其可被分析，以判定對應的實體位址之位置（方塊230）。例如，如同下面進一步所討論的，各TLB項目可包括表示對應的分頁是否存在

於本地（亦即，系統記憶體）或遠端記憶體中之資訊。若實體位址為存在於系統記憶體中（菱形240），則控制通往方塊245，其中，可實施對系統記憶體的記憶體存取請求，因此，所請求的資料可被提供為請求者的回應（方塊250）。

若在菱形240，反而被判定實體位址不在系統記憶體中，則控制反而通往方塊260。在方塊260，可準備反向代理執行請求，以將記憶體存取請求傳送至包括資料的遠端記憶體（例如，加速器的本地記憶體）。在不同的實施中，此請求可被通向非同調互連，例如，作為專門的載入/儲存請求。在此反向代理執行請求於加速器上被處理之後，控制通往方塊270，其中，反向代理執行請求的結果被接收（亦即所請求的資料被接收），且回應可被提供給請求者，以上所討論的有關於方塊250。雖然以圖2的實施例中之特定實施來予以顯示，但是本發明的範圍在此方面並未限制。

現在參考圖3，所顯示的是繪示依據本發明的實施例之反向代理執行的操作之方塊圖。如圖3中所顯示，當存取（1）自CPU 120至MMU 125的虛擬位址（V.A.）X之請求失敗時（例如，查表顯示此分頁為存在於此加速器卡的本地記憶體中），反向代理執行請求被準備且傳送（2）至加速器卡150。依序地，加速器160處理此請求，以將記憶體存取請求（3）傳送至MMU 165，其依序存取（4）本地記憶體180中之所請求的分頁，使得所請求的資料可被

傳回至 CPU 120 (5)。要注意的是，來自此加速器的資料可透過 DMA 而被傳送至主控端 CPU，或根據主控端內部之美化的 (glorified) LD/ST 實施 (例如，在微碼流程中) 來輪詢全部。換言之，一旦 CPU 將存取傳出至遠端記憶體位置，CPU 便將此存取轉回至微碼常式，以實施美化的 LD/ST，其將等待 DMA 中斷，或實施主動輪詢，以獲得來自此加速器的「返回」資料。(V.A.) X 的記憶體存取之完成係以透過於應用程式 (其存取虛擬位址 X 處所識別的記憶體位置) 之方式來予以實施。

整體而言，RPE 操作的動作類似通過非一致性記憶體架構 (NUMA) 系統的長傳統記憶體存取操作。潛在的通向機制可根據此架構的特性而變化。在 PCIe™ 架構的情況中，由於根物件 (系統) 與子物件 (加速器) 之間的非對稱複雜 (其中，即使 CPU 不能經常存取任何之加速器的本地記憶體，但是此加速器可存取系統記憶體的範圍)，所以 RPE 效能機制的不同最佳化可藉由使用系統記憶體的部分，或加速器的本地記憶體，作為私有記憶體來予以達成。在一個實施例中，系統記憶體的部分可被保留作為用於遠端加速器本地記憶體的快取記憶體。或者，私有記憶體區域可被配置來用做為緩衝區，以保持於遠端被存取的虛擬分頁。對於例如被對映至遠端分頁之虛擬位址 X 的存取而言，會導致整個分頁暫時地被讀取至本地緩衝區中，其中，其將可用於未來的存取，以降低遠端記憶體存取。

在一個實施例中，代理執行機制可被使用於處理加速

器的定序器上所引起之分頁錯誤情況中，其表示該錯誤可被傳至CPU以供處理。這樣意謂加速器的定序器上之MMU為與CPU的MMU一致，且全部指向OS的相同分頁表。虛擬分頁的分頁錯誤（是否藉由CPU或加速器上的操作所引起）致使CPU使用傳統的分頁處理機制，以使此分頁進入記憶體中。若此錯誤源自加速器的定序器上之存取，則CPU可將新分頁安置於遠端加速器的本地實體記憶體中。否則，此分頁可被放置於系統記憶體中。對映至遠端加速器的本地記憶體之虛擬位址的CPU上之非錯誤存取將保證對映至加速器上的實體分頁，因而確保代理執行的完成。

現在參考圖4，所顯示的是依據本發明的實施例之代理執行的方法之流程圖。如同在圖4中所看到的，當加速器所想要的資料不存在於其本地記憶體中時，方法300可被使用來實施代理執行。

如同在圖4中所看到的，方法300可藉由接收來自加速器的記憶體存取請求而開始（方塊310）。其然後可被判定用於此請求的虛擬位址之項目為存在於此加速器的TLB中（菱形350）。若如此，則此加速器的本地記憶體可使用此虛擬位址（方塊370），及提供給請求者的回應（方塊380）來予以存取。

否則，若項目不是存在於TLB中，則控制通往方塊330，其中，代理執行請求可被傳送至CPU（方塊330）。假設所請求的轉譯不存在於CPU的MMU中，則會執行分頁漏失處理程式，以獲得項目（方塊335）。再者，對應於此

虛擬位址的分頁可自系統記憶體而被移至加速器的本地記憶體（方塊340）。然後恢復訊息可自CPU而被傳送至加速器（方塊350）。因此，加速器會重試對其TLB的記憶體請求（方塊360）。因為項目現在為存在於MMU中，所以可實施對本地記憶體的記憶體存取請求，以獲得所請求的資料（方塊370）。因此，包括所請求的資料之回應可被提供給請求者（方塊380）。

現在參考圖5，所顯示的是繪示依據本發明的實施例之代理執行操作的方塊圖。如圖5中所視，代理執行會出現在加速器請求存取（1）不存在於本地MMU 165中的虛擬位址（V.A）X的時候。因此，此分頁錯誤上的代理執行請求（2）被傳送至CPU 120。圖5顯示MMU 165直接將代理執行請求傳送至CPU 120的實施。然而，當此代理係以分頁移動（page-walking）碼（根據MMU如何被管理之微碼、韌體或軟體）來予以實施時，則此碼可傳送代理請求。CPU 120將此請求（3）傳送至本地MMU 125，其依序存取有關所存取的虛擬位址（4）之系統記憶體130中的分頁。如所視，當此請求為用於資料的讀取時，整個分頁可自系統記憶體130而被傳送（5）至本地記憶體180。在一個實施例中，資料的轉移可為透過CPU 120，其將DMA程式化，以使來自系統記憶體130中的區域之資料複製到本地記憶體180中的區域。另一種是，CPU 120可藉由執行個別「美化的」載入/儲存指令序列（例如，使用如上述的微碼來實施）來實施複製。之後，CPU 120可將恢復訊息（6

）傳送至加速器 160（其依序重試 MMU 165 的存取（7）），其現在找出目前的轉譯，且傳送此請求（8），以獲得來自本地記憶體 180 的對應資料。要注意的是，為了能存取本地記憶體 180 中的資料，MMU 165 被使得可定址的。CPU 120 可直接實施個別「美化的」LD/ST，以使來自系統記憶體 130 上的分頁表之轉譯項目更新成 MMU 165。另一種是，CPU 120 可將包含轉譯的分頁表或分頁表的子組透過資料轉移而複製到加速器側，而至本地記憶體 180，然後恢復加速器 160，其的分頁移動程式（page walker）將使現在存在於本地之分頁表移動。

區別 CPU 上的虛擬位址存取是否為本地（在系統記憶體中）或遠端（在加速器的記憶體中）之資訊可來自 OS，其自基本輸入/輸出系統（BIOS）中得知此種資訊，BIOS 對有關係統記憶體組態，有完全的認識。為了支援 RPE，BIOS 可列舉加速器上之公布的記憶體大小。此操作類似在開機時間，藉由 BIOS 所實施之唯讀記憶體（ROM）/隨機存取記憶體（RAM）的晶片選取。BIOS 然後會回報系統記憶體及加速器本地記憶體的總和，且將記憶體的何範圍為本地系統記憶體，而何者為遠端通知 OS。

在不同的實施例中，用於 BIOS 的系統層級狀態（亦即，一組描述符架構狀態）（稱為記憶體分割區描述符）可記錄此範圍資訊（例如，在最低限度，用於系統記憶體的範圍資訊），因此來自此範圍的任何實體位址將被識別為遠端。在一個實施例中，此資訊可被儲存於 BIOS 內建的資

料結構中。記憶體描述符也可被儲存為軟體及微碼可存取之機器特定暫存器中的私有狀態。要注意的是，在OS開始之前，此種範圍資訊首先係藉由BIOS來予以建立，所以這些狀態的使用不會依賴OS。換言之，RPE機制可與甚至不瞭解遠端與本地的記憶體之間的不同之傳統的OS一起運作。

對於藉由CPU所處理的任何給定LD/ST而言，使各TLB轉譯也將實體位址與記憶體分割區描述符做比較，以決定是否其為本地系統記憶體或遠端記憶體是麻煩的。反而，此種查核可離開MMU中的臨界路徑而被實施，且僅會在填滿新TLB項目之後，才出現於分頁移動上。在某些實施例中，各TLB項目可包括屬性位元，以表示對應的項目是否為在遠端或本地系統記憶體中。當安置新TLB項目時，分頁移動程式可對著記憶體分割區描述符，實施分頁表中之實體位址範圍的範圍查核。要注意的是，即使OS不在本地或遠端對映的分頁之間做區別，此機制仍運作。

在某些實施例中，OS可藉由使用加速器本地記憶體來處理加速器本地記憶體的使用之後的策略，以僅保持加速器頻繁地存取之應用程式的碼及資料之子組。若OS不知道，則例如執行時期層或其他實體的區域性原則將協助使工作集移至靠近存取更頻繁地發生於系統記憶體或加速器本地記憶體中之處。

此外，如上所述，OS分頁表格式可包括屬性位元，以表示對應的分頁是否係儲存於本地或遠端記憶體中。當OS

建構虛擬位址至實體位址的對映時，此位元可被標記，且對於各實體分頁而言，OS會查核記憶體分割區描述符，以將此分頁標記為本地或遠端。以此方式，不需對任何安置的TLB項目實施範圍查核。為了讓應用程式能存取加速器上的記憶體，CPU會分析屬性位元，使得其可將載入/儲存傳送至給定的虛擬位址，而至遠端實體記憶體位置。此外，屬性位元也會追蹤在遠端實施之存取的數量，使OS軟體能根據遠端存取的數量來實施策略，使得若遠端存取的數量超過特定臨限值，則此分頁會被遷移至另一位置。

雖然藉由實施虛擬分頁的保護（諸如，將分頁標記為不可存取或不存在的）且處理所產生的錯誤來實施遠端記憶體存取是可行的，但是因為每次記憶體存取發生時，分頁錯誤處理程式需執行，所以存取延遲增加。反而，使用本發明的實施例，CPU可將遠端記憶體位置的位址提交給匯流排控制器。其將此存取指向加速器的記憶體位置。例如，CPU可藉由存取匯流排控制器中所界定之暫存器的標準集來直接重新指向載入/儲存，以存取遠端記憶體位置，而不用來自軟體的任何協助來完成載入/儲存。此資料轉移可為藉由DMA（大批轉移），或快取線精細度（*cacheline granularity*）的純量轉移。使虛擬分頁自遠端記憶體位置透通地移至本地記憶體位置（且反之亦然）之能力讓軟體（例如，應用程式）能與加速器一起共用資料，而不用詳盡地管理資料的移動。在缺少連接至此系統或進入未回應狀態的加速器中，位址轉譯單元產生表示為什麼

載入/儲存失敗的原因之分頁錯誤。

現在參考圖6，所顯示的是依據本發明的一個實施例之分頁表中的範例項目400。如同所看到的，各項目可包括分頁基礎位址（PBA）欄位410，其儲存指向記憶體中所儲存之分頁的第一位址之PBA。此外，各項目可包括加速器欄位420（其儲存有關於存在於此系統中的加速器之n位元加速器識別符（ID））、本地/遠端欄位430（用以儲存，例如，n位元指示符，有關於對應的虛擬分頁是否係儲存於本地記憶體，或可能的多個遠端記憶體的其中之一遠端記憶體中）、計數器欄位440（用以儲存m位元計數器，其計數實施遠端記憶體的存取之數量（使得此計數器僅在此分頁為在遠端記憶體中的時候動作））、以及屬性欄位450（用以儲存表示此分頁的不同屬性之不同的位元）。

在一個實施例中，當存取計數器的值到達零時，例外會增加。此例外讓OS軟體（例如，負責用於分頁遷移的虛擬記憶體管理核心）能夠根據給定的虛擬分頁之存取的數量來管理遷移策略。亦即，此軟體可管理應用程式運作的虛擬位址空間，使得虛擬位址空間可對映設置成更靠近CPU，或更靠近加速器的實體記憶體分頁。對於耦接至PCIe™匯流排的加速器而言，因為此匯流排為非同調，所以潛在的執行時期軟體可實施以軟體為基礎的同調機制。對於任何共用資料結構之競爭的存取而言，可使用類似號誌（semaphore）的同步化控制，使得生產者執行緒直到

立即將此資料交遞至消費者，才會發佈此號誌。在生產者發佈此號誌之前，其需要將有關共用資料的所有髒快取線刷新至記憶體中。這保證當加速器上的消費者執行緒開始存取來自記憶體中的共用資料時，即使主CPU與加速器之間的架構不支援快取記憶體同調，但是此資料是同調的。反之，當加速器完成處理共用資料時，可使用類似的同步化及刷新（flush）機制，以確保以記憶體為基礎的資料同調。CPU與加速器之間的架構應該為快取記憶體同調（例如，PCIe的後代），在處理器發佈此號誌之前的交遞之後，不會立即需要將髒線刷新至記憶體。

在具有OS支援的實施例中，加速器上之記憶體的配置及管理可與此OS的記憶體管理器合作來予以實施，此OS的記憶體管理器配置及管理系統記憶體分頁，系統記憶體分頁被提供給應用程式，且管理藉由CPU所利用的分頁表，以使虛擬位址轉譯至實體位址。記憶體管理器也處理由於存取遠端記憶體的重新指向而出現之例外，且管理CPU與加速器之間的實體分頁之遷移之後的策略。分頁遷移策略可根據工作負載的行為而變化，且可潛在地被改變，以降低遠端存取的數量（在系統記憶體之對應的分頁之移動之前），或實施第一接觸策略，以使此分頁移至有存取的最大數量之位置。碼及唯讀資料分頁可被複製於多個記憶體中，以防止實體分頁來回地不必要移動。只有在程式的執行期間被處理之包含資料的資料分頁係根據資料分頁的存取之區域性而來回地遷移。

現在參考圖 7，所顯示的是依據本發明的一個實施例之處理器的方塊圖。如圖 7 中所顯示，處理器 500 可為多階段管線的亂序（out-of-order）處理器。在圖 7 中，處理器 500 係以相當簡化的圖式來予以顯示，以繪示依據本發明的一個實施例之與代理執行及反向代理執行結合所使用的不同特性。

如圖 7 中所顯示的，處理器 500 包括前端單元 510，其可被使用來提取要被執行的巨集指令，且準備其供稍後於此處理器中使用。例如，前端單元 510 可包括提取單元 504、指令快取記憶體 506、及指令解碼器 508。在某些實施中，前端單元 510 可另包括伴隨微碼儲存器以及微操作儲存器的追蹤快取記憶體。提取單元 504 可，例如，自記憶體或指令快取記憶體 506 中提取巨集指令，且將其饋入至指令解碼器 508，以將其解碼成原始碼，亦即，由此處理器所執行的微操作。前端單元 510 另包括依據本發明的實施例之 MMU 509，用以儲存包括處理如在此上所述的代理執行及反向代理執行之增加的狀態資訊。根據此種資訊，包括一個或多個加速器的系統記憶體或本地記憶體之共用虛擬記憶體空間中的操作可有效地被實施，而不需要程式設計者對移動資料的支援。

耦接於前端單元 510 與執行單元 520 之間的是亂序（OOO）引擎 515，其可被使用來接收微指令，且準備其供執行之用。更特別的是，OOO 引擎 515 可包括不同的緩衝區，用以重新排序微指令流程，且配置用於執行所需的不

同資源，以及用以將邏輯暫存器的重新命名提供至諸如暫存器檔案530及延伸暫存器檔案535之不同的暫存器檔案內之儲存器位置中。暫存器檔案530可包括用於整數及浮點操作的分離暫存器檔案。延伸暫存器檔案535可提供用於向量大小的單元（例如，每個暫存器之256或512個位元）之儲存。

不同的資源可存在於其他專用硬體之中的執行單元520中，執行單元520包括，例如，不同的整數、浮點、及單一指令多個資料（SIMD）的邏輯單元。結果可被提供給撤離（retirement）邏輯，亦即重新排序緩衝區（ROB）540。更特別而言，ROB 540可包括用以接收與被執行的指令相關聯之資訊的不同陣列及邏輯。此資訊然後係藉由ROB 540來予以檢查，以判定指令是否可被有效地撤離，及產生實施用於此處理器的架構狀態之資料，或所出現的一個或多個例外是否阻止指令的適當撤離。當然，ROB 540可處理其它與撤離相關的操作。

如圖7中所顯示，ROB 540係耦接至快取記憶體550，其在一個實施例中，可為低階快取記憶體（例如，L1快取記憶體），然而本發明的範圍在此方面並未限制。再者，執行單元520可被直接耦接至快取記憶體550。自快取記憶體550中，會出現與較高階快取記憶體、系統記憶體等等的資料相通訊。雖然以圖7的實施例中之此高階來予以顯示，但是瞭解本發明的範圍在此方面並未限制。

例如，在某些實施中，加速器可與處理器一起被整合

於晶片上。例如，在一種架構中，多核心處理器可包括伴隨著加速器（其可為例如圖形處理器的異質核心，或其他專用處理單元）之一些個別處理器核心。一般而言，代理執行及反向代理執行的操作會以如上述之用於晶片上加速器（其可藉由任何型式的互連（包括同調或非同調鏈結）而被耦接至核心）上之相同的方式而出現。

現在參考圖 8，所顯示的是依據本發明的另一個實施例之系統的方塊圖。如圖 8 中所視，系統 600 包括多核心處理器 610，其可被形成於單一半導體晶粒上，且包括不同的組件。特別是，在圖 8 的實施中，處理器 610 可包括複數個核心 620_a-620_n ，其的每一個核心可包括 TLB 625_a-625_n ，其可包括具有諸如以上所討論之轉譯及額外的欄位之項目。依序地，核心可被耦接至共用快取記憶體 640，其可為共用最後階快取記憶體，原因是個別核心的每一個核心可包括其本身的快取記憶體。如同進一步所看到的，處理器 610 也可包括加速器。在所顯示的實施例中，顯示兩個此類的加速器，然而本發明的範圍在此方面並未限制。加速器至核心的連接可為任何型式的互連，諸如同調或非同調鏈結，例如，PCIeTM鏈結、共用匯流排互連、或等等。加速器 630_a 及 630_b 係顯示成每一個包括具有如上述的項目之 TLB 635。此外，處理器 610 可包括記憶體控制器 650。

在一種實施中，處理器 610 可被耦接至記憶體 660，其可為可被分割成多個分割區（例如，包括可與處理器核心相關聯的第一分割區 665_a ，及可與加速器相關聯的第二分

割區 665_b) 之系統記憶體。當然，與核心及加速器相關聯的記憶體可例如透過不同的埠而被不同地組構，且作為不同的記憶體模組，或等等。處理器 610 可進一步被耦接至晶片組 670，其依序可被耦接至不同的週邊裝置，諸如輸入/輸出裝置、儲存裝置、及其他的加速器等。

因此，實施例可在可包括整合的加速器，或透過可為同調或非同調的鏈結而耦接的加速器之不同的系統中，提供用於代理執行及反向代理執行的處理。

實施例可以碼來予以實施，且可被儲存於儲存媒體上，儲存媒體具有儲存於其上的指令，其可被使用來將系統程式化，以實施指令。儲存媒體可包括，但不受限於，任何型式的碟片（包括軟碟、光碟、固態硬碟（SSD）、光碟唯讀記憶體（CD-ROM）、光碟可覆寫（CD-RW）、及磁性光學碟片）、半導體裝置（諸如唯讀記憶體（ROM）、諸如動態隨機存取記憶體（DRAM）的隨機存取記憶體（RAM）、靜態隨機存取記憶體（SRAM）、可抹除可程式化唯讀記憶體（EPROM）、快閃記憶體、電氣式可抹除可程式化唯讀記憶體（EEPROM）、磁性或光學卡）、或適用用於儲存電子指令之任何其他型式的媒體。

雖然本發明已關於有限數目的實施例來予以說明，但是熟習此項技術者將瞭解的是，由此有許多的修改及變化。其係意謂後附的申請專利範圍涵蓋如落入此本發明的真實精神及範圍內的所有此類修改及變化。

【圖式簡單說明】

圖 1 係依據本發明的一個實施例之系統的方塊圖。

圖 2 係依據本發明的實施例之反向代理執行的方法之流程圖。

圖 3 係繪示依據本發明的實施例之反向代理執行操作的方塊圖。

圖 4 係依據本發明的實施例之代理執行的方法之流程圖。

圖 5 係繪示依據本發明的實施例之代理執行操作的方塊圖。

圖 6 係依據本發明的一個實施例之分頁表的範例項目。

圖 7 係依據本發明的一個實施例之處理器的方塊圖。

圖 8 係依據本發明的實施例之包括晶片上加速器的系統之方塊圖。

【主要元件符號說明】

100：系統

110：主個人電腦（PC）平台

120：CPU

125：記憶體管理單元（MMU）

130：系統記憶體

140：非同調互連

150：加速器卡

155：橋接器
 160：智慧財產權（IP）區塊
 165：MMU
 170：晶片（SoC）架構
 180：本地記憶體
 400：項目
 410：分頁基礎位址（PBA）欄位
 420：加速器欄位
 430：本地/遠端欄位
 440：計數器欄位
 450：屬性欄位
 500：處理器
 504：提取單元
 506：指令快取記憶體
 508：指令解碼器
 509：MMU
 510：前端單元
 515：亂序（OOO）引擎
 520：執行單元
 530：暫存器檔案
 535：延伸暫存器檔案
 540：重新排序緩衝區（ROB）
 550：快取記憶體
 600：系統

610：多核心處理器

620_a：核心

620_n：核心

625_a：TLB

625_n：TLB

630_a：加速器

630_b：加速器

635_a：TLB

635_b：TLB

640：共用快取記憶體

650：記憶體控制器

660：記憶體

665_a：第一分割區

665_b：第二分割區

670：晶片組

七、申請專利範圍：

1.一種為本地記憶體與遠端實體記憶體間之共用虛擬記憶體提供硬體支援之方法，該方法包含：

接收包括虛擬位址的記憶體存取請求；

分析對應於處理器的轉譯後備緩衝區（TLB）中所儲存之該虛擬位址的項目，以根據該TLB項目之位置屬性而判定對應於該虛擬位址的實體位址（PA）是否存在於與該處理器相關聯的本地記憶體，或與透過非同調鏈結而被耦接至該處理器的加速器相關聯之遠端記憶體中，以指示該PA是否存在於該本地記憶體或一或多個遠端記憶體的其中一個遠端記憶體中，該本地記憶體及該遠端記憶體共同形成共用虛擬記憶體空間；以及

若該PA係存在於該遠端記憶體中，則將反向代理執行請求傳送至該遠端記憶體，以實施該記憶體存取請求。

2.如申請專利範圍第1項之方法，另包含根據該TLB項目的位置指示器來判定該PA是否存在於該本地記憶體或該遠端記憶體中。

3.如申請專利範圍第2項之方法，另包含判定複數個遠端記憶體之各遠端記憶體的何者與加速器相關聯，其中，使用指示該加速器與該遠端記憶體相關聯之該TLB項目的識別符欄位來找到該PA。

4.如申請專利範圍第3項之方法，另包含分析該TLB項目的計數器，其指示藉由該處理器之該遠端記憶體的該PA之存取的數目。

5.如申請專利範圍第4項之方法，另包含當該計數器到達臨限值時，將來自該遠端記憶體之該PA之資訊移至該本地記憶體，而未涉及於該處理器上執行的應用程式。

6.如申請專利範圍第1項之方法，另包含設定包括該本地記憶體及該遠端記憶體之系統的記憶體組態，以指示與該本地記憶體相關聯的第一實體位址範圍，及與該遠端記憶體相關聯的第二實體位址範圍。

7.如申請專利範圍第6項之方法，另包含存取分頁移動（page walk）上的該記憶體組態，以獲得用於該記憶體存取請求的轉譯，且當該PA為在該第一實體位址範圍內時，儲存該TLB中的項目，該TLB包括該轉譯，及具有用以表示該PA為在該本地記憶體中之第一值的位置指示器。

8.一種為本地記憶體與遠端實體記憶體間之共用虛擬記憶體提供硬體支援之設備，該設備包含：

處理器，具有包括第一記憶體管理單元（MMU）的第一核心，該第一MMU包括用以儲存虛擬位址至實體位址轉譯的複數個項目，其中，各項目包括用以儲存表示該對應的項目之記憶體位置是否存在於耦接至該處理器的本地記憶體中，或耦接至透過非同調互連而被耦接至該處理器的加速器之遠端記憶體中的第一指示器之位置欄位，及用以儲存與該遠端記憶體相關聯之加速器的識別符之識別符欄位，其中，該處理器係要執行處理程式，以將用於該遠端記憶體中之記憶體位置的記憶體請求轉換成用於透過該非同調互連之通訊的直接記憶體存取交易。

9.如申請專利範圍第8項之設備，其中，該第一MMU的各項目另包括計數器，用以儲存藉由該處理器之該遠端記憶體之該記憶體位置之存取的數目之計數。

10.如申請專利範圍第8項之設備，其中，該本地記憶體為系統記憶體，而該遠端記憶體為該加速器的本地記憶體。

11.如申請專利範圍第10項之設備，其中，該系統記憶體及該遠端記憶體包含單一虛擬位址空間。

12.如申請專利範圍第8項之設備，其中，當該第一MMU的項目表示該記憶體位置為在該遠端記憶體中時，該處理器係要透過該非同調互連的協定而將記憶體請求通向該加速器。

13.如申請專利範圍第8項之設備，其中，該加速器包括第二MMU，且回應於該第二MMU上的分頁錯誤，該加速器係要請求該處理器解決該分頁錯誤。

14.如申請專利範圍第13項之設備，其中，該處理器係要致使該本地記憶體將與該分頁錯誤的位址相關聯之記憶體分頁轉移至該遠端記憶體，且要致使被傳送至該加速器的該記憶體分頁之轉譯，以用於該第二MMU中的儲存。

15.如申請專利範圍第8項之設備，其中，該處理器係要使用該第一MMU的項目中之資訊來直接定址存在於該遠端記憶體中的記憶體位置。

16.如申請專利範圍第8項之設備，其中，該處理器係要使用該第一MMU的項目中之資訊，而不使用於該處理器

上執行的使用者層級的應用程式，將反向代理執行請求發送至該加速器，以存取該加速器的該遠端記憶體中所儲存之資料。

17.一種為本地記憶體與遠端實體記憶體間之共用虛擬記憶體提供硬體支援之系統，該系統包含：

處理器，具有包括第一記憶體管理單元（MMU）的第一核心，該第一MMU包括用以儲存虛擬位址至實體位址轉譯的複數個項目，其中，各項目包括用以儲存表示該對應的項目之記憶體位置是否存在於系統記憶體中或與加速器組件相關聯的第二記憶體中之指示器的位置欄位，及用以儲存藉由該處理器之該第二記憶體的該記憶體位置之存取的數目之計數的計數器，其中，當該計數器到達臨限值時，該處理器係要發出異常，以致使儲存在該記憶體位置中之資料從存在於該第二記憶體中的該記憶體位置移到存在於該系統記憶體中的記憶體位置；

該加速器組件，包括第二處理器，及第二MMU，該加速器組件透過鏈結而被耦接至該處理器；以及

該系統記憶體，被耦接至該處理器，其中，該系統記憶體包含動態隨機存取記憶體（DRAM）。

18.如申請專利範圍第17項之系統，其中，該處理器係要使用該第一MMU的項目中之資訊來直接定址存在於該第二記憶體中的記憶體位置。

19.如申請專利範圍第17項之系統，其中，該系統記憶體及該第二記憶體包含單一虛擬位址空間。

20.如申請專利範圍第18項之系統，其中，該處理器係要回應於發生在該計數器到達臨限值之時的中斷，將來自存在於該第二記憶體中的該記憶體位置之資訊移至存在於該系統記憶體中的記憶體位置，而不涉及於該處理器上執行的應用程式。

21.如申請專利範圍第17項之系統，其中，該處理器及該加速器組件係由單一半導體晶粒所構成。

22.如申請專利範圍第21項之系統，其中，該第二記憶體為該系統記憶體的分割部分。

23.一種為本地記憶體與遠端實體記憶體間之共用虛擬記憶體提供硬體支援之設備，該設備包含：

處理器，具有第一核心，其包括第一邏輯，用以當該記憶體存取請求的記憶體位置存在於與該加速器相關聯的記憶體中時，將用於資料的記憶體存取請求傳送至耦接至該第一核心的加速器，透過於請求該資料之於該第一核心上執行的使用者層級的應用程式，且回應於該記憶體存取請求而自該加速器中接收該記憶體位置中所儲存的該資料，其中，該處理器可接達之系統記憶體的總和及與該加速器相關聯之該記憶體，連同該總和之對該處理器而言為本地的第一範圍以及該總和之對該處理器而言為遠端的第二範圍之通知，皆被回報至作業系統。

24.如申請專利範圍第23項之設備，其中，該第一核心包括第一儲存器，其包括用以儲存虛擬位址至實體位址轉譯的複數個項目，其中，各項目包括用以表示該對應的

項目之記憶體位置是否存在於與該第一核心相關聯的第一記憶體中，或與耦接至該處理器的加速器相關聯之記憶體中的至少一個指示器，及與該記憶體相關聯的該加速器之識別符。

25.如申請專利範圍第24項之設備，其中，該第一儲存器的各項目另包括計數器，用以儲存藉由該第一核心之與該加速器相關聯的該記憶體之該記憶體位置的存取之數目的計數。

26.如申請專利範圍第25項之設備，其中，若該計數器的值大於臨限值，則該記憶體位置中所儲存的該資料即將被移至該第一記憶體中的第二記憶體位置。

27.如申請專利範圍第26項之設備，其中，該記憶體存取請求及該資料移動的發送係透過於請求該資料的該使用者層級的應用程式。

28.如申請專利範圍第24項之設備，其中，當該第一儲存器的項目表示該記憶體位置為在與該加速器相關聯的該記憶體中時，該處理器係要透過耦接該處理器與該加速器的互連之協定而將記憶體存取請求通向該加速器。

29.如申請專利範圍第23項之設備，其中，該處理器包含多核心處理器，該多核心處理器包括該第一核心及該加速器，該多核心處理器係形成於單一半導體晶粒上。

30.如申請專利範圍第23項之設備，其中，該第一記憶體及與該加速器相關聯的該記憶體包含單一虛擬位址空間。

100

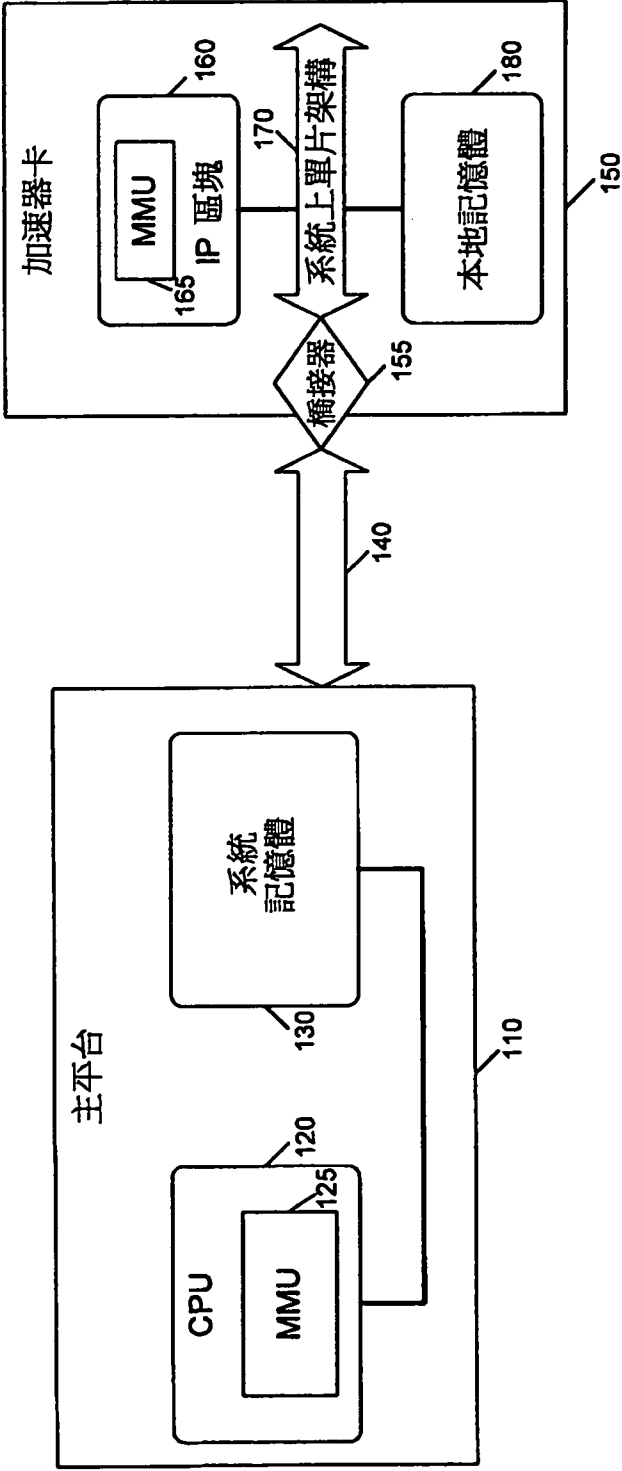


圖1

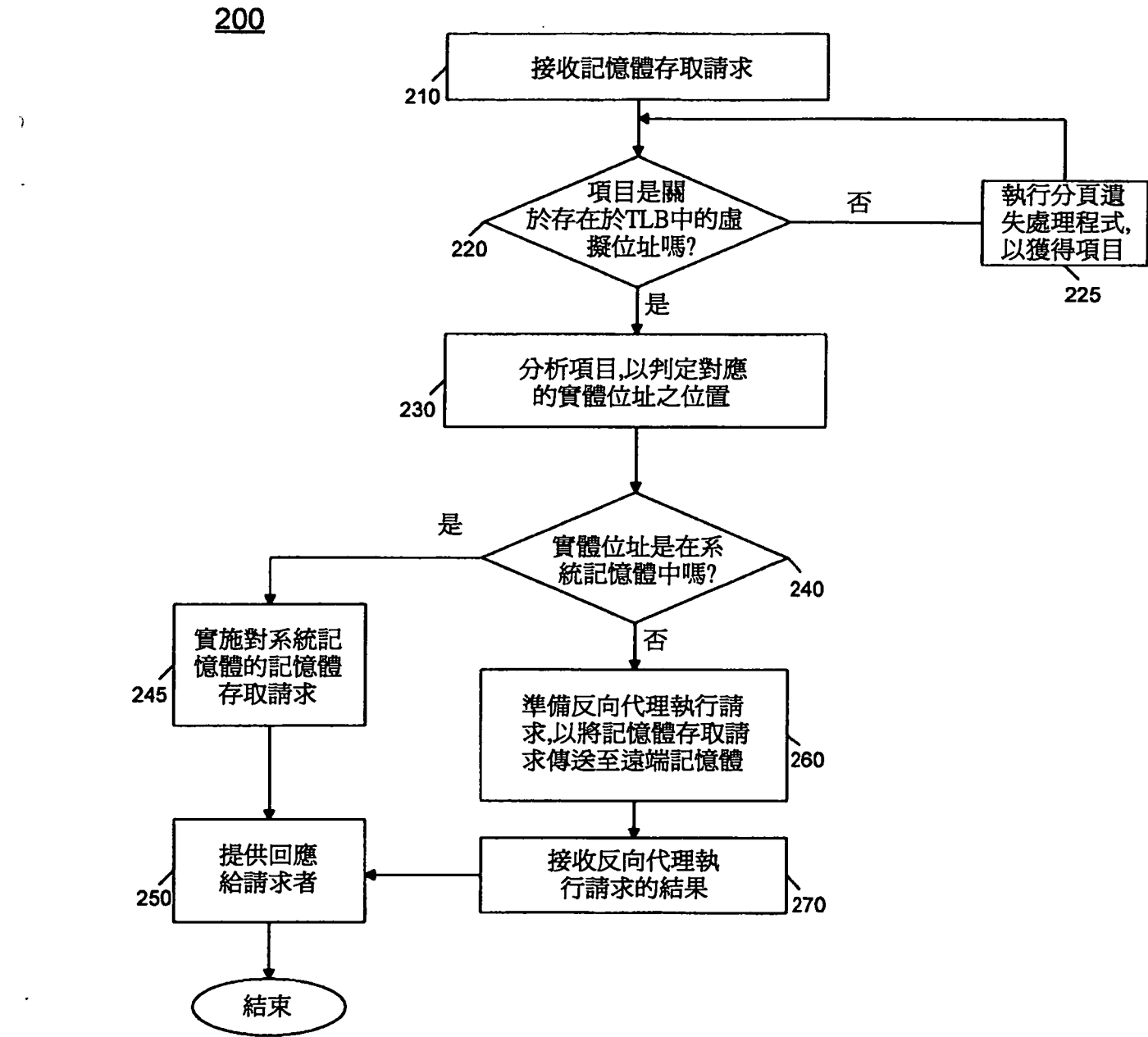


圖2

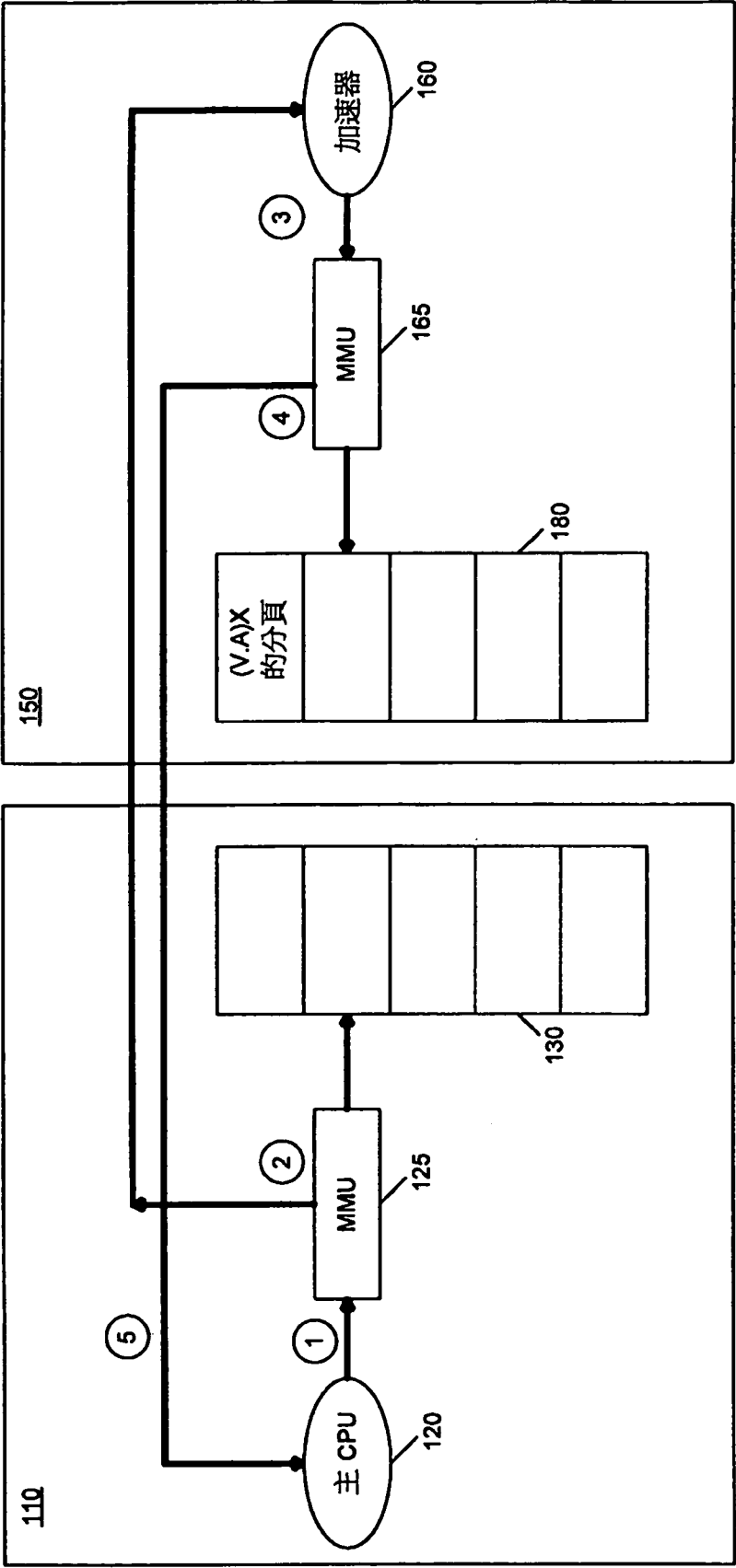


圖3

300

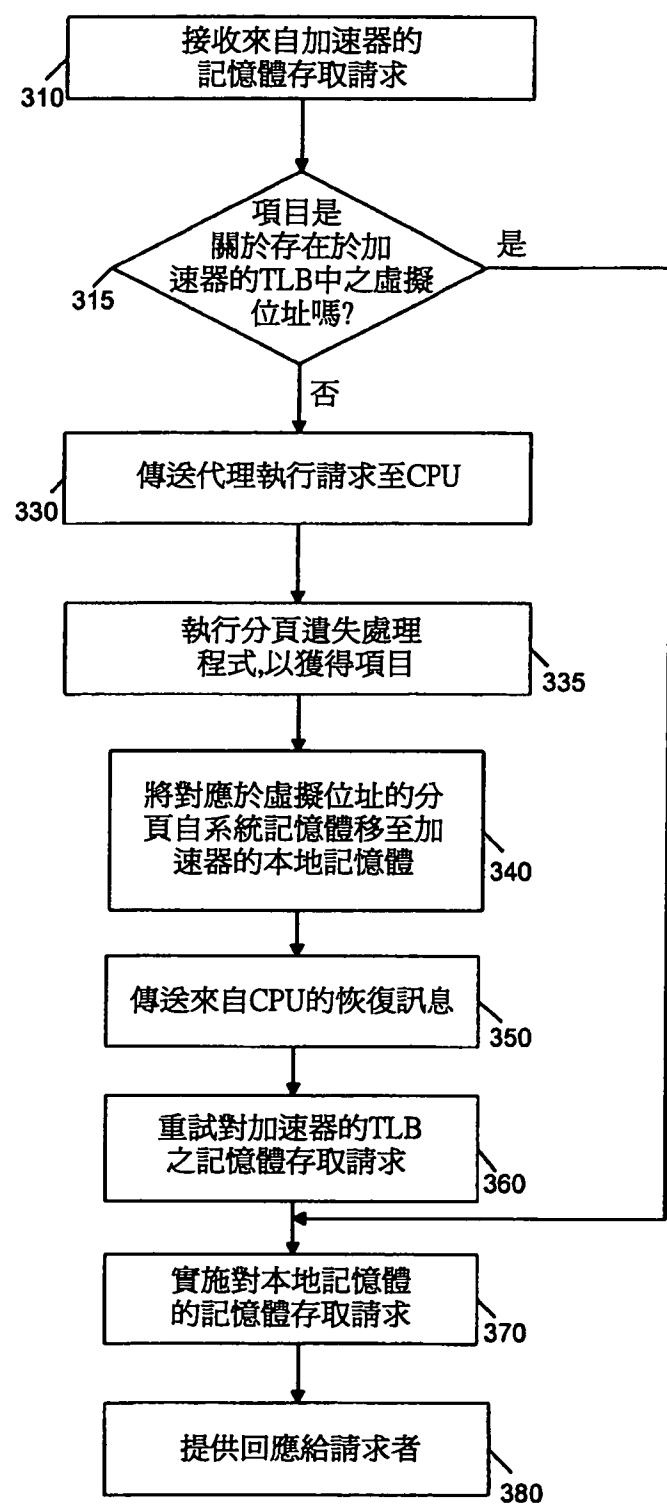


圖 4

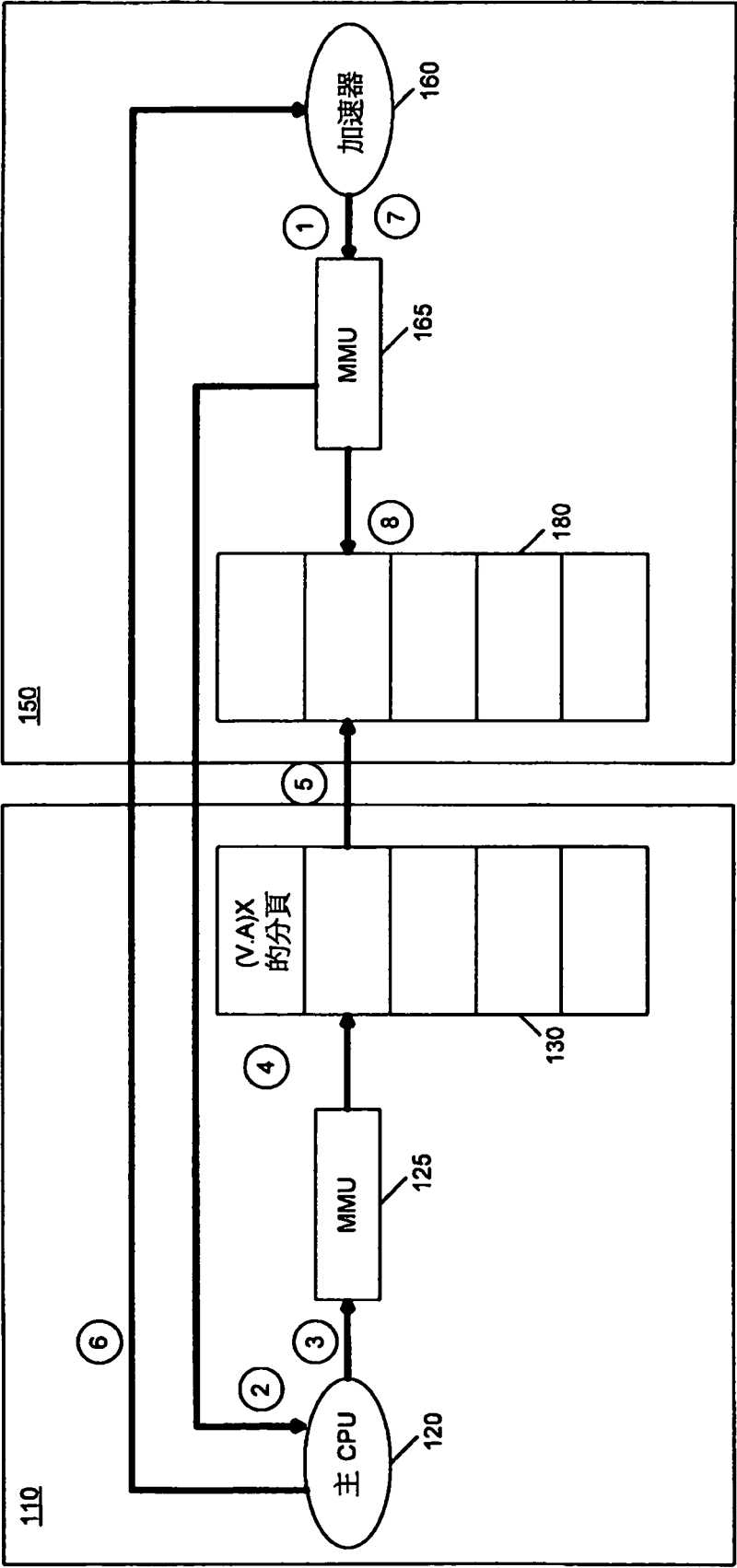


圖5

400

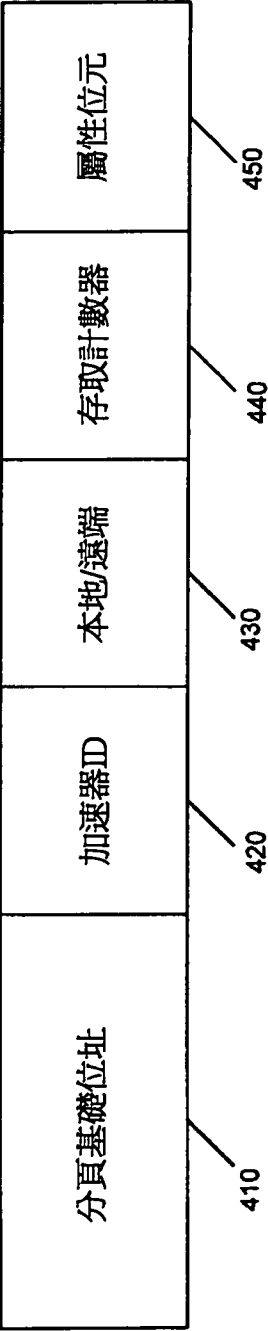


圖6

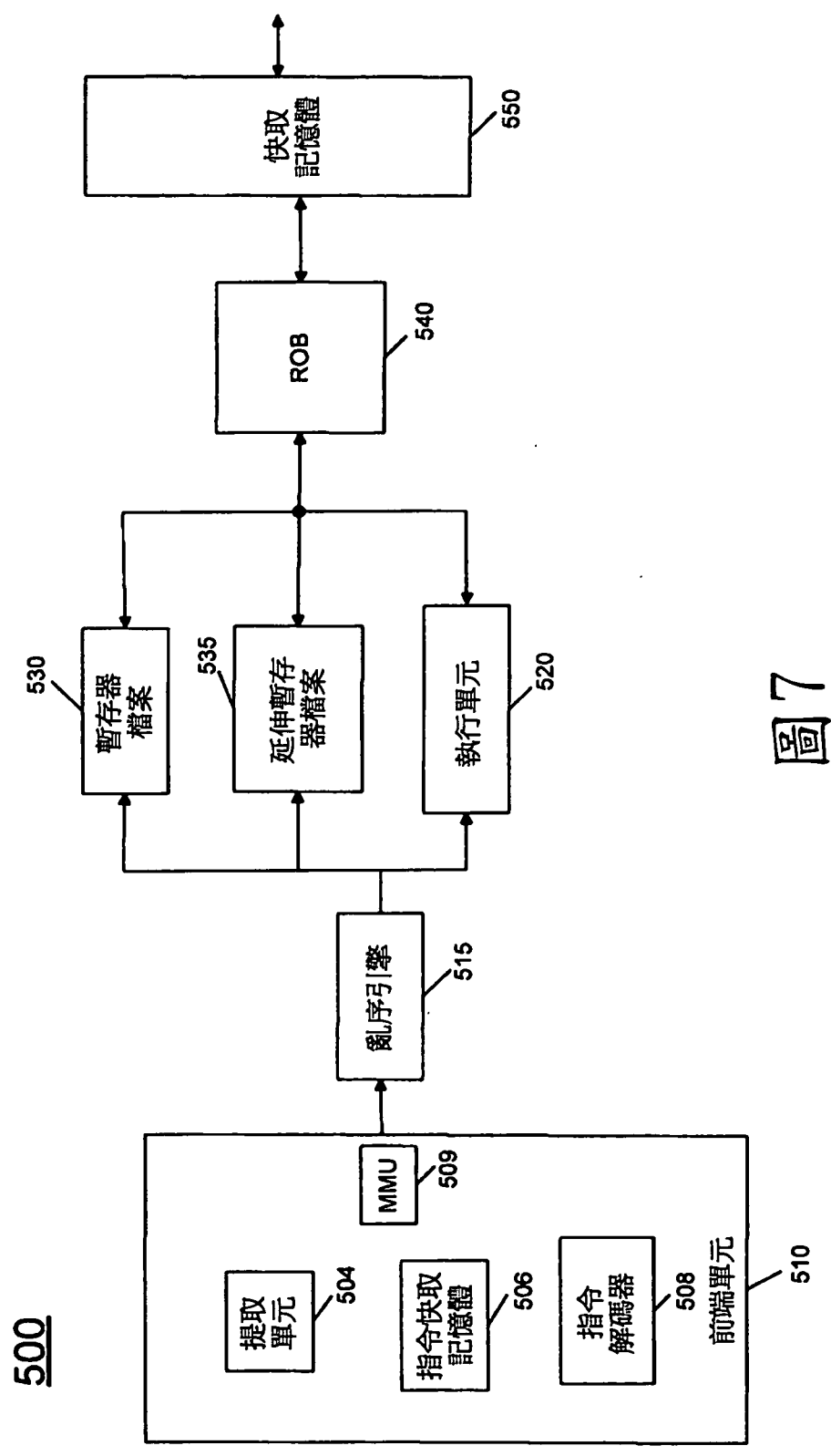


圖7

600

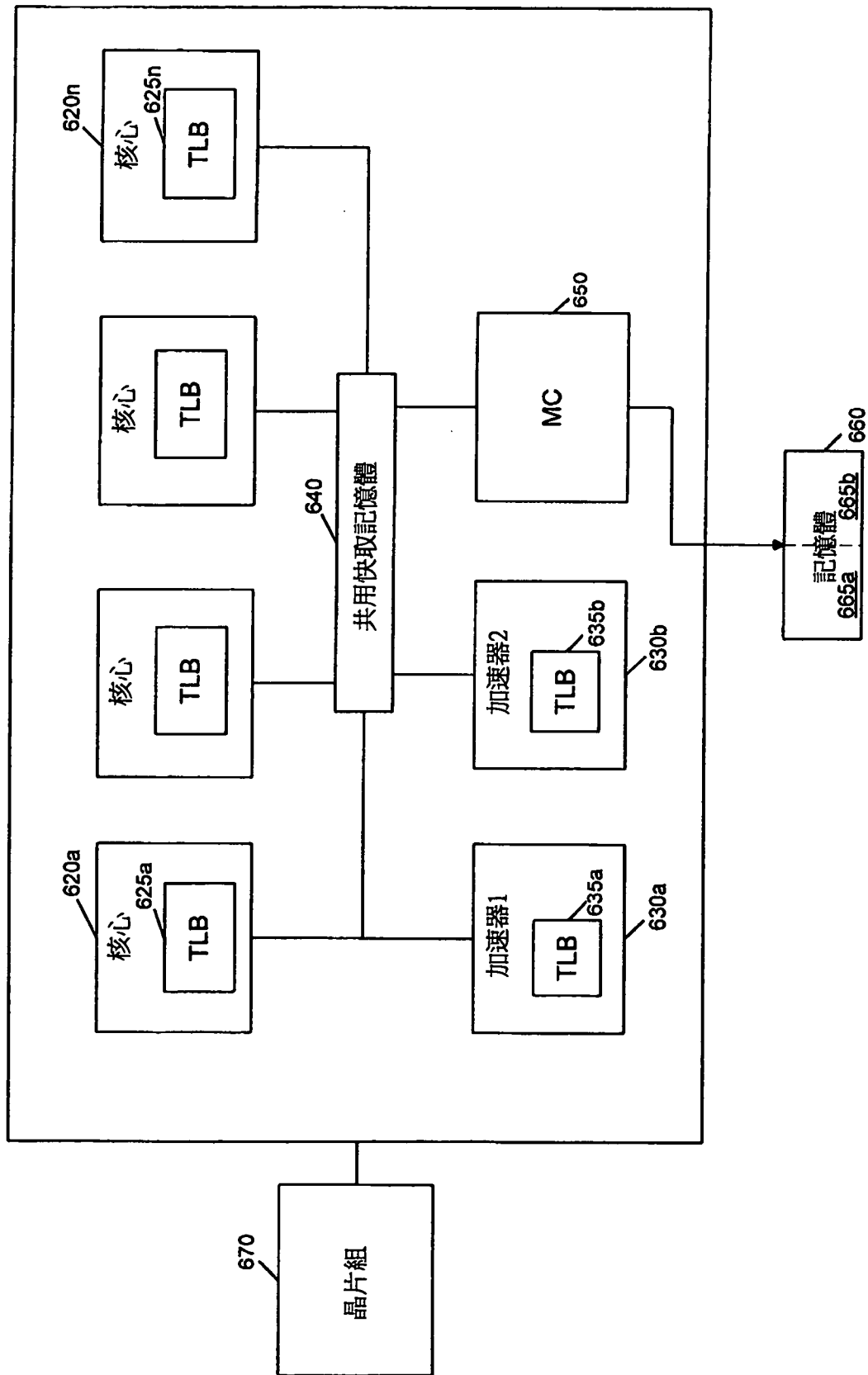


圖8