



(19)中華民國智慧財產局

(12)發明說明書公開本

(11)公開編號：TW 201327402 A1

(43)公開日：中華民國 102 (2013) 年 07 月 01 日

(21)申請案號：101133283

(22)申請日：中華民國 101 (2012) 年 09 月 12 日

(51)Int. Cl.：

G06F9/46 (2006.01)

G06F9/28 (2006.01)

G06F15/16 (2006.01)

(30)優先權：2011/12/20

日本

2011-279022

(71)申請人：富士通股份有限公司 (日本) FUJITSU LIMITED (JP)

日本

(72)發明人：鯉沼秀之 KOINUMA, HIDEYUKI (JP)；岡田誠之 OKADA, SEISHI (JP)；杉崎剛
SUGIZAKI, GO (JP)

(74)代理人：憚軼群；陳文郎

申請實體審查：有 申請專利範圍項數：10 項 圖式數：26 共 82 頁

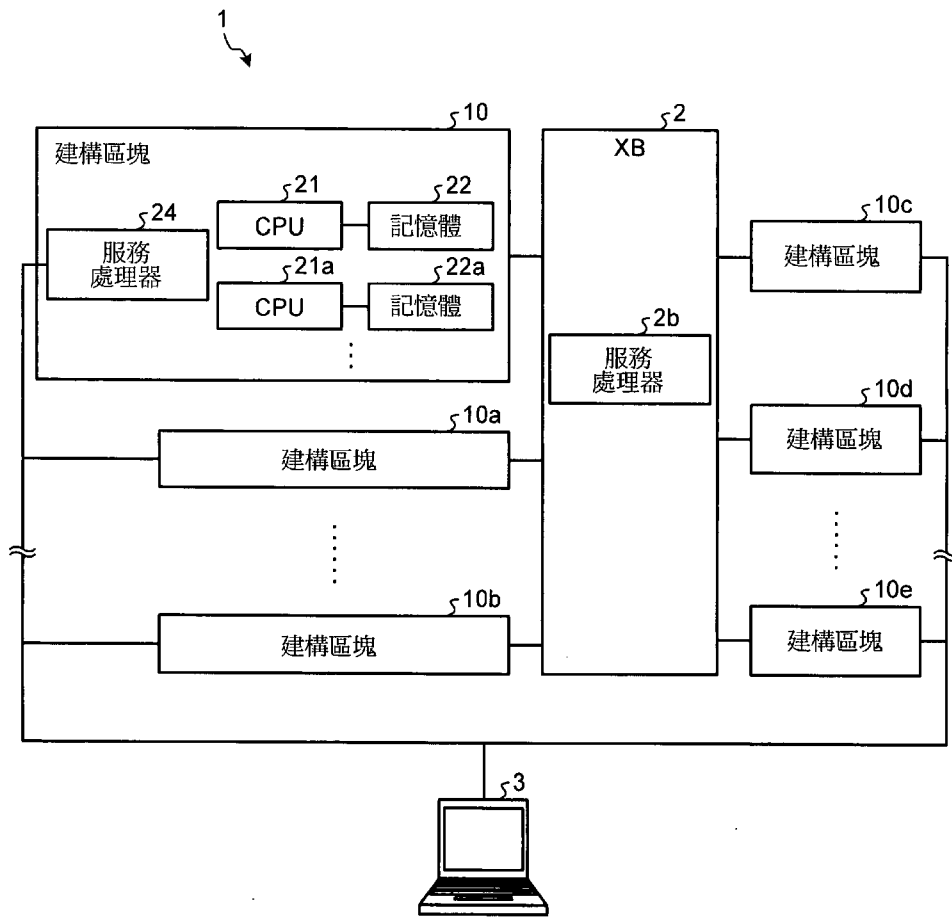
(54)名稱

資訊處理裝置及記憶體存取方法

INFORMATION PROCESSING APPARATUS AND MEMORY ACCESS METHOD

(57)摘要

一節點包括一第一轉換單元，其執行一邏輯位址與一實體位址間之轉換。該節點包括一第二轉換單元，執行該實體位址與用以識別含括在多個節點各自中的一處理器的處理器識別資訊間之轉換。該節點包括一發送單元，其發送包括該實體位址及該處理器識別資訊的傳輸資訊用以儲存由該實體位址所指示之一儲存區。該節點包括一本地決定單元，基於含括在由該接收單元所接收的該傳輸資料中的該實體位址，決定藉接收自另一個節點的該傳輸資料所指示的一存取是否為對一本地區之一存取或對一共用區之一存取。



- 1：資訊處理系統
- 2：縱橫式開關(XB)
- 2b：服務處理器
- 3：管理終端
- 10：建構區塊
- 10a-e：建構區塊
- 21：中央處理單元(CPU)
- 21a：中央處理單元(CPU)
- 22：記憶體
- 22a：記憶體
- 24：服務處理器



(19)中華民國智慧財產局

(12)發明說明書公開本

(11)公開編號：TW 201327402 A1

(43)公開日：中華民國 102 (2013) 年 07 月 01 日

(21)申請案號：101133283

(22)申請日：中華民國 101 (2012) 年 09 月 12 日

(51)Int. Cl.：

G06F9/46 (2006.01)

G06F9/28 (2006.01)

G06F15/16 (2006.01)

(30)優先權：2011/12/20

日本

2011-279022

(71)申請人：富士通股份有限公司 (日本) FUJITSU LIMITED (JP)

日本

(72)發明人：鯉沼秀之 KOINUMA, HIDEYUKI (JP)；岡田誠之 OKADA, SEISHI (JP)；杉崎剛
SUGIZAKI, GO (JP)

(74)代理人：憚軼群；陳文郎

申請實體審查：有 申請專利範圍項數：10 項 圖式數：26 共 82 頁

(54)名稱

資訊處理裝置及記憶體存取方法

INFORMATION PROCESSING APPARATUS AND MEMORY ACCESS METHOD

(57)摘要

一節點包括一第一轉換單元，其執行一邏輯位址與一實體位址間之轉換。該節點包括一第二轉換單元，執行該實體位址與用以識別含括在多個節點各自中的一處理器的處理器識別資訊間之轉換。該節點包括一發送單元，其發送包括該實體位址及該處理器識別資訊的傳輸資訊用以儲存由該實體位址所指示之一儲存區。該節點包括一本地決定單元，基於含括在由該接收單元所接收的該傳輸資料中的該實體位址，決定藉接收自另一個節點的該傳輸資料所指示的一存取是否為對一本地區之一存取或對一共用區之一存取。

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：101133783

※申請日：101-9-12

※IPC 分類：G06F 9/46 (2006.01)

一、發明名稱：(中文/英文)

G06F 9/58 (2006.01)

G06F 15/16 (2006.01)

資訊處理裝置及記憶體存取方法

INFORMATION PROCESSING APPARATUS AND MEMORY ACCESS METHOD

二、中文發明摘要：

一節點包括一第一轉換單元，其執行一邏輯位址與一實體位址間之轉換。該節點包括一第二轉換單元，執行該實體位址與用以識別含括在多個節點各自中的一處理器的處理器識別資訊間之轉換。該節點包括一發送單元，其發送包括該實體位址及該處理器識別資訊的傳輸資訊用以儲存由該實體位址所指示之一儲存區。該節點包括一本地決定單元，基於含括在由該接收單元所接收的該傳輸資料中的該實體位址，決定藉接收自另一個節點的該傳輸資料所指示的一存取是否為對一本地區之一存取或對一共用區之一存取。

三、英文發明摘要：

A node includes a first converting unit that performs conversion between a logical address and a physical address. The node includes a second converting unit that performs conversion between the physical address and processor identification information for identifying a processor included in a each of a plurality of nodes. The node includes a transmitting unit that transmits transmission data including the physical address and the processor identification information for accessing a storing area indicated by the physical address. The node includes a local determining unit that determines whether an access, indicated by the transmission data received from another nodes, is an access to a local area or an access to a shared area based on the physical address included in the transmission data received by the receiving unit.

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：

1...資訊處理系統

2...縱橫式開關(XB)

2b、24...服務處理器

3...管理終端

10、10a-e...建構區塊

21、21a...中央處理單元(CPU)

22、22a...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：

六、發明說明：

【發明所屬之技術領域】

發明領域

此處討論之實施例係有關於一種資訊處理裝置及一種記憶體之存取方法。

【先前技術】

發明背景

相關地，已知對稱性多重處理器(SMP)技術，其中多個算術處理單元共用一個主儲存裝置。至於應用SMP技術的資訊處理系統實例，已知一種資訊處理系統，其中多個節點各自包括一算術處理單元及一主儲存裝置彼此透過同一個匯流排相連結，及算術處理單元透過該匯流排共用各個主儲存裝置。

於此種資訊處理系統中，使用使用監聽系統維持藉各個節點的該算術處理單元快取資料的同調。但依據監聽系統，藉各個算術處理單元快取的資料之更新態係透過匯流排交換，據此，隨著節點數目的增加，匯流排變成瓶頸，因而記憶體存取效能低劣。

為了避免此種匯流排的瓶頸，非一致記憶體存取(NUMA)技術為已知，其中多個節點係使用互連體互連，該等節點的算術處理單元共用該等節點的主儲存裝置。

於應用此種NUMA技術的資訊處理系統中，各個節點的主儲存裝置之儲存區係獨特地對映至一共通實體位址空間。如此各個節點的算術單元識別一節點，於該處存在有

由一存取目標的實體位址所表示之一儲存區，及透過該互連體存取該經識別節點的主儲存裝置。

專利文獻1：日本公開專利案第2000-235558號

非專利文獻1：電腦架構：量化辦法，第二版，John L. Hennessy, David A. Patterson，§8.4。

此處，依據前述NUMA技術，並不維持由各個節點的算術處理單元所快取的資料之同調。如此，可考慮採用快取同調NUMA(ccNUMA)，其中含括有維持由各個節點的算術處理單元所快取的資料之同調的機構。

但於應用ccNUMA的一種資訊處理系統中，各個節點識別存在有一存取目標的一儲存區的一節點，據此，位址轉換須以高效率實施。此外，有些情況其中各個節點將該主儲存裝置劃分為只由該節點使用之一儲存區，及由其它節點一起共用的一儲存區。於此種情況下，各個節點需要有效地決定一儲存區是否為由其它節點所一起共用的一儲存區。

依據本發明之一面向，可有效地從事由各個算術處理單元所做的記憶體存取。

【發明內容】

發明概要

依據該等具體例之一態樣，一種資訊處理裝置包括多個節點，及互相連結該等多個節點的一互連體。該等多個節點各自包括一處理器一儲存單元、及執行一邏輯位址與一實體位址間之轉換的一第一轉換單元。該等多個節點各

自包括一第二轉換單元，其執行該實體位址與用以識別含括在該等多個節點各自中的一處理器之處理器識別資訊間之轉換。該等多個節點各自包括一發送單元，其發送包括該實體位址及用以存取由該實體位址所指示的一儲存區之該處理器識別資訊的傳輸資料。該等多個節點各自包括一接收單元，其係接收從另一個節點透過該互連體發送的該傳輸資料。該等多個節點各自包括一本地決定單元，其基於含括在由該接收單元所接收的該傳輸資料的該實體位址，決定一存取是否為一存取至從含括該儲存單元的該節點所能存取的該儲存單元之一本地區，或為一存取至從該等多個節點所能存取的該儲存單元之一共用區。

圖式簡單說明

圖1為略圖例示說明依據第一實施例一資訊處理系統之一實例；

圖2為略圖例示說明依據第一實施例一建構區塊的功能組態；

圖3為略圖例示說明依據第一實施例分配給該等建構區塊的記憶體之實體位址之範圍；

圖4為略圖例示說明依據第一實施例由該資訊處理系統分配給各個記憶體之實體位址；

圖5為第一略圖例示說明分配實體位址中之一變化；

圖6為第二略圖例示說明分配實體位址中之一變化；

圖7為略圖例示說明依據第一實施例一CPU的功能組態；

圖8為略圖例示說明依據第一實施例儲存在節點對映

圖中之資訊之一實例；

圖9為第一略圖例示說明儲存在節點對映圖中之資訊之一變化實例；

圖10為第二略圖例示說明儲存在節點對映圖中之資訊之一變化實例；

圖11A為略圖例示說明一快取標籤之一實例；

圖11B為略圖例示說明依據第一實施例由CPU所發送之一封包；

圖12為略圖例示說明依據第一實施例使用CPU發送一要求之處理實例；

圖13為略圖例示說明依據第一實施例當CPU接收一封包時所執行的處理實例；

圖14為流程圖例示說明一節點對映圖設定處理之處理流程；

圖15為流程圖例示說明一共用區控制處理之處理流程；

圖16為流程圖例示說明一共用記憶體分配處理；

圖17為流程圖例示說明一共用記憶體附接處理；

圖18為流程圖例示說明使用一共用記憶體的應用程式之處理程序；

圖19為流程圖例示說明在節點間脫離一共用記憶體之處理程序；

圖20為流程圖例示說明在節點間釋放一共用記憶體之處理程序；

圖21為流程圖例示說明一要求簽發處理之處理流程；

圖22為流程圖例示說明當接收一要求時執行的一處理流程；

圖23為流程圖例示說明當CPU接收到回覆時執行的一處理流程；

圖24為略圖例示說明依據第二實施例之一資訊處理系統；

圖25為略圖例示說明區劃之一實例；

圖26A為略圖例示說明由區劃#A之CPU所儲存的一節點對映圖之一實例；

圖26B為略圖例示說明表示區劃#A之一節點對映圖之一實例；及

圖26C為略圖例示說明表示區劃#B之一節點對映圖之一實例。

【實施方式】

較佳實施例之詳細說明

將參考附圖解說較佳實施例。

[a] 第一實施例

首先，在描述依據本案之實施例前，將敘述於相關資訊處理系統中的特定問題實例。舉例言之，相關資訊處理系統將藉一中央處理單元(CPU)輸出的用以存取一共用記憶體區的一邏輯位址轉換成一共用記憶體空間位址。然後，藉將該共用記憶體空間位址轉換成一實體位址，該資訊處理系統識別一儲存區，該儲存區為該CPU的存取目標。

然而依據下述技術其中如前述，該邏輯位址被轉成共

用記憶體空間位址，及轉換後的共用記憶體空間位址係被轉成實體位址，轉換該位址需要的硬體資源數量龐大。此外，依據下述技術其中如前述，該邏輯位址被轉成共用記憶體空間位址，及轉換後的共用記憶體空間位址係被轉成實體位址，轉換該位址需要的時間增加。

此外，當CPU快取共用記憶體空間的資料時，相關資訊處理系統藉將快取記憶體資訊傳輸給全部CPU而維持同調性。但依據如前述快取記憶體資訊傳輸給全部CPU的技術中，出現瓶頸，記憶體存取效能劣化。此外，在相關資訊處理系統中，當安裝的CPU數目增加時，匯流排資訊通量係與CPU數目成比例地增加，因此，出現瓶頸，記憶體存取效能劣化。

又復，舉例言之，一節點儲存核心資料及使用者資料於本地一區，該區只由該節點存取。據此，為了獲得儲存於該本地區的資料之安全性，且提升對軟體錯誤之抗性，各個節點須決定屬於存取目標的一儲存區是否係為可由其它節點所可存取的一共用記憶體區或一本地記憶體區。

據此，在相關資訊處理系統中，儲存於本地區的資料係經組配成可被快取，而儲存於共用區的資料係經組配成不可被快取。但依據前述將儲存於共用區的資料組配成不可被快取的技術，存取記憶體的延遲時間增加。此外，每次從任何其它節點存取記憶體時決定一存取目標是否為共用區或本地區之情況下，用於決定處理的電路大小增加，存取延遲時間增加。

此外，在相關資訊處理系統中，每次一節點存取含括在任何其它節點的記憶體時，該節點需要一特定通道裝置或直接記憶體存取(DMA)架構引擎程式的執行，據此，記憶體存取效能低劣。又復，在相關資訊處理系統中，含括在記憶體中欲組配為共用區的該儲存區面積係被固定設定。據此，例如在相關資訊處理系統中，藉增加一節點而未停止該系統，則共用區不會增加。

又復，在相關資訊處理系統中，增加用以經由通道或DMA路徑而執行記憶體存取的硬體。據此，在相關資訊處理系統中，安裝的硬體係與其中記憶體在節點不被共享的一種系統的硬體有顯著差異。結果，在相關資訊處理系統中，於記憶體係在節點間共享的情況下，程式諸如作業系統(OS)須做顯著改變。

於後文呈示之詳細說明部分中，解決前述問題的資訊處理系統之一實例將描述為第一實施例。首先，將參考圖1說明資訊處理系統之該組態實例。圖1為一略圖例示說明依據第一實施例資訊處理系統之一實例。於圖1例示說明之實例中，資訊處理系統1包括一縱橫式開關(XB) 2及多個建構區塊10至10e。建構區塊10至10e係經由一管理網絡而連結至一管理終端3。此外，XB 2包括一服務處理器2b。

建構區塊10包括多個CPU 21至21c、多個記憶體22至22c、及一服務處理器24。此處，其它建構區塊10至10e具有與建構區塊10相同的組態，而將不再呈示其詳細說明。又復，於圖1例示說明之實例中，將不顯示CPU 21b及21b

及記憶體22b及22c。

XB 2為縱橫式開關，係將建構區塊10至10e彼此連結。此外，含括於XB 2的服務處理器2b乃係管理含括於建構區塊10至10e的服務處理器之一服務處理器，換言之，用作為主站的服務處理器。此外，管理終端3為一終端，該終端係透過管理網絡而設定或控制含括於建構區塊10至10e的服務處理器。此外，於其中少數節點為互連的小型組態之情況下，建構區塊可直接地彼此連結而無需透過XB 2。

建構區塊10至10e獨立地操作作業系統。換言之，由建構區塊10至10e所操作的作業系統係以區劃操作，而該等區劃針對該等建構區塊為彼此相異。此處，從該操作的作業系統觀看，一區劃表示當同一個作業系統操作時作為一個系統操作的一組建構區塊。

舉例言之，建構區塊10及10a操作為區劃#A，而建構區塊10b至10d操作為區劃#B。於此種狀態下，藉建構區塊10操作的OS識別建構區塊10及10a操作為一個系統，而藉建構區塊10b操作的OS識別建構區塊10b至10d操作為一個系統。

其次，將參考圖2描述建構區塊的組態。圖2為一略圖例示說明依據第一實施例建構區塊的功能組態。於圖2例示說明之實例中，建構區塊10包括一節點20、該服務處理器24、XB連結單元27及27a、及一周邊組件互連體快速(PCIe)連結單元28。

該節點20包括多個CPU 21至21c、多個記憶體22至22c、及一通訊單元23。此外，該服務處理器24包括一控制

單元25及一通訊單元26。於圖2例示說明之實例中，CPU 21至21c係直接連結彼此且係連結至通訊單元23。記憶體22至22c係連結CPU 21至21c。

CPU 21至21c係連結至XB連結單元27或XB連結單元27a。XB連結單元27及27a可為相同XB連結單元。此外，CPU 21至21c係連結至PCIe連結單元28。又復，通訊單元23係連結至含括於服務處理器24中的通訊單元26。該控制單元25、通訊單元26、通訊單元23、與CPU 21至21c例如係透過聯合測試行動小組(JTAG)或內置積體電路(I2C)互連。

舉例言之，於圖2例示說明之實例中，CPU 21至21c為執行算術處理的算術處理單元。此外，獨立的記憶體22至22c係連結至CPU 21至21c。又復，CPU 21至21c運用記憶體22至22c及含括於其它建構區塊10a至10e的記憶體作為共用記憶體。此外，容後詳述，CPU 21至21c具有一節點對映圖，其中一實體位址係與一識別身分(CPUID)彼此相連結，該CPUID係為與該實體位址相連結的該CPU的識別符。

舉例言之，當與作為存取目標的該實體位址相連結的CPUID係表示含括在節點20以外的一節點中的一CPU時，CPU 21透過XB連結單元27及XB 2而傳輸一記憶體存取要求給其它節點。當與作為存取目標的該實體位址相連結的CPUID係表示CPU 21至21c中之一者時，CPU 21透過該等CPU間之直接連結而傳輸一記憶體存取要求。換言之，當與作為存取目標的該實體位址相連結的CPUID係為CPU 21以外的一CPU，且表示在與CPU21的該節點相同節點20呈

示的一CPU時，CPU 21透過該等CPU間之直接連結而傳輸一記憶體存取要求。

此外，於CPU 21接收來自另一節點的針對與其相連結的一記憶體之一要求之情況下，CPU 21從連結其上的記憶體22讀出資料作為要求目標，且傳輸該資料給一要求源。

此外，於執行的應用程式要求配置一共用記憶體之情況下，CPU 21至21c具有配置一共用記憶體的功能，該共用記憶體係藉彼此通訊而由一應用程式使用。此外，CPU 21至21c各自係假設具有執行與相關CPU相同的處理之功能，諸如使用TLB執行位址轉換，及當發生TLB遺漏例外時執行捕獲處理。

記憶體22至22c乃由包括在該資訊處理系統1中的全部CPU所共享的記憶體。此外，於資訊處理系統1中，其中建構區塊10至10e的服務處理器係對映至相同實體位址空間的該等實體位址係配置給包括於全部建構區塊10至10e中的記憶體。換言之，具有彼此不重疊的數值之該等實體位址係配置給包括於資訊處理系統1中的全部記憶體。

此外，記憶體22至22c中之各者將該儲存區之一部分組配為由資訊處理系統1的全部CPU所使用的一共用區，而將其它部分組配為CPU 21至21c存取的一本地區，其中儲存核心資料或使用者資料。此外，於記憶體22至22c中，在由資訊處理系統1所運用的實體位址空間，其中位在一特定位置的一位元具有相同值範圍內的該等實體位址係配置給共用區。此外，於記憶體22至22c中，其中位在一特定位置的一

位元具有與配置給共用區的該等實體位址不同值範圍內的該等實體位址係配置給本地區。

舉例言之，於記憶體22至22c中，其中第46個位元為「0」的各個實體位址係配置給本地區。此外，其中第46個位元為「1」的各個實體位址係配置給共用區。至於更進一步細節實例，含括於「0」至「0x63ff ffff ffff」的實體位址係配置給在該實體位址空間內記憶體22至22c的本地區。含括於「0x6400 000 0000」至「0x1 27ff ffff ffff」的實體位址係配置給在該實體位址空間內記憶體22至22c的共用區。

此外，於資訊處理系統1中，含括於互異範圍內的實體位址係配置給針對建構區塊10至10e的記憶體。後文中，將參考附圖描述於資訊處理系統1中配置給針對建構區塊10至10e的記憶體的實體位址之範圍。

圖1為一略圖例示說明依據第一實施例，配置給建構區塊的記憶體的實體位址之範圍。於圖3例示說明之實例中，各個建構區塊係表示為一個建構區塊(BB)。此處，BB#0表示建構區塊10，BB#1表示建構區塊10a，及BB#15表示建構區塊10e。換言之，於圖3例示說明之實例中，假設資訊處理系統1具有16個建構區塊。

此外，於圖3例示說明之實例中，高達4兆位元組(TB)的記憶體係假設可安裝在各個建構區塊上。於後文呈示之描述中，為了簡化一記憶體位址的表示型態，例如「 2^{42} 」的位址係表示為「4 TB」的位址。

於圖3例示說明之實例中，於含括於建構區塊10的記憶

體22至22c中，含括於實體位址空間內從位址「0」至「4 TB-1」之範圍的實體位址係配置給本地區。此外，於含括於建構區塊10的記憶體22至22c中，含括於實體位址空間內從位址「64 TB」至「68 TB-1」之範圍的實體位址係配置給共用區。

此外，於含括於建構區塊10a的記憶體中，含括於實體位址空間內從位址「4 TB」至「8 TB-1」之範圍的實體位址係配置給本地區。此外，於含括於建構區塊10a的記憶體中，含括於實體位址空間內從位址「68 TB」至「72 TB-1」之範圍的實體位址係配置給共用區。

又復，於含括於建構區塊10a的記憶體中，含括於實體位址空間內從位址「60 TB」至「64 TB-1」之範圍的實體位址係配置給本地區。此外，於含括於建構區塊10a的記憶體中，含括於實體位址空間內從位址「124 TB」至「128 TB-1」之範圍的實體位址係配置給共用區。

結果，如圖4之例示說明，於資訊處理系統1中，該實體位址空間係配置給含括於建構區塊10至10e中的全部記憶體。圖4為一略圖例示說明依據第一實施例，藉該資訊處理系統配置給該等記憶體之實體位址。

更明確言之，於圖4例示說明之實例中，資訊處理系統1組配在「0」至「256 TB-1」之實體位址中從「0」至「64 TB-1」之範圍作為欲配置給該本地區的實體位址。此外，資訊處理系統1組配從「64 TB」至「128 TB-1」之範圍作為欲配置給該共用區的實體位址。

換言之，當一最低有效位元係組配為位元0時，資訊處理系統1配置其中位元46為「0」之一範圍給該本地區，及配置其中位元46為「1」之一範圍給該共用區。此外，資訊處理系統1運用從位址「128 TB」至「256 TB-1」之一範圍作為I/O空間。

此外，於圖3及4例示說明之實例僅為舉例說明，資訊處理系統1可運用其它配置方法。後文中，將參考附圖說明藉資訊處理系統1所執行的實體位址配置之變化實例。

圖5為第一略圖例示說明配置一實體位址的變化例。於圖5例示說明之實例中，於含括於建構區塊10至10e的記憶體中，從位址「0」至「4 TB-1」之範圍的實體位址係配置給本地區。此外，於含括於建構區塊10的記憶體22中，含括於從位址「4 TB」至「8 TB-1」之範圍的實體位址係配置給共用區。

此外，於圖5例示說明之實例中，於含括於建構區塊10a的記憶體中，含括於從位址「8 TB」至「12 TB-1」之範圍的實體位址係配置給共用區。此外，於圖5例示說明之實例中，於含括於建構區塊10e的記憶體中，含括於從位址「64 TB」至「68 TB-1」之範圍的實體位址係配置給共用區。

結果，於圖5例示說明之實例中，資訊處理系統1配置於實體位址空間內從位址「0」至「4 TB-1」之範圍的實體位址給本地區，及配置從位址「4 TB」至「128 TB-1」之範圍的實體位址給共用區。此外，於圖5例示說明之實例中，資訊處理系統1使用從位址「128 TB」至「256 TB-1」

之範圍作為I/O空間。換言之，當最低有效位元係組配為位元0時，資訊處理系統1配置其中位元42為「0」的一範圍給該本地區，及配置其中位元42為「1」的一範圍給該共用區。

圖6為第二略圖例示說明配置實體位址的變化例。於圖6例示說明之實例中，於含括於建構區塊10至10e的記憶體中，含括於從位址「0」至「4 TB-1」之範圍的實體位址係保留為I/O空間。此外，於圖6例示說明之實例中，於含括於建構區塊10至10e的記憶體中，含括於從位址「4 TB」至「8 TB-1」之範圍的實體位址係配置給本地區。

此外，於圖6例示說明之實例中，於含括於建構區塊10a的記憶體22至22c中，含括於從位址「8 TB」至「12 TB-1」之範圍的實體位址係配置給共用區。又復，於圖6例示說明之實例中，於含括於建構區塊10a的記憶體中，含括於從位址「12 TB」至「16 TB-1」之範圍的實體位址係配置給共用區。此外，於圖6例示說明之實例中，於含括於建構區塊10e的記憶體中，含括於從位址「68 TB」至「72 TB-1」之範圍的實體位址係配置給共用區。

結果，於圖6例示說明之實例中，資訊處理系統1配置於實體位址空間內從位址「0」至「4 TB-1」之範圍的實體位址作為I/O空間，及配置從位址「4 TB」至「8 TB-1」之範圍的實體位址給本地區。此外，於圖5例示說明之實例中，資訊處理系統1使用從位址「8 TB」至「256 TB-1」之範圍給共用區。換言之，當最低有效位元係組配為位元0時，資訊處理系統1配置其中位元43為「0」的一範圍給該

本地區，及配置其中位元43為「1」的一範圍給該共用區。

回頭參考圖2，控制單元25控制建構區塊10。舉例言之，控制單元25執行建構區塊10的功率管理，監視且控制出現在建構區塊10內部的異常等。此外，控制單元25係透過管理網絡而連結至該等服務處理器的管理終端3及控制單元，該等控制單元係含括於其它建構區塊10至10e且可執行由管理終端3所指示的控制，且執行與建構區塊10至10e相連結的控制。此外，控制單元25可與在CPU 21至21c上運作的作業系統通訊。

此外，於第一實施例中，雖然含括於建構區塊10至10e的服務處理器係透過管理網絡而彼此連結，但此一實施例並非限制性。例如服務處理器可透過連結建構區塊10至10e彼此的XB而彼此通訊。

又復，控制單元25係透過通訊單元26及通訊單元23而存取CPU 21至21c。然後，容後詳述，控制單元25藉由更新存在建構區塊10至10e上的節點對映圖等而執行建構區塊中CPU的控制。

此外，通訊單元23透過含括於服務處理器24的通訊單元26而傳遞從控制單元25發射的控制信號給CPU 21至21c。通訊單元26而傳遞從控制單元25發射的控制信號給含括於節點20的通訊單元23。XB連結單元27及27a連結CPU 21至21c至XB 2，且中繼含括於建構區塊10至10e內CPU間的通訊。PCIe連結單元28中繼CPU 21至21c之接取至輸出入(I/O)裝置。

其次，將參考圖7描述CPU 21至21c的功能組態。圖7為一略圖例示說明依據第一實施例，CPU的功能組態。此處因CPU 21a至21c具有與CPU 21之功能相同的功能，將不再呈示其描述。於圖7例示說明之實例中，將不顯示連結服務處理器24與CPU 21彼此的通訊單元23及26。

於圖7例示說明之實例中，CPU 21包括一算術處理單元30、一路由器40、一記憶體存取單元41、及一PCIe控制單元42。算術處理單元30包括一算術單元31、一層級1 (L1) 快取記憶體32、一L2快取記憶體33、一節點對映圖34、一位址轉換單元35、一快取目錄管理單元36、及一封包控制單元37。封包控制單元37包括一要求產生單元38及一要求接收單元39。PCIe控制單元42包括一要求產生單元43及一PCIe匯流排控制單元44。

首先說明含括於算術處理單元30的該節點對映圖34。於該節點對映圖34中，連結至具有一實體位址表示的一儲存區的一記憶體的CPU，該實體位址及該CPU之CUID係彼此連結儲存。後文中，將參考附圖描述儲存於節點對映圖34的資訊實例。

圖8為一略圖例示說明依據第一實施例，儲存於節點對映圖的資訊實例。於圖8例示說明之實例中，於節點對映圖34中，一位址、有效性、一節點ID、一CUID係彼此相連結儲存。此處，在各個分錄的位址儲存資訊，表示含括多個為連續的位址之一位址區。

舉例言之，資訊處理系統1將一實體位址空間分割成為

具有相等大小的位址區，分配給全部記憶體，且分配識別符諸如#0、#1、#2給該等位址區。然後，資訊處理系統1將表示該等位址區的識別符儲存在包括於節點對映圖34的各個分錄的位址。

此外，於各個分錄的有效性中，儲存一有效性位元，其指示由該實體位址所表示之一儲存區是否為可存取。舉例言之，由該實體位址所表示之一儲存區為由該等CPU所共用的一共用區之情況下，一有效性位元(例如「1」)指示儲存許可存取。

節點ID為一識別符，表示存在有該實體位址所被分配的記憶體之一節點。CPUID為一識別符，表示一CPU連結至該實體位址所被分配的記憶體。換言之，於節點對映圖34中儲存有資訊，表示一CPU連結至一記憶體，該記憶體的實體位址為作為存取目標之一實體位址。

例如，於圖8例示說明之實例中，於節點對映圖34中，表示具有「#0」識別符一位址區係存在於一節點，該節點的節點ID為「0」，具有「0」CPUID的一CPU係對該節點作存取。此外，於節點對映圖34中，表示具有「#1」識別符一位址區係存在於一節點，該節點的節點ID為「0」，具有「1」CPUID的一CPU係對該節點作存取。又復，於節點對映圖34中，因具有「#2」識別符一位址區乃將不藉CPU 21進行存取的一位址區，或不被對映的一位址區，表示節點ID及CPUID尚未設定於其上。

當節點對映圖34可表示作為存取目標的該實體位址連

結的一CPU時，資訊可以本實施例之形式以外的任意形式登錄。後文中，將參考圖9及10描述節點對映圖34的變化例。

圖9為第一略圖例示說明儲存在節點對映圖中之資訊之變化例。於圖9例示說明之實例中，節點對映圖34儲存各個分錄與彼此相聯結的有效性、一起始位址、一位址罩、一節點ID、及一CPUID。此處，於該起始位址中，儲存包括於該位址區的全部實體位址之一最小實體位址。

於該位址罩中，儲存表示由該CPU所管理的該實體位址之範圍之一位址罩。舉例言之，當一分錄的位址罩為「0xffff ffff ffff 0000」時，表示重合該相同分錄的起始位址，在上48位元一位址區係由相同分錄的CPUID所表示之一CPU所管理。

例如，於圖9例示說明之實例中，節點對映圖34表示從一位址「0x00000」至藉使用一位址罩「0x3fff」遮罩該位址所得的一位址之一範圍作為一第一分錄，換言之，直至「0x03fff」的一範圍係為一個位址區。此外，節點對映圖34表示從「0x00000」至「0x03fff」之一位址區係存在於由「0」節點ID所表示的一節點，及該位址區係藉具有「0」CPUID之一CPU存取。

同理，節點對映圖34表示從「0x10000」至「0x13fff」之一位址區係存在於由「1」節點ID所表示的一節點，及該位址區係藉具有「4」CPUID之一CPU存取。此外，節點對映圖34表示從「0x14000」至「0x17fff」之一位址區係存在於由「1」節點ID所表示的該節點，及該位址區係藉具有「5」

CPUID 之一 CPU 存取。又復，節點對映圖 34 表示從「0x20000」至「0x21fff」之一位址區係存在於由「2」節點ID所表示的一節點，及該位址區係藉具有「8」CPUID之一CPU存取。

當如圖9之例示說明一位址區係藉在節點對映圖34中的一一起始位址及一位址記號表示時，基於邏輯和與邏輯積的組合，可決定實體位址是否含括於各個位址區，據此而容易地組配電路。

圖10為第二略圖例示說明儲存在節點對映圖中之資訊之變化例。於圖10例示說明之實例中，一分錄係儲存在節點對映圖34帶有彼此相聯結的有效性、一起始位址、一長度、一節點ID、及一CPUID。此處，長度為用以設定一位址區的大小之資訊。

舉例言之，於起始位址為「0x12 0000」及長度為「0x1 ffff」之情況下，由該相同分錄的該CPUID所表示之一CPU分配從「0x12 0000」至「0x13 ffff」的實體位址給一接受管理的記憶體。

例如，於圖10例示說明之實例中，節點對映圖34表示從「0x00000」之一位址其中長度係含括於「0x3fff」的範圍，亦即直至「0x03fff」之一範圍乃一個位址區，作為一第一分錄。此外，節點對映圖34表示從「0x00000」至「0x03fff」之該位址區係存在於由「0」節點ID所表示的一節點，及該位址區係藉具有「0」CPUID之一CPU存取。

同理，節點對映圖34表示從「0x10000」至「0x13fff」

之一位址區係存在於由「1」節點ID所表示的一節點，及該位址區係藉具有「4」CUID之一CPU存取。此外，節點對映圖34表示從「0x14000」至「0x17ff」之一位址區係存在於由「1」節點ID所表示的該節點，及該位址區係藉具有「5」CUID之一CPU存取。又復，節點對映圖34表示從「0x20000」至「0x202ef」之一位址區係存在於由「2」節點ID所表示的一節點，及該位址區係藉具有「8」CUID之一CPU存取。

當如圖10之例示說明一位址區係藉在節點對映圖34中的一起始位址及一長度表示時，各個位址區的長度可以彈性方式設定。換言之，當該位址區係在節點對映圖34以一起始位址及一位址罩表示時，指定於一範圍之一存取區，其中從該最低有效位元(LSB)算起多個1為連續。另一方面，當各個位址區係藉一起始位址及一長度表示時，各個位址區的長度可任意地設定。

回頭參考圖7，算術單元31乃一算術裝置的核心，該核心係藉從事算術處理程序而執行作業系統或應用程式。此外，於從事資料讀取之情況下，算術單元31輸出該儲存區之一邏輯位址，其中作為讀取目標的資料係儲存至該位址轉換單元35。

L1快取記憶體32為一快取記憶體儲存裝置，該裝置暫時性儲存資料或目錄中常用的資料。雖然類似L1快取記憶體32，L2快取記憶體33暫時性儲存資料或目錄中常用的資料，但L2快取記憶體33具有比L1快取記憶體32更大的儲存

容量，及比L1快取記憶體32更低的讀/寫速度。此處，一目錄為一資訊，該目錄表示具有快取資料儲存在記憶體22的各個儲存區或快取資料的更新狀態之一CPU。

位址轉換單元35運用轉譯後備緩衝器(TLB)，將由該算術單元31所輸出之一邏輯位址轉換成一實體位址。舉例言之，位址轉換單元35包括一TLB，該TLB儲存其中一邏輯位址與一實體位址彼此相聯結的一分錄，且輸出從該算術單元31所獲得的與該邏輯位址相聯結而儲存的該實體位址給快取目錄管理單元36。此外，當發生TLB遺漏時，位址轉換單元35引發捕獲處理，結果，於TLB中發生TLB遺漏該處登錄實體位址與邏輯位址之關係。

此外，當來自由CPU 21所執行的一應用程式要求該共享記憶體的分配時，位址轉換單元35從事下列處理。位址轉換單元35在該TLB中設定一分錄，其中在存取由CPU 21至21c所共用的一共用區時，由一應用程式所使用的一邏輯位址係與分配給該共用區之一範圍中之一實體位址彼此相聯結。

另一方面，當從一應用程式或一OS要求一本地區的配置時，位址轉換單元35從事下列處理。位址轉換單元35在該TLB中設定一分錄，其中在存取由該CPU 21所專用的一本地區時由一應用程式所使用或藉在CPU 21上運作的作業系統所使用的一邏輯位址係與分配給該共用區之一實體位址彼此相聯結。

快取目錄管理單元36管理快取記憶體資料及目錄。更

明確言之，快取目錄管理單元36從位址轉換單元35獲得一實體位址，該實體位址係藉轉換由算術單元31所輸出的該邏輯位址而獲得。

於該實體位址係從位址轉換單元35所獲得之情況下，快取目錄管理單元36查核一目錄，來查核由該實體位址表示的資料狀態是否為正常。當由該實體位址所表示的資料係快取在L1快取記憶體32或L2快取記憶體33時，快取資料係輸出給算術單元31。

另一方面，當由該實體位址所表示的資料係不快取在L1快取記憶體32或L2快取記憶體33時，快取目錄管理單元36決定由該實體位址所表示的儲存區是否存在於記憶體22。當由該實體位址所表示的儲存區係不存在於記憶體22時，快取目錄管理單元36參考該節點對映圖34。

此外，快取目錄管理單元36識別一範圍的一分錄，該範圍係包括藉參考該節點對映圖34所獲得的該實體位址。然後，快取目錄管理單元36決定所識別的分錄之CPUID是否為該CPU 21的CPUID。其後，當所識別的分錄之CPUID係為該CPU 21的CPUID時，快取目錄管理單元36輸出實體位址給記憶體存取單元41。

此外，當所識別的分錄之CPUID係非為該CPU 21的CPUID時，快取目錄管理單元36執行下列處理。快取目錄管理單元36獲得所識別的分錄之CPUID及節點ID。然後，快取目錄管理單元36將所獲得的CPUID及實體位址輸出給封包控制單元37。

當儲存在藉輸出實體位址所表示的該儲存區的資料係從記憶體存取單元41或封包控制單元37獲得時，快取目錄管理單元36將所獲得的資料儲存在L1快取記憶體32及L2快取記憶體33。然後，快取目錄管理單元36將快取在L1快取記憶體32的資料輸出給算術單元31。

此外，當一實體位址係從封包控制單元37獲得時，換言之，當獲得作為來自另一CPU的記憶體存取要求的目標之一實體位址時，快取目錄管理單元36執行下列處理。快取目錄管理單元36基於該所獲得的實體位址之位在一預定位置的一位元係為「0」或為「1」而決定該所獲得的實體位址是否為分配給該本地區之一實體位址。

舉例言之，於圖3及4例示說明之該等範圍的實體位址係分配給資訊處理系統1的記憶體之情況下，當最低有效位元為位元0時，快取目錄管理單元36決定位元46是否為「0」或為「1」。當位元46為「0」時，快取目錄管理單元36決定該所獲得的實體位址乃分配給該本地區的一實體位址。於此種狀態下，快取目錄管理單元36指示封包控制單元37發射一否定回應(存取錯誤)給該要求來源。

另一方面，當位元46為「1」時，快取目錄管理單元36決定該所獲得的實體位址乃分配給該共用區的一實體位址。於此種狀態下，快取目錄管理單元36獲得儲存在由該所獲得的實體位址所表示的該儲存區內的資料，輸出所獲得的資料給封包控制單元37，及指示封包控制單元37傳輸該資料給該要求來源。

此外，於儲存在記憶體22的資料係欲被存取之情況下，快取目錄管理單元36從事一項處理，該處理維持儲存在由該實體位址所表示的該儲存區內的資料與所快取的資料間之同調。舉例言之，快取目錄管理單元36參照表示該快取資料狀態之一快取標籤及針對各個快取分錄之一目錄。此外，快取目錄管理單元36基於該快取標籤及該目錄，從事維持快取同調之處理及記憶體存取處理。

此處，圖11A為一略圖例示說明該快取標籤之一實例。於圖11A例示說明之實例中，快取標籤包括一退化旗標、一錯誤檢查及校正(ECC)查核位元、一指令提取(IF)/操作代碼(opcode)、一L1快取態、一L2快取態、及位址資訊(AA)。

此處，退化旗標為表示退化與否的一快取行退化資訊。ECC查核位元為對該冗餘添加的一查核位元。IF/操作代碼為顯示該資料是否為一指令的資訊。

此外，AA為位址資訊，詳言之，該實體位址之一訊框位址係儲存於其中。L1快取態及L2快取態為表示儲存在L1快取記憶體32及L2快取記憶體33的資料狀態。

舉例言之，表示「修正(M)」、「互斥(E)」、「共用(S)」及「無效(I)」中之一者一位元係以L1快取態及L2快取態儲存。此處，「修正」表示其中任一個CPU快取資料而所快取的資料已經更新之一狀態。此外，於該所快取的資料狀態為「修正」之情況下須執行回寫。

「互斥」表示一狀態，其中任一個且只有一個CPU擁有且快取資料，且所快取的資料尚未經更新。「共用」表示

多個CPU快取資料，且所快取的資料尚未經更新的狀態。此外，「無效」表示其中該快取記憶體狀態尚未經登錄之一狀態。

該目錄管理2位元之CK位元，63位元之PRC及4位元之UE。此處，CK位元表示其中快取資料態係經編碼的資訊。PRC為表示具有一相對應快取行的快取資料作為位元對映圖之CPU位元的資訊。UE為表示目錄異常及其因素的資訊。

快取目錄管理單元36識別具有儲存在該所獲得的實體位址之快取資料、該快取資料之狀態等的一CPU。然後，快取目錄管理單元36基於該快取資料之狀態，藉簽發一刷新要求等而從事更新儲存在記憶體的資料之處理程序，來維持該快取資料與儲存在記憶體的資料間之同調。隨後，快取目錄管理單元36輸出該資料給該要求來源。

此處，將敘述運用該快取目錄管理單元36維持快取同調之處理實例。舉例言之，快取目錄管理單元36指示要求產生單元38發出一指令，該指令指示具有快取資料其狀態係經修正(M)的一CPU來執行回寫。然後，快取目錄管理單元36更新該資料狀態，且依據更新後的狀態來從事一處理程序。由快取目錄管理單元36發送的或接收的要求及命令型別容後詳述。

於一實體位址及一CPUID係獲取自快取目錄管理單元36之情況下，要求產生單元38產生一封包，其中儲存已經獲得的該實體位址及該CPUID，換言之，係為一記憶體存取要求之一封包。然後，要求產生單元38將所產生的封包

發送給路由器40。

此處圖11B為一略圖例示說明依據第一實施例一封包係藉一CPU發送。於圖11B例示說明之實例中，該實體位址係標示以PA。於圖11B例示說明之實例中，要求產生單元38產生其中儲存一CUID、一實體位址、及表示該要求內容的資料之一要求，且輸出所產生的要求給路由器40。於此種情況下，路由器40通過XB連結單元27將由要求產生單元38所產生的要求輸出給XB 2。然後，XB 2傳輸該要求給儲存在該要求裡的該CUID所表示之該CPU。

當從快取目錄管理單元36接收到簽發用來維持同調的一要求或一命令的一指令時，要求產生單元38係依據該指令而產生該要求或命令。然後，要求產生單元38依據該指令，透過路由器40、XB連結單元27、及XB 2，將該已經產生的要求或命令傳輸給CPU。此外，於資料係得自I/O裝置之情況下，要求產生單元38輸出對該I/O之一存取要求給路由器40。

回頭參考圖7，當透過XB 2、XB連結單元27及路由器40接收由另一個CPU所輸出的一封包時，要求接收單元39獲得含括於所接收封包內的一實體位址。然後，要求接收單元39輸出而得實體位址給快取目錄管理單元36。此外，於接收到從另一CPU所傳輸的資料之情況下，要求接收單元39將所接收的資料輸出給快取目錄管理單元36。

此外，當接收到用來維持同調的一要求或命令時，要求接收單元39將所接收的要求或命令輸出給快取目錄管理

單元36。又復，於針對I/O存取要求的回覆或資料係接收自路由器40之情況下，要求接收單元39將已接收的回覆或資料輸出給快取目錄管理單元36。於此種情況下，例如快取目錄管理單元36執行輸出所獲得的資料給記憶體存取單元41因而將儲存在記憶體22之處理。

當接收到由含括在封包控制單元37的要求產生單元38所輸出之該封包時，路由器40輸出所接收的要求給XB連結單元27。此外，路由器40透過XB連結單元27輸出由另一個CPU所發送的封包或資料給要求接收單元39。又復，路由器40將由封包控制單元37輸出給I/O等的該封包輸出給PCIe控制單元42。此外，當從I/O發送的回覆等係接收自PCIe控制單元42時，路由器40將已接收的回覆等輸出給封包控制單元37。

記憶體存取單元41乃所謂的記憶體存取控制器(MAC)，控制對記憶體22的存取。舉例言之，當一實體位址係接收自快取目錄管理單元36時，記憶體存取單元41從記憶體22獲取儲存在所接收的實體位址之資料，且輸出所獲取的資料給快取目錄管理單元36。記憶體存取單元41可形成共用區欲運用記憶體鏡射功能而變成冗餘。

當一I/O存取要求係透過路由器40獲得時，含括在PCIe控制單元42的要求產生單元43產生一要求，該要求係欲發送給作為該存取要求之目標的該I/O裝置，且輸出所產生的要求給PCIe匯流排控制單元44。當獲取由該要求產生單元43所產生的要求時，PCIe匯流排控制單元44透過PCIe連結

單元28傳輸該要求給該I/O裝置。

其次，將參考圖12敘述CPU 21發送一要求給另一個CPU之處理實例。圖4為一略圖例示說明CPU依據第一實施例發送一要求之處理實例。舉例言之，如於圖12以(A)標示，一分錄係從服務處理器24設定至節點對映圖34，其中存取一記憶體而對該記憶體分配該實體位址之一CPU的CPUID與該實體位址係彼此相聯結。

此外，算術單元31執行算術處理，如圖12標示以(B)，輸出一邏輯位址，該邏輯位址乃位址轉換單元35的存取目標。然後，位址轉換單元35將該邏輯位址轉換成一實體位址，且如圖12標示以(C)，將已轉換的實體位址輸出至快取目錄管理單元36。

此處，當從位址轉換單元35獲取該實體位址時，如圖12中標示以(D)，快取目錄管理單元36藉參考節點對映圖34獲得一CPUID，其係與所獲取的實體位址相聯結。然後，於所獲取的CPUID非為該CPU 21的CPUID之情況下，如圖12中標示以(E)，快取目錄管理單元36輸出所獲取的CPUID及實體位址給封包控制單元37。

即便於此種情況下，要求產生單元38產生一封包，該封包內儲存獲取自快取目錄管理單元36的實體位址及CPUID，且如圖12中標示以(F)，輸出所產生的封包給路由器40。然後，如圖12中標示以(G)，路由器40輸出獲取自要求產生單元38的封包給XB連結單元27。其後，如圖12中標示以(H)，XB連結單元27輸出所獲取的封包給XB 2。然後，

XB 2遞送該封包給由儲存在該封包內的該CPUID所表示的一CPU。

其次，將參考圖13描述當CPU 21從另一個CPU接收一封包時的處理實例。圖13為一略圖例示說明當該CPU依據第一實施例接收一封包時所執行的處理實例。舉例言之，如圖13中標示以(I)，要求接收單元39接收來自另一個CPU的一封包，其中儲存CPU 21的CPUID及分配給記憶體22的實體位址。

於此種情況下，要求接收單元39從所接收的封包獲得一實體位址，且如圖13中標示以(J)，輸出該所獲得的實體位址給快取目錄管理單元36。然後，快取目錄管理單元36決定該所獲得的實體位址的位元46是否為「0」或「1」。

換言之，如圖3及4之例示說明，當資訊處理系統1設定實體位址係分配給共用區及本地區時，快取目錄管理單元36無需識別該實體位址的全部位元。換言之，藉利用快取目錄管理單元36決定位元46是否為「0」或「1」，可準確地決定由該實體位址所表示的一儲存區係為共用區或本地區。

當所接收的實體位址的位元46係為「1」時，快取目錄管理單元36決定存取共用區。於此種情況下，如圖13標示以(K)，快取目錄管理單元36決定儲存在由該實體位址所表示的儲存區之資料是否快取在L1快取記憶體32及L2快取記憶體33。

此外，於資料係決定為非經快取的情況下，如圖13標示以(L)，快取目錄管理單元36將該實體位址輸出給記憶體

存取單元41。然後，如圖13標示以(M)，記憶體存取單元41從記憶體22獲取儲存在由該實體位址所表示的儲存區中的資料，且輸出所獲取的資料給快取目錄管理單元36。

然後，於資料係獲取自L1快取記憶體32、L2快取記憶體33、或記憶體存取單元41之情況下，快取目錄管理單元36輸出所獲取的資料給封包控制單元37，且指示封包控制單元37將該所獲取的資料傳輸給作為要求源的該CPU。

例如，CPU 21至21c、通訊單元23、服務處理器24、控制單元25、通訊單元26、XB連結單元27、及PCIe連結單元28為電子電路。此外，算術單元31、位址轉換單元35、快取目錄管理單元36、封包控制單元37、要求產生單元38、及要求接收單元39為電子電路。

又復，路由器40、記憶體存取單元41、PCIe控制單元42、要求產生單元43、及PCIe匯流排控制單元44為電子電路。此處，作為電子電路之一實例，使用一積體電路諸如特定應用積體電路(ASIC)或可現場程式規劃陣列(FPGA)、一中央處理單元(CPU)、一微處理單元(MPU)等。

此外，記憶體22至22a為半導體記憶體裝置，諸如隨機存取記憶體(RAM)、唯讀記憶體(ROM)、快閃記憶體等。又復，L1快取記憶體32及L2快取記憶體33為高速半導體記憶體裝置，諸如靜態隨機存取記憶體(SRAM)。

其次，將簡短描述使用CPU 21至21c維持快取同調之處理。於如下呈示的描述中，假設資訊處理系統1的各個CPU藉使用伊利諾協定而維持快取同調。

此外，於如下呈示的描述中，含括於資訊處理系統1的各個記憶體係假設識別為具有可藉全部CPU快取的一空間之一記憶體。又復，於如下呈示的描述中，透過設置在CPU內部的MAC，實體上直接連結至儲存資料的一記憶體之一CPU(其為存取目標)係稱作原位CPU，而要求存取的一CPU將稱作為本地CPU。

此外，已經發送一要求給原位CPU且已經完成快取資料的CPU，將稱作遠端CPU。此外，有些情況其中本地CPU與原位CPU為相同CPU，且有些情況其中本地CPU與遠端CPU為相同CPU。

舉例言之，本地CPU決定作為存取目標的該實體位址係藉參考其節點對映圖而分配給原位CPU所存取的一記憶體。然後，本地CPU簽發一要求，其中該實體位址係儲存至該原位CPU。此外，至於由本地CPU所簽發的要求，有多種型別的要求。據此，含括於原位CPU的快取目錄管理單元依據所獲得的要求型別而執行快取同調控制程序。

舉例言之，作為由本地CPU所簽發的要求型別，有共用型提取存取、互斥型提取存取、快取無效要求、快取置換要求等。共用型提取存取為用以執行「移進共用」的要求，且係當資料係從由該原位CPU所存取的記憶體讀取出時簽發的要求。

此外，互斥型提取存取例如為用以執行「移進互斥」的要求，且係當資料係當儲存資料於由該原位CPU所存取的記憶體時資料載入快取記憶體時簽發的要求。快取無效

要求例如為執行「移出」的要求，且係當對原位CPU做要求來失效一快取行時簽發的要求。當接收到快取無效要求時，有一種情況為原位CPU簽發無效要求給遠端CPU，或有一種情況為原位CPU簽發用以失效該快取的一命令。

快取置換要求例如為執行「回寫」的要求，且係當更新快取資料時簽發，換言之，在修正態的快取資料係回寫到由該原位CPU所存取的記憶體。此外，快取置換要求例如為執行「回刷新」的要求，且係當快取資料尚未被更新時簽發，換言之，於共用態或互斥態的快取被拋棄。

於前述要求係接收自本地CPU之情況下，為了處理該要求，原位CPU簽發一命令給本地CPU或遠端CPU。此處，為了依據獲得的要求型別來執行快取同調控制處理，原位CPU簽發多型命令。舉例言之，原位CPU簽發「移出與旁路共享」，因而將遠端CPU所快取的資料負載至本地CPU。

此外，例如，原位CPU失效除了本地CPU以外的全部遠端CPU的快取，及隨後原位CPU簽發「移出與旁路互斥」而傳輸資料給本地CPU。又復，原位CPU簽發「移出帶有失效」，要求遠端CPU來失效該快取。當原位CPU簽發「移出帶有失效」時，全部CPU的快取針對一目標位址係在無效態。

此外，原位CPU簽發「移出用以刷新」，要求遠端CPU來失效該快取行。於原位CPU簽發「移出用以刷新」之情況下，形成一種狀態，其中目標資料係僅由原位CPU快取。此外，於目標資料狀態為「共用」之情況下，原位CPU簽發「緩衝無效」，要求遠端CPU拋棄該快取。

原位CPU依據要求型別，藉簽發前述命令，過渡由各個CPU所快取的資料狀態。此外，於接收一命令之情況下，本地CPU或遠端CPU執行由該命令所表示之一處理，及過渡藉此所快取的資料狀態。

隨後，本地CPU或遠端CPU發送一命令完成的回覆或資料添加完成的回覆給原位CPU。此外，於執行命令處理程序後，原位CPU或遠端CPU發送一要求資料添加的回覆給本地CPU。

CPU之處理流程

其次，將參考圖14描述設定含括於資訊處理系統1的各個CPU中的節點對映圖34之一處理程序的流程圖。圖14為流程圖例示說明一節點對映圖設定處理之流程。於如下呈示之詳細說明中，一個CPU與由該CPU所存取之一記憶體所組成的集合將稱作為一節點。此外，於如下呈示之詳細說明中，將說明一個實例，其中一新節點係添加至資訊處理系統1。

首先，於步驟S101，資訊處理系統1的操作員添加一新節點。其次，於步驟S102，建構區塊10至10e的服務處理器讀取所添加節點的硬體組態。其次，於步驟S103，資訊處理系統1的操作員指示服務處理器分配含括在該新節點的記憶體的一共用區。

其次，於步驟S104，資訊處理系統1的操作員指示該新節點的服務處理器輸入電力。然後，於步驟S105，建構區塊10至10e之服務處理器基於運用I2C讀取組態資訊，設定

含括於建構區塊10至10e的CPU之節點對映圖34。隨後，於步驟S106，資訊處理系統1輸入電力給建構區塊10至10e及結束處理程序。

其次，將參考圖15描述使用資訊處理系統1來控制共用區之處理流程。圖15為流程圖例示說明共用區控制處理之流程。首先，於步驟S201，資訊處理系統1執行依據來自一應用程式的要求而在多個節點間分配一共用記憶體之處理。其次，於步驟S202，資訊處理系統1執行附接在多個節點間共用的一共用記憶體之處理。

其後，於步驟S203，由含括在資訊處理系統1的CPU所執行的應用程式使用記憶體。其次，於步驟S204，資訊處理系統1從事一共用記憶體脫離處理。其後，於步驟S205，資訊處理系統1從事一共用記憶體解除處理且結束處理程序。此外，步驟S201及S205之處理只可藉在該共用記憶體的原位節點上跑的應用程式執行；或當實際上並未執行任何操作時，也可執行在該共用記憶體的原位節點以外的節點上跑的應用程式。

其次，將參考圖16說明圖15中於步驟S201表示的共用記憶體分配處理之流程。圖16為流程圖例示說明共用記憶體之分配處理。舉例言之，於圖16例示說明之實例中，於步驟S301，由CPU 21執行的應用程式要求OS執行在節點間分配一共用記憶體的處理。

然後，於步驟S302，由CPU 21所執行的OS分配一記憶體具有來自用於共用區的該實體位址一區之要求大小。其

次，於步驟S303，該OS遞送由OS所分配的一共用記憶體之管理ID給該應用程式及結束共用記憶體之分配處理程序。

其次，將參考圖17描述例示說明圖15的步驟S202之在節點間附接一共用記憶體的處理流程。圖17為流程圖例示說明共用記憶體之附接處理程序。首先，於步驟S401，一應用程式遞送一管理ID給OS且要求附接一共用記憶體的處理流程。於此種情況下，於步驟S402，OS與在其它節點執行的作業系統通訊，及獲得與管理ID相對應的一實體位址。

此處，於OS與在其它節點執行的作業系統通訊之情況下，使用透過區域網路(LAN)等的通訊、透過服務處理器24在節點間的通訊等。舉例言之，可組配使得在各個節點執行的作業系統(OS)設定一特定共用區作為用在節點間之通訊的一區，及藉將資訊儲存於或讀取自該設定區而從事通訊。

其次，於步驟S403，OS決定與該實體位址相對應的一邏輯位址(虛擬位址)且從事其分配。舉例言之，藉CPU 21執行的OS藉運用位址轉換單元35而設定該實體位址及邏輯位址的TLB。

此外，由CPU 21至21c所使用的邏輯位址針對各個CPU可能係在重疊範圍，或可能係在彼此相異範圍。又復，由CPU 21至21c所使用的邏輯位址可藉OS裡的應用程式指定。其後，於步驟S404，OS遞送該邏輯位址之值給應用程式且結束該處理程序。

其次，將參考圖18描述例示說明圖15的步驟S203之在節點間使用一共用記憶體之應用程式的處理流程。圖18為

流程圖例示說明使用一共用記憶體之應用程式的處理程序。舉例言之，於步驟S501，由CPU 21執行的應用程式簽發一邏輯位址，且存取由該邏輯位址所表示之一儲存區。

然後，於步驟S502，CPU 21決定是否出現TLB遺漏。於步驟S503，於出現TLB遺漏之情況下(步驟S502為是)，CPU 21執行捕獲處置程序，及設定一邏輯位址及一實體位址的集合分錄於TLB。

其次，於步驟S504，應用程式再度簽發一邏輯位址，及通常地運用TLB，透過將該邏輯位址轉換成一實體位址而存取該共用記憶體。另一方面，於步驟S505，當不出現TLB遺漏時(步驟S502為否)，通常執行對共用記憶體的存取，及結束處理。

其次，將參考圖19描述例示說明圖15的步驟S204之在節點間脫離一共用記憶體的處理流程。圖19為流程圖例示說明在節點間，共用記憶體之脫離處理程序。舉例言之，於步驟S601，由CPU 21所執行的一應用程式要求OS執行脫離處理程序，節點間的共用記憶體之一邏輯位址或一管理ID係經指定。

然後，於步驟S602，由CPU 21所執行的OS刷新該快取記憶體。換言之，於解除共用記憶體的分配後再度做共用記憶體的分配之情況下，未做共用記憶體的分配之情況下，當在該共用記憶體的原位節點之一CPU重新軟體啟動時，擔憂快取記憶體與實際記憶體狀態可能彼此不匹配。據此，OS刷新該快取記憶體，藉此防止其中快取記憶體與

實際記憶體狀態彼此不匹配的狀態。

然後，於步驟S603，OS解除在節點間該共用記憶體的分配，換言之，於由該應用程式所使用的範圍內該邏輯位址之分配，及去除與該被釋放的邏輯位址有關的TLB之分錄。此外，於步驟S604，OS執行節點間之通訊，藉此通知該應用程式已經完成目標實體位址(PA)的使用。然後，於步驟S605，當原位節點透過節點間之通訊認知最末使用者脫離已經被解除的共用記憶體時，OS解除對一經指定的共用記憶體之記憶體分配。此外，步驟S605之處理係有關於圖20例示說明之步驟S702的處理。

於步驟S603之後，即便針對在該節點已經脫離的記憶體位址出現TLB遺漏之情況下(步驟S502為是)，OS並不設定相對應於已經在TLB釋放的邏輯位址之一實體位址。於此種情況下，步驟S504的處理通常不結束，反而出現存取錯誤。此外，與步驟S402相反地，於脫離完成後，OS執行節點間之通訊，藉此通知該應用程式已經完成對該共用記憶體的實體位址之存取。在其中於原位節點該共用記憶體已經被釋放的狀態中，於該應用程式相對應於共用記憶體的最末使用者之情況下，該原位節點被要求執行釋放處理。

其次，將參考圖20描述例示說明圖15的步驟S205表示的在節點間釋放該共用記憶體的處理流程。圖20為流程圖例示說明在節點間，共用記憶體之釋放處理程序。舉例言之，於步驟S701，由CPU 21所執行的應用程式要求OS執行在節點間釋放該共用記憶體的處理。然後，於步驟S702，

於指定的共用區之全部使用者皆脫離之情況下，OS釋放分配及結束處理。當脫離尚未完成時，處理結束而不執行分配釋放處理程序。於步驟S605，執行完成分配的實際處理。

其次，將參考圖21說明一處理流程，其中CPU 21發送一記憶體存取要求給其它CPU。圖21為流程圖例示說明一要求簽發處理流程。舉例言之，於步驟S801，CPU 21的算術單元簽發一邏輯位址。

然後，於步驟S802，位址轉換單元35將該邏輯位址轉成一實體位址。其次，於步驟S803，快取目錄管理單元36獲得實體位址及管理快取目錄。換言之，快取目錄管理單元36過渡由該所獲得的實體位址表示的儲存區之快取態。

其次，於步驟S804，快取目錄管理單元36決定該所獲得的實體位址是否為藉參考節點對映圖34所分配給任何其它節點的一實體位址。於步驟S805，當該所獲得的實體位址係經決定為非為在分配給位在任何其它節點的一實體位址時(步驟S804為否)，快取目錄管理單元36使用該所獲得的實體位址從事記憶體存取。

另一方面，於步驟S806，當該所獲得的實體位址係經決定為在分配給位在任何其它節點的一實體位址時(步驟S804為是)，快取目錄管理單元36獲得CUID其係與得自節點對映圖34的該實體位址相聯結。然後，於步驟S807，封包傳輸單元產生一封包，其中儲存CUID及實體位址，換言之，記憶體存取要求發送記憶體存取要求給XB 2，及結束處理程序。

其次，將參考圖22說明當CPU 21接收來自任何其它CPU的記憶體存取要求時所執行的處理流程。圖22為流程圖例示說明當接收要求時執行的處理流程。於圖22例示說明之實例中，將描述當CPU 21接收來自任何其它CPU的「移進共用」或「移進互斥」時執行的處理流程。舉例言之，於步驟S901，CPU 21透過XB 2接收來自任何其它CPU的要求。

於此種情況下，於步驟S902，CPU 21決定屬於要求目標的該實體位址之一預定位元是否為「1」，藉此決定屬於要求目標的該實體位址是否相對應於一本地區。於步驟S903，當作為要求目標的該實體位址係相對應於一本地區時(步驟S902為是)，CPU 21回送否定回覆給屬於要求源的該CPU及結束處理。

另一方面，於步驟S904，於作為要求目標的該實體位址係不相對應於該本地區之情況下(步驟S902為否)，CPU 21管理快取目錄用來維持同調。此外，於步驟S905，CPU 21決定由該實體位址所表示的儲存區之狀態。

然後，於步驟S906，CPU 21依據所決定的狀態簽發一命令，及於步驟S907，CPU 21過渡該狀態。其後，CPU 21發送儲存在由該實體位址所表示的儲存區中的資料給該要求源的CPU作為回覆及結束處理。

其次，將參考圖23描述當CPU 21接收到回覆時所執行的處理流程。圖23為流程圖例示說明當CPU接收到回覆時所執行的處理流程。例如於步驟S1001，CPU 21接收回覆。於此種情況下，於步驟S1002，CPU 21決定回覆內容是否為

正常。

於步驟S1003，若回覆內容為正常，換言之，接收到作為要求目標的資料之情況下(步驟S1002為是)，則CPU 21使用該資料從事正常操作及結束處理。另一方面，於步驟S1004，當接收到否定回覆時(步驟S1002為否)，CPU 21決定否定回覆的理由是否為存取錯誤。

於步驟S1005，當否定回覆的理由非為存取錯誤(步驟S1004為否)時，CPU 21執行正常錯誤處置及結束處理。另一方面，於步驟S1006，當否定回覆的理由非為存取錯誤(步驟S1004為是)時，CPU 21設定在一錯誤暫存器中出現錯誤的實體位址，執行捕獲處理及結束處理。

第一實施例之優點

如前文描述，資訊處理系統1包括CPU 21至21c、記憶體22至22c、及連結CPU 21至21c彼此的XB 2。此外，CPU 21包括邏輯位址與實體位址間轉換的位址轉換單元，及用在實體位址與CPUID間之轉換的節點對映圖34。

CPU 21發送含括一實體位址及一CPUID的一要求封包。此外，於一要求封包係接收自另一CPU之情況下，CPU 21基於儲存在所接收的封包裡的該實體位址，而決定作為一存取目標的一儲存區是否為共用區或本地區。

藉此方式，資訊處理系統1可使用小量硬體有效地存取節點間的共用記憶體。換言之，於資訊處理系統1中，因CPU 21使用用以執行實體位址與CPUID間之轉換的節點對映圖34來執行位址轉換，故可有效從事記憶體存取。

此外，於CPU 21存取由另一個CPU所存取的共用區之情況下，CPU 21可傳輸封包，其中一實體位址及一CPUID只儲存至XB 2。據此，資訊處理系統1可有效地從事記憶體存取。

又復，於CPU 21接收來自另一個CPU的要求封包之情況下，資訊處理系統1基於儲存在該所接收的封包的實體位址，而決定作為一存取目標的一儲存區是否為共用區或本地區。據此，資訊處理系統1可維持儲存在本地區的核心資料及使用者資料的安全度位準為高。此外，因資訊處理系統1組配全部記憶體為可快取，故容易掩蓋記憶體存取的延遲時間。

使用類似用來存取記憶體22的方法，CPU 21存取由另一個CPU所存取的記憶體共用區。換言之，即便於作為存取目標的該儲存區係存在於記憶體22或另一個記憶體之情況下，含括於CPU 21的算術單元31可只輸出邏輯位址。

據此，即便於未進行互斥地控制I/O等的處理程序、程式規劃等之情況下，資訊處理系統1容易地存取共用區，據此，可改進記憶體存取效能。此外，即便於所執行的程式或OS係不修正的情況下，CPU 21可合宜地使用共用記憶體，結果，類似一般情況，可執行預提取處理，藉此可改進記憶體的存取效能。

此外，資訊處理系統1分配其預定位元為「1」的實體位址給共用區，及分配其預定位元為「0」的實體位址給本地區。據此，唯有藉決定該實體位址的預定一個位元是否

為「1」，CPU 21可容易地決定作為存取目標的實體位址是否為共用區的實體位址。結果，資訊處理系統1可執行有效記憶體存取。

另一方面，當來自另一CPU的記憶體存取目標經決定為存取本地區時，CPU 21回送否定回覆。據此，資訊處理系統1阻止存取共用區以外的一區，藉此可避免錯誤。

此外，快取目錄管理單元36使用節點對映圖34，將一實體位址轉換成儲存在與其相聯結的節點對映圖34之一CPUID。據此，CPU 21可識別一CPU，該CPU存取作為存取目標的實體位址被分配的該記憶體。

又復，建構區塊10至10e各自係包括改寫該節點對映圖34的一服務處理器。據此，資訊處理系統1可自由地分配一本地區及一共用區給各個記憶體22至22c。舉例言之，當記憶體22具有4 TB容量時，資訊處理系統1可組配具有任意容量的一儲存區來在各節點間共享，例如，1 TB係分配給本地區，及3 TB係分配給共用區。

此外，即便於增加新CPU及新記憶體，或移開CPU或記憶體之情況下，資訊處理系統1可透過服務處理器以容易方式分配本地區及共用區。

又復，藉使用一目錄管理CPU快取儲存在記憶體22的資料，CPU 21控制快取同調。據此，即便於含括在資訊處理系統1的CPU數目增加之情況下，資訊處理系統1可有效地維持快取同調而不增加XB 2的資料通量。

更明確言之，於資訊處理系統1中，CPU間的通訊係限

於快取已更新資料的一遠端CPU與一原位CPU，或一遠端CPU、一原位CPU與快取已更新資料的一本地CPU。據此，資訊處理系統1可有效地維持快取同調。

此外，於發生快取遺漏之情況下，CPU 21決定發生快取遺漏之該實體位址是否為分配給由另一個CPU存取的一記憶體的一實體位址。於發生快取遺漏之該實體位址被決定為分配給由另一個CPU存取的一記憶體的一實體位址之情況下，CPU 21將該實體位址轉換成一CUID，產生其中儲存該實體位址及該CUID之一封包，及傳輸該所產生的封包。據此，CPU 21可存取一記憶體而不執行剩餘位址轉換處理。

又復，於所執行的應用程式要求獲取一共用區的情況下，CPU 21設定一TLB，該TLB係用來執行由該應用程式所使用的一邏輯位址與分配給該共用區的一實體位址間之轉換。據此，CPU 21可存取一記憶體而在考慮存取共用區或本地區時無需修正所執行的應用程式或OS。

[b]第二實施例

至目前為止，雖然已經描述本發明之實施例，但可以各種形式從事前述實施例以外的實施例。如此，後文中，屬於本發明之另一個實施例將描述為第二實施例。

(1)建構區塊

前述資訊處理系統1含括建構區塊10至10e，各有四個CPU。但實施例並非囿限於此，建構區塊10至10e各自可具有任意數目的CPU及由各個CPU存取的記憶體。此外，CPU

及記憶體無需具有一對一的對應關係，直接存取記憶體的CPU可以是全部CPU的一部分。

(2)共用區及本地區的分配

前文描述的實體位址對共用區及本地區之分配僅為一個實例，及資訊處理系統1可分配任意實體位址給各個區。

舉例言之，資訊處理系統1可分配其最低有效的一個位元為「0」的實體位址給共用區，及分配其最低有效的一個位元為「1」的實體位址給本地區。於此種情況下，藉由決定該實體位址的最低有效的一個位元為「0」或「1」，各個CPU容易地決定該存取目標是否為共用區。

此外，資訊處理系統1可分配含括在實體位址空間的第一半的任意實體位址給該共用區，及分配含括在實體位址空間的一第二半的任意實體位址給該本地區。於此種情況下，各個CPU可容易地藉決定實體位址的最高的一個位元是否為「0」或「1」而決定一存取目標是否為該共用區。又復，資訊處理系統1可分配含括在實體位址空間的一第一半的任意實體位址給該本地區，及分配含括在實體位址空間的一第二半的任意實體位址給該共用區。

換言之，雖然資訊處理系統1可分配任意實體位址給該共用區及本地區，但藉分配其一預定位元具有一相同值的實體位址給共用區，且分配該預定位元具有與共用區不同值的實體位址，可容易地決定一存取目標是否為該共用區或本地區。

(3)由CPU發送的封包

前述CPU 21發送含括一CUID及一PS的封包作為一記憶體存取要求。但實施例並非圍限於此。換言之，CPU 21可輸出其中儲存任意資訊的一封包，只要可獨特地識別存取一記憶體其乃一存取目標的一CPU即可。

此外，例如，CPU 21可將一CUID轉換成一虛擬連結(VC) ID及儲存該VCID。又復，CPU 21可儲存資訊，諸如表示封包裡的資料長度之一長度資訊。

(4)由CPU簽發指令

如前文說明，CPU 21至21c中之各者藉簽發一要求或一指令而維持快取同調。但前述要求或指令僅供舉例說明，例如CPU 21至21c可簽發比較與調換(CAS)指令。

如前述，當CPU 21至21c簽發CAS指令時，即便當互斥控制競爭頻繁地出現在多個CPU間時，各項處理程序係在各個CPU的快取記憶體中執行。結果，CPU 21至21c可防止因記憶體存取的發生所造成的延遲，且可防止CPU間的異動處理壅塞。

(5)透過超級管理器之控制

於前述資訊處理系統1中，已經描述一個實例，其中藉OS對屬於硬體的該位址轉換單元35作存取。但實施例並非圍限於此，舉例言之，操作虛擬機器的超級管理器(HPV)可存取位址轉換單元35。

換言之，在HPV操作的一節點，OS並不直接操作CPU 21至21c的硬體資源諸如快取記憶體或記憶體管理單元(MMU)，反而要求超級管理器進行操作。如前述，當透過

超級管理器接受控制時，CPU 21至21c各自將一虛擬位址轉換成一實際位址(RA)，其後，將該實際位址轉換成一實體位址。

此外，在HPV操作的該節點，中斷處理不會直接中斷OS，反而係中斷超級管理器(HPV)。於此種情況下，HPV讀取出該OS之一中斷處理程序且執行該中斷。此外，藉前述HPV執行的處理為執行用來操作虛擬機器的已知處理。

(6)使用區劃的處理

於前文說明的資訊處理系統1中，藉使用一個節點對映圖，CPU 21至21c各自發送一記憶體存取。但實施例並非囿限於此。舉例言之，可組配使得建構區塊10至10e操作為多個節點組群，針對各個節點組群組配一個邏輯區劃，其操作同一個韌體(超級管理器)。

於此種情況下，CPU 21至21c各自具有表示存取目的地之CPU的一節點對映圖，及表示在同一個邏輯區劃內部之CPU的一節點對映圖。如上，因CPU 21至21c各自具有表示含括在相同邏輯區劃內部的CPU的一節點對映圖，故CPU可識別不欲傳輸超出該邏輯區劃的特定封包的傳輸範圍，諸如通知錯誤、停機要求、及復置要求封包的出現。

後文中將描述表示含括在相同邏輯區劃內部的CPU的一節點對映圖。圖24為略圖例示說明依據第二實施例之一資訊處理系統。如圖24之例示說明，建構區塊10及10a操作邏輯區劃#A，及建構區塊10b至10d操作邏輯區劃#B。

此處，於邏輯區劃#A中，多個定義域#A至#C及韌體#A

操作。此外，於邏輯區劃#B中，多個定義域#D至#G及韌體#B操作。此處，韌體#A及韌體#B例如為超級管理器。此外，於邏輯區劃#A中，應用程式及作業系統(OS)操作；而於其它定義域#B至#G中之各者，類似定義域#A，應用程式及OS操作。

換言之，定義域#A至#G為虛擬機器，其中應用程式及OS係分別地獨立地操作。此處，雖然含括於建構區塊10的CPU 21至21c可傳輸前述特別封包給含括於區劃#A的CPU，但CPU 21至21c不會傳輸前述特別封包給含括於區劃#B的CPU。

據此，建構區塊10至10d各自的CPU具有一節點對映圖表示含括在同一個邏輯區劃內的CPU的CUID。舉例言之，CPU 21具有一節點對映圖34，其中一實體位址及連結至一記憶體含括由該實體位址所表示之一儲存區的一CPU的該CUID係關聯彼此儲存。此外，CPU 21具有一節點對映圖34，其中儲存含括在與CPU21同一個邏輯區劃亦即區劃#A內的CPU的CUID。類似節點對映圖34，節點對映圖34a假設係由服務處理器24設定。

後文中，將參考附圖描述含括在相同邏輯區劃內的CPU的CUID的節點對映圖之一個實例。圖25為略圖例示說明區劃之實例。舉例言之，於圖25例示說明之實例中，區劃#A包括建構區塊#0。此外，於建構區塊#0中，CPU #0及位址區「#0」具有所分配的記憶體。

此外，區劃#B包括建構區塊#1及#2。此處，於建構區

塊#1、CPU #4、CPU #5、及位址區「#1」具有所分配的記憶體，及位址區「#2」具有所分配的記憶體。此外，位址區「#1」之所分配的記憶體係藉CPU #4存取，及位址區「#2」之所分配的記憶體係藉CPU #5存取。又復，於建構區塊#2中，CPU #8及位址區「#3」具有所分配的記憶體。

其次，將參考圖26A至26C描述圖25例示說明的含括於CPU #0之節點對映圖及含括於CPU #4之節點對映圖。首先，將參考圖26A及26B描述由區劃#A的CPU所儲存的節點對映圖。圖26A為略圖例示說明由區劃#A的CPU所儲存之節點對映圖之一實例。圖26B為略圖例示說明表示區劃#A的節點對映圖之一實例。

於後文呈示之說明中，節點ID「0」表示建構區塊#0，節點ID「1」表示建構區塊#1，節點ID「2」表示建構區塊#2。此外，CUID「0」為CPU #0的CUID，CUID「4」為CPU #4的CUID，CUID「5」為CPU #5的CUID，及CUID「8」為CPU #8的CUID。

舉例言之，於圖26A例示說明之實例中，節點對映圖34例示說明位址區「0」係存在於建構區塊#0且係藉CPU #0存取。此外，節點對映圖34例示說明位址區「1」係存在於建構區塊#1且係藉CPU #4存取。又復，節點對映圖34例示說明位址區「2」係存在於建構區塊#2且係藉CPU #5存取。此外，節點對映圖34例示說明位址區「3」係存在於建構區塊#3且係藉CPU #8存取。

圖26B為略圖例示說明表示區劃#A的一個節點對映圖

之一實例。如圖26B之例示說明表示區劃#A的該節點對映圖包括有效性、節點ID、及CPUID於各個分錄。舉例言之，於圖26B例示說明之實例中，該節點對映圖表示建構區塊#0的CPU #0係含括於區劃#A。

舉例言之，於圖25例示說明之實例中，CPU #0包括圖26A及26B例示說明的節點對映圖。當欲做記憶體存取時，CPU #0識別藉使用圖26A表示的節點對映圖之該存取目的地的CPU。另一方面，於特定封包只欲傳輸給位在相同區劃內部的CPU之情況下，CPU #0識別藉使用圖26B表示的節點對映圖之該傳輸目的地的CPU。換言之，CPU #0傳輸特定封包給位在於由圖26B例示說明的節點對映圖所表示的區劃#A內部之CPU作為一例。

另一方面，為了做記憶體存取，CPU #4包括圖26A及26C例示說明的節點對映圖。此處，圖26C為略圖例示說明表示區劃#B的一節點對映圖實例。於圖26C例示說明之實例中，表示區劃#B之一節點對映圖表示建構區塊#1的CPU #4及#5及建構區塊#2的CPU #8係存在於區劃#B。CPU #4傳輸特定封包給位在於由圖26C例示說明的節點對映圖所表示的區劃#B內部之CPU作為一例。

如前述，CPU #1及#4儲存其中一位址區及一CPUID係彼此相聯結的該節點對映圖及表示一區劃的該節點對映圖。然後，CPU #1及#4藉使用其中一位址區及一CPUID係彼此相聯結的該節點對映圖而直接地存取含括在其它節點的記憶體。此外，CPU #1藉使用表示區劃#A之一節點對映

圖而傳輸特定封包。又復，CPU #4藉使用表示區劃#B之一節點對映圖而傳輸特定封包。

如前述，各個CPU可包括一節點對映圖，其針對含括該CPU的各個區劃具有不同值。於各個CPU具有該節點對映圖具有針對含括該CPU的各個區劃的不同值之情況下，可防止特定封包係傳輸超出該區劃。

此外，類似第一實施例，各個CPU可表示一位址區，該位址區藉一起始位址及一位址罩或一起始位址與一長度表示一存取目標。換言之，CPU #1及CPU #4使用表示一位址區的該節點對映圖作為存取目標，該位址區係為藉使用一起始位址及位址罩或一起始位址與長度表示一存取目標。此外，CPU #1及CPU #4藉使用表示互異區劃的節點對映圖來傳送特定封包。

依據一實施例，可有效地執行藉各個算術處理單元所做記憶體存取。

【圖式簡單說明】

圖1為略圖例示說明依據第一實施例一資訊處理系統之一實例；

圖2為略圖例示說明依據第一實施例一建構區塊的功能組態；

圖3為略圖例示說明依據第一實施例分配給該等建構區塊的記憶體之實體位址之範圍；

圖4為略圖例示說明依據第一實施例由該資訊處理系統分配給各個記憶體之實體位址；

圖5為第一略圖例示說明分配實體位址中之一變化；

圖6為第二略圖例示說明分配實體位址中之一變化；

圖7為略圖例示說明依據第一實施例一CPU的功能組態；

圖8為略圖例示說明依據第一實施例儲存在節點對映圖中之資訊之一實例；

圖9為第一略圖例示說明儲存在節點對映圖中之資訊之一變化實例；

圖10為第二略圖例示說明儲存在節點對映圖中之資訊之一變化實例；

圖11A為略圖例示說明一快取標籤之一實例；

圖11B為略圖例示說明依據第一實施例由CPU所發送之一封包；

圖12為略圖例示說明依據第一實施例使用CPU發送一要求之處理實例；

圖13為略圖例示說明依據第一實施例當CPU接收一封包時所執行的處理實例；

圖14為流程圖例示說明一節點對映圖設定處理之處理流程；

圖15為流程圖例示說明一共用區控制處理之處理流程；

圖16為流程圖例示說明一共用記憶體分配處理；

圖17為流程圖例示說明一共用記憶體附接處理；

圖18為流程圖例示說明使用一共用記憶體的應用程式之處理程序；

圖19為流程圖例示說明在節點間脫離一共用記憶體之

處理程序；

圖20為流程圖例示說明在節點間釋放一共用記憶體之處理程序；

圖21為流程圖例示說明一要求簽發處理之處理流程；

圖22為流程圖例示說明當接收一要求時執行的一處理流程；

圖23為流程圖例示說明當CPU接收到回覆時執行的一處理流程；

圖24為略圖例示說明依據第二實施例之一資訊處理系統；

圖25為略圖例示說明區劃之一實例；

圖26A為略圖例示說明由區劃#A之CPU所儲存的一節點對映圖之一實例；

圖26B為略圖例示說明表示區劃#A之一節點對映圖之一實例；及

圖26C為略圖例示說明表示區劃#B之一節點對映圖之一實例。

【主要元件符號說明】

1...資訊處理系統	27...XB連結單元
2...縱橫式開關(XB)	28... 周邊組件互連體快速
2b、24...服務處理器	(PCIe)連結單元
3...管理終端	30...算術處理單元
10、10a-e...建構區塊	31...算術單元
21、21a-c...中央處理單元(CPU)	32...L1快取記憶體
22、22a-c...記憶體	33...L2快取記憶體

34、34a...節點對映圖	42...PCIe控制單元
35...位址轉換單元	44...PCIe匯流排控制單元
36...快取目錄管理單元	S101-106 、 S201-205 、
37...封包控制單元	S301-303 、 S401-404 、
38、43...要求產生單元	S501-504 、 S601-605 、
39...要求接收單元	S701-702 、 S801-807 、
40...路由器	S901-908 、 S1001-1006...
41...記憶體存取單元	步驟

七、申請專利範圍：

1. 一種資訊處理裝置，其係包含：

多個節點；及

一互連體，其互相連結該等多個節點，

其中該等多個節點各自包括：

一處理器，

一儲存單元，

一第一轉換單元，其執行一邏輯位址與一實體位址間之轉換，

一第二轉換單元，其執行該實體位址與用以識別含括在該等多個節點各自中的一處理器之處理器識別資訊間之轉換，

一發送單元，其發送包括該實體位址及用以存取由該實體位址所指示的一儲存區之該處理器識別資訊的傳輸資料，

一接收單元，其係接收從另一個節點透過該互連體發送的該傳輸資料，及

一本地決定單元，其基於含括在由該接收單元所接收的該傳輸資料的該實體位址，決定一存取是否為一存取至從含括該儲存單元的該節點所能存取的該儲存單元之一本地區，或為一存取至從該等多個節點所能存取的該儲存單元之一共用區。

2. 如申請專利範圍第1項之資訊處理裝置，其中

一儲存單元之該共用區係分配給其位在一預定位

置的一位元具有一相同值的實體位址，

該儲存單元之該本地區係分配給實體位址，該等實體位址位在一預定位置的一位元係具有與該等實體位址之分配給該共用區的該預定位置之該位元值不同的一值，及

該本地決定單元係依據位在含括於該傳輸資料的該實體位址的該預定位置之該位元的一值，而決定一存取是否為存取至該本地區或存取至該共用區。

3. 如申請專利範圍第1項之資訊處理裝置，其中

該本地區及該共用區係分配給含括在該等多個節點各自的儲存單元之全部實體位址，及

該本地決定單元係依據含括於該傳輸資料的該實體位址的一最高有效位元的一值，而決定一存取是否為存取至該本地區或存取至該共用區。

4. 如申請專利範圍第1項之資訊處理裝置，其中

該發送單元係發送一否定回覆，指示於該本地決定單元決定該存取係存取該本地區之情況下，一存取係不允許存取該傳輸資料之一傳輸源的一節點。

5. 如申請專利範圍第1項之資訊處理裝置，其係進一步包含：

一儲存裝置，其係儲存該處理器識別資訊及分配給該節點的該儲存單元之一實體位址，包括由該處理器識別資訊表示之該處理器彼此相聯結，其中

該第二轉換單元係將該實體位址轉換成儲存在與該實體位址相聯結的該儲存裝置之該處理器識別資訊。

6. 如申請專利範圍第5項之資訊處理裝置，其係進一步包含：

一控制裝置，其改寫該處理器識別資訊及儲存在該儲存裝置的該實體位址。

7. 如申請專利範圍第1項之資訊處理裝置，其中該等多個節點各自包括一目錄控制單元，其藉使用一目錄表示一節點從含括於該節點的該儲存單元快取該資料，而維持藉該等節點中之任一者所快取的資料身分(identity)。

8. 如申請專利範圍第1項之資訊處理裝置，其中該等多個節點各自係進一步包含：

一快取儲存單元，其係從含括在該等多個節點的該等儲存單元中快取資料；及

一決定單元，於出現一快取錯誤之情況下，決定出現該快取錯誤的該實體位址是否為含括在其它節點中之任一者的該儲存單元之一實體位址，其中

於該決定單元已經決定出現該快取錯誤的該實體位址係為含括在該等其它節點中之任一者的該儲存單元之該實體位址的情況下，該第二轉換單元將該實體位址轉換成該處理器識別資訊。

9. 如申請專利範圍第2項之資訊處理裝置，其中該處理器執行一作業系統設定該第一轉換單元，因而於一應用程式要求該共用區的獲取之情況下，執行由該應用程式所使用的該邏輯位址與分配給該共用區的該實體位址間之轉換。

10. 一種藉多個節點各自執行的記憶體存取方法，該方法係

包含：

在一邏輯位址與一實體位址間轉換，及在該實體位址與用以識別含括在該等多個節點各自中的一處理器之一處理器識別資訊轉換；

發送包括該實體位址及用以存取由該實體位址所指示的一儲存區之該處理器識別資訊的一傳輸資料；

接收從另一個節點透過該互連體發送的該傳輸資料；及

基於含括在由該接收單元所接收的該傳輸資料的該實體位址，決定一存取是否為一存取至從含括該儲存單元的該節點所能存取的該儲存單元之一本地區，或為一存取至從該等多個節點所能存取的該儲存單元之一共用區。

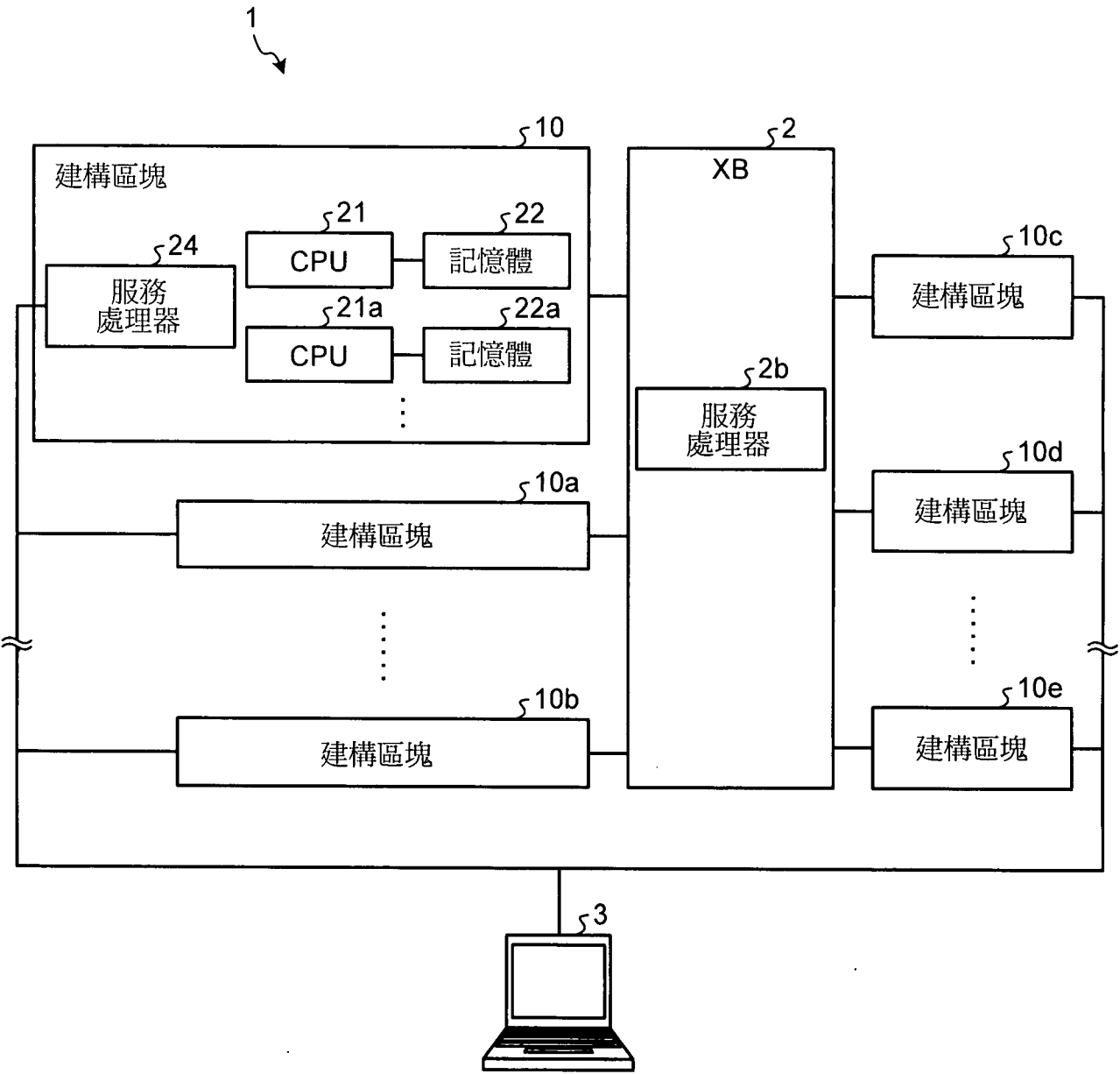


圖 1

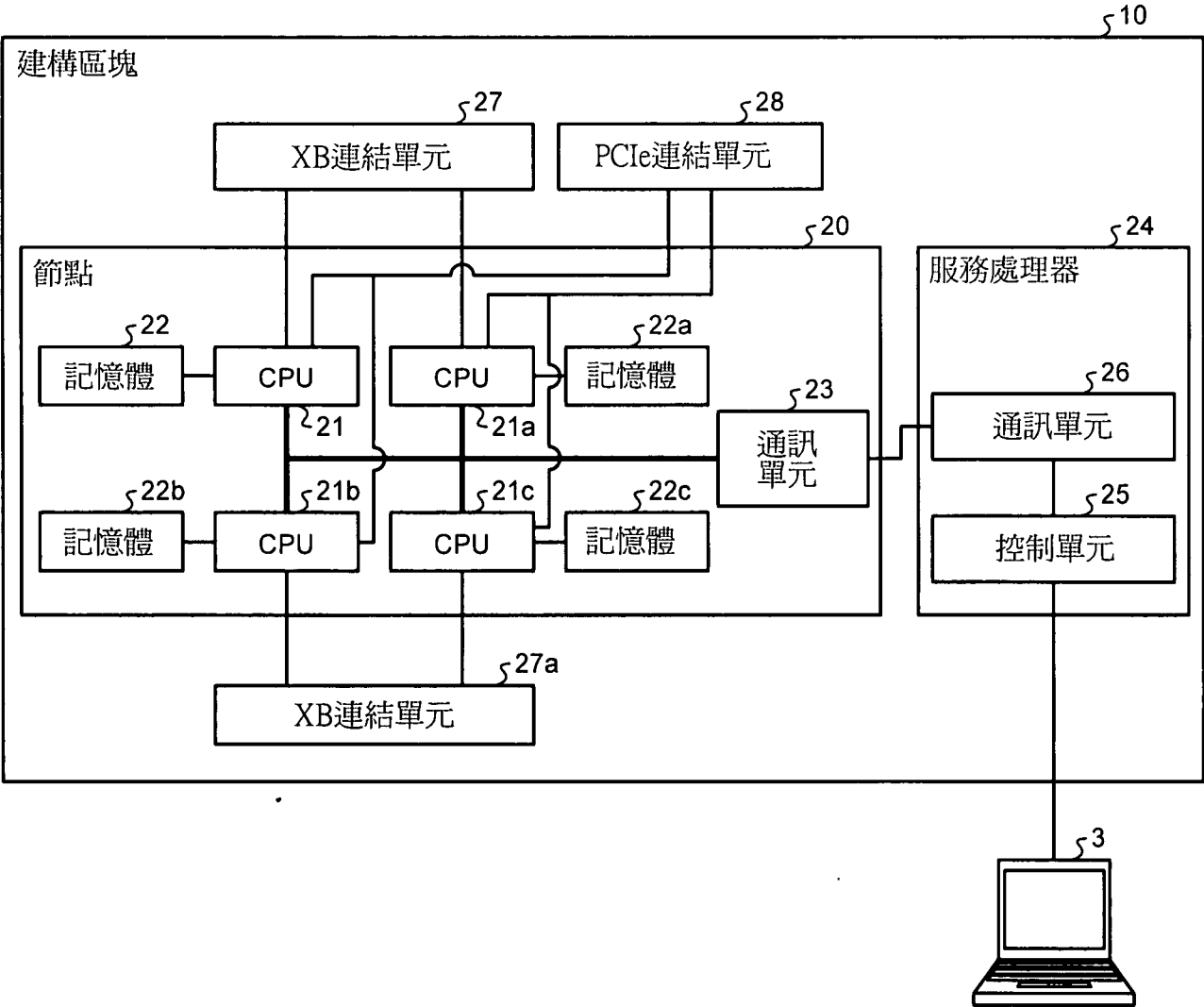


圖 2

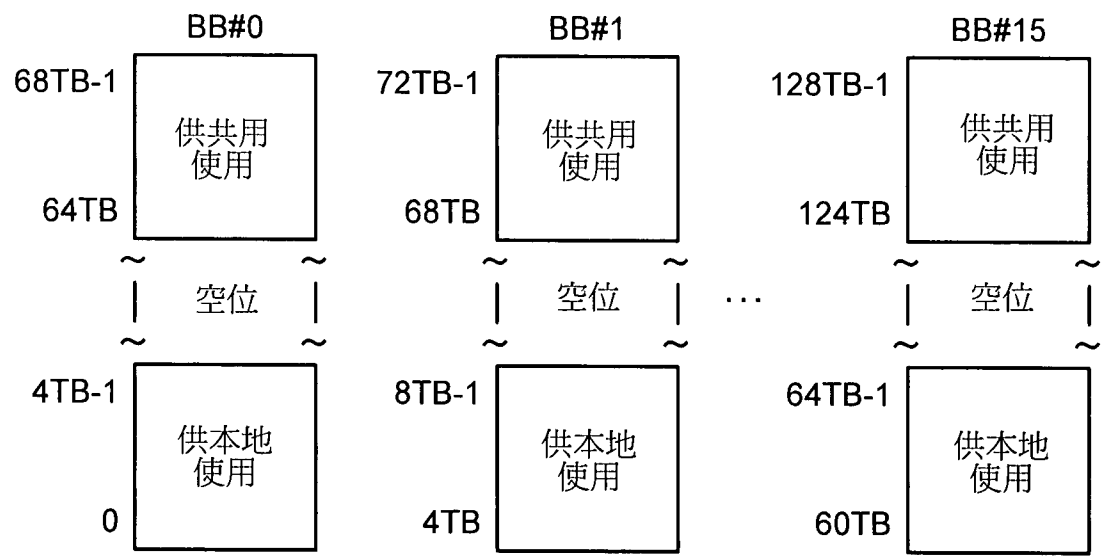


圖 3

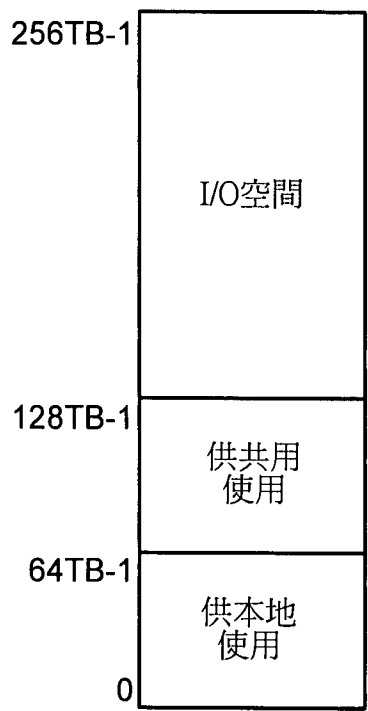


圖 4

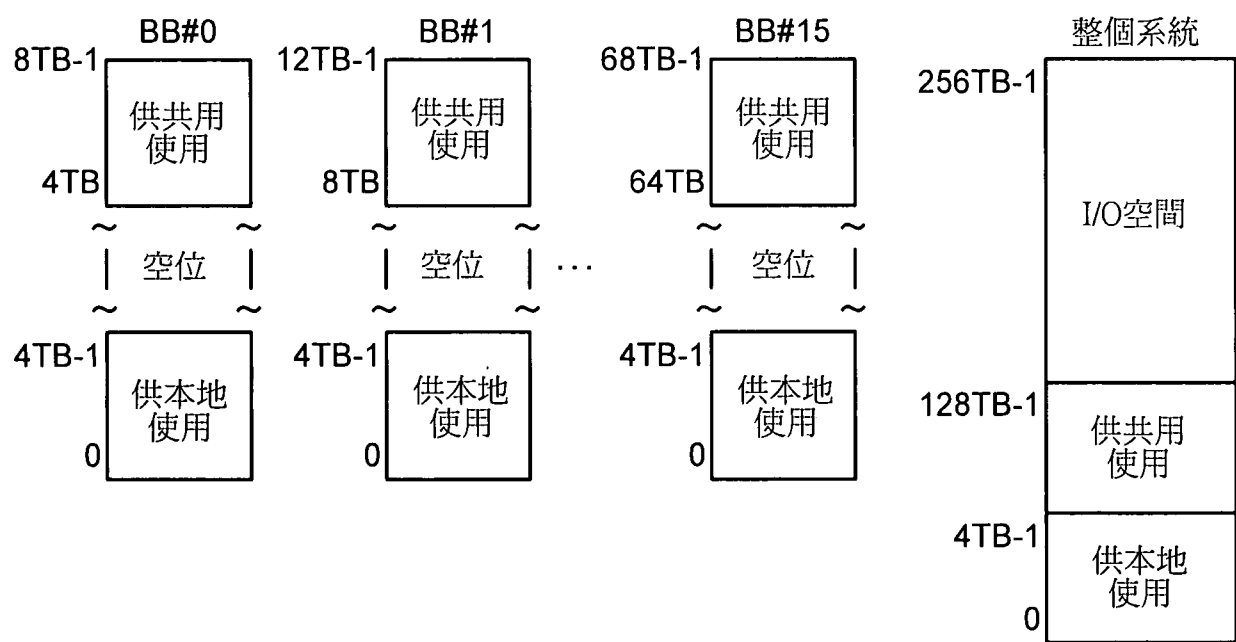


圖 5

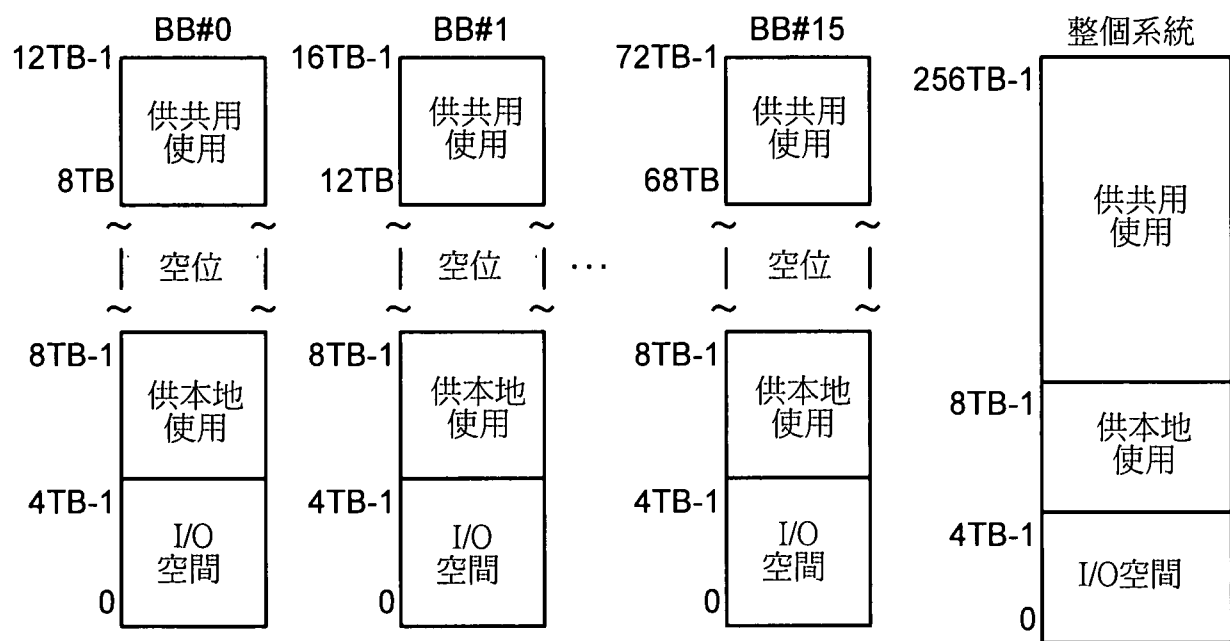
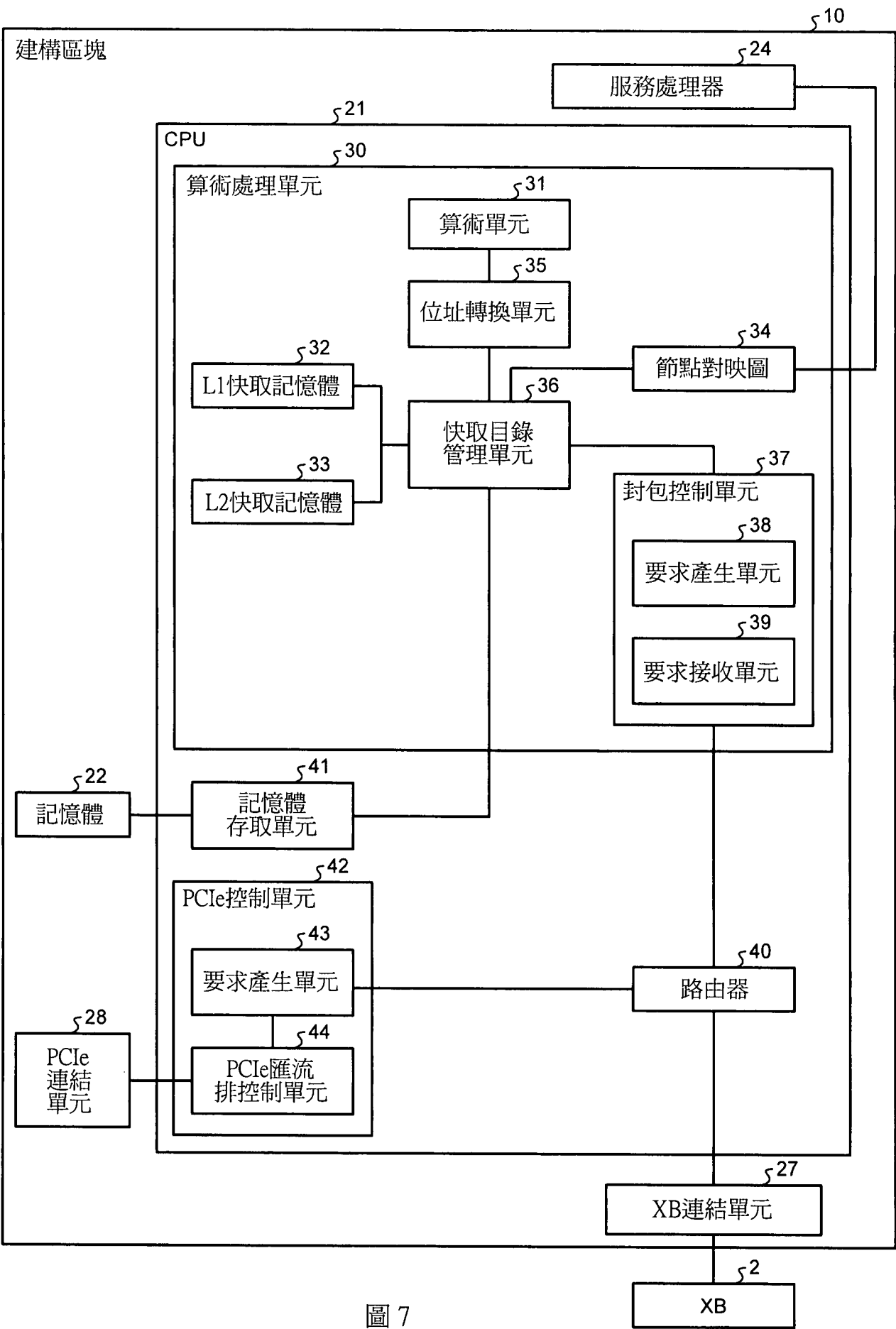


圖 6



位址	有效	節點ID	CPU ID
#0	1	0	0
#1	1	0	1
#2	0		

⋮

圖 8

有效	起始位址	位址罩	節點ID	CPU ID
1	0x00000	0x3fff	0	0
1	0x10000	0x3fff	1	4
1	0x14000	0x3fff	1	5
1	0x20000	0x1fff	2	8

⋮

圖 9

有效	起始位址	長度	節點ID	CPU ID
1	0x00000	0x3fff	0	0
1	0x10000	0x3fff	1	4
1	0x14000	0x3fff	1	5
1	0x20000	0x2ef	2	8

⋮

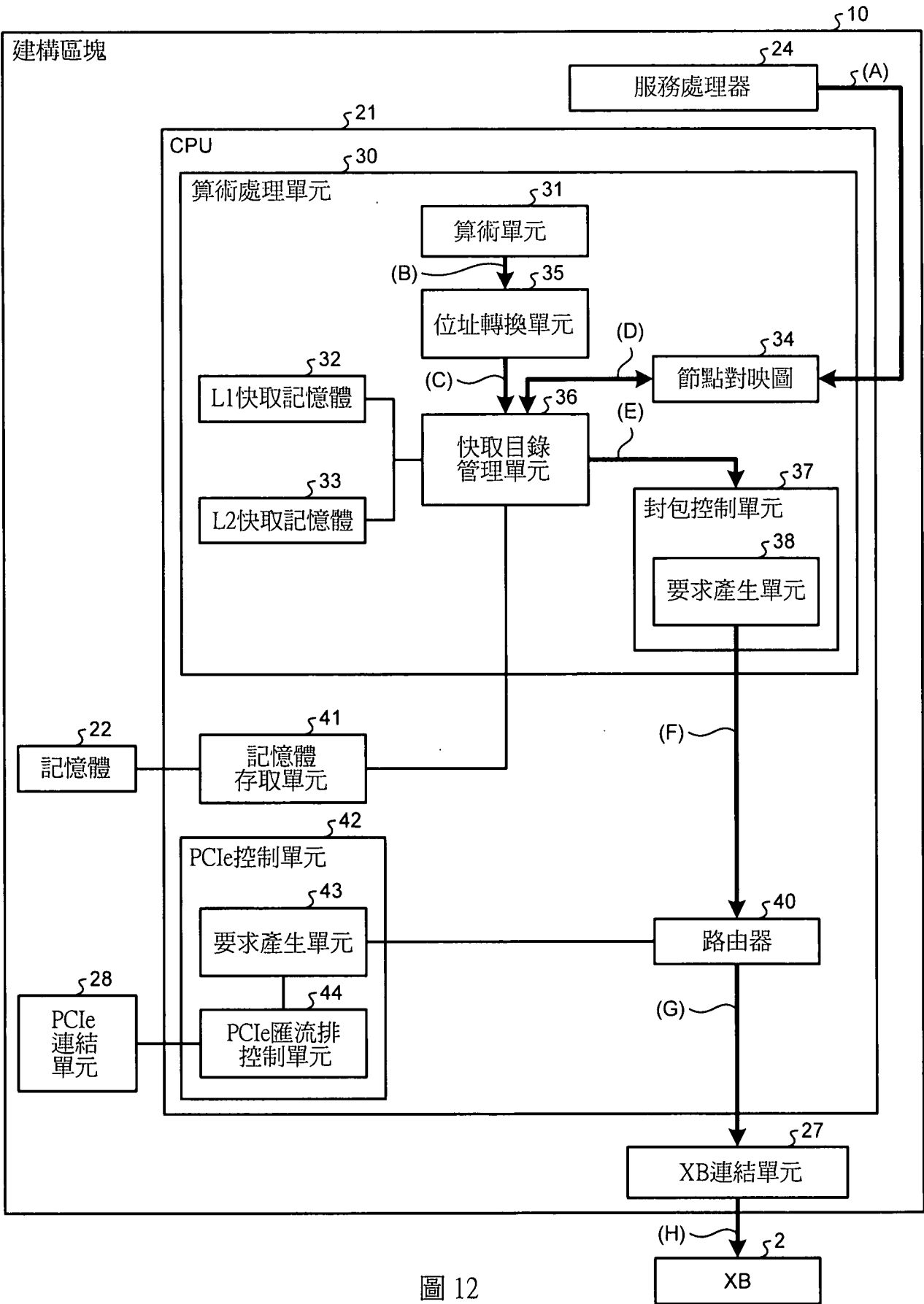
圖 10

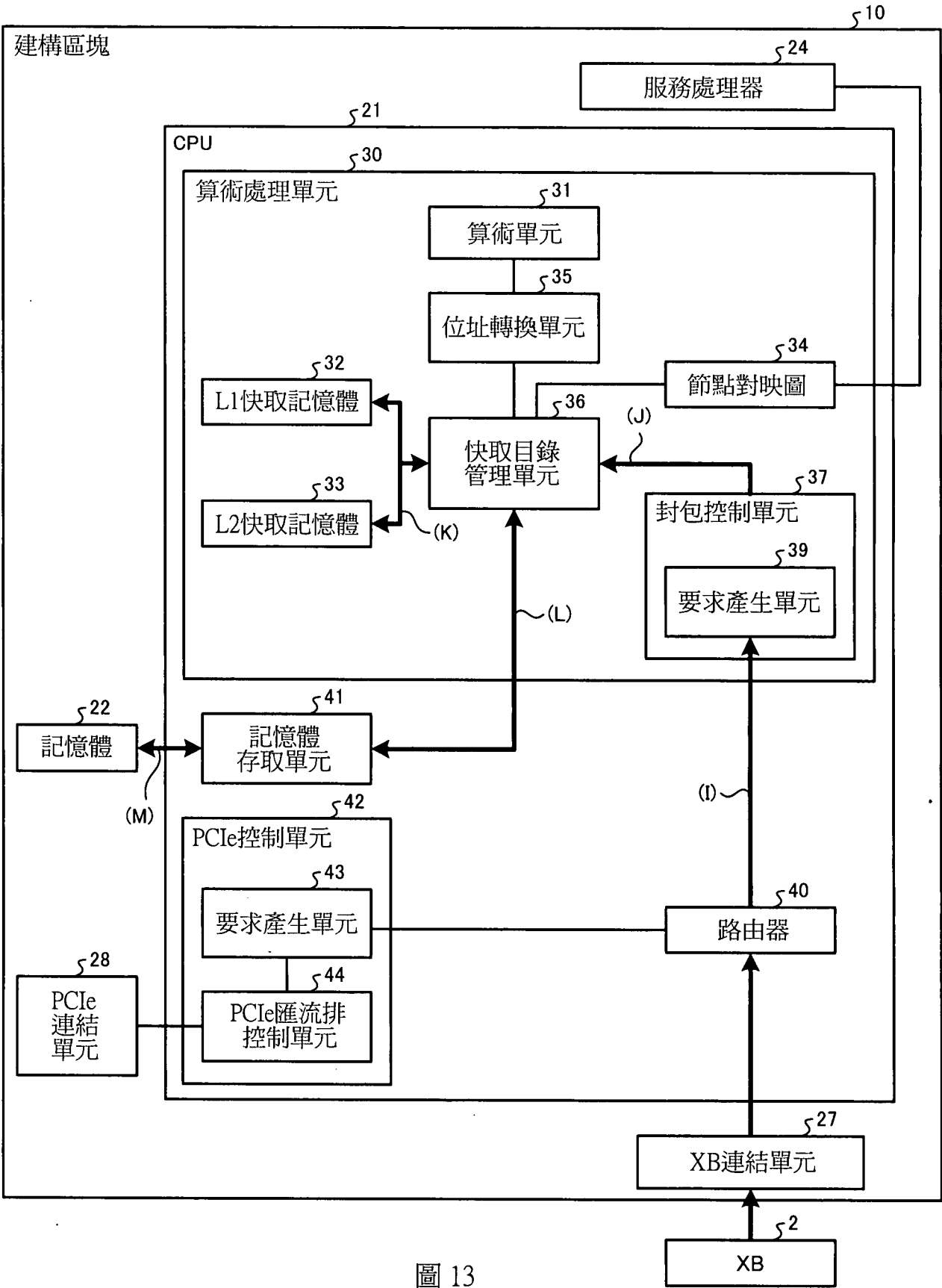
退化 旗標	ECC 查核 位元	IF/OFF 代碼	L1 快取態	L2 快取態	AA
----------	-----------------	--------------	-----------	-----------	----

圖 11A

CPU ID	PA	資料
--------	----	----

圖 11B





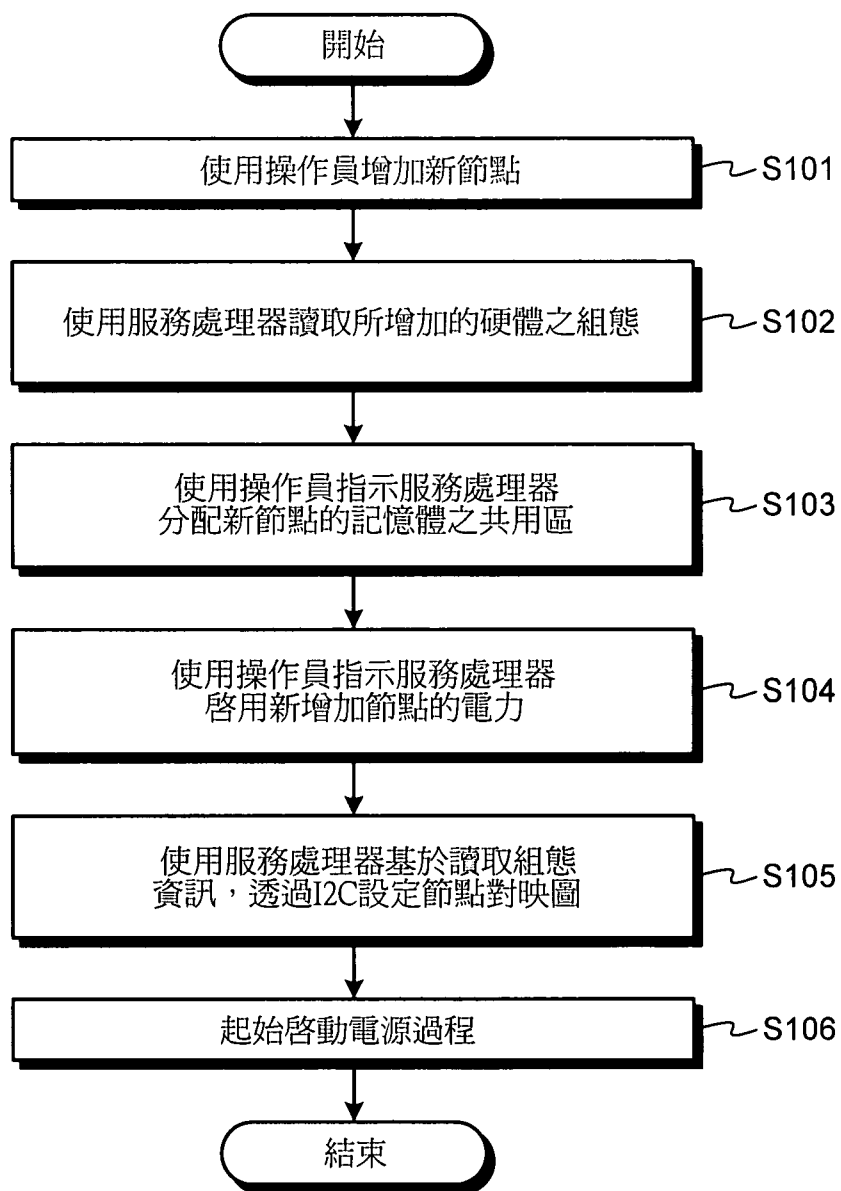


圖 14

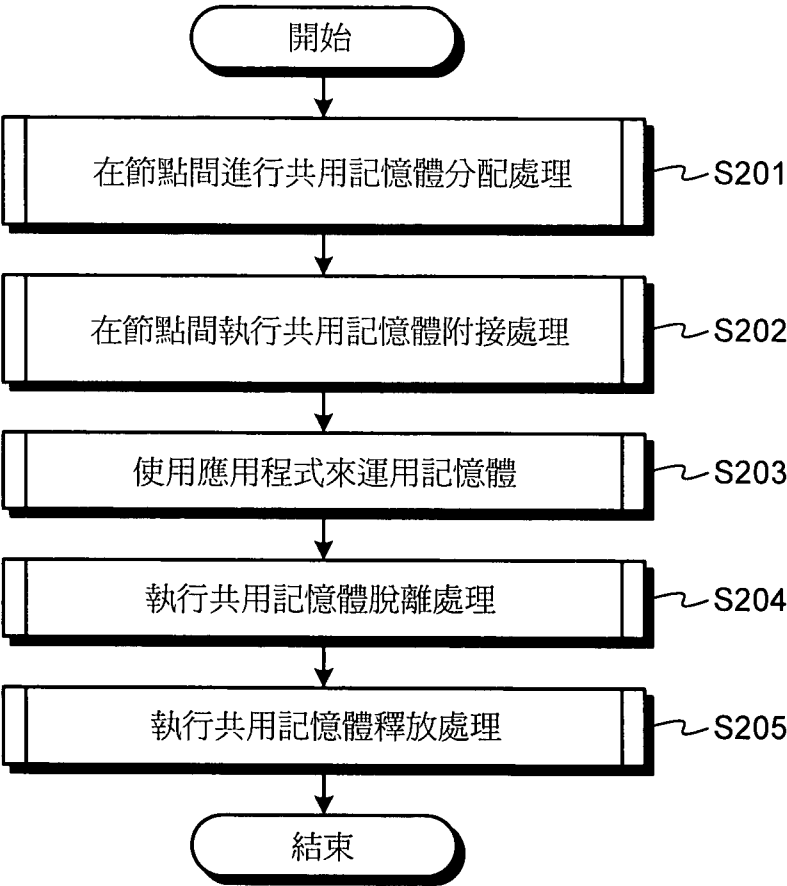


圖 15

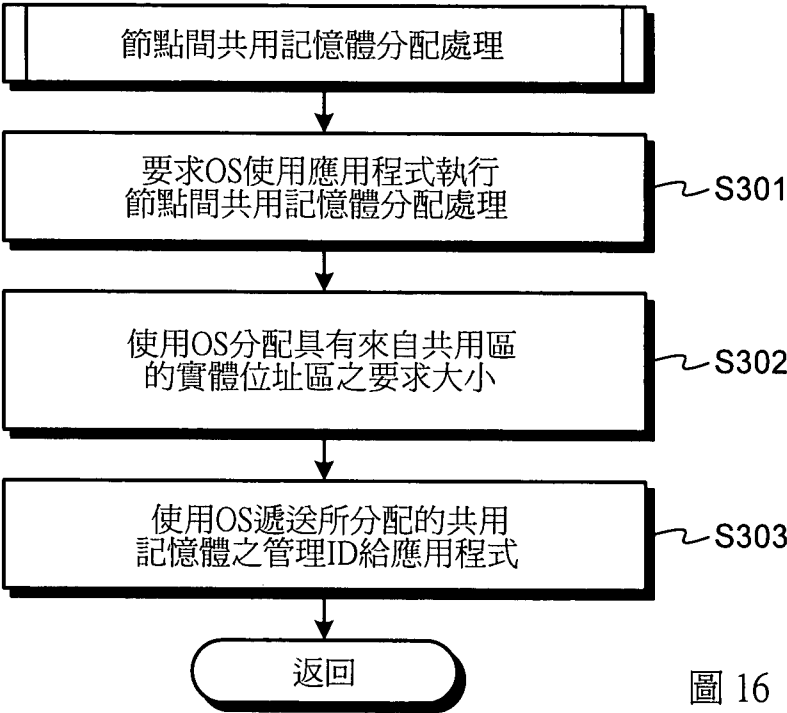


圖 16

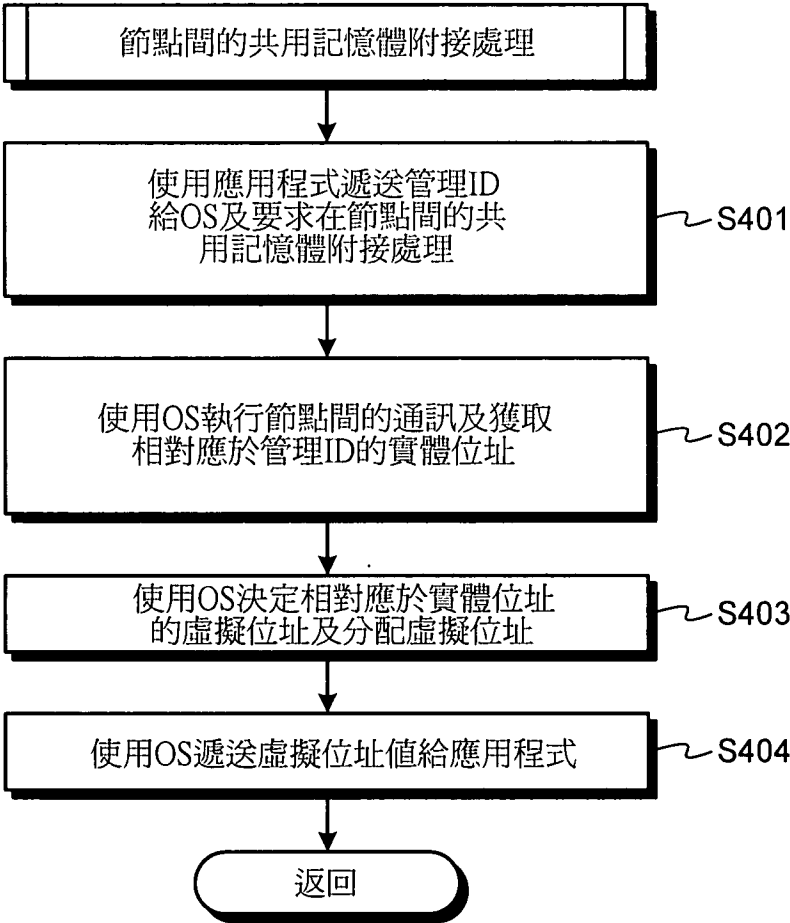


圖 17

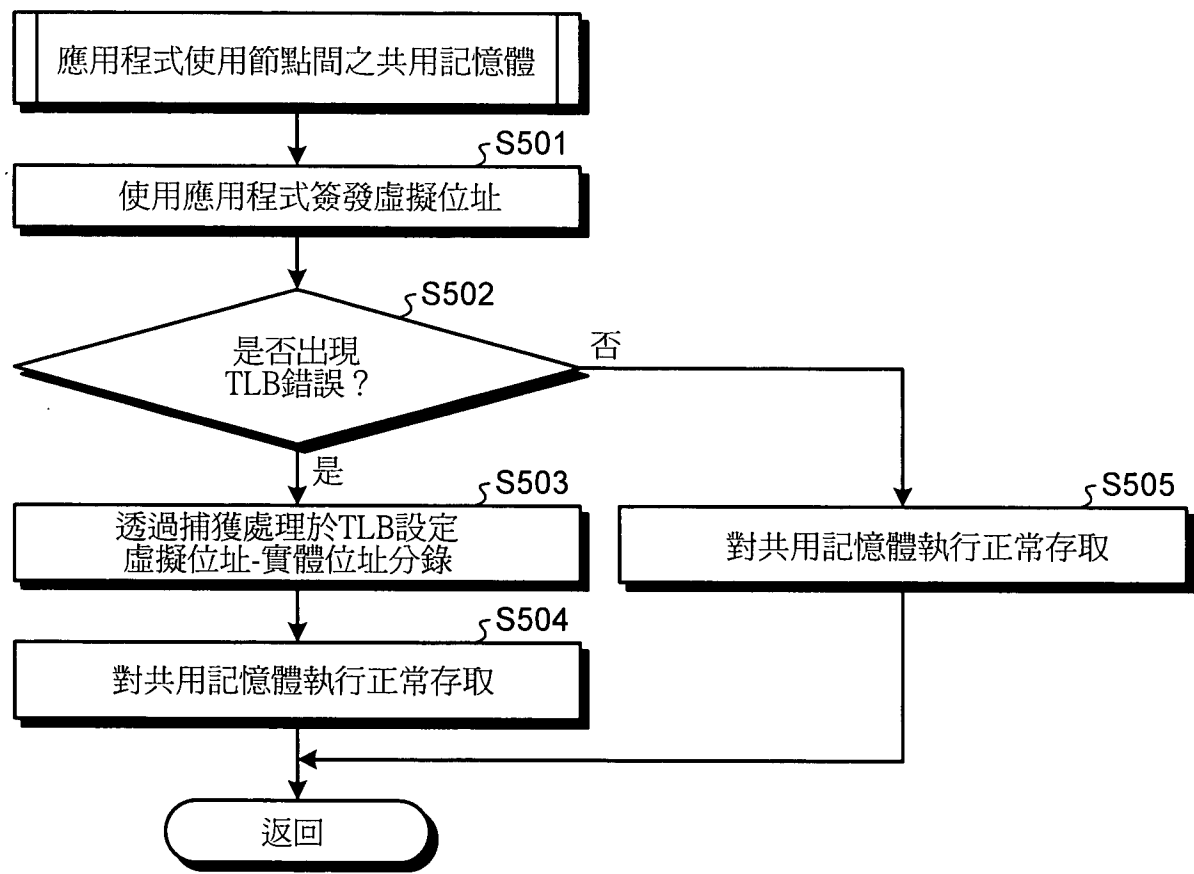


圖 18

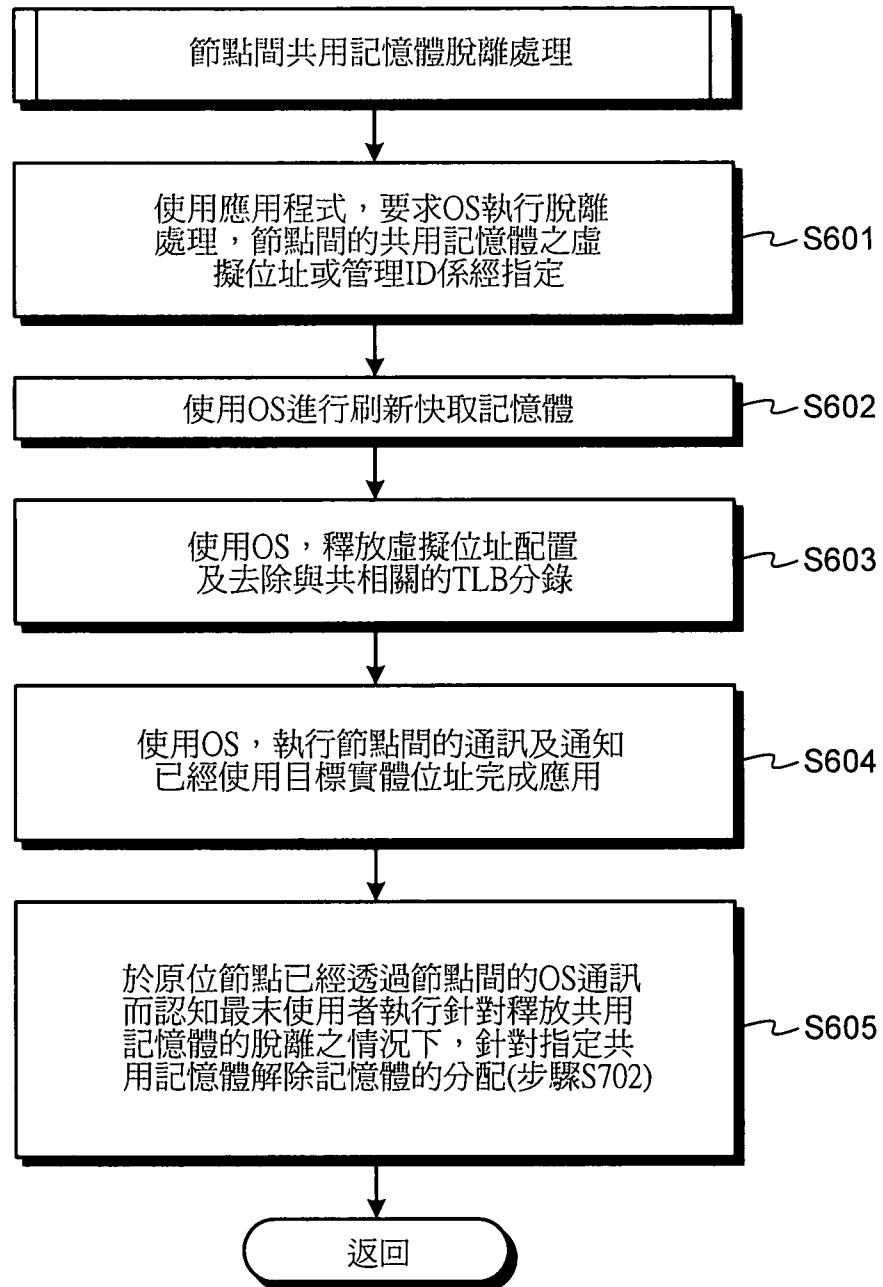


圖 19

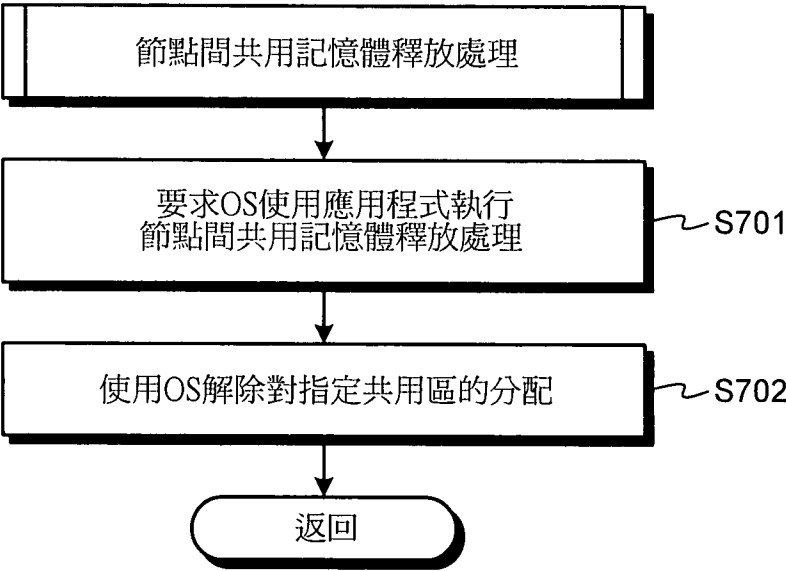


圖 20

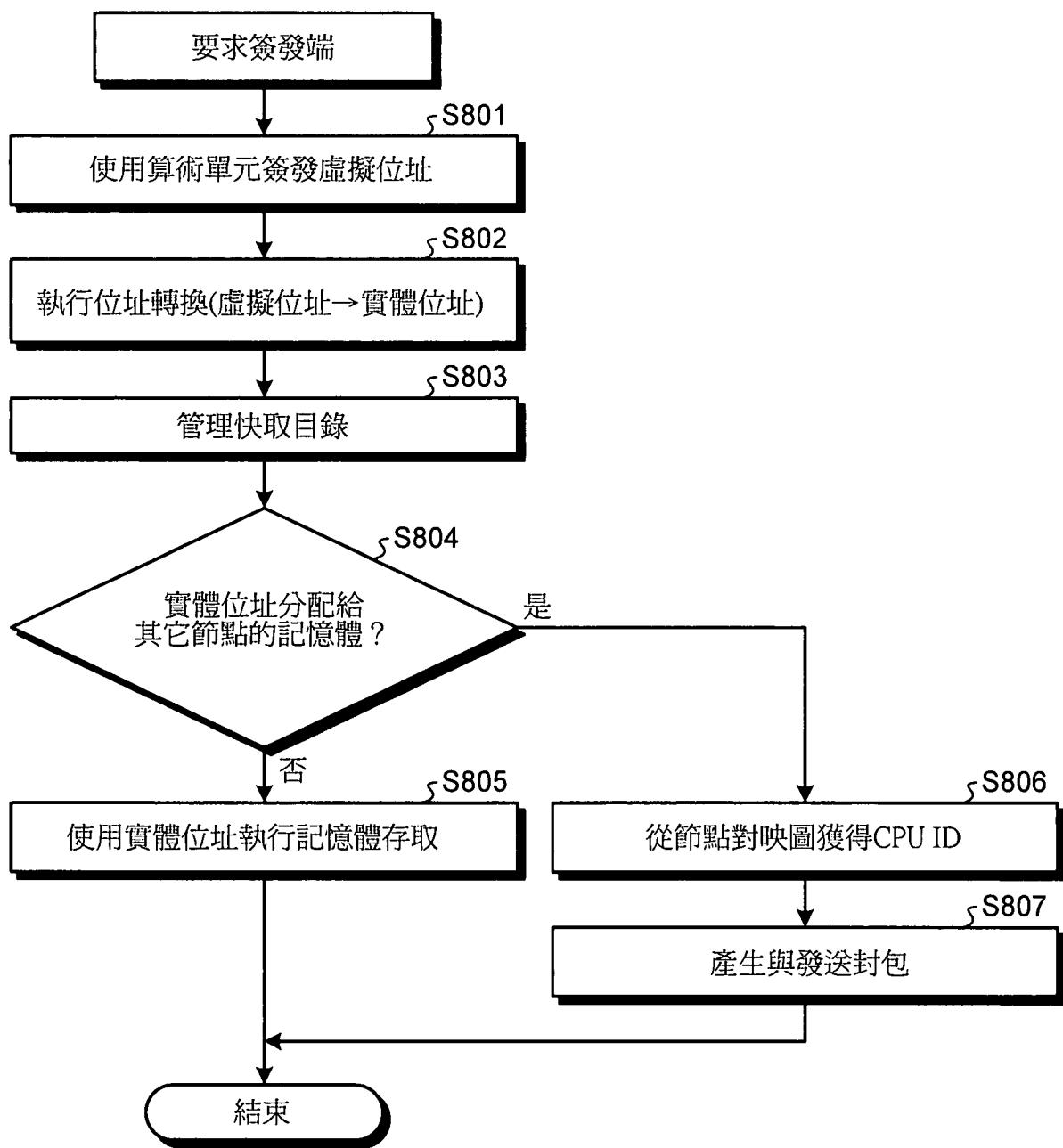


圖 21

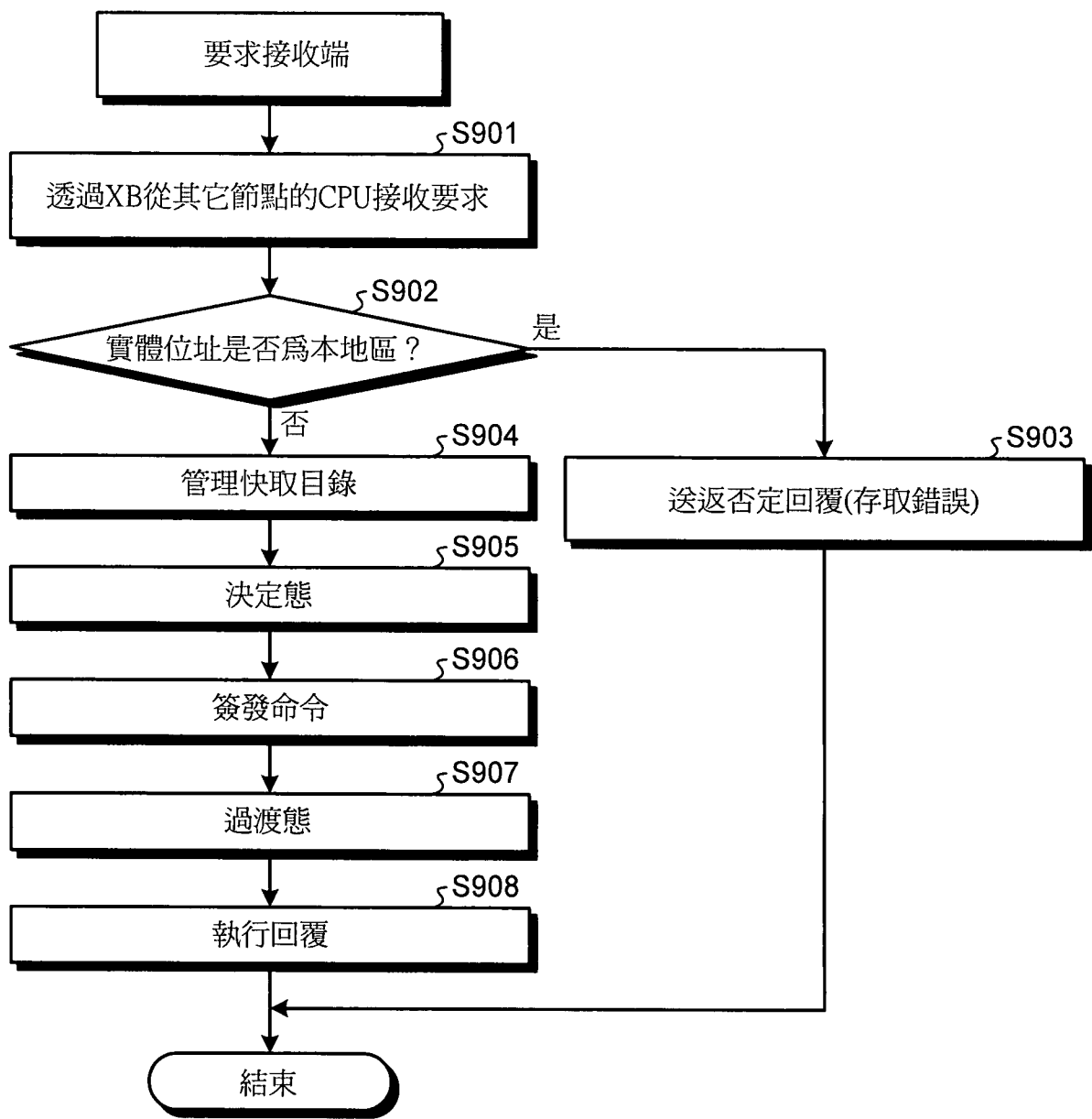


圖 22

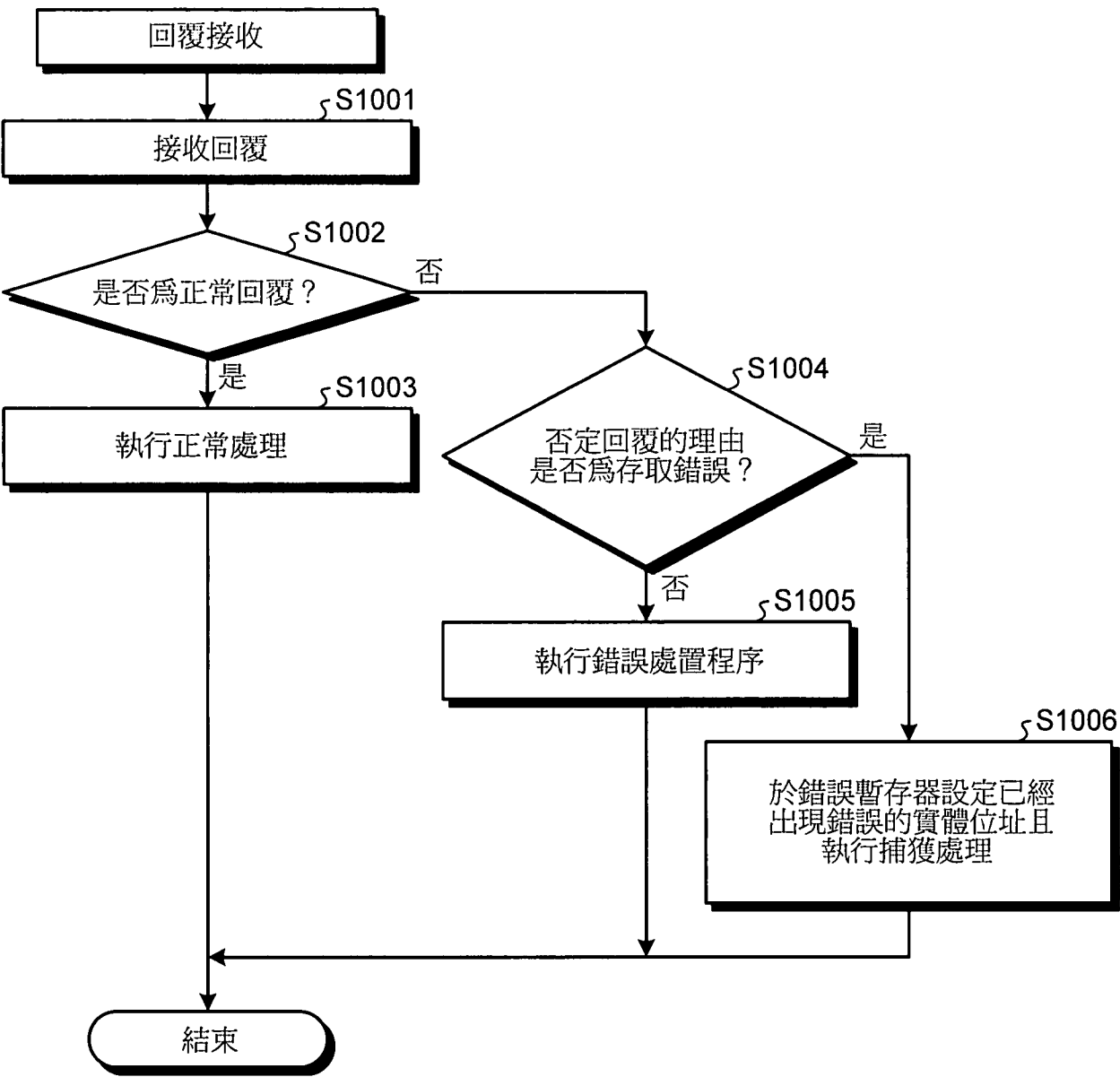


圖 23

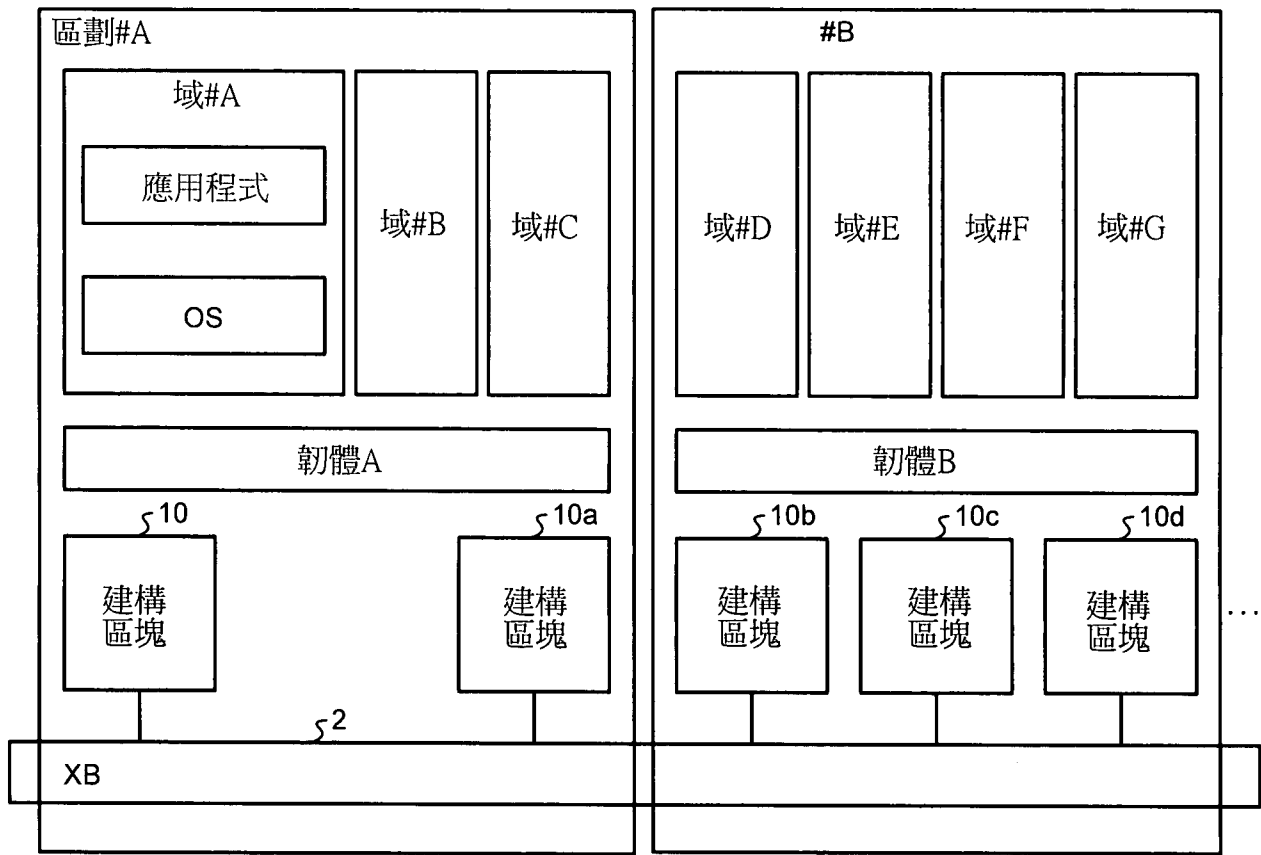


圖 24

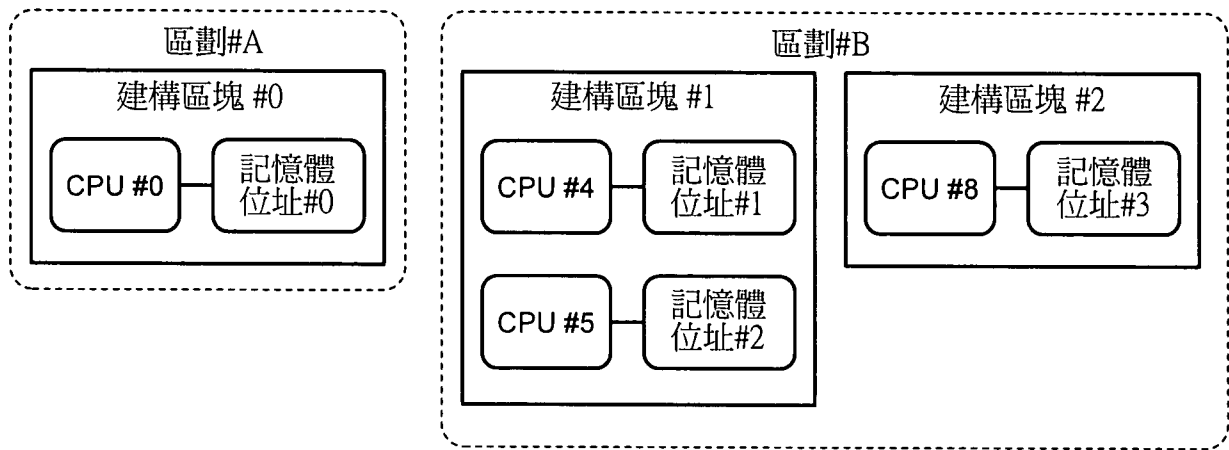


圖 25

位址	有效	節點ID	CPU ID
#0	1	0	0
#1	1	1	4
#2	1	1	5
#3	1	2	8

⋮

圖 26A

分錄	有效	節點ID	CPU ID
#0	1	0	0
#1	0		
#2	0		
#3	0		

⋮

圖 26B

分錄	有效	節點ID	CPU ID
#0	1	1	4
#1	1	1	5
#2	1	2	8
#3	0		

⋮

圖 26C