I461911

(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 I461911 B**

(45)公告日： 中華民國 103 (2014) 年 11 月 21 日

(21)申請案號：100134352 (22)申請日： 中華民國 100 (2011) 年 09 月 23 日

(51)Int. Cl.： ***G06F12/02 (2006.01)***

(30)優先權：2010/09/24 美國 12/890,585

(71)申請人：英特爾公司 (美國) INTEL CORPORATION (US)
美國

(72)發明人：興頓 格倫 HINTON, GLENN (US)；派瑟塞拉希 麥德文 PARTHASARATHY, MADHAVAN (US)；派瑟塞拉希 拉傑什 PARTHASARATHY, RAJESH (IN)；史瓦米那森 木蘇庫瑪 SWAMINATHAN, MUTHUKUMAR (IN)；拉馬努金 拉傑 RAMANUJAN, RAJ (US)；茲梅爾曼 大衛 ZIMMERMAN, DAVID (US)；史密斯 賴瑞 O SMITH, LARRY O. (US)；莫加 亞得連 C MOGA, ADRIAN C. (RO)；卡普 史考特 J CAPE, SCOTT J. (US)；道爾 韋恩 A DOWNER, WAYNE A. (US)；查裴爾 羅伯特 S CHAPPELL, ROBERT S. (US)

(74)代理人：惲軼群；陳文郎

(56)參考文獻：

TW	200708952A	TW	200842579A
US	6360303B1	US	7428626B2
US	2003/0005249A1	US	2010/0082883A1

審查人員：陳泰龍

申請專利範圍項數：25 項 圖式數：14 共 61 頁

(54)名稱

用於實施微分頁表之裝置、方法與系統

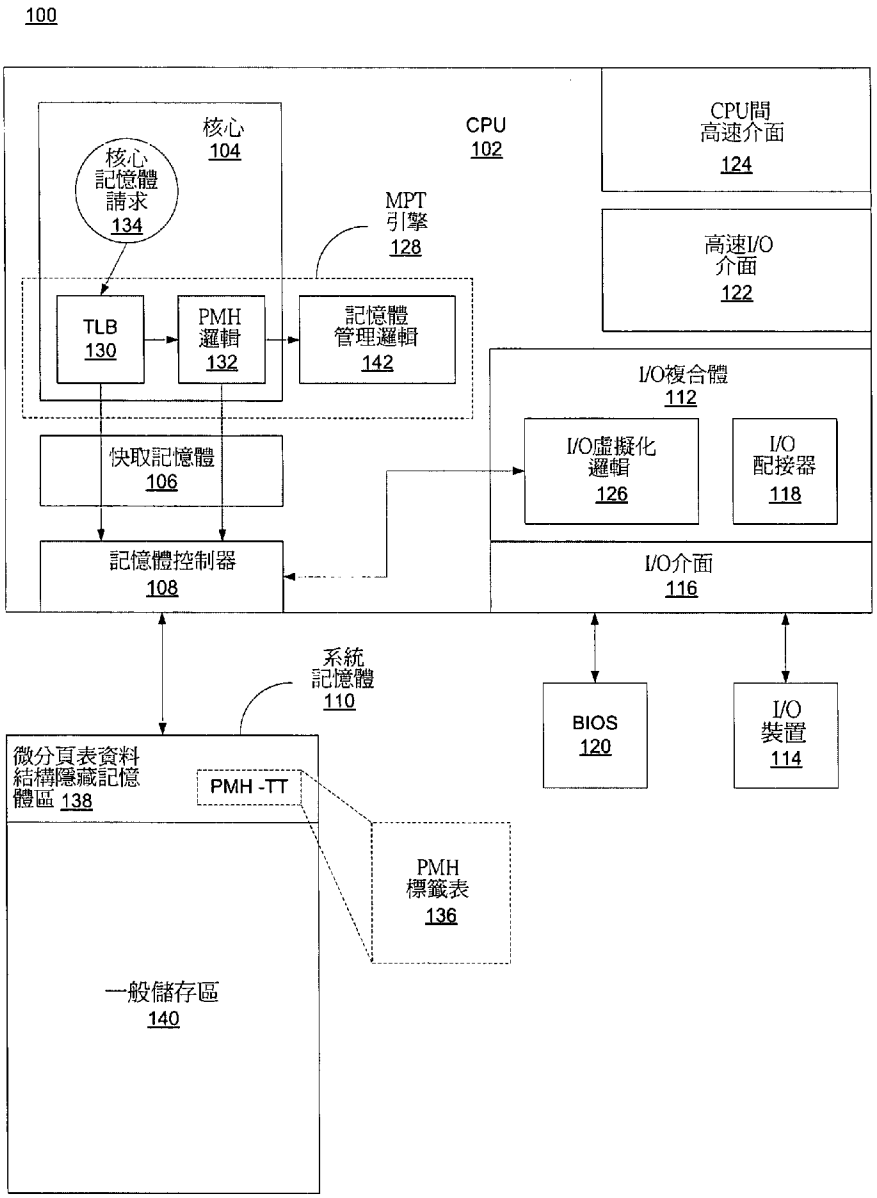
APPARATUS, METHOD, AND SYSTEM FOR IMPLEMENTING MICRO PAGE TABLES

(57)摘要

揭示裝置、方法、機器可讀取媒體、及系統。於一個實施例中，該裝置乃微分頁表引擎，其包括針對於總體記憶體位址空間內之分頁可接收一記憶體分頁請求之邏輯。該裝置也包括轉譯後備緩衝器(TLB)可儲存記憶體分頁位址轉譯。此外，該裝置也具有回應於該 TLB 不儲存針對由該記憶體分頁請求所指稱的該記憶體分頁之記憶體分頁位址轉譯而執行在分頁失誤處理器標籤表內之微實體位址詢查之分頁失誤處理器。該裝置也包括可管理該分頁失誤處理器標籤表分錄之記憶體管理邏輯。該微分頁表引擎允許該 TLB 成為確定於二層級記憶體階層組織內之資料是否在記憶體之熱區域或在記憶體之冷區域之代理器。當資料係在記憶體之冷區域時，該微分頁表引擎將該資料提取至該熱記憶體及然後熱記憶體區塊被推出至該冷記憶體區。

An apparatus, method, machine-readable medium, and system are disclosed. In one embodiment the apparatus is a micro-page table engine that includes logic that is capable of receiving a memory page request for a page in global memory address space. The apparatus also includes a translation lookaside buffer (TLB) that is capable of storing one or more memory page address translations. Additionally, the apparatus also has a page miss handler capable of performing a micro physical address lookup in a page miss handler tag table in response to the TLB not storing the memory page address translation for the page of memory

referenced by the memory page request. The apparatus also includes memory management logic that is capable of managing the page miss handler tag table entries. The micro-page table engine allows the TLB to be an agent that determines whether data in a two-level memory hierarchy is in a hot region of memory or in a cold region of memory. When data is in the cold region of memory, the micro-page table engine fetches the data to the hot memory and a hot memory block is then pushed out to the cold memory area.



第 1 圖

100・・・電腦系統

102・・・中央處理單元(CPU)

104・・・核心

106・・・快取記憶體

108・・・記憶體控制器

110・・・系統記憶體

112・・・輸入/輸出(I/O)複合體

114・・・I/O 裝置

116・・・I/O 介面

118・・・I/O 配接器

120・・・基本輸出入系統(BIOS)

122・・・高速 I/O 介面

124・・・CPU 間高速介面

126・・・I/O 虛擬化邏輯

128・・・微分頁表(MPT)引擎

130・・・轉譯後備緩衝器(TLB)

132・・・分頁失誤處理器(PMH)邏輯

134・・・核心記憶體請求

136・・・PMH 標籤表

138・・・隱藏記憶體區

140・・・一般儲存區

142・・・記憶體管理邏輯(MML)

I461911

公告本

發明專利說明書

（本說明書格式、順序，請勿任意更動，※記號部分請勿填寫）

※申請案號：100134752

※申請日：100.9.23　※IPC分類：G06F12/02 (2006.01)

一、發明名稱：(中文/英文)

用於實施微分頁表之裝置、方法與系統/APPARATUS, METHOD, AND SYSTEM FOR IMPLEMENTING MICRO PAGE TABLES

二、中文發明摘要：

揭示裝置、方法、機器可讀取媒體、及系統。於一個實施例中，該裝置乃微分頁表引擎，其包括針對於總體記憶體位址空間內之分頁可接收一記憶體分頁請求之邏輯。該裝置也包括轉譯後備緩衝器(TLB)可儲存記憶體分頁位址轉譯。此外，該裝置也具有回應於該TLB不儲存針對由該記憶體分頁請求所指稱的該記憶體分頁之記憶體分頁位址轉譯而執行在分頁失誤處理器標籤表內之微實體位址詢查之分頁失誤處理器。該裝置也包括可管理該分頁失誤處理器標籤表分錄之記憶體管理邏輯。該微分頁表引擎允許該TLB成為確定於二層級記憶體階層組織內之資料是否在記憶體之熱區域或在記憶體之冷區域之代理器。當資料係在記憶體之冷區域時，該微分頁表引擎將該資料提取至該熱記憶體及然後熱記憶體區塊被推出至該冷記憶體區。

三、英文發明摘要：

An apparatus, method, machine-readable medium, and system are disclosed. In one embodiment the apparatus is a micro-page table engine that includes logic that is capable of receiving a memory page request for a page in global memory address space. The apparatus also includes a translation lookaside buffer (TLB) that is capable of storing one or more memory page address translations. Additionally, the apparatus also has a page miss handler capable of performing a micro physical address lookup in a page miss handler tag table in response to the TLB not storing the memory page address translation for the page of memory referenced by the memory page request. The apparatus also includes memory management logic that is capable of managing the page miss handler tag table entries. The micro-page table engine allows the TLB to be an agent that determines whether data in a two-level memory hierarchy is in a hot region of memory or in a cold region of memory. When data is in the cold region of memory, the micro-page table engine fetches the data to the hot memory and a hot memory block is then pushed out to the cold memory area.

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...電腦系統
102...中央處理單元(CPU)
104...核心
106...快取記憶體
108...記憶體控制器
110...系統記憶體
112...輸入/輸出(I/O)複合體
114...I/O裝置
116...I/O介面
118...I/O配接器
120...基本輸出入系統(BIOS)
122...高速I/O介面
124...CPU間高速介面
126...I/O虛擬化邏輯
128...微分頁表(MPT)引擎
130...轉譯後備緩衝器(TLB)
132...分頁失誤處理器(PMH)邏輯
134...核心記憶體請求
136...PMH標籤表
138...隱藏記憶體區
140...一般儲存區
142...記憶體管理邏輯(MML)

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

六、發明說明：

【發明所屬之技術領域】

發明領域

本發明係有關於在電腦系統體現之記憶體分頁表。

【先前技術】

發明背景

近代電腦系統結合複雜的記憶體管理體系來處理系統記憶體在系統的多個組件間之共享。電腦系統可包括數個多核心處理器，此處各個核心(亦即各個硬體執行緒)需要存取記憶體。舉例言之，在系統上跑的作業系統以及潛在地虛擬機器監視器二者皆包括輔助管理系統記憶體在全部硬體執行緒間共享之邏輯。此種記憶體管理經常並不考慮系統中記憶體實際上如何布局的實體約束。舉例言之，可有記憶體節電能力其允許記憶體排組電力下降至低功率態來節省平台電力。於另一實施例中，系統中可有多個記憶體實體型別(亦即非同質記憶體系統而不是同質記憶體系統)。電腦系統之記憶體次系統之各樣實體體現無法透過此處討論之手段從目前可利用之標準記憶體管理獲得如此高的效果。

【發明內容】

依據本發明之一實施例，係特地提出一種微分頁表引擎裝置，其係包含：用以針對於總體記憶體位址空間內之一分頁接收一記憶體分頁請求之邏輯；一轉譯後備緩衝器(TLB)，用以儲存一記憶體分頁位址轉譯；一分頁失誤處理

器邏輯，用以回應於該TLB不儲存針對由該記憶體分頁請求所參照的該記憶體分頁之記憶體分頁位址轉譯而執行在一分頁失誤處理器標籤表內之一微實體位址詢查；及一記憶體管理邏輯，用以管理在該分頁失誤處理器標籤表內的分錄。

圖式簡單說明

本發明係藉附圖例示說明舉例說明而非限制性，附圖中類似的元件符號係指相似的元件，及附圖中：

第1圖描述體現微分頁表之電腦系統之一個實施例。

第2圖描述體現微分頁表之電腦系統之又一個實施例。

第3圖描述體現微分頁表之電腦系統之另一個實施例。

第4圖例示說明分頁失誤處理器標籤表之一實施例。

第5圖例示說明針對階級分開，體現微分頁表之電腦系統之一個實施例。

第6圖例示說明至少部分利用在階級分開，體現微分頁表之電腦系統之一個實施例。

第7圖例示說明當至少部分體現在階級分開時，分頁失誤處理器標籤表之一實施例。

第8圖為用以處理熱分頁失誤之一實施例之流程圖。

第9圖例示說明當回應於熱分頁失誤時，由微分頁表引擎所利用的若干額外微分頁表資料結構之實施例。

第10圖為維護程序之實施例之流程圖，該維護程序係用來對多個記憶體分頁提供以冷至熱記憶體分頁資料移轉期間利用作為熱分頁的能力。

第11圖例示說明於維護程序期間，由微分頁表引擎所利用的若干額外微分頁表資料結構之實施例。

第12A至12D圖例示說明微分頁表引擎處理邏輯可利用來確定何時回復記憶體分頁供使用之流程圖之若干實施例。

第13圖描述在電腦系統內部管理二層級記憶體次系統之微分頁表實施例。

第14圖描述相變記憶體特定記憶體次系統之實施例。

【實施方式】

較佳實施例之詳細說明

描述體現微分頁表之裝置、方法、系統、及機器可讀取媒體。

透過使用微分頁表，該微分頁表將記憶體之軟體觀點對映至記憶體之實體體現，電腦系統可以有效方式體現額外硬體及韌體邏輯來管理記憶體。微分頁表體現的架構可包含於處理器核心及非核心(uncore)之某些邏輯來管理隱藏資料結構之額外集合。此等隱藏資料結構對在電腦上跑的上方作業系統及應用程式為透明。

過去當操作CPU接收到來自作業系統的記憶體請求時，該記憶體請求包含線性記憶體位址。此一線性記憶體位址並非該所請求記憶體位置的實際實體位址，反而是由電腦系統中的作業系統所利用的位址。為了獲得實際實體位址，CPU內部的邏輯取專該線性位址，執行行進通過該標準記憶體分頁表來找到該實體分頁。於許多實施例中，體現微分頁表的處理器要求行進中的額外步驟。通常在分

頁行進詢查程序結束時的實體位址(平台實體位址(PPA))實際上是從真正實體位址移開的一個層級，今日可稱作微實體位址(MPA)。

在CPU體現之微分頁表引擎邏輯利用具有PPA作為索引的分頁失誤處理器標籤表來找出MPA位址。藉由增加迂回程度達一個額外層級，微分頁表引擎內部的邏輯可執行完全不為系統的任何其它硬體或軟體所知曉的實體記憶體之大量記憶體管理。接著將做下列之深度描述：於若干不同通用用途電腦系統的微分頁表之構思布局，及微分頁表之若干不同體現，及其如何利用來對電腦記憶體管理提供額外效果。

製作微分頁表允許在記憶體的近部分(例如高效能/高耗電量記憶體)及遠部分(例如低效能/低耗電量記憶體)間分裂記憶體的潛在透明方式。此一層額外記憶體管理可允許記憶體次系統成本、記憶體耗電量、及記憶體效能之最佳化。

微分頁表一般體現

第1圖描述體現微分頁表之電腦系統之一個實施例。

顯示電腦系統100。該電腦系統可以是桌上型、伺服器、工作站、膝上型、掌上型、電視機上盒、媒體中心、遊戲機台、整合式系統(諸如於汽車)、或其它型別之電腦系統。於若干實施例中，電腦系統100包括一或多個中央處理單元(CPU)。雖然於許多實施例中可能有多個CPU，但於第1圖所示實施例中，只顯示CPU 102以求清晰。CPU 102可

以是英特爾公司(Intel Corporation) CPU或其它品牌CPU。於不同實施例中，CPU 102包括一或多個核心。再度為求簡明，CPU 102顯示包括單一核心(核心104)。

於許多實施例中，核心104包括內部功能區塊，諸如一或多個執行單元、報廢單元、通用及特用暫存器集合等。若核心104為多執行緒或超執行緒，則各個硬體執行緒也可視為「邏輯」核心。

CPU 102也可包括一或多個快取記憶體，諸如快取記憶體106。於圖中未顯示之許多實施例中，快取記憶體106以外的額外快取記憶體係體現為使得在核心的執行單元與記憶體間存在有多階快取記憶體。於不同實施例中，快取記憶體106可以不同方式分配。此外，於不同實施例中，快取記憶體106可以是多個不同尺寸中之一者。舉例言之，快取記憶體106可以是8百萬位元組(MB)快取記憶體、16 MB快取記憶體等。此外，於不同實施例中，快取記憶體可以是直接對映快取記憶體、完全聯結快取記憶體、多路集合聯結快取記憶體、或有另一型對映之快取記憶體。於包括多個核心之其它實施例中，快取記憶體106可包括由全部核心共享的一個大型部分，或可劃分成數個分開功能截割片(例如各個核心有一個截割片)。快取記憶體106也可包括由全部核心共享的一個部分及每個核心有數個功能截割片之若干其它部分。

於許多實施例中，CPU 102包括整合式系統記憶體控制器108來提供與系統記憶體110溝通之介面。於圖中未顯示

之其它實施例中，記憶體控制器108可位在電腦系統100它處的分開晶片。

系統記憶體110可包括動態隨機存取記憶體(DRAM)諸如一型雙倍資料率(DDR) DRAM、非依電性記憶體諸如快閃記憶體、相變記憶體(PCM)、或其它型別之記憶體技術。系統記憶體110可以是通用記憶體來儲存欲藉CPU 102操作的資料及指令。此外，可有在電腦系統100內部的其它潛在裝置，其有能力讀及寫至系統記憶體，諸如具直接記憶體存取(DMA)能力之I/O(輸入/輸出)裝置。

耦聯CPU 102與系統記憶體110之鏈路(亦即匯流排、互連體等)可包括能夠傳輸資料、位址、控制、及時鐘資訊之一或多個光學、金屬、或其它導線(亦即線路)。

I/O複合體112允許CPU 102與一或多個I/O裝置諸如I/O裝置114間之溝通。於第1圖所示實施例中，I/O裝置114係透過I/O介面116而通訊式耦接I/O複合體112及CPU 102其餘部分。I/O複合體112可以是包含數個I/O主機配接器及其它I/O電路之I/O中樞器介面來提供CPU 102與許多I/O次系統間的存取。舉例言之，I/O複合體112可包含平台控制器中樞器(PCH)。更明確言之，I/O複合體112可提供耦接至一或多個I/O互連體(亦即I/O匯流排)的多個I/O裝置與該CPU 102間之一般通訊介面。為了達成此項目的，針對各個所利用的I/O協定，I/O中樞器複合體可具有至少一個整合式I/O配接器。可有多個I/O裝置通訊式耦接至I/O介面116，但為求清晰只顯示I/O裝置114。

第1圖中顯示為I/O複合體112內部的整合式I/O配接器的I/O配接器118將利用在CPU 102內部的主機通訊協定轉譯成與特定I/O裝置諸如I/O裝置118可相容的協定。給定I/O配接器可轉譯的若干協定包括周邊組件互連體(PCI)-快速、通用串列匯流排(USB)、串列進階技術附接(SATA)、小型電腦系統介面(SCSI)、廉價碟片冗餘陣列(RAID)、及1394「火線(Firewire)」等。此外，可有一或多個無線協定I/O配接器。無線協定之實例有藍牙、基於IEEE 802.11之無線協定、及小區式協定等。

於許多實施例中，BIOS 120(基本輸出入系統)係耦接至I/O複合體112。BIOS係韌體儲存於電腦系統，BIOS含有指令來於軟體啟動程序期間初始化關鍵電腦系統組件。BIOS 120通常係儲存在快閃記憶體裝置內部，但設計來以非依電性方式儲存資訊的其它此等儲存裝置可也利用來儲存BIOS。此外，雖然未顯示於第1圖，超出BIOS的其它韌體也可儲存在耦接至CPU 102的快閃裝置諸如擴充式韌體。

除了I/O介面116之外，可有整合於CPU 102的其它介面來提供通訊介接CPU 102外部的一或多個鏈路。高速I/O介面122可通訊式耦接CPU 102至到高速I/O次系統諸如圖形次系統及/或網路次系統之一或多個鏈路。舉例言之，高速I/O介面可以是單通道或多通道高速雙向串列介面，諸如PCI-快速。CPU間高速介面124可提供耦接至一或多個額外CPU之鏈路介面且允許進行CPU間通訊。例如，CPU間高速介面可以是快速路徑互連體(QPI)或其它類似介面。

於許多實施例中，電腦系統100包括可提供虛擬化環境之硬體及軟體邏輯，具有在虛擬機器(VM)環境中跑的一或多個客端作業系統(OS)。虛擬機器監視器(VMM)或超監督器可在系統內部的邏輯體現來隔離各個VM的作業環境(亦即使得在其內部跑的各個VM及OS及應用程式係與存在於系統的其它VM隔離且不知曉存在於系統的其它VM)。

製作無接縫式虛擬化環境所要求的區域中之一者為虛擬化I/O。I/O虛擬化邏輯126提供虛擬化及隔離I/O次系統內的I/O裝置諸如I/O裝置114之能力。於若干實施例中，I/O虛擬化邏輯包括英特爾VT-d架構。

由I/O裝置所啟動之裝置移轉(直接記憶體存取(DMA))及中斷乃要求裝置與給定VM隔離的關鍵處理程序。於許多實施例中，I/O虛擬化邏輯126可允許系統軟體製作多個DMA保護域。保護域是配置系統記憶體子集的隔離環境。取決於軟體使用模型，DMA保護域可表示配置給VM之記憶體，或由在VM裡跑的客端OS驅動程式所配置的或作為VMM或超監督器本身一部分的DMA記憶體。I/O虛擬化邏輯126使得系統軟體指定一或多個I/O裝置給保護域。藉由約束從非指定給該保護域的I/O裝置存取保護域的實體記憶體，達成DMA隔離。

用於中斷處理，I/O虛擬化邏輯126可修改中斷訊息格式成為DMA寫入請求，該請求包括「訊息識別符」而非實際中斷屬性。類似任何DMA請求，寫入請求可載明產生該中斷之I/O裝置功能之請求器ID。然後，當該中斷請求係由

I/O虛擬化邏輯126接收時，透過中斷表結構而重新對映中斷。在中斷重新對映表中的各個分錄係相對應於得自裝置之獨特中斷訊息識別符，包括任一種需要的中斷屬性(例如目的地CPU、向量等)。

於第1圖所示實施例中，I/O虛擬化邏輯126透過I/O介面116接收來自一或多個I/O裝置之請求。I/O虛擬化邏輯126在允許此等請求發送通過至記憶體控制器108之前，如前文說明處理此等請求。

於許多實施例中，微分頁表(MPT)引擎128邏輯係在核心104體現而使用可能的隱藏(對OS及應用程式隱藏)分頁表結構而提供硬體管理的記憶體位址空間。MPT引擎128卷擬化記憶體位址空間如由在電腦系統100上跑的OS及應用程式軟體所見。更明確言之，包括跑一或多個應用程式/處理程序的作業系統的軟體假設其可要求直接存取系統記憶體110的實體位址，但MPT引擎128提供隱藏迂回層級使得實體記憶體可與作業系統中的核心所知曉的實體記憶體布局分開地管理。此種硬體體現的可能隱藏之針對記憶體迂回層級管理全部記憶體相干性及系統記憶體110各區間的資料移動。

MPT引擎128包括各個核心諸如核心104內部之經修改的轉譯後備緩衝器(TLB) 130及分頁失誤處理器(PMH)邏輯132。

於許多實施例中，TLB 130通常考慮CPU快取記憶體，記憶體管理硬體使用該CPU快取記憶體來改良線性位址轉

譯速度。TLB 130包括固定數目的插槽，該等插槽含有分頁表分錄，其通常將線性位址對映至平台實體位址(PPA)。又復，TLB 130乃內容可定址記憶體(CAM)，其中搜尋鍵為線性位址，而搜尋結果為平台實體位址。若從核心記憶體請求134所請求的線性位址係存在於TLB，則CAM搜尋獲得匹配，其為TLB命中。否則，若所請求的位址不在TLB，則CAM搜尋得到TLB失誤。

若有TLB命中，則已經進行線性位址→平台實體位址轉譯，轉譯係儲存在TLB 130。若有TLB失誤，據此，轉譯不儲存，然後要求MPT引擎128邏輯使用OS建立的分頁表執行分頁行進來取回平台實體位址。一旦分頁行進已經完成，找到正確平台實體位址，然後轉譯可儲存入TLB。

但於電腦系統100中，平台實體位址並非用來存取組成系統記憶體110的記憶體裝置之實際實體記憶體位址。反而一旦MPT引擎128已經接收轉譯之平台實體位址，然後該平台實體位址用作為進入儲存在隱藏記憶體區138的PMH標籤表(TT) 136的索引(亦即搜尋鍵)來將真正位址取回入實體記憶體。於本文件全文可利用的其它實施例中，記憶體區138及可能的記憶體區儲存的部分或全部結構並非隱藏，反而為作業系統及軟體應用程式所可見。

更明確言之，PMH TT 136具有多個分錄。各個分錄儲存微實體位址(MPA)，該位址係直接對應至實體記憶體空間。如此，需要至少兩次位址轉譯來從線性位址獲得微實體位址(MPA)。當使用MPT引擎128時，額外表詢查為用來

將平台實體位址轉譯成微實體位址(MPA)的表詢查。換言之，於電腦系統100的基本位址轉譯係以下述順序處理：線性位址→平台實體位址→微實體位址(MPA)。平台實體位址至微實體位址轉譯(使用PMH TT 136)乃全然聯結的對映關係。如此，在PMH TT 136的任何分錄可儲存任何微實體位址(MPA)。一旦微分頁表(MPT)引擎已經從PMH-TT取回微實體位址(MPA)，該引擎將MPA儲存在TLB供隨後存取，於該處事直接完成從線性位址至MPA的轉譯。於其它實施例中，PMH TT 136可以更受約束的集合聯結性體現。

於許多實施例中，儲存全部需要的MPT資料結構諸如PMH TT 136之隱藏記憶體區138為軟體(例如作業系統(OS))所不可見，特別由MPT引擎128利用，經由額外位址迂回層(亦即使用在隱藏區138中的表之額外轉譯)來管理實體記憶體。

於下述實施例中，平台係在虛擬化環境運轉及記憶體位址請求係接收自客端OS(或更通常客端VM)，有第二額外位址轉譯係使用指稱為擴充分頁表(EPT)的另一個位址轉譯表。更明確言之，接收自客端OS的線性位址使用OS分頁表，透過標準分頁行進而首先轉譯成客端實體位址(GPA)，然後客端實體位址藉使用EPT而轉譯成平台實體位址(PPA)，及最後，平台實體位址係藉使用PMH TT 136而轉譯成相對應的微實體位址(MPA)。

記憶體管理邏輯(MML) 142駐在CPU 102。於若干實施例中，MML 142係在核心104、核心104外側(例如非核心(uncore))體現，或可能甚至跨核心及非核心二者體現。非

核心通常係指CPU內部非實際上在核心的邏輯/電路。舉例言之，許可一給定CPU與其它CPU的核心間通訊之某些I/O電路可位在該CPU的非核心。MML 142為協助管理MPT資料結構的MPT引擎128之一部分，諸如PMH TT 136。於許多實施例中，容後詳述，記憶體分頁將以某些方式標示，及儲存在該等分頁上的資料可能需交換出至其它分頁。此等記憶體分頁交換移轉的管理大部分可藉MML 142處理。於第1圖所示實施例中，MML 142包含非核心的硬體電路、儲存於非核心的微代碼、或二者的組合。於圖中未顯示之其它實施例中，組成MML 142的電路及/或微代碼駐在該核心。又有其它實施例中，MML 142可跨據核心及非核心。

又復，有PMH TT 136以及儲存在隱藏記憶體區138的額外資料結構可普遍性地指稱為由MPT引擎128管理的「微分頁表」。

第1圖描述具有含單一核心的單一CPU之電腦系統方便解說，但另一個較為複雜的實施例係例示說明於如下第3圖。回頭參考第1圖，CPU 102通常係耦接至系統板(亦即主機板)。主機板雖然未顯示於第1圖，但可包括插槽，該插槽係設計來確保源自於CPU 102的各個外部電力及通訊接腳與電腦系統中其它組件間之接觸。此一插槽大致上係將整合成CPU 102的全部電路通訊式耦接組件，諸如系統記憶體(例如一般儲存區140)、I/O複合體112、及於圖中未顯示之其它額外組件。於許多情況下，系統資源諸如記憶體的配置可基於遵照插槽布局。如此，除非另行註明，否則述

及系統組織結構時，CPU與插槽係可互換使用。

體現微分頁表之電腦系統其它實施例係顯示於第2及3圖。

第2圖描述體現微分頁表之電腦系統之另一實施例。

第2圖例示說明之電腦系統係顯示第1圖所示電腦系統，但I/O複合體及整合於I/O複合體的電路乃與CPU 102分開的外部I/O複合體200。於許多實施例中，I/O虛擬化邏輯202及I/O配接器204二者皆係整合入分開的I/O複合體200。此等組件之功能可與如上於第1圖中就I/O複合體112、I/O配接器118及I/O虛擬化邏輯126所述者相同，唯有此等功能組件係位在電腦系統內部不同位置。於又其它實施例中，未顯示於第1及2圖，I/O複合體200可部分體現在CPU 102晶粒上及部分體現在CPU 102外部。

第3圖描述體現微分頁表之電腦系統之另一實施例。

第1及2圖例示說明之電腦系統限於顯示含單一核心之單一CPU。如所述，如此僅是為了例示說明目的。於許多實施例中，體現微分頁表的電腦系統可以是有許多核心及許多CPU的電腦。舉例言之，第3圖例示說明有四個CPU(A0、A1、B0、及B1)之電腦系統。CPU A0及A1駐在節點A內部，CPU B0及B1駐在節點B內部。四個CPU皆係透過高速CPU間介面(I/F)而彼此溝通。在節點間，高速介面係路由通過針對各個節點之節點控制器(節點控制器A及B)。

於第3圖所示實施例中，各個CPU包括四個分開的核心(核心A0a、A0b、A0c、及A0d係在CPU A0；核心A1a、A1b、

A1c、及A1d係在CPU A1；核心B0a、B0b、B0c、及B0d係在CPU B0；及核心B1a、B1b、B1c、及B1d係在CPU B1)。至少部分針對微分頁表引擎邏輯係駐在16個核心各自的內部。

於許多實施例中，有每個CPU必須存取的單一通用系統記憶體300。雖然於圖中未顯示，但可有隱藏記憶體區(第1圖之138)在系統記憶體300內部。又復，為求圖式簡明於第3圖並未顯示在各個CPU內部的額外組件(例如快取記憶體)。

第4圖例示說明分頁失誤處理器標籤表之一實施例。於許多實施例中，PMH-TT 400針對全部實體記憶體404的各個分頁儲存一個分錄402。如第4圖所示，實體記憶體404多次係由多個階級(諸如階級0-7)組成。各個PMH-TT分錄(例如402)包括MPA其參考在實體記憶體404多個階級中之一者的特定分頁。於若干實施例中，分錄402也包括狀態資訊來將記憶體的目前狀態儲存於微實體位址(MPA)。舉例言之，第4圖顯示分錄402之若干細節實例。於第4圖中，狀態包含三個位元，而此等位元的組合可顯示於不同狀態的分錄，諸如污穢、犧牲、回收、閂鎖、提取、零等所列舉的狀態。其中許多狀態實例將如下就不同聚焦MPT實施例詳加解說。於其它實施例中，狀態可能並非單純編碼，此處3位元可象徵8態，反而針對各個態可有一或多個分開位元(但未顯示此一實施例)。

回頭參考第4圖所示分頁詢查例示說明，位址請求到達MPT引擎(第1圖之128)邏輯。舉例言之，線性位址可以是來

自OS的位址請求。若在平台上無虛擬化，則分頁行進將產生平台實體位址(PPA)，討論如前。PPA可用作為索引檢索進入PMH-TT 400來取回含有微實體位址(MPA)的相關分頁，該MPA係直接指稱實體記憶體內的位置(例如實體記憶體分頁位址)。

另一方面，若在平台上有個虛擬化層級，該虛擬化層級可包括一或多個虛擬機器及虛擬機器管理器(VMM)，則額外採取中介分頁行進通過VMM維持的分頁表集合。更明確言之，於本實施例中，分頁行進通過OS維持的分頁表406，係指行進通過分頁表，而該等分頁表為VM已知供給線性位址的OS在其上跑。於此種情況下，從線性位址分頁行進所產生的位址稱作為客端OS實體位址(GPA)。GPA並非直接檢索進入PMH-TT，原因在於在客端OS下方有個VMM層，該VMM層管理本身的分頁表集合，該分頁表集合為在可能存在於平台的許多虛擬機器中之一者上跑的客端OS所不知曉。一般而言，VMM分頁表係指稱擴充分頁表(EPT)，然後GPA將用作為分頁行進通過EPT的索引來產生PPA而檢索至PMH-TT 400。當處理虛擬化平台時此一額外分頁行進步驟通常為標準步驟。

任一種情況下，一旦已經產生PPA，則PPA係利用作為索引檢索至PMH-TT 400來找出含有直接參考實體記憶體404的MPA之該分錄。

為了使得記憶體以前述方式針對全部軟體為虛擬化，記憶體及支援MPT引擎的資料結構須經啟動。於許多實施

例中，於電腦系統的軟體啟動期間，BIOS對電腦提供以指令集來初始化存在於系統的許多組件。

許多電腦系統中，BIOS的主要構面乃記憶體參考代碼(MRC)。MRC係有關於記憶體初始化，及包括有關記憶體設定值、時間、驅動、及記憶體控制器之細節操作的資訊。為了支援MPT，電腦系統中的BIOS可經更新而公開CPU-插槽特定記憶體位址區域。如此可包括公開存在於系統的各個記憶體階級之實體位址範圍，以及初步隱藏部分系統記憶體(第1圖之隱藏記憶體區130)為全部軟體所不可見來體現PMH-TT 400及其它MPT資料結構，容後詳述。特別係針對PMH-TT 400，識別身分對映關係可用於初始化目的。

於許多實施例中，PPA至MPA的對映為完全聯結的對映。如此，給定PPA可索引檢索至整個PMH-TT 400內部的恰一個位置，但於該位置的分錄可對映至記憶體內的任一個任意MPA位置。

此種額外迂回層級係隱藏在硬體，因此在平台上操作的客端OS、VM、VMM、超監督器、或潛在任何其它軟體組成可能完全不知曉該額外轉譯層。換言之，從軟體觀點，相信PPA乃檢索記憶體分頁的實際實體位址。取而代之，當體現PMH-TT 400時，MPA為記憶體內部的實際實體位址。額外迂回層可允許許多項應用用途，否則當限於軟體解決辦法時，該等應用用途為顯著較無效率或可能無法達成。

微分頁表階級

第5圖例示說明針對階級分開，體現微分頁表之電腦系

統之一個實施例。

階級分開涉及允許記憶體階級以使用性排優先順位。為何可體現階級優先排序有多個原因。舉例言之，於許多實施例中，從電力觀點，同時利用電腦系統的全部系統記憶體階級並非經常性地有效。舉例言之，於伺服器中，有許多記憶體階級可資利用，但伺服器不會同時使用或至少不會高度使用。於此種情況下，若使用系統中記憶體階級之一子集，則存在伺服器上的工作負荷可未顯示效能降級。如此，可達成排某些階級的使用性優先順位超過其它階級，而於較低使用性期間，存在於記憶體次系統的某些低優先順位排序則可從積極使用脫離。排序從積極使用脫離允許記憶體次系統電力管理方案將不使用的階級置於較低功率態歷經長時間，藉此降低整個電腦系統之耗電量。

於其它實施例中，存在有交插記憶體存取架構而非NUMA架構。如此，對記憶體組態並無約束，組態將依體現而異。舉例言之，於若干實施例中，可有多個CPU共享單一DIMM。於其它實施例中，針對各個CPU可使用多個DIMM。於又其它實施例中，DIMM數目與CPU數目間可有一對一相關性。

轉向參考第5圖，顯示體現階級分開之多CPU插槽電腦系統之一個實施例。

CPU A 500橫過點對點鏈路504而耦接至CPU B 502。CPU A 500也係直接地耦接至記憶體A 504及間接地耦接至記憶體B 508(透過鏈路504及CPU B 502)。CPU B 502係直

接地耦接至記憶體B 508及間接地耦接至記憶體A 504(透過鏈路504及CPU A 500)。當記憶體用於工作負荷時，若CPU A 500將利用記憶體A 506及若CPU B 502將利用記憶體B 508，則通常更為有效。此點係由於在NUMA環境中每個CPU的記憶體所在位置。如此於許多實施例中，CPU A可優先使用位在記憶體A 506的記憶體階級，而CPU B可優先使用位在記憶體B 508的記憶體階級。

於許多實施例中，階級分開有助於體現遵照每個CPU之階級優先排序(如此也指稱「每個插槽」，原因在於各個CPU係藉其本身的插槽而耦接至系統板)。遵照每個插槽的階級優先排序係藉利用在各個CPU的MPT引擎來動態地優先排序由該CPU所使用的階級順序而達成。MPA位址空間跨據全部插槽。如此，全部插槽所看到的位址空間範圍為相同，但針對每個插槽可優先排序位址空間的不同部分。

舉例言之，於第5圖所示二插槽系統中，CPU A具有顯示在其下方的CPU A/插槽1 MPA位址空間。如此顯示從位址0至位址2十億位元組(GB)-1，亦即插槽1的下NUMA區域，係在插槽1的熱區域。CPU B具有CPU B/插槽2 MPA位址空間，其顯示位址2GB至位址4GB-1，亦即插槽2的下NUMA區域，係在插槽2的熱區域。

更明確言之，記憶體的首4GB為下NUMA區域。頭兩個十億位元組係對映至插槽1，及次兩個十億位元組係對映至插槽2。於第5圖所示實施例中，有高於4GB的額外位址空間。為了位址空間的均勻分裂，上NUMA區域可在全部

插槽間均等劃分。又復，上NUMA區域將等於記憶體頂部(ToM)減下NUMA區域頂部(例如4GB)。於第5圖所示實施例中，共有16GB位址空間，而下NUMA區域為4GB。如此，頂12GB(16GB-4GB)係在上NUMA區域。然後12GB一分為二來在兩個插槽間分配上NUMA區域，使得各個插槽有6GB可定址上NUMA區域位址空間。因此，插槽1可配置從4GB至10GB-1的位址空間，及插槽2可配置從10GB至記憶體頂部於本實例為16GB-1的位址空間。

基於使用狀況，熱及冷區域的大小可隨時間而異。於第5圖所示實施例中，在做記憶體的複製時，熱區域包含總位址空間的一半而冷區域包含另一半。如此，插槽1的上NUMA區域具有位址4GB至6GB-1在熱區域及6GB至10GB-1在冷區域，及插槽2具有位址10GB至12GB-1在熱區域及12GB至16GB-1在冷區域。但須瞭解在不同實施例中可改變各個個別NUMA區域之大小、記憶體大小、及插槽數目。

此外，因記憶體的熱及冷區域可能可做動態調整，故各個插槽的熱及冷區域大小也可變。雖然針對各個插槽有熱及冷區域，但每個插槽大小的變異性可以是對稱性或非對稱性。舉例言之，於某些情況下，針對各個插槽的熱及冷區域大小橫過各個插槽經常性為相同。如此，若熱區域跨據可定址記憶體的25%至50%，則此項改變可針對全部插槽同時進行(橫過各插槽之對稱性處理)。另一方面，於其它情況下，針對各個插槽的熱及冷區域大小係分開地維持(橫過各插槽之非對稱性處理)。舉例言之，若插槽1具有重載

工作負荷而插槽2具有輕載工作負荷，則比較插槽2的可定址記憶體空間，插槽1的熱區域將跨據更高百分比之插槽1的可定址記憶體空間。

一般而言，PMH-TT的身分對映關係係儲存於記憶體位址空間之熱區域，及MPT資料結構本身包括PMH-TT也係儲存於記憶體位址空間之熱區域。於許多實施例中，資料結構跨據兩個插槽各自的熱區域，如第5圖所示。

第6圖例示說明至少部分利用在階級分開，體現微分頁表之電腦系統之一個實施例。第6圖之電腦系統係類似第1圖例示說明之電腦系統。全部主要組件係以相似方式使用，且可述及於前文描述。第6圖增加階級分開體現於系統記憶體110的特定區劃。顯示作用態記憶體600區劃及非作用態記憶體602區劃。作用態記憶體600區劃包括目前由CPU 102所利用的該等記憶體階級(亦即位址空間的熱區域)。而非作用態記憶體602區劃包括目前非由CPU 102所利用的該等記憶體階級(亦即位址空間的冷區域)。

當較多階級被使用時，作用態記憶體600區劃將增加，而當使用較少階級時，非作用態記憶體602將增加。作用態及非作用態記憶體部分間之大小潛在改變的粒度乃體現之特定粒度。舉例言之，若階級分裂係用在電力管理，則作用態對非作用態階級的改變粒度可映射可分開管理電力的階級數目。若系統記憶體有16階級且此等階級係以4個為一組而耦接至系統板上的電力平面，則表示在任何給定時間，0、4、6、12或16階級可以是作用態，針對非作用態階級反

之亦然。另一方面，若系統記憶體內部的電源供應器有更細粒度，則可有更多個選項。若各個階級可以分開電力控制，則可有16個不同作用態對非作用態記憶體部分組合。

可也聚焦在以記憶體模組為基礎的粒度上，將允許經由單一模組(例如DIMM)而耦接至系統板的全部記憶體作為一組進行電力管理。

另一方面，作用態及非作用態階級可以每個CPU基於在以NUMA為基礎的系統中之效能而管理。舉例言之，回頭參考第5圖，於二CPU系統中，記憶體A 506係在CPU A 500本地，故表示記憶體A 506的位址空間初步可標示為CPU A 500之作用態記憶體。而記憶體B 508係在CPU A 500遠端，故表示記憶體B 508的位址空間初步可標示為CPU A 500之非作用態記憶體。但若工作負荷要求增加針對CPU A 500之記憶體使用，則記憶體B 508可具有部分標示切換成作用態記憶體。針對CPU B 502初步使用記憶體B 508作為作用態記憶體及記憶體A 506作為非作用態記憶體，此種利用本地記憶體位址空間亦為真。

現在轉向參考第7圖，本圖例示說明當至少部分體現在階級分開時，分頁失誤處理器標籤表之一實施例。

分頁行進處理從初步輸入線性位址至MPA係類似第4圖所述及例示說明之處理程序。處理程序的主要步驟係以類似方式完成，可參考前文說明。第5圖包括額外階級分裂體現細節。

顯示在第7圖右側(亦即階級0至7)的實體記憶體包含微

實體位址(MPA)空間冷區域500及MPA空間熱區域502。冷區域可視為非作用態記憶體區域及熱區域可視為作用態記憶體區域。如此，階級0及1為目前作用態，而階級2至7為目前非作用態。

702細節顯示狀態資訊所利用的一個可能態為「冷」位元。針對在PMH-TT 704的各個分錄，此一位元可指示該分錄是否檢索至MPA空間的冷區域或MPA空間的熱區域。於系統之初始化期間，PMH-TT 704中相對應於冷區域的階級之各個分錄初步可使用冷位元(例如冷位元=「1」)設定作為冷MPA(cMPA)。而其餘分錄可設定作為熱位元(例如冷位元=「0」或存在位元=「1」)。

當系統首次啟動時，對各階級可有初態，有關該階級(及其包含的全部MPA分錄)係在整體系統記憶體的冷區域或熱區域內部。操作期間當記憶體的使用樣式改變時(例如重載記憶體工作負荷、輕載記憶體工作負荷、閒置等)，MPT引擎邏輯可決定收縮或擴展記憶體702的熱區域。如此，於系統操作期間，記憶體的熱及冷區域可動態調整。如此可能係植基於效能策略(亦即當效能降級時，熱區域擴大來補償)或電力策略(系統閒置期間，更多階級加至冷區域潛在使用低功率模式至少用在記憶體次系統部分)。除了此等情況外，記憶體之分裂階級有多項其它潛在用途。

熱及冷MPA轉譯對系統是否使用虛擬化並無關聯，且要求額外GPA→PPA分頁行進步驟。本圖式係特別顯示單一實例之該方式用於例示說明目的。

於許多實施例中，只有熱分頁轉譯係儲存在TLB。如此，當請求存取係在記憶體冷區域於MPA位址之一實體分頁時，發生階級分裂失誤。因熱分頁係特別利用來由CPU或其它匯流排主機裝置做一般性存取，故儲存在所請求分頁的資料係從記憶體冷分頁移至熱分頁。於許多實施例中，記憶體熱區域經常性維持總熱空間的某個百分比作為自由記憶體分頁，來在來自冷分頁的資料需被存取時利用於資料交換之用。此一自由熱分頁百分比可從占總熱分頁的極小百分比(例如一個自由熱分頁)高達占熱分頁整個範圍的相當百分比(例如熱分頁總數的10%為自由)之範圍。

第8圖為用以處理熱分頁失誤之一實施例之流程圖及第9圖例示說明當回應於熱分頁失誤時，由微分頁表引擎所利用的若干額外微分頁表資料結構之實施例。為求清晰，熱分頁失誤只是對在記憶體冷區域的實體分頁之記憶體分頁請求。第8圖係與第9圖有關。更明確言之，第8圖例示說明之處理程序顯示藉由利用第9圖所示資料結構，MPT引擎邏輯如何處理熱分頁失誤。第8圖例示說明之處理程序可藉與MPT引擎有關的處理邏輯而執行。此一處理邏輯可包含硬體、軟體、韌體或該等三個邏輯形式之任何組合。又復，第8及9圖中有以字母標示的小圈(例如A、B、C等)。此等以字母標示的圓形為感興趣項目有關當執行第8圖的逐一方塊處理流程時，哪些資料結構及資料移轉係由處理邏輯利用。

現在轉向參考第8圖，處理程序始於處理邏輯確定所接收的記憶體請求是否靶定於非在TLB的記憶體分頁(處理方

塊800)。如前述，於許多實施例中，TLB不儲存冷分頁轉譯，如此於本實施例中，若有TLB命中，則本質上所請求的記憶體分頁已經駐在記憶體之熱區域內部。若有TLB命中，則此處理程序結束，原因在於處理邏輯不處理熱分頁失誤。利用處理方塊800，原因在於雖然處理方塊802將確定所請求的記憶體分頁是否在熱或冷區域，但處理方塊802要求額外詢查時間，若有TLB命中，則不需要此一額外詢查時間。

繼續處理(假設有TLB失誤)，其次藉由檢查與PMH-TT(第9圖之900)相聯結的狀態資訊，處理邏輯特別確定所請求的記憶體分頁是否在熱或冷區域(處理方塊802)。此項確定要求詢查在PMH-TT 900的記憶體實體分頁。詢查係以細節描述於如上第7圖。更明確言之，處理邏輯利用記憶體分頁請求之PPA來索引檢索入PMH-TT 900找到該特定分錄。

該特定分錄包括MPA及該特定記憶體分頁之狀態資訊。於若干實施例中，狀態資訊包括存在位元(P)，使用所請求的記憶體分頁之PPA，若該P位元為設定(P=1)則指示記憶體分頁係在熱區域；或若該P位元為清除(P=0)則指示記憶體分頁係在冷區域。於許多其它實施例中，額外利用冷位元(C)，若該C位元為設定(C=1)則指示記憶體分頁係在冷區域；或若該C位元為清除(C=0)則指示記憶體分頁係在熱區域。舉例言之，如第8圖之處理程序所示，處理邏輯確定是否P=0。藉由查看PMH-TT 900內部在PPA檢索位置(項目A)，處理邏輯確定此點。

若該P位元為清除(P=0)，則處理邏輯將PMH-TT 900閂鎖在PPA檢索位置(處理方塊804)。PMH-TT 900需被閂鎖來允許處理邏輯啟動冷至熱記憶體分頁移轉。若PMH-TT 900不被閂鎖在PPA檢索位置，可能遭致冷至熱變遷的訛誤，例如原因在於另一個實體同時試圖做類似的存取。於許多實施例中，藉將在PMH-TT 900中PPA檢索位置(項目B)的狀態資訊位元設定為「提取」(F=1)可達成閂鎖。

然後，處理邏輯從PMH-TT 900提取cMPA(處理方塊808)。[於許多實施例中，此一提取程序係從cMPA記憶體位置(項目C)提取資料。該資料可置於緩衝器邏輯供暫時儲存]。其次，處理邏輯將cMPA實體記憶體分頁位址載入冷自由列表(處理方塊808)，換言之，藉第9圖資料移轉D例示說明，於PMH-TT 900的cMPA位址係拷貝至冷自由列表資料結構(第9圖之902)。冷自由列表儲存實體記憶體分頁位址，該位址係在記憶體冷區域但已經成為記憶體請求標的，如此於該分頁的資料已經要求移轉至熱記憶體區域分頁。一旦冷區域分頁不再要求繼續保有該資料(原因在於該資料複本已經拷貝入緩衝器)，此時冷記憶體分頁可自由被覆寫，因而其位址係置於冷自由列表。

然後，處理邏輯從熱自由列表資料結構(第9圖之904)提取自由熱MPA(hMPA)實體記憶體分頁位址(第9圖項目E)(處理方塊810)。熱自由列表包括可供此一處理程序寫入的熱區域記憶體分頁。如下第10圖及第11圖描述熱自由列表904如何進駐。然後處理邏輯將資料從cMPA記憶體分頁

拷貝至hMPA記憶體分頁(處理方塊812及第9圖項目F)。一旦已經進行冷至熱記憶體分頁資料移轉，則處理邏輯更新且解鎖PMH-TT 900(處理方塊814及第9圖項目G)。

於許多實施例中，PMH-TT 900更新將狀態資訊設定為存在(P=1)及不提取(F=0)在hMPA位址的記憶體分頁來解鎖與更新該分頁。此外，於此處理程序之前，在hMPA位址的熱區域記憶體分頁係在熱自由列表，原因在於其可用於冷至熱移轉；但現在因該分頁已經用於移轉，故該分頁正在使用中而不再自由。如此，處理邏輯從熱自由列表904移除hMPA位址。

現在轉向參考第10圖及第11圖，第10圖為維護程序之實施例之流程圖，該維護程序係用來對多個記憶體分頁提供以冷至熱記憶體分頁資料移轉期間利用作為熱分頁的能力及第11圖例示說明於維護程序期間，由微分頁表(MPT)引擎所利用的若干額外微分頁表資料結構之實施例。第10圖係與第11圖有關。更明確言之，第10圖例示說明之處理程序顯示藉利用第11圖所示資料結構，MPT引擎邏輯如何前進通過維護方法。第10圖例示說明之處理程序可藉與MPT引擎有關的處理邏輯執行。此一處理邏輯可包含硬體、軟體、韌體或此三邏輯形式的任一項組合。又復，類似第8圖及第9圖，第10圖及第11圖中有以字母標示的小圈(例如A、B、C等)。此等以字母標示的圓形為感興趣項目有關當執行第10圖的逐一方塊處理流程時，哪些資料結構及資料移轉係由處理邏輯利用。

第10圖之處理程序始於處理邏輯配置hMPA用於熱分頁失誤上的本地描述符表(LDT)(處理方塊1000及項目A)。LDT(第11圖之1102)含有於PMH-TT(第11圖之1100)中之一分錄子集。於LDT之特定分錄子集為實際上使用的該等分錄子集(亦即「存在」或「熱」位元經設定)。一般而言，LDT 1102許可快速詢查來讓邏輯確定是否存在有分錄。沒有LDT 1102，PMH-TT需要被搜尋來確定關注的分錄是否存在，原因在於於PMH-TT中，全部記憶體位址空間皆被參照，於許多實施例中，大部分PMH-TT分錄進入空白(亦即「存在」或「熱」位元經清除)。一旦得自標籤表詢查的MPA位址，於PMH-TT的PPA檢索位置的狀態資訊中發現P=0(及/或C=1)(第10圖及第11圖二者中處理方塊1000及項目A)時，確定熱分頁失誤，如上於第8圖及第9圖詳細說明。標記於PMH-TT(第11圖之1100)從熱分頁變遷至冷分頁的資料可配置於熱分頁失誤(如上於第8圖及第9圖說明)。LDT(第11圖之1102)使用MPA位址檢索。更明確言之，配置的hMPA係用作為索引來檢索至LDT 1102找到PPA。如此，雖然PMH-TT 1100儲存實體MPA位址，且係使用PPA位址檢索，但LDT 1102為相反，原因在於其儲存PPA位址且係使用MPA位址檢索。

於不同實施例中，在LDT 1102的插槽配置可在各次熱分頁失誤時發生，或以其它方式發生，諸如在某個數目的熱分頁失誤後一次配置數個插槽。在PPA記憶體位置儲存於LDT 1102後的某個時間，MPT引擎處理邏輯將選擇在LDT

1102的一或多個PPA記憶體位置用於犧牲(第10圖及第11圖中處理方塊1002及項目B)。階級分裂犧牲乃儲存冷熱記憶體分頁的資料移轉入冷記憶體分頁之處理程序，故熱記憶體分頁被釋放供未來要求冷至熱分頁移轉時使用。

用於犧牲的選擇程序可以是數個實施例中之一者。舉例言之，MPT引擎處理邏輯可追蹤自各個熱分頁已經由CPU存取以來已經有多長時間；及基於該項資料，犧牲者包括已經於非作用態歷經最長時間的該等熱分頁。於另一個實例中，MPT引擎處理邏輯可追蹤資料的本地性，及將資料維持於熱分頁，該等熱分頁為作用態被利用且被簇集在一起於相對相鄰的實體記憶體分頁位置，而犧牲不接近熱分頁群簇的其它熱分頁。雖然此處未討論，藉MPT引擎處理邏輯可執行多個其它型別的犧牲演算法。

於許多實施例中，藉處理邏輯設定該分頁的狀態資訊犧牲位元為V=1，選擇LDT 1102插槽用於犧牲。當V係設定為「1」時，可提示邏輯將hMPA移轉至犧牲者列表1104結構供儲存，或處理邏輯可藉其它手段開始移轉至犧牲者列表1104，且於移轉完成時將犧牲者列表改成「1」。犧牲者列表1104係用來儲存標示用於熱至冷區域變遷的犧牲化實體記憶體分頁位址集合。雖然標示為犧牲者，但藉由清除犧牲位元(V=0)，及設定PMH-TT 1100中的回收位元(R=1)，及針對該分頁移除犧牲者列表1104分錄，MPT引擎處理邏輯可回收在犧牲者列表1104內的任何給定犧牲分頁。回收犧牲分頁的處理程序允許CPU及/或I/O DMA存取目前在

TLB的具轉譯之作用態分頁。如此允許回收分頁而無需通過TLB失誤詢查程序。

犧牲者可位在犧牲者列表1104內，只要有空間及/或有足夠數目的可用自由熱分頁即可。於某一點，犧牲者列表1104可成長至該點，列表裡的犧牲者數目超過臨界值，或自由熱分頁數目降至低於另一個臨界值。一旦已經通過此等臨界值中之一者，則MPT引擎處理邏輯可處理相當大量的犧牲分頁資料移轉至冷空間。完成此種熱至冷資料移轉，原因在於熱區域記憶體分頁的資料首先須移轉至冷區域記憶體分頁的資料，然後該熱區域記憶體分頁才可視為「自由」。

如此，轉向參考第10圖，MPT引擎處理邏輯須從犧牲者列表1104選出一或多個犧牲者用於資料移轉至記憶體之冷區域(處理方塊1004及第11圖之資料移轉項目C)。當將一分錄從犧牲者列表1104移動至污穢列表1106時要求TLB擊倒(shootdown)。因此，若TLB擊倒一起處理一群分頁而非處理各分頁來限制系統時間通常為更有效，因TLB擊倒處理程序導致緩慢。TLB擊倒要求處理器間中斷來掃除受影響的TLB詢查轉譯。在針對一記憶體分頁的TLB擊倒後，針對在犧牲者列表1104裡的MPA實體記憶體分頁分錄可移轉入污穢列表1106。此項移轉也涉及修改各分錄的狀態資訊來清除犧牲者列表(V=0)及設定污穢列表(D=1)。如前文討論，於許多實施例中，只有熱分頁轉譯係快取在TLB，如此雖然從hMPA至cMPA的熱至冷區域資料移轉要求TLB擊

倒，但冷至熱移動不要求TLB擊倒。

儲存於污穢列表的此等經TLB擊倒處理的分頁分錄也可藉CPU或I/O DMA存取而回收。再度，回收程序只要求移除在污穢列表的分頁分錄且更新該特定分頁的PMH-TT 1100分錄來清除污穢位元(D=0)及設定回收位元(R=0)。如此，犧牲及污穢列表分錄二者皆可被回收。一旦狀態位元被更新，回收分錄係指可使用而彷彿未曾犧牲的分錄。

在一給定時間，於污穢列表1106中的各個分錄需要其資料從hMPA記憶體分頁位置拷貝至得自冷自由列表1108的擇定自由cMPA記憶體分頁位置(處理方塊1006及第11圖之項目D)。如前述，如此在自由cMPA分頁形成曾經儲存在hMPA分頁的資料拷貝，來允許釋放hMPA分頁供未來使用。一旦發生資料移轉至記憶體冷區域，則處理邏輯以新cMPA資訊更新PMH-TT 1100(處理方塊1008及第11圖之項目E)。PMH-TT 1100利用LDT 1102更新來將所用cMPA索引設定至拷貝的hMPA的PPA索引。如此大致上重新對映PMH-TT 1100分錄，故詢查該PPA位址現在將指向現在已有資料拷貝的所用cMPA而非舊hMPA。

最後，MPT引擎處理邏輯然後將以hMPA資訊更新熱自由列表1110(處理方塊1010及第11圖之項目F)。儲存冷記憶體分頁在hMPA位址的資料現在安全地儲存於新cMPA分頁，hMPA分頁現在自由用作為自由熱記憶體區域分頁。如此，因此理由故，hMPA分頁係儲存於熱自由列表1110。熱自由列表1110分頁不再於TLB。如此允許TLB擊倒數目的減

少，原因在於從熱自由列表占用的所需熱分頁不需要額外TLB擊倒。反而，在犧牲者列表與污穢列表間進行TLB擊倒程序，此處一大群分頁分錄可在單次擊倒期間處理。

如前文討論，回收位元特徵允許拋棄具有該位元設定的犧牲者列表及污穢列表分頁分錄。拷貝處理方塊(處理方塊1006及第11圖之項目D)並不拷貝回收的分頁。於若干實施例中，MPT引擎拋棄在拷貝處理方塊回收的分頁。於其它實施例中，在進行回收之後，回收的分頁係從犧牲者列表及污穢列表移除。

雖然PMH-TT是通用結構，但通常第9圖及第11圖所示其它結構係遵照插槽定位。因此，一般而言，主記憶體係遵照插槽配置，而隱藏記憶體區(第1圖之138)包括通用PMH-TT及本地額外結構(例如本地描述符表(LDT)、犧牲者列表、熱自由列表等)。於若干實施例中，電腦系統針對系統中的多個CPU中之一者，儲存單一通用PMH-TT在記憶體儲存區中之一者的某個位置。於其它實施例中，通用PMH-TT的本地複本係遵照CPU儲存在各個隱藏記憶體區，及廣播更新訊息區在CPU間發送來更新其PMH-TT的本地複本，故整個系統操作中全部複本保持相同，但可能不常見。通常，PMH-TT係在系統的多個核心/插槽間劃分，使得使得PMH-TT存取具有命中插槽本地的記憶體相較於另一插槽上的記憶體相等機率。

移動於一個列表之分錄至另一列表的時間可藉一或多個臨界值規定。第12A至12D圖例示說明微分頁表引擎處理

邏輯可利用來確定何時回復記憶體分頁供使用(亦即記憶體分頁整頓)之流程圖之若干實施例。各項處理程序可藉處理邏輯執行，處理邏輯可以是硬體、軟體、韌體或該等三個邏輯形式之任何組合。此等略圖之各圖中，所指稱的「臨界值」是個可在任何給定時間確定之值，無論係在軟體啟動之前利用測試程序來決定最佳臨界值用來維持峰值效能，或在運轉時間使用演算法分析動態決定來基於目前工作負荷確定臨界值是否需要增減。於若干實施例中，並非利用全部臨界值測試，反而只利用臨界值測試之一子集(例如一或多個)。

第12A、12B、12C、及12D圖間關注的各個「臨界值以及任何其它非圖像臨界值機率彼此可具相似值或相異值。針對經MPT引擎維持列表資料結構的整頓部分，可利用多個不同臨界值，各個臨界值係特定體現。第12A至12D圖只提供就MPR引擎而言，可如何利用臨界值的若干實例。於所討論的許多實施例中，一或多個分錄係經拷貝/移動。實際上，拷貝/移動的列表分錄數目經常為列表中分錄的總數(亦即列表完全經掃除)。於其它實施例中，在一個區塊有被拷貝/移動之列表分錄的設定最大數目，及若分錄總數超過最大值，則針對任何單一處理程序，利用許可一次拷貝/移動的最大數目，及一或多個後續處理可用來處理任何其餘列表分錄。

第12A圖例示說明處理程序之一實施例確定何時犧牲者列表已經到達臨界值而開始整頓。於許多實施例中，MPT

引擎處理邏輯確定犧牲者列表裡的分錄總數已經達到高臨界值(處理方塊1200)。若確定屬於此種情況，則處理邏輯選擇犧牲者列表裡的一或多個分錄來移進記憶體冷區域(處理方塊1202)。然後處理邏輯執行所選分錄的TLB擊倒(處理方塊1204)，及完成整頓處理程序。

第12B圖例示說明處理程序之一實施例確定何時污穢列表已經到達高臨界值而開始整頓。於許多實施例中，MPT引擎處理邏輯確定犧牲者列表裡的分錄總數已經達到高臨界值(處理方塊1206)。若確定屬於此種情況，則處理邏輯拷貝一或多個污穢列表的分錄至記憶體冷區域(處理方塊1208)及處理完成。

第12C圖例示說明處理程序之一實施例確定何時熱自由列表已經到達低臨界值而開始整頓。於許多實施例中，MPT引擎處理邏輯確定於熱自由列表中的分錄總數已減至低於最小臨界要求值的數目(處理方塊1210)。若判定屬於此種情況，則處理邏輯從LDT選擇一或多個分錄用犧牲(處理方塊1212)。犧牲化選擇開始前文就第10圖詳加說明的處理程序。返回第12C圖，一旦已經選擇一或多個分錄，則處理邏輯拷貝所選分錄進入犧牲者列表(處理方塊1214)。

於許多實施例中，若熱自由列表中不含足夠的分錄，首先自第12C圖處理收集犧牲者，然後，一旦已經收集，處理邏輯對犧牲者執行第12A圖之處理來將犧牲者移動至熱自由列表。

最後，第12D圖例示說明處理程序之另一實施例確定何

時熱自由列表已經到達低臨界值而開始整頓。如同第12C圖，第12D圖之MPT引擎處理邏輯確定於熱自由列表中的分錄總數已減至低於最小臨界要求值的數目(處理方塊1216)。若判定屬於此種情況，則處理邏輯從記憶體熱區域拷貝一或多個污穢列表分錄至記憶體冷區域(處理方塊1218)，及處理完成。總而言之，在此等經拷貝的分錄能夠重新用在熱自由列表之前要求轉譯後備緩衝器(TLB)擊倒。

二層級記憶體

微分頁表引擎也可利用來體現二層級記憶體(2LM)記憶體次系統。第13圖描述在電腦系統內部管理二層級記憶體次系統之微分頁表實施例。

第13圖例示說明之電腦系統係類似第1圖所示電腦系統，許多組件提供類似功能。本圖的其它版本所顯示的某些元件諸如I/O次系統並未特別顯示於第13圖以求清晰。雖然於圖中未顯示，但類似第1圖及第2圖所示I/O次系統通常係在第13圖所示系統體現。

於許多實施例中，分開的記憶體控制器亦即記憶體控制器B 1300駐在電腦系統100內。記憶體控制器B 1300係透過高速I/O介面122耦接至CPU 102，或記憶體控制器B可在CPU晶粒體現而非呈分開晶粒，但圖中未顯示此一實施例。高速I/O介面可以是若干型別互連體中之一者，諸如PCI-快速、QPI等。記憶體控制器B 1300轉而對第二記憶體諸如記憶體B 1302提供控制。記憶體B 1302可以是與記憶體A 110不同型別的記憶體。舉例言之，記憶體A 110可包含一型

DRAM，但記憶體B 1302可包含一型非依電性記憶體。於不同實施例中，記憶體B 1302可以是相變記憶體(PCM)、其它型別非依電性記憶體、或標準DRAM或低功率DRAM等。

於許多實施例中，包含一般儲存區B 1304的記憶體B 1302可視為電腦系統100內部的主記憶體，其中包含一般儲存區A 140的記憶體A 110可體現為DRAM快取記憶體。該快取記憶體包含DRAM，及於許多實施例中包含多個十億位元組(GB)儲存空間，能夠吸收電腦系統100常規操作期間的大部分寫入。於記憶體B為非依電性記憶體的實施例中，此項吸收效應協助最小化非依電性記憶體B 1302的磨耗，有助於最小化下列效應：相變記憶體(PCM)或其它型別非依電性記憶體(NVM)的有限寫入使用壽命及隱藏寫至此等型別記憶體的長延遲時間。

如前文詳細說明，使用MPT引擎體現二層級記憶體(2LM)通常係以階級分裂之相似方式發揮效果。大致上，MPT引擎128可設定記憶體空間熱區域對映至一般儲存區A 140，及記憶體空間冷區域對映至一般儲存區B 1304。如此於許多實施例中，一般儲存區A乃熱區域，而一般儲存區B乃冷區域。來自記憶體存取冷區域的資料係以記憶體分頁移轉調換而帶入熱區域。

在分頁行進詢查處理後，一旦發現記憶體請求不存在TLB，也發現不存在記憶體熱區域(一般儲存區A 140)時，MML 142可走出至記憶體控制器B 1300且請求存取一般儲存區B 1304中的分頁。

利用針對記憶體冷區域的第二記憶體控制器及第二記憶體，前文於第8圖及第9圖所述標準熱分頁失誤處理流程恰重新施加至第13圖之體現。此外，對調記憶體熱及冷分頁的整頓通常也同等適用。於許多2LM實施例中，有MML 142執行的額外處理。當記憶體B 1302為非依電性時，磨耗均勻演算法可結合入MML 142內部的邏輯。於極少至無活性期間，MPT引擎128的MML 142部分可指令記憶體控制器B 1300重新分配儲存在記憶體B 1302內部的資料部分來均勻分散磨耗於組成全部記憶體B 1302的相變記憶體(PCM)裝置間。

第14圖描述特定PCM記憶體次系統之一實施例。記憶體B 1302顯示為x16 PCM裝置組態。於許多其它圖中未顯示的實施例中，PCM裝置可以堆疊數個裝置高來更進一步增加儲存量而存取時間只有相對小的延遲增加。記憶體控制器B 1300係藉鏈路1400耦接至CPU。從CPU裡的邏輯發出請求至控制器。記憶體控制器B 1300可包含數個DMA單元(單元1-N)，該等DMA單元係耦接至記憶體B 1302中的整合式鏈路1502(亦即匯流排)。該等DMA單元可以串接方式工作，發送請求給記憶體B 1302及接收送回的資料。然後，此一資料橫過鏈路1400送回給CPU。

本發明實施例之元件也可提供作為儲存機器可執行指令之機器可讀取媒體。機器可讀取媒體可包括但非限於快閃記憶體、光碟、光碟-唯讀記憶體(CD-ROM)、數位影音碟(DVD) ROM、隨機存取記憶體(RAM)、可抹除可規劃唯

讀記憶體(EPROM)、可電氣抹除可規劃唯讀記憶體(EEPROM)、磁卡或光卡、傳播媒體、或適合儲存電子指令的其它型別機器可讀取媒體。

於前文描述及申請專利範圍中可使用「包括」及「包含」等術語連同其衍生詞且意圖視為彼此的同義詞。此外，於後文說明及申請專利範圍中可使用「耦接」及「連結」等詞連同其衍生詞。須瞭解此等術語絕非意圖為彼此的同義詞。反而於特定實施例中，「連結」可用來指示二或多個元件係彼此直接實體接觸或電氣接觸。「耦接」可用來指示二或多個元件係彼此直接實體接觸或電氣接觸。但「耦接」一詞也表示二或多個元件並非彼此直接接觸，但仍然彼此協作、互動、或通訊。

於前文說明中，若干術語係用來描述本發明之實施例。舉例言之，「邏輯」一詞係表示硬體、韌體、軟體(或其任一種組合)執行一或多項功能。例如「硬體」包括但非限於積體電路、有限狀態機、或甚至綜合邏輯。積體電路可呈處理器形式，諸如微處理器、特定應用積體電路、數位信號處理器、微控制器等。

須瞭解本說明書全文中述及「一個實施例」或「一實施例」表示結合該實施例描述的特定特徵、結構或特性係含括於至少一個本發明之實施例。因此強調且須瞭解於本說明書各部分兩次或更多次述及「一實施例」或「一個實施例」或「另一實施例」並非必然全部指稱同一個實施例。又復，特定特徵、結構或特性視合宜而可組合於一或多個

本發明之實施例。

同理，須瞭解於前文本發明之實施例之描述中，偶爾多個特徵結構結合在單一實施例、圖式、或其說明中以求揭示文的流暢，協助各個發明構面中之一或多者的瞭解。但此種揭示方法不應解譯為反映出請求專利的主旨意圖需要比較申請專利範圍各項中更多個特徵結構。反而如下申請專利範圍反映出本發明之構面係在少於單一前文揭示實施例的全部特徵。如此，詳細說明部分後方之申請專利範圍藉此明確結合於本詳細說明部分。

【圖式簡單說明】

第1圖描述體現微分頁表之電腦系統之一個實施例。

第2圖描述體現微分頁表之電腦系統之又一個實施例。

第3圖描述體現微分頁表之電腦系統之另一個實施例。

第4圖例示說明分頁失誤處理器標籤表之一實施例。

第5圖例示說明針對階級分開，體現微分頁表之電腦系統之一個實施例。

第6圖例示說明至少部分利用在階級分開，體現微分頁表之電腦系統之一個實施例。

第7圖例示說明當至少部分體現在階級分開時，分頁失誤處理器標籤表之一實施例。

第8圖為用以處理熱分頁失誤之一實施例之流程圖。

第9圖例示說明當回應於熱分頁失誤時，由微分頁表引擎所利用的若干額外微分頁表資料結構之實施例。

第10圖為維護程序之實施例之流程圖，該維護程序係

用來對多個記憶體分頁提供以冷至熱記憶體分頁資料移轉期間利用作為熱分頁的能力。

第11圖例示說明於維護程序期間，由微分頁表引擎所利用的若干額外微分頁表資料結構之實施例。

第12A至12D圖例示說明微分頁表引擎處理邏輯可利用來確定何時回復記憶體分頁供使用之流程圖之若干實施例。

第13圖描述在電腦系統內部管理二層級記憶體次系統之微分頁表實施例。

第14圖描述相變記憶體特定記憶體次系統之實施例。

【主要元件符號說明】

100...電腦系統

102...中央處理單元(CPU)

104...核心

106...快取記憶體

108...記憶體控制器

110、300...系統記憶體

112、200...I/O複合體

114...I/O裝置

116...I/O介面

118、204...I/O配接器

120...基本輸出入系統(BIOS)

122...高速I/O介面

124...CPU間高速介面

126、202...I/O虛擬化邏輯

128...微分頁表(MPT)引擎

130...轉譯後備緩衝器(TLB)

132...分頁失誤處理器(PMH)邏輯

134...核心記憶體請求

136、400、704、900、1100...分頁失誤處理器(PMH)標籤表(TT)、PMH-TT

138...隱藏記憶體區

140...一般儲存區

142...記憶體管理邏輯(MML)

202...I/O虛擬化邏輯

402...分錄

404...實體記憶體

406...OS維持的分頁表

500...CPU A

502...CPU B

504...點對點(P2P)鏈路

506...記憶體A

508...記憶體B

600...作用態記憶體

602...非作用態記憶體

700...微實體位址(MPA)空間冷區域

702...微實體位址(MPA)空間熱區域

704...PMH標籤表

706...熱微實體位址(hMPA)

708...MPT引擎分頁行動OS分頁表

710...分頁行動VMM分頁表

800-814、1000-1010、1200- 1218...處理方塊

902...冷自由列表資料結構

904...熱自由列表資料結構

1102...本地描述符表(LDT)

1104...犧牲者列表

1106...污穢列表

1108...冷自由列表

1110...熱自由列表

1300...記憶體控制器B

1302...記憶體B

1304...一般儲存區B

1400...鏈路

1402...整合型鏈路

七、申請專利範圍：

1. 一種微分頁表引擎裝置，其係包含：

邏輯，用以針對一記憶體分頁接收一記憶體分頁請求，其中該記憶體分頁請求包括該記憶體分頁之一線性位址；

一轉譯後備緩衝器(TLB)，用以儲存一或更多記憶體分頁位址轉譯，該等記憶體分頁位址轉譯經組態以將線性位址轉譯成對映至一記憶體之微實體位址的平台實體位址；

一分頁失誤處理器標籤表，經組態以儲存多個分錄(entries)，該等分錄經組態以檢索平台實體位址至微實體位址，其中，除了檢索一平台實體位址至一微實體位址之一索引，每一分錄包括一與該微實體位址相關聯之記憶體分頁的狀態資訊；

一分頁失誤處理器邏輯，用以回應於該TLB不儲存針對由該記憶體分頁請求所參照的該記憶體分頁之該記憶體分頁位址轉譯而執行在該分頁失誤處理器標籤表內之一微實體位址詢查；及

一記憶體管理邏輯，用以實施該記憶體之階級分裂(rank shedding)，包括更新在該分頁失誤處理器標籤表之該等分錄中的該狀態資訊以反映關於執行在該記憶體上之操作的階級分裂之結果。

2. 如申請專利範圍第1項之裝置，其中該分頁失誤處理器標籤表係位在一系統記憶體之可由該微分頁表引擎存

取之一隱藏區。

3. 如申請專利範圍第2項之裝置，其中該分頁失誤處理器標籤表係完全聯結。

4. 如申請專利範圍第1項之裝置，其中該記憶體係可劃分成一或更多作用態或非作用態區域，且該狀態資訊包含指示與一微實體位址相關聯之一記憶體分頁是否在一作用態或非作用態區域之狀態資訊。

5. 如申請專利範圍第4項之裝置，其中該記憶體管理邏輯係可操作來藉將資料從於一作用態區域之一第一記憶體分頁移轉至於一非作用態區域之一第二記憶體分頁而釋放該第一記憶體分頁，且更新該分頁失誤處理器標籤表以反映該移轉。

6. 如申請專利範圍第4項之裝置，其中該記憶體管理邏輯係可操作來回應於靶定於該非作用態區域之一第三記憶體分頁的所接收之記憶體請求而將資料從於該非作用態區域之該第三記憶體分頁移轉至於該作用態區域之該第一記憶體分頁，且更新該分頁失誤處理器標籤表以反映該移轉，包括在該更新期間閂鎖該分頁失誤處理器標籤表。

7. 如申請專利範圍第4項之裝置，其中該記憶體包含實體記憶體之二部分，一實體記憶體A及一實體記憶體B，其中，於操作期間，該記憶體管理邏輯係可操作來將該實體記憶體A標示為一作用態區域且將該實體記憶體B標示為一非作用態區域。

8. 如申請專利範圍第1項之裝置，其中該等微實體位址界定用於該記憶體之一微實體位址空間，包括在多個處理器之中共享之該微實體位址空間之一範圍，及其中該記憶體管理邏輯係可操作來優先排序該等多個處理器之每一者的該微實體位址空間之多個部分。

9. 一種用於實施微分頁表之方法，其係包含：

在線性位址與平台實體位址之間儲存一或更多記憶體分頁位址轉譯於一轉譯後備緩衝器(TLB)，其中該等平台實體位址對映至一記憶體之微實體位址；

儲存分錄於一分頁失誤處理器標籤表以檢索平台實體位址至微實體位址，其中，除了檢索一平台實體位址至一微實體位址之一索引，每一分錄包括一與該微實體位址相關聯之記憶體分頁的狀態資訊；

實施該記憶體之階級分裂，包括更新在該分頁失誤處理器標籤表中的該狀態資訊以反映關於執行在該記憶體上之操作的階級分裂之結果；

接收針對在該記憶體中之一記憶體分頁的一記憶體分頁請求，該記憶體分頁請求包括一線性位址；

回應於該記憶體分頁請求，搜尋該TLB以定位對映至該記憶體之一微實體位址的相對應平台實體位址；以及

回應於該TLB不儲存針對由該記憶體分頁請求所參照的該記憶體分頁之該記憶體分頁位址轉譯，搜尋該分頁失誤處理器標籤表以獲得對於該TLB之更新。

10. 如申請專利範圍第9項之方法，其中該分頁失誤處理器標籤表係位在一系統記憶體之可由該微分頁表引擎存取之一隱藏區。

11. 如申請專利範圍第10項之方法，其中該分頁失誤處理器標籤表係完全聯結。

12. 如申請專利範圍第9項之方法，其中實施包含將該記憶體劃分成一或更多作用態或非作用態區域。

13. 如申請專利範圍第12項之方法，其係進一步包含：

藉將資料從於一作用態區域之一第一記憶體分頁移轉至於一非作用態區域之一第二記憶體分頁而釋放該第一記憶體分頁，且更新該分頁失誤處理器標籤表以反映該移轉。

14. 如申請專利範圍第13項之方法，其係進一步包含：

回應於靶定於該非作用態區域之一第三記憶體分頁的所接收之記憶體請求而將資料從於該非作用態區域之該第三記憶體分頁移轉至於該作用態區域之該第一記憶體分頁，且更新該分頁失誤處理器標籤表以反映該移轉，包括在該更新期間閂鎖該分頁失誤處理器標籤表。

15. 如申請專利範圍第12項之方法，其中該記憶體包含不同記憶體技術的實體記憶體之二部分，一實體記憶體A及一實體記憶體B，及其中該方法進一步包含將該實體記憶體A標示為一作用態區域且將該實體記憶體B標示為一非作用態區域。

16. 如申請專利範圍第9項之方法，其中該等微實體位址界定用於該記憶體之一微實體位址空間，包括在多個處理器之中共享之微實體位址空間之一範圍，及其中該方法進一步包含優先排序該等多個處理器之每一者的該微實體位址空間之多個部分。

17. 一種具有指令儲存於其上之非暫態機器可讀取媒體，該等指令當由一機器執行時，使得該機器進行下列動作：

在線性位址與平台實體位址之間儲存一或更多記憶體分頁位址轉譯於一轉譯後備緩衝器(TLB)，其中該等平台實體位址對映至一記憶體之微實體位址；

儲存分錄於一分頁失誤處理器標籤表以檢索平台實體位址至微實體位址，其中，除了檢索一平台實體位址至一微實體位址之一索引，每一分錄包括一與該微實體位址相關聯之記憶體分頁的狀態資訊；

實施該記憶體之階級分裂，包括更新在該分頁失誤處理器標籤表中的該狀態資訊以反映關於執行在該記憶體上之操作的階級分裂之結果；

接收針對在該記憶體中之一記憶體分頁的一記憶體分頁請求，該記憶體分頁請求包括一線性位址；

回應於該記憶體分頁請求，搜尋該TLB以定位對映至該記憶體之一微實體位址的相對應平台實體位址；以及

回應於該TLB不儲存針對由該記憶體分頁請求所參照的該記憶體分頁之該記憶體分頁位址轉譯，搜尋該

分頁失誤處理器標籤表以獲得對於該TLB之更新。

18. 如申請專利範圍第17項之非暫態機器可讀取媒體，其中該分頁失誤處理器標籤表係位在一系統記憶體之可由該微分頁表引擎存取之一隱藏區。

19. 如申請專利範圍第18項之非暫態機器可讀取媒體，其中該分頁失誤處理器標籤表係完全聯結。

20. 如申請專利範圍第17項之非暫態機器可讀取媒體，其中實施包含將該記憶體劃分成一或更多作用態或非作用態區域。

21. 如申請專利範圍第20項之非暫態機器可讀取媒體，其中進一步使得該機器進行下列動作：

藉由將資料從於一作用態區域之一第一記憶體分頁移轉至於一非作用態區域之一第二記憶體分頁而釋放該第一記憶體分頁，且更新該分頁失誤處理器標籤表以反映該移轉。

22. 如申請專利範圍第21項之非暫態機器可讀取媒體，其中進一步使得該機器進行下列動作：

回應於靶定於該非作用態區域之一第三記憶體分頁的所接收之記憶體請求而將資料從於該非作用態區域之該第三記憶體分頁移轉至於該作用態區域之該第一記憶體分頁，且更新該分頁失誤處理器標籤表以反映該移轉，包括在該更新期間閂鎖該分頁失誤處理器標籤表。

23. 如申請專利範圍第22項之非暫態機器可讀取媒體，其中

該記憶體包含不同記憶體技術的實體記憶體之二部分，一實體記憶體A及一實體記憶體B，及其中進一步使得該機器將該實體記憶體A標示為一作用態區域且將該實體記憶體B標示為一非作用態區域。

24. 如申請專利範圍第17項之非暫態機器可讀取媒體，其中該等微實體位址界定用於該記憶體之一微實體位址空間，包括在多個處理器之中共享之微實體位址空間之一範圍，及其中進一步使得該機器優先排序該等多個處理器之每一者的該微實體位址空間之多個部分。

25. 一種用於實施微分頁表之系統，其係包含：

一記憶體，其中該記憶體包括一隱藏部分，其用以儲存至少一個經組態以儲存多個分錄的分頁失誤處理器標籤表，該等分錄經組態以檢索平台實體位址至該記憶體的微實體位址，其中，除了檢索一平台實體位址至一微實體位址之一索引，每一分錄包括一與該微實體位址相關聯之記憶體分頁的狀態資訊；及

一處理器，該處理器包括：

邏輯，用以針對一記憶體分頁接收一記憶體分頁請求，其中該記憶體分頁請求包括該記憶體分頁之一線性位址；

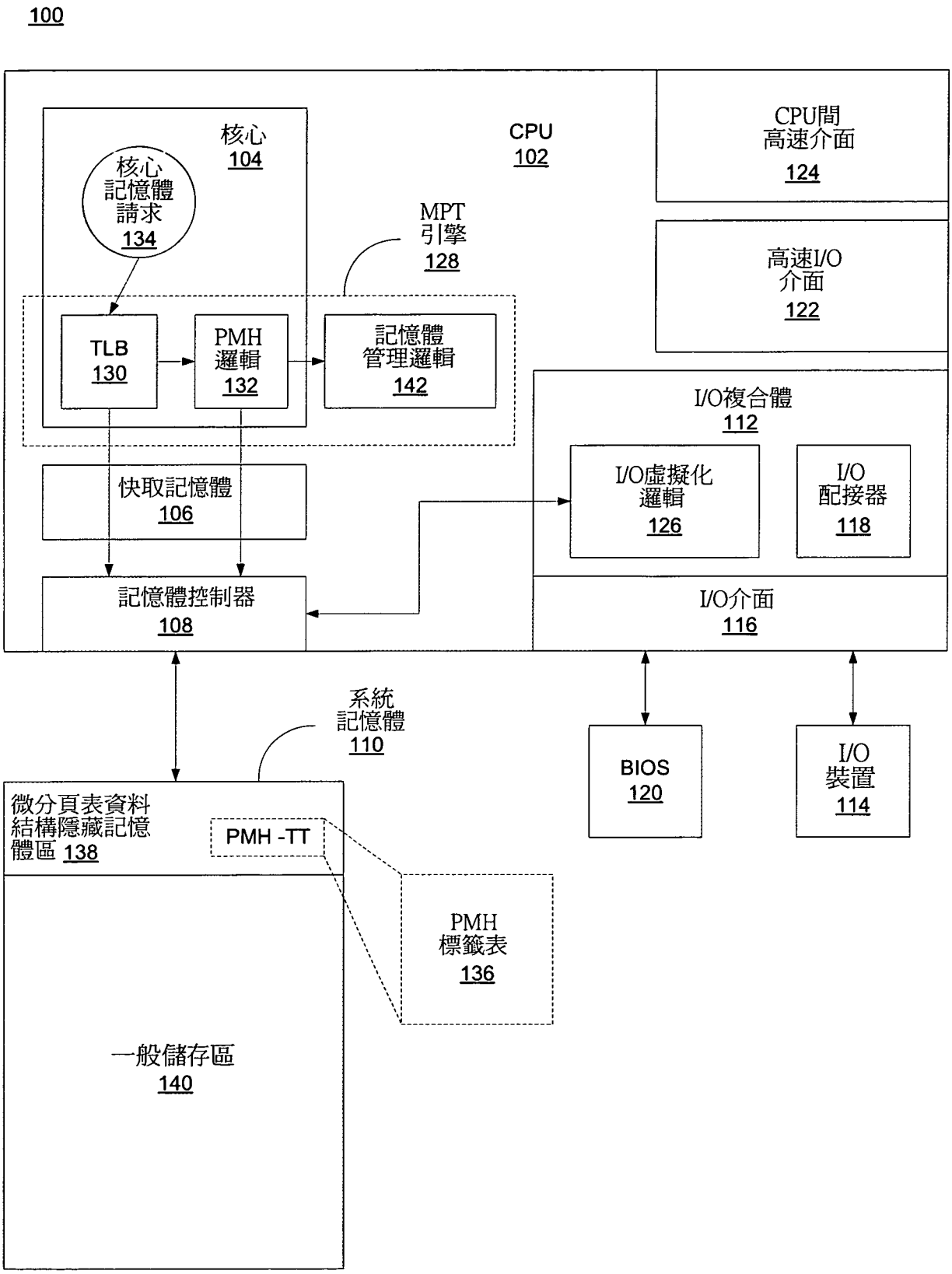
一轉譯後備緩衝器(TLB)，用以儲存一或更多記憶體分頁位址轉譯，該等記憶體分頁位址轉譯經組態以將線性位址轉譯成平台實體位址；

一分頁失誤處理器邏輯，用以回應於該TLB不

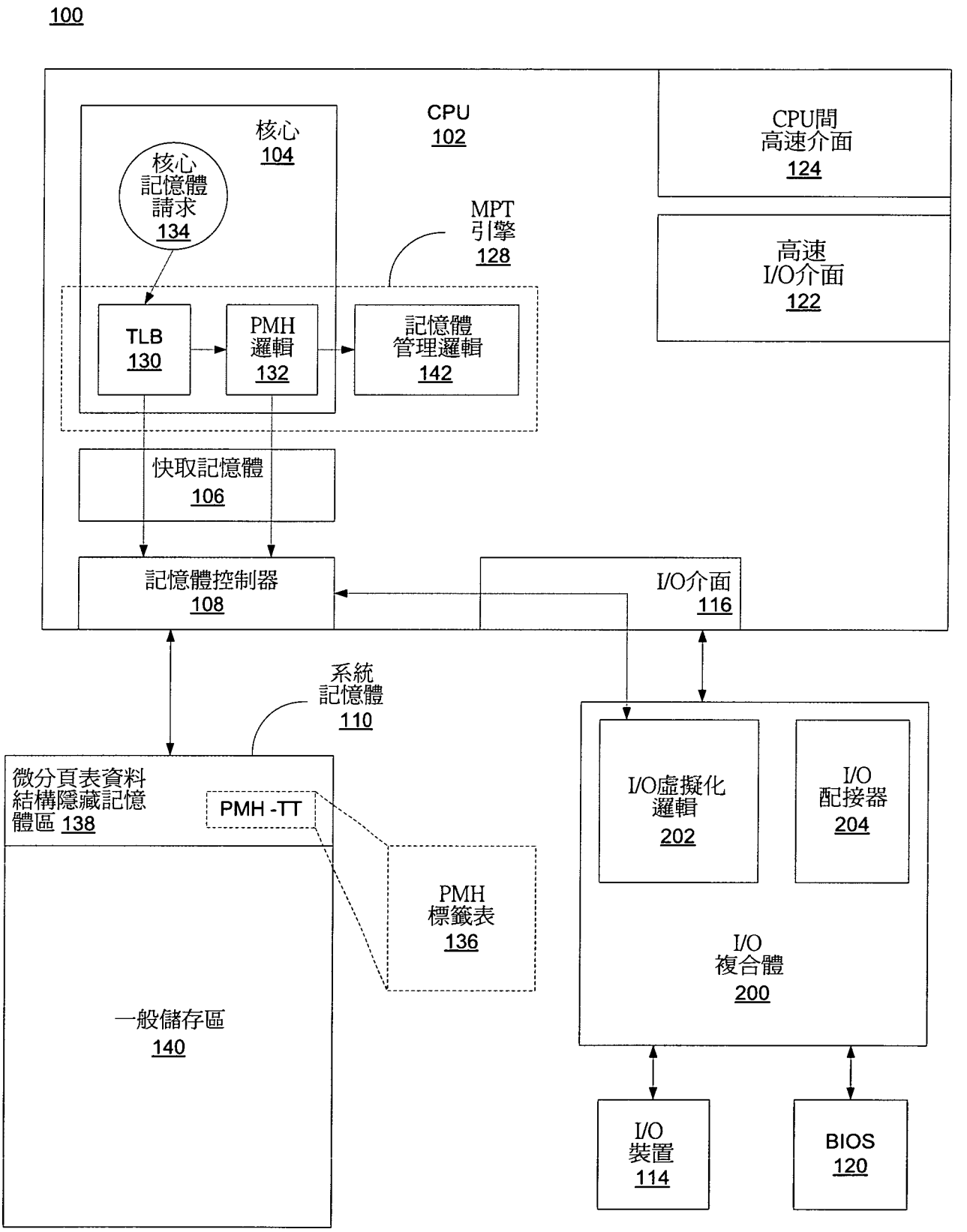
儲存針對由該記憶體分頁請求所參照的該記憶體分頁之該記憶體分頁位址轉譯而執行在該分頁失誤處理器標籤表內之一微實體位址詢查；及

一記憶體管理邏輯，用以實施該記憶體之階級分裂，包括更新在該分頁失誤處理器標籤表之該等分錄中的該狀態資訊以反映關於執行在該記憶體上之操作的階級分裂之結果。

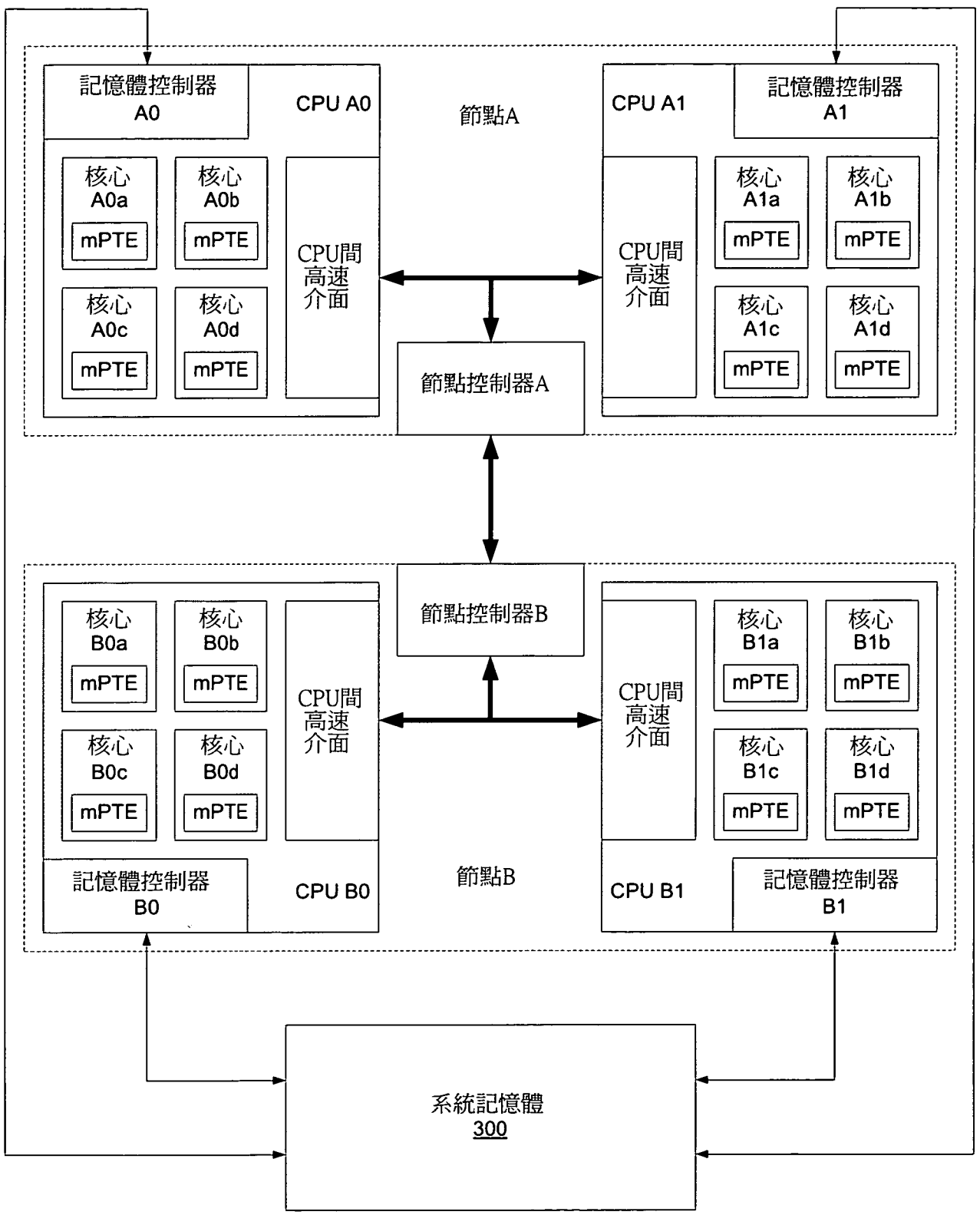
八、圖式：


100
核心
104
核心
記憶體
請求
134
CPU
102
CPU間
高速介面
124
MPT
引擎
128
高速I/O
介面
122
TLB
130
PMH
邏輯
132
記憶體
管理邏輯
142
I/O複合體
112
I/O虛擬化
邏輯
126
I/O
配接器
118
快取記憶體
106
記憶體控制器
108
I/O介面
116
系統
記憶體
110
BIOS
120
I/O
裝置
114
微分頁表資料
結構隱藏記憶
體區 138
PMH -TT
PMH
標籤表
136
一般儲存區
140

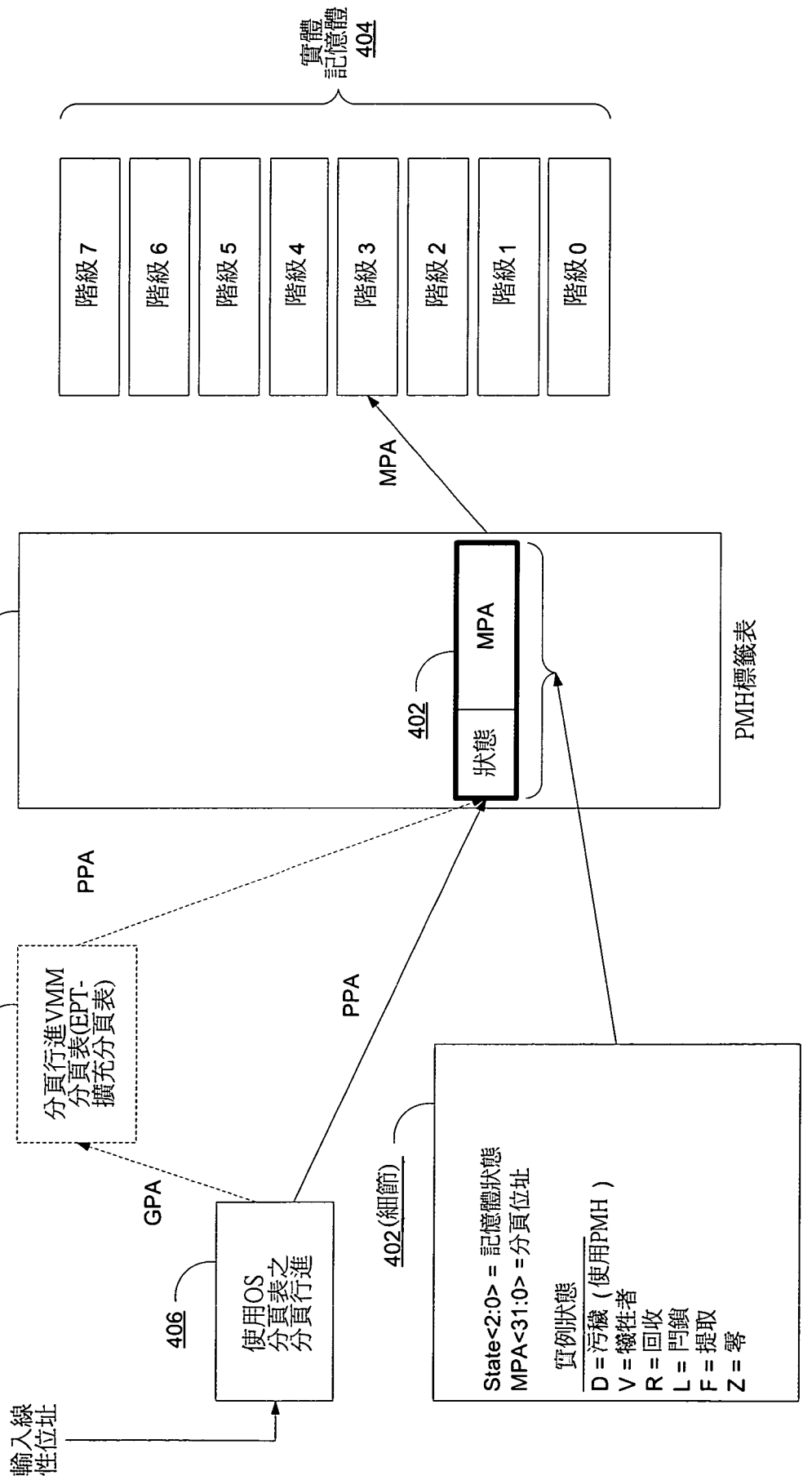
第 1 圖


100
CPU
102
核心
104
核心
記憶體
請求
134
MPT
引擎
128
TLB
130
PMH
邏輯
132
記憶體
管理邏輯
142
CPU間
高速介面
124
高速
I/O介面
122
快取記憶體
106
記憶體控制器
108
I/O介面
116
系統
記憶體
110
微分頁表資料
結構隱藏記憶
體區 138
PMH -TT
PMH
標籤表
136
一般儲存區
140
I/O虛擬化
邏輯
202
I/O
配接器
204
I/O
複合體
200
I/O
裝置
114
BIOS
120

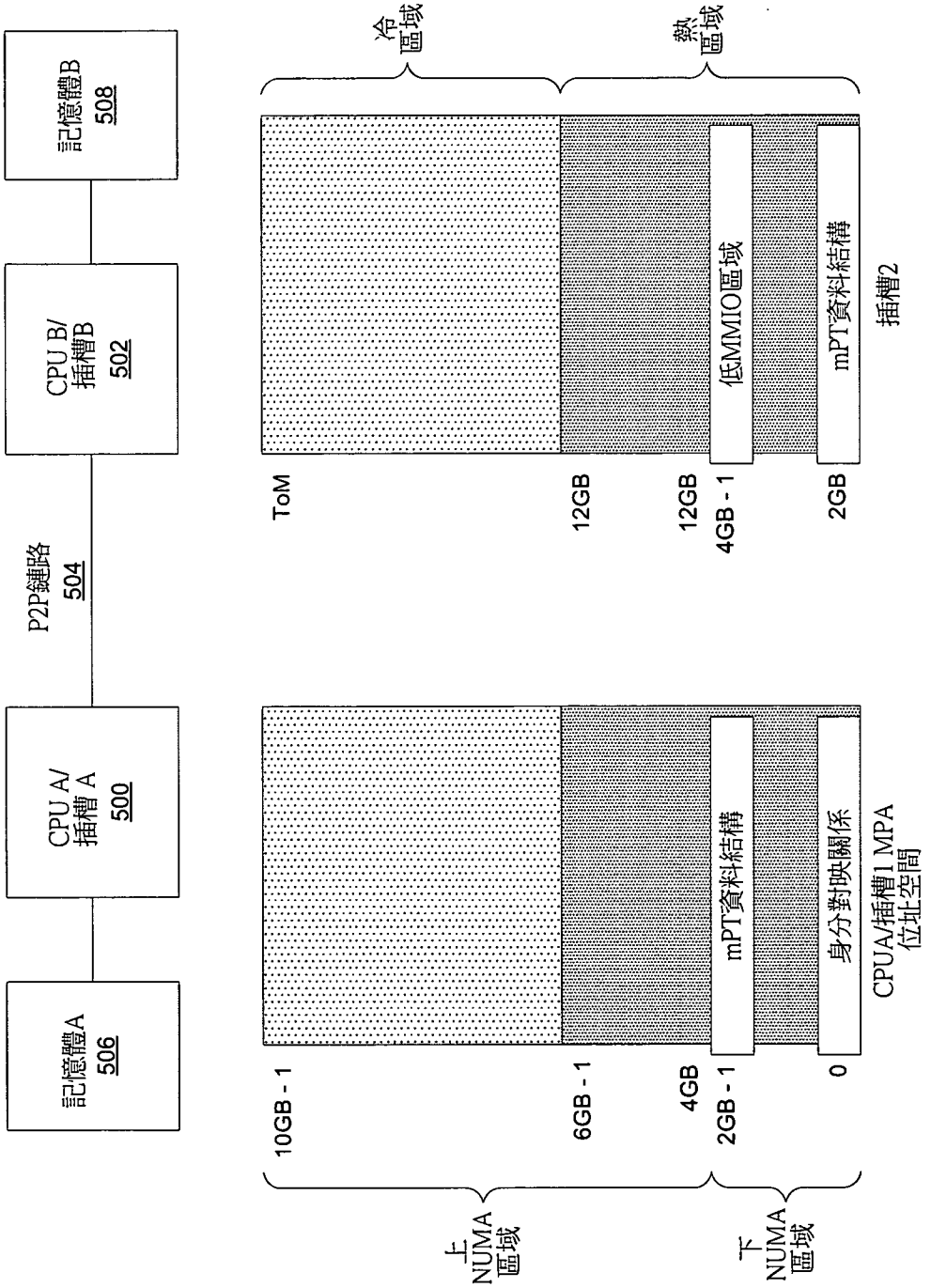
第 2 圖


記憶體控制器
A0
CPU A0
節點A
CPU A1
記憶體控制器
A1
核心
A0a
mPTE
核心
A0b
mPTE
CPU間
高速
介面
核心
A0c
mPTE
核心
A0d
mPTE
節點控制器A
CPU間
高速
介面
核心
A1a
mPTE
核心
A1b
mPTE
核心
A1c
mPTE
核心
A1d
mPTE
節點控制器B
核心
B0a
mPTE
核心
B0b
mPTE
CPU間
高速
介面
核心
B0c
mPTE
核心
B0d
mPTE
CPU間
高速
介面
核心
B1a
mPTE
核心
B1b
mPTE
核心
B1c
mPTE
核心
B1d
mPTE
記憶體控制器
B0
CPU B0
節點B
CPU B1
記憶體控制器
B1
系統記憶體
300

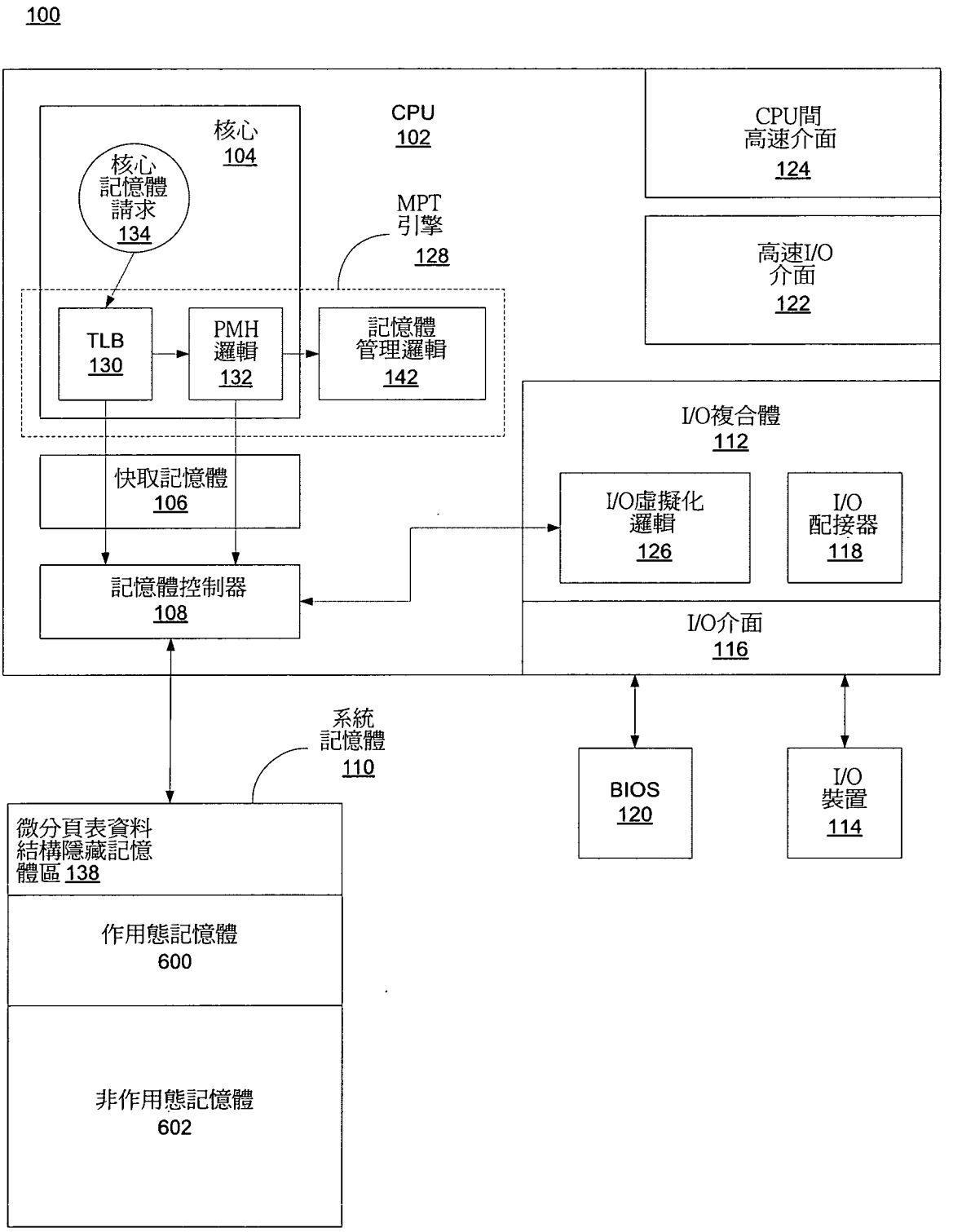
第 3 圖


實體記憶體 404
階級 7
階級 6
階級 5
階級 4
階級 3
階級 2
階級 1
階級 0
MPA
400
402
狀態
MPA
PMH標籤表
PPA
408
分頁行進VMM 分頁表(EPT-擴充分頁表)
PPA
402 (細節)
State<2:0> = 記憶體狀態
MPA<31:0> =分頁位址
實例狀態
D = 污穢 (使用PMH)
V = 機性者
R = 回收
L = 門鎖
F = 提取
Z = 零
GPA
406
使用OS 分頁表之分頁行進
輸入線性位址

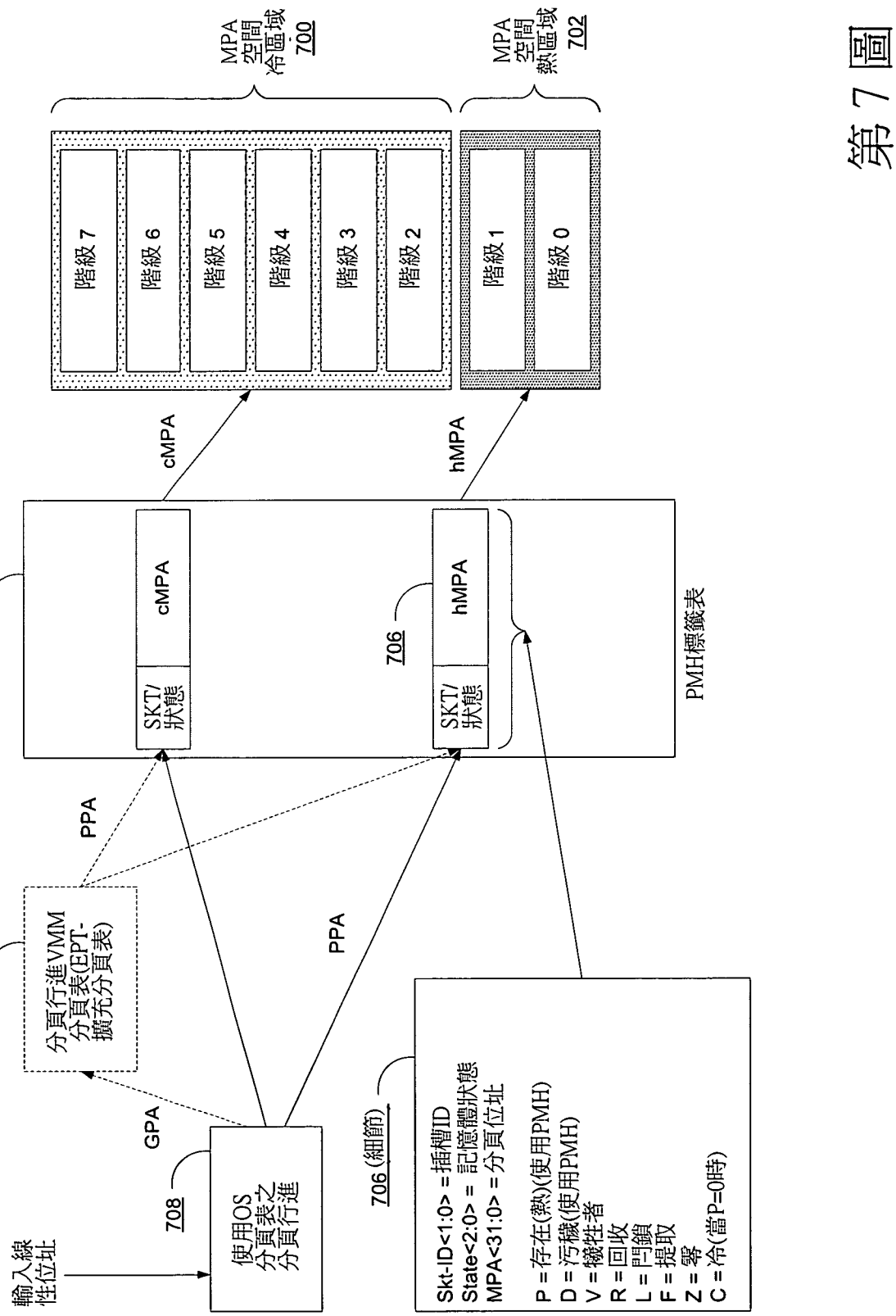
第 4 圖


記憶體A
506
CPU A/
插槽 A
500
P2P鏈路
504
CPU B/
插槽B
502
記憶體B
508
上
NUMA
區域
下
NUMA
區域
10GB - 1
6GB - 1
4GB
2GB - 1
0
mPT資料結構
身分對映關係
CPUA/插槽1 MPA
位址空間
冷
區域
熱
區域
ToM
12GB
12GB
4GB - 1
2GB
低MMIO區域
mPT資料結構
插槽2

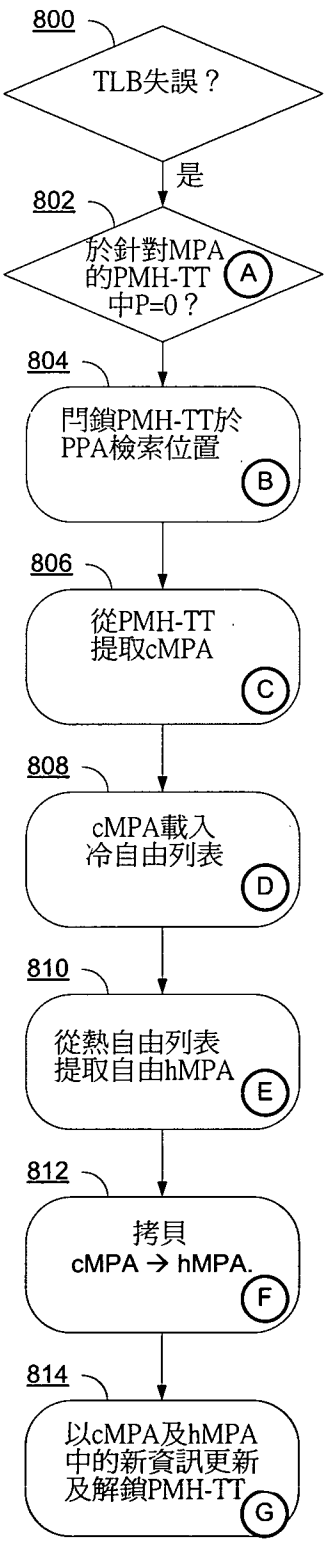
第 5 圖


100
CPU
102
核心
104
核心
記憶體
請求
134
MPT
引擎
128
TLB
130
PMH
邏輯
132
記憶體
管理邏輯
142
CPU間
高速介面
124
高速I/O
介面
122
I/O複合體
112
I/O虛擬化
邏輯
126
I/O
配接器
118
I/O介面
116
快取記憶體
106
記憶體控制器
108
系統
記憶體
110
BIOS
120
I/O
裝置
114
微分頁表資料
結構隱藏記憶
體區 138
作用態記憶體
600
非作用態記憶體
602

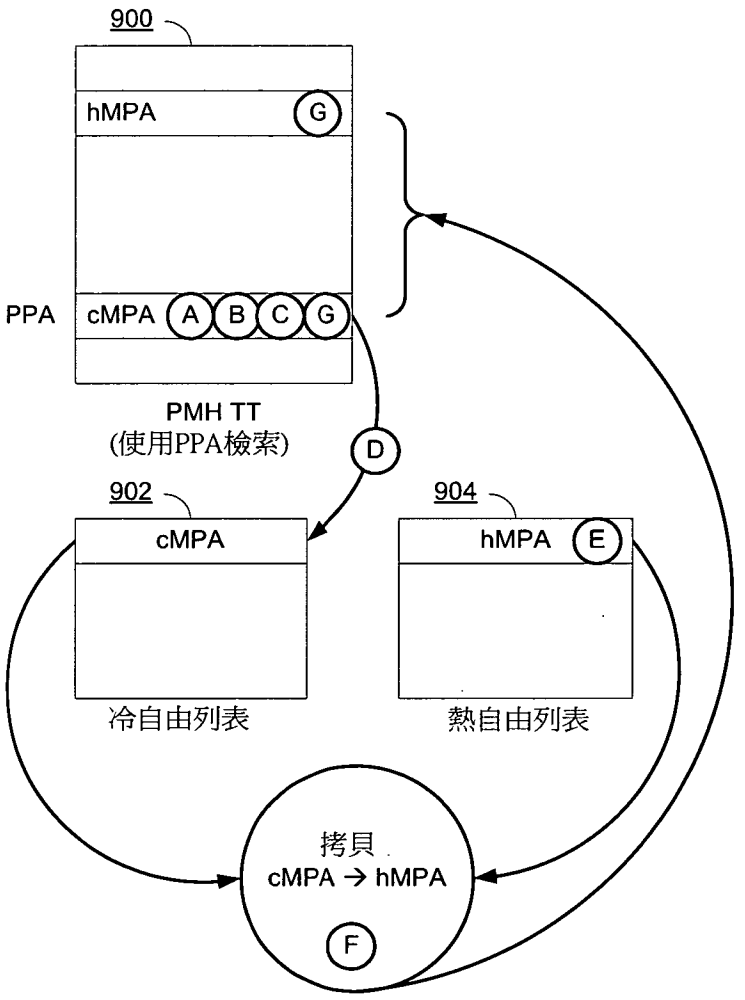
第 6 圖


MPA
空間
冷區域
700
MPA
空間
熱區域
702
階級 7
階級 6
階級 5
階級 4
階級 3
階級 2
階級 1
階級 0
cMPA
hMPA
704
SKT/
狀態
cMPA
706
SKT/
狀態
hMPA
PMH標籤表
PPA
710
分頁行進VMM
分頁表(EPT-
擴充分頁表)
GPA
708
使用OS
分頁表之
分頁行進
輸入線
性位址
PPA
706 (細節)
Skt-ID<1:0> = 插槽ID
State<2:0> = 記憶體狀態
MPA<31:0> = 分頁位址
P = 存在(熱)(使用PMH)
D = 污穢(使用PMH)
V = 穩性者
R = 回收
L = 閂鎖
F = 提取
Z = 零
C = 冷(當P=0時)

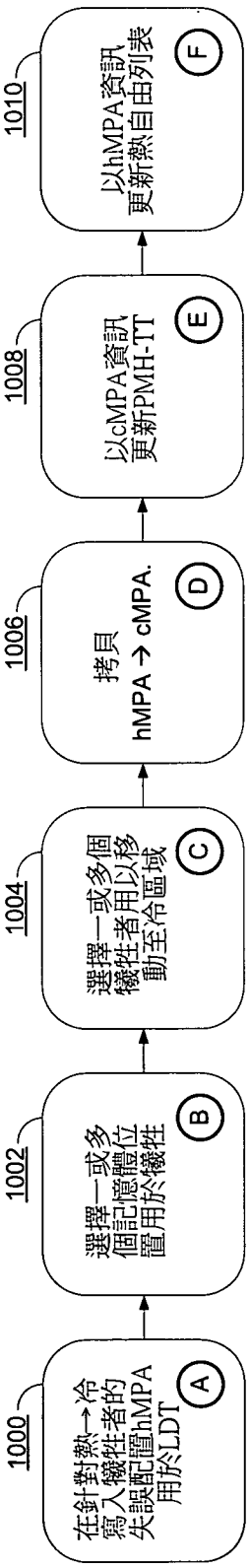
第 7 圖


800
TLB失誤？
是
802
於針對MPA的PMH-TT中P=0？
A
804
閂鎖PMH-TT於PPA檢索位置
B
806
從PMH-TT提取cMPA
C
808
cMPA載入冷自由列表
D
810
從熱自由列表提取自由hMPA
E
812
拷貝
cMPA → hMPA.
F
814
以cMPA及hMPA中的新資訊更新及解鎖PMH-TT
G

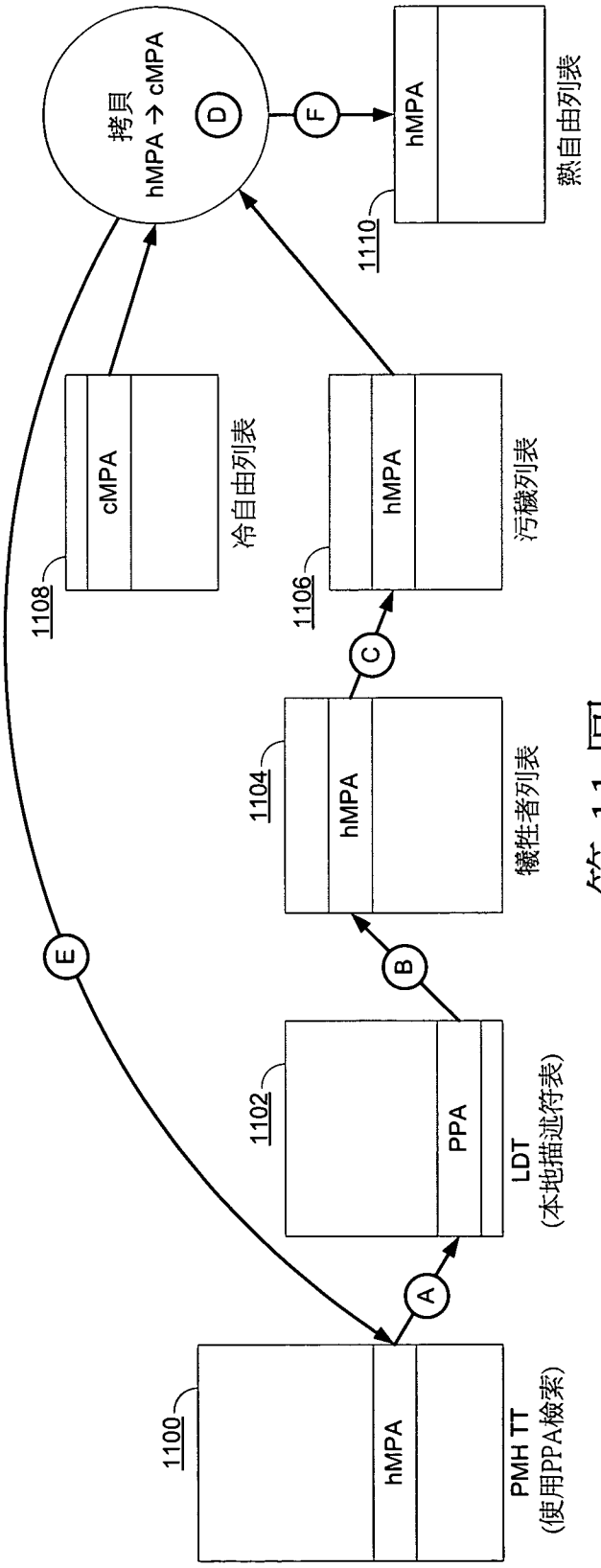
第 8 圖


900
hMPA
G
PPA
cMPA
A
B
C
G
PMH TT
(使用PPA檢索)
D
902
cMPA
冷自由列表
904
hMPA
E
熱自由列表
拷貝
cMPA → hMPA
F

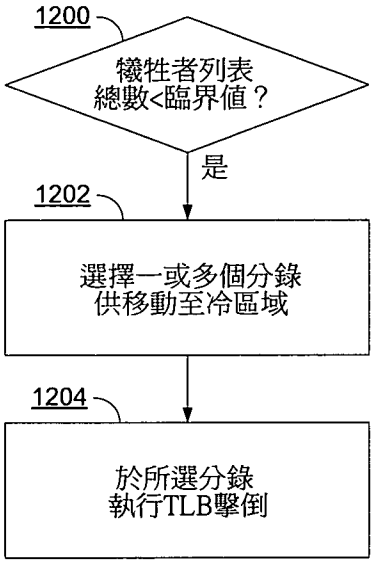
第 9 圖


1000
在針對熱→冷寫入襪性者的失誤配置hMPA用於LDT
A
1002
選擇一或多個記憶體位置用於襪性
B
1004
選擇一或多個襪性者用以移動至冷區域
C
1006
拷貝
hMPA → cMPA.
D
1008
以cMPA資訊更新PMH-TT
E
1010
以hMPA資訊更新熱自由列表
F

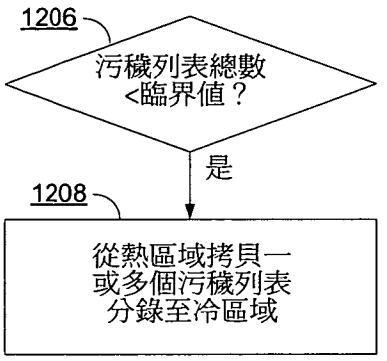
第 10 圖


1100
hMPA
PMH TT
(使用PPA檢索)
A
1102
PPA
LDT
(本地描述符表)
B
1104
hMPA
襪性者列表
C
1106
hMPA
污穢列表
1108
cMPA
冷自由列表
D
拷貝
hMPA → cMPA
E
F
1110
hMPA
熱自由列表

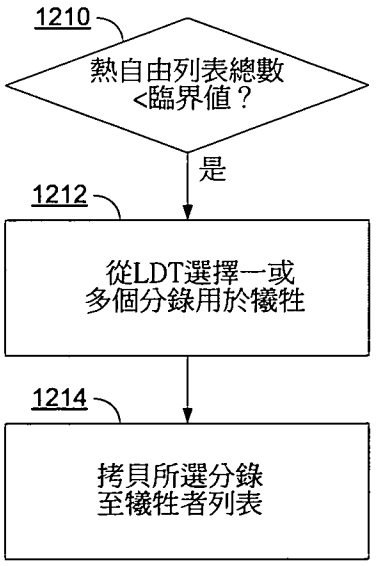
第 11 圖



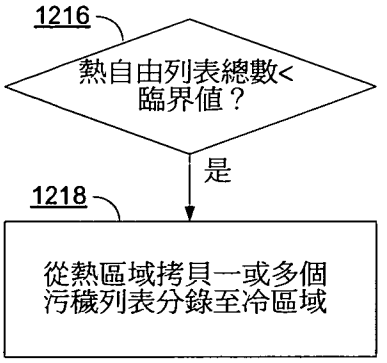
第 12A 圖



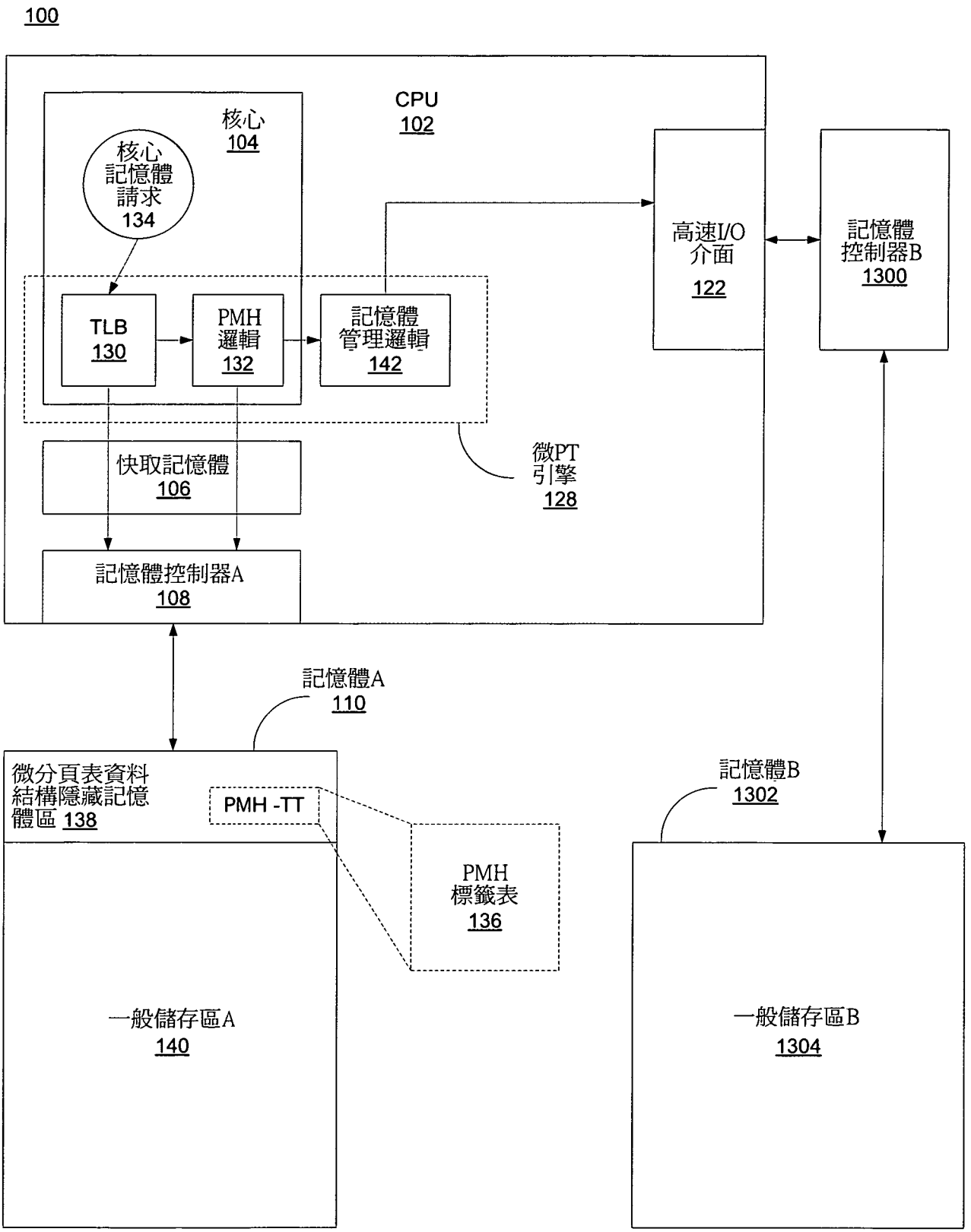
第 12B 圖



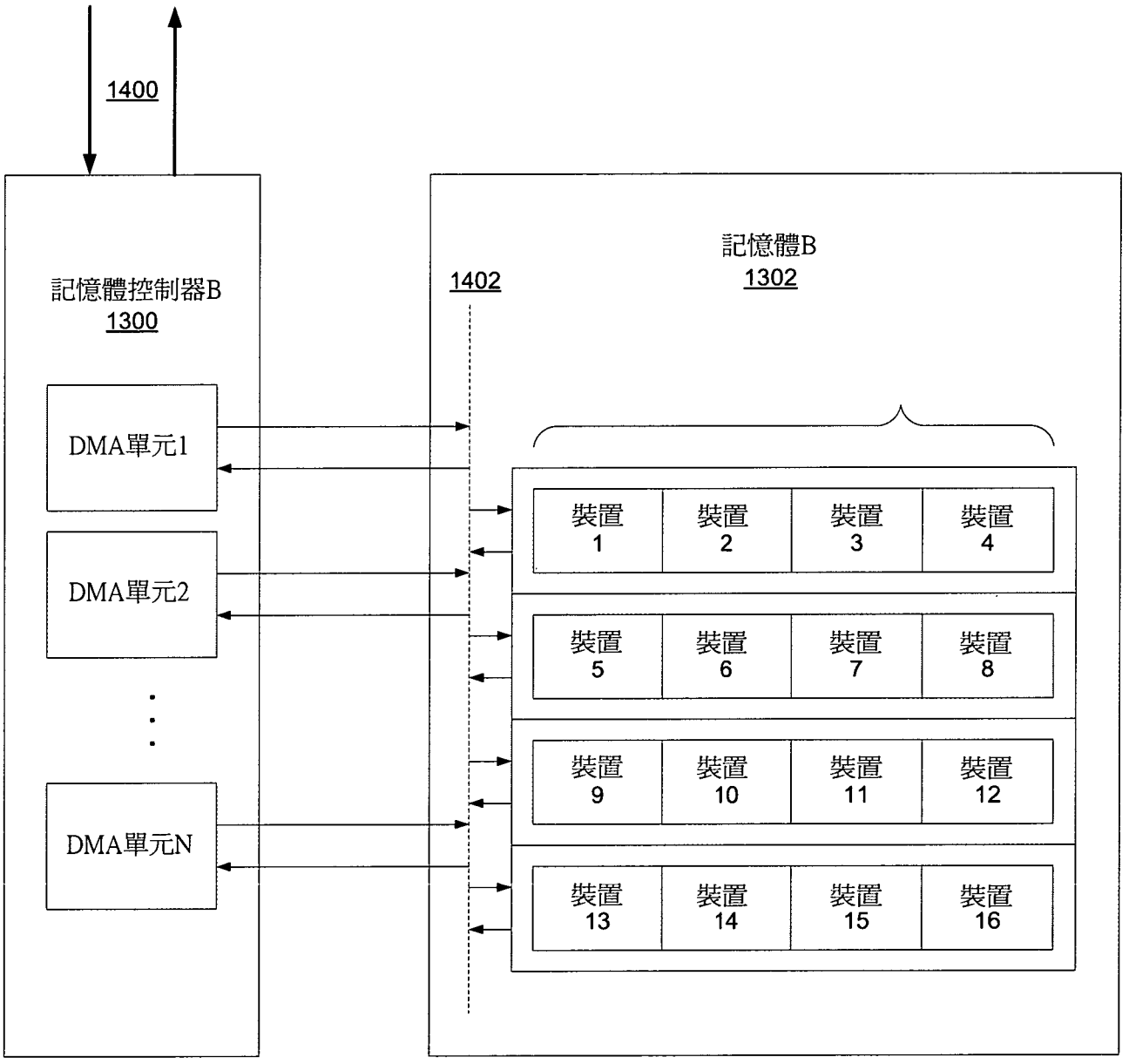
第 12C 圖



第 12D 圖


100
CPU
102
核心
104
核心
記憶體
請求
134
TLB
130
PMH
邏輯
132
記憶體
管理邏輯
142
高速I/O
介面
122
記憶體
控制器B
1300
微PT
引擎
128
快取記憶體
106
記憶體控制器A
108
記憶體A
110
微分頁表資料
結構隱藏記憶
體區 138
PMH -TT
PMH
標籤表
136
記憶體B
1302
一般儲存區A
140
一般儲存區B
1304

第 13 圖


1400
記憶體控制器B
1300
DMA單元1
DMA單元2
DMA單元N
1402
記憶體B
1302
裝置 1
裝置 2
裝置 3
裝置 4
裝置 5
裝置 6
裝置 7
裝置 8
裝置 9
裝置 10
裝置 11
裝置 12
裝置 13
裝置 14
裝置 15
裝置 16

第 14 圖