



(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 I453625 B

(45) 公告日：中華民國 103 (2014) 年 09 月 21 日

(21) 申請案號：101133281

(22) 申請日：中華民國 101 (2012) 年 09 月 12 日

(51) Int. Cl. : G06F21/62 (2013.01)

G06F21/79 (2013.01)

(30) 優先權：2011/12/28 日本

2011-290025

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

日本

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

(74) 代理人：惲軼群；陳文郎

(56) 參考文獻：

TW 201135493A1

US 5900019

US 8086585B1

審查人員：王鵬翔

申請專利範圍項數：9 項 圖式數：38 共 0 頁

(54) 名稱

資訊處理裝置及未授權存取之防止方法

INFORMATION PROCESSING APPARATUS AND UNAUTHORIZED ACCESS PREVENTION METHOD

(57) 摘要

一種資訊處理系統包括具有一第一節點及一第二節點之節點，該等節點各自包括一處理器及一記憶體，其中至少一部分區係設定作為一共用記憶體區。該第一節點儲存用以存取該第二節點中之一記憶體的識別資訊，且藉附上該識別資訊而發送通訊資料。該第二節點儲存識別資訊，用以控制許可存取在該第二節點中的該共用記憶體區，及基於附在該通訊資料的該識別資訊及儲存在該第二節點的該識別資訊而決定是否許可存取在該第二節點中的該共用記憶體區。

An information processing apparatus includes nodes having a first node and a second node each of which includes a processor and a memory in which at least a part of area is set as a shared memory area. The first node stores identification information used for accessing a memory in the second node, and transmits communication data by attaching the identification information. The second node stores identification information used for controlling permission to access the shared memory area in the second node, and determines whether or not an access to the shared memory area in the second node is permitted on the basis of the identification information that is attached to the communication data and the identification information stored in the second node.

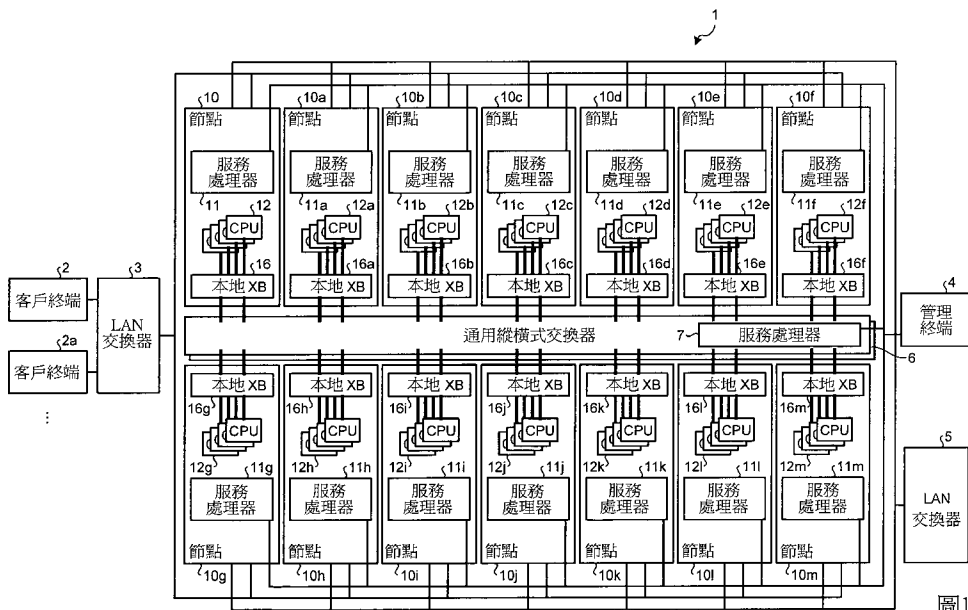


圖1

- 1 . . . 資訊處理系統
- 2、2a . . . 用戶終端
- 3、5 . . . 區域網路 (LAN)交換器
- 4 . . . 管理終端
- 6 . . . 通用縱橫式交換器
- 7、11、11a-m . . . 服務處理器
- 10、10a-m . . . 節點
- 12、12a-m . . . 中央處理單元(CPU)
- 16、16a-m . . . 本地縱橫式交換器(XB)

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：101133281

※申請日：01.9.12

※IPC 分類：

G06F 2/62 (2013.01)
G06F 2/79 (2013.01)

一、發明名稱：(中文/英文)

資訊處理裝置及未授權存取之防止方法

INFORMATION PROCESSING APPARATUS AND UNAUTHORIZED
ACCESS PREVENTION METHOD

二、中文發明摘要：

一種資訊處理系統包括具有一第一節點及一第二節點之節點，該等節點各自包括一處理器及一記憶體，其中至少一部分區係設定作為一共用記憶體區。該第一節點儲存用以存取該第二節點中之一記憶體的識別資訊，且藉附上該識別資訊而發送通訊資料。該第二節點儲存識別資訊，用以控制許可存取在該第二節點中的該共用記憶體區，及基於附在該通訊資料的該識別資訊及儲存在該第二節點的該識別資訊而決定是否許可存取在該第二節點中的該共用記憶體區。

三、英文發明摘要：

An information processing apparatus includes nodes having a first node and a second node each of which includes a processor and a memory in which at least a part of area is set as a shared memory area. The first node stores identification information used for accessing a memory in the second node, and transmits communication data by attaching the identification information. The second node stores identification information used for controlling permission to access the shared memory area in the second node, and determines whether or not an access to the shared memory area in the second node is permitted on the basis of the identification information that is attached to the communication data and the identification information stored in the second node.

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 | 7、11、11a-m...服務處理器 |
| 2、2a...用戶終端 | 10、10a-m...節點 |
| 3、5...區域網路(LAN)交換器 | 12、12a-m...中央處理單元(CPU) |
| 4...管理終端 | 16、16a-m...本地縱橫式交換器(XB) |
| 6...通用縱橫式交換器 | |

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資訊處理裝置及未授權存取之防止方法。

【先前技術】

發明背景

已知一種具有叢集組態之習知資訊處理系統，其中各自具有至少一個中央處理單元(CPU)及一記憶體的多個節點係使用區域網路(LAN)執行，或執行含括在該等節點中之無限頻帶(InfiniBand)及作業系統(OS)。使用具有此種叢集組態之資訊處理系統，各個CPU係共用節點中之一記憶體且係並列執行處理。

圖33為示意圖例示說明一種具有叢集組態之資訊處理系統。於圖33例示說明之實例中，資訊處理系統包括藉LAN或無限頻帶而彼此連結的節點50至52。節點50包括一記憶體50a、一OS 50b、及一CPU 50c。同理，節點51包括一記憶體51a、一OS 51b、及一CPU 51c。節點52包括一記憶體52a、一OS 52b、及一CPU 52c。

節點50至52分別地執行OS 50b至52b，且OS 50b至52b彼此係各自獨立，藉此防止在OS 50b至52b中之一者出現的故障傳播至整個資訊處理系統。此外，一或多個CPU 50c至52c係分別地含括於節點50至52，共用於各個節點中的記憶體50a至52a，且透過該LAN或無限頻帶而存取含括於其

它節點中的記憶體。

圖34為示意圖例示說明節點間之記憶體存取。舉例言之，節點50將由在節點50中跑的一應用程式所簽發的記憶體存取請求，透過一驅動程式、OS、及硬體而發送給節點51。藉此，節點51透過該硬體、OS、及驅動程式報告接收自該節點50的請求給在節點51中之一應用程式，且讓該應用程式執行該記憶體存取。

此時，於該具有叢集組態之資訊處理系統中，因記憶體存取係透過於欲存取的節點的應用程式執行，故記憶體存取的延遲增加，因而應用程式的架構變複雜化。據此，有一種具有共用記憶體組態之已知資訊處理系統，其中多個節點係藉互連體連結，及於各個節點中的應用程式係直接地存取記憶體，該等記憶體執行對映至相同記憶體位址空間。

圖35為示意圖例示說明一種使用共用記憶體組態技術之資訊處理系統。於圖35例示說明之實例中，節點53係藉一互連體而連結至節點54，及遠端記憶體係對映至一位址空間，該位址空間係可由節點53及節點54各自執行的應用程式直接地存取。類似於有關在各個節點中的記憶體的記憶體存取，由節點53及節點54所執行的應用程式各自簽發有關該遠端記憶體之記憶體存取請求。

此外，有一種已知技術係藉設定存取一共用記憶體區的一儲存鑰來防止未授權存取非由應用程式存取的一記憶體區。圖36為示意圖例示說明一種未授權記憶體存取之防

止功能。舉例言之，於圖36例示說明之實例中，OS事先設定一個4-位元儲存鑰在2千位元組或4千位元組的記憶體區。又復，當OS設定一程式計數器時，OS在一程式狀態字組中針對由該應用程式所使用的該記憶體區設定一儲存鑰用作為一存取鑰。

然後，一個指令單元讀取該程式計數器，及簽發記憶體存取請求給記憶體存取單元。此外，當記憶體存取單元從該指令單元獲得該記憶體存取請求時，記憶體存取單元決定設定在該程式狀態字組的存取鑰是否匹配該儲存鑰。

若該記憶體存取單元決定該存取鑰確實匹配該儲存鑰，則該記憶體存取單元許可該記憶體存取，及輸出儲存在欲存取的記憶體區之資料給該指令單元。此外，若該存取鑰並不匹配該儲存鑰，則該記憶體存取單元不允許該記憶體存取，及防止未授權存取不欲由該應用程式所存取的記憶體區。

專利文獻1：國際專利申請案第2009-537879號之日本公告案

專利文獻2：日本專利公開案第05-204762號

專利文獻3：日本專利公開案第2000-235558號

但有個問題，當由一節點所執行的應用程式出現故障而因而不可能與另一個節點進行通訊時，前述具有共用記憶體組態的資訊處理系統無法正確地執行故障切換，其中欲藉該故障節點執行的一處理係移交給另一個節點。

後文中，將參考圖37A及37B對無法正確地執行故障切

換的情況作說明。圖37A為第一示意圖例示說明其中故障切換並非正常地執行的一個實例。圖37B為第二示意圖例示說明其中故障切換並非正常地執行的一個實例。於圖37A及37B例示說明之實例中將對下述情況作說明，其中由節點55執行的應用程式已經出現問題，因而不可能與其它節點進行通訊。

舉例言之，如圖37A之例示說明，節點56從由節點55所執行的該應用程式接收記憶體存取；但因不可能與節點55進行通訊，故節點56判定該節點55已經故障。於此種情況下，如圖37B之例示說明，執行故障切換，其中由該節點55所執行的處理係移交給一節點57。

但因節點56無法與節點55通訊，故節點56不許可執行欲被停止的節點的記憶體存取從節點55的記憶體存取。據此，出現分裂腦狀態，其中節點55及節點57二者皆同時執行在相對應節點55及57中的該應用程式，因而無法正確地執行故障切換。

因而可瞭解使用在具有共用記憶體組態之該資訊處理系統中前述設定一儲存鑰的技術，及藉改變該儲存鑰，可防止從另一個節點存取在一個節點中之記憶體。但使用前述設定儲存鑰的技術，單一OS設定一儲存鑰及一應用程式鑰二者。於是，當在不同節點亦即記憶體存取源之一節點及欲存取之一節點執行OS時，因儲存鑰的不變，故難以防止從故障節點進行存取。

因此，於本發明之實施例之一個面向中，一個目的係

正確地執行在共用記憶體組態中之故障切換。

【發明內容】

發明概要

依據一實施例之一面向，一種資訊處理裝置包括多個節點具有一第一節點及一第二節點，該等節點各自包括一處理器及一記憶體，其中至少一部分區係設定作為一共用記憶體區及連結該等節點之一互連體。該第一節點包括一第一儲存單元，於該儲存單元中儲存用以存取在該第二節點之一記憶體的識別資訊；及一發送單元，其係藉附上儲存在該第一儲存單元的該識別資訊而發送欲發送的通訊資料給該第二節點。該第二節點包括一第二儲存單元，於該儲存單元中儲存識別資訊，該識別資訊係用以控制許可從另一個節點存取在該第二節點中之一記憶體內的該共用記憶體區；及一控制單元，其係基於附上至從該第一節點所發送的該通訊資料的該識別資訊及儲存在該第二儲存單元之該識別資訊而決定是否許可存取在該第二節點中之一記憶體內的該共用記憶體區。

圖式簡單說明

圖1為示意圖例示說明依據第一實施例之一種資訊處理系統；

圖2為示意圖例示說明依據第一實施例一個節點之功能組態；

圖3為示意圖例示說明依據第一實施例記憶體之一種配置；

圖4為示意圖例示說明一種記憶體對映圖其中依據第一實施例的資訊處理系統係對映至各個記憶體；

圖5為示意圖例示說明依據第一實施例藉該資訊處理系統所配置的共用記憶體區；

圖6為示意圖例示說明依據第一實施例一種共用記憶體區管理表；

圖7為示意圖例示說明依據第一實施例一種記憶體符記管理表；

圖8為示意圖例示說明依據第一實施例一種CPU之實例；

圖9為示意圖例示說明依據第一實施例針對一存取符記表之一暫存器的組態；

圖10為示意圖例示說明於該存取符記表中的暫存器欄位；

圖11為示意圖例示說明依據第一實施例一種記憶體符記檢查器之實例；

圖12為示意圖例示說明一種記憶體存取請求之實例；

圖13為示意圖例示說明一種藉操作代碼指示之處理實例；

圖14為示意圖例示說明藉一存取決定控制器進行決定之一實例；

圖15為示意圖例示說明由原位狀態機執行的處理之內容；

圖16為示意圖例示說明節點間的記憶體存取之處理流

程；

圖17為示意圖例示說明一種停止從故障節點執行記憶體存取之處理程序；

圖18為第一示意圖例示說明一種防止其中已經出現故障的一節點執行記憶體存取之處理；

圖19為第二示意圖例示說明一種防止其中已經出現故障的一節點執行記憶體存取之處理；

圖20為一序列圖例示說明一種獲得一存取符記之處理流程；

圖21為一序列圖例示說明一種存取多個節段之處理流程；

圖22為一序列圖例示說明一種脈絡切換之流程；

圖23為一序列圖例示說明一種改變一記憶體符記之處理流程；

圖24為一序列圖例示說明一種改變該記憶體符記之處理之變化例；

圖25為一序列圖例示說明一種中斷一故障節點的連結之處理流程；

圖26為流程圖例示說明一種改變記憶體符記之處理流程；

圖27為流程圖例示說明一種更新共用記憶體節段管理表及記憶體符記管理表之處理流程；

圖28為示意圖例示說明該共用記憶體節段管理表之更新；

圖29為示意圖例示說明該記憶體符記管理表之更新；

圖30為示意圖例示說明一多執行緒能存取符記表之實例；

圖31為示意圖例示說明該記憶體符記檢查器之變化例；

圖32為示意圖例示說明一記憶體符記檢查器，其只允許先前許可的節點來執行記憶體存取；

圖33為示意圖例示說明一種具有叢集組態之資訊處理系統；

圖34為示意圖例示說明節點間之記憶體存取；

圖35為示意圖例示說明一種使用共用記憶體組態技術之資訊處理系統；

圖36為示意圖例示說明一種未授權記憶體存取之防止功能；

圖37A為第一示意圖例示說明其中故障切換並非正常地執行的一個實例；及

圖37B為第二示意圖例示說明其中故障切換並非正常地執行的一個實例。

【實施方式】

較佳實施例之詳細說明

將參考附圖解釋本發明之較佳實施例。

[a] 第一實施例

於後述第一實施例中，將參考圖1描述一種資訊處理系統之實例。圖1為示意圖例示說明依據第一實施例之一種資

訊處理系統。於圖1例示說明之實例中，資訊處理系統1包括一客戶終端2、一客戶終端2a、一區域網路(LAN)交換器3、一管理終端4、一LAN交換器5、一雙重通用縱橫式交換器6、及節點10至10m。通用縱橫式交換器6包括一服務處理器7。

於該第一實施例中，一種節點是一個單位，其中跑單一獨立的作業系統(OS)，且只包括用以跑該OS的最少需要裝置。容後詳述，該節點包括一或多個CPU、記憶體、及硬碟驅動裝置(HDD)或替代該HDD的外部儲存裝置。該節點也可為單一實體裝置或可為實體上從多個裝置藉使用纜線連結組成。

節點10至10m係透過服務LAN而連結至LAN交換器3及客戶終端2及2a。此外，節點10至10m係透過內部LAN而連結至LAN交換器5。此外，節點10至10m係透過管理LAN而連結至管理終端4。又復，節點10至10m係連結至雙重通用縱橫式交換器6，且係透過互連而彼此連結。

客戶終端2及2a各自係透過服務LAN而獲得來自節點10至10m中之各者的服務。雖然並未例示說明於圖1，資訊處理系統1也可包括任意數目的額外客戶終端。

節點10包括一服務處理器11、一CPU 12、及一本地縱橫式交換器(XB)16。類似節點10，其它節點10a至10m分別地包括服務處理器11a至11m、CPU 12a至12m、及本地XB 16a至16m。

服務處理器11至11m各自係透過管理LAN而連結至管

理終端4及執行管理控制，諸如於節點10至10m各自的電源供應器的管理。含括在通用縱橫式交換器6的服務處理器7係操作為服務處理器11至11m的主站且執行總體管理控制。由服務處理器7、及11至11m執行的管理控制之實例包括電源供應器管理、復置、操作模式的改變、節點之增刪的設定、錯誤日誌的集合、及簡併性之操作控制。

此外，節點10至10m獨立地執行作業系統(OS)。雖然並未例示說明於圖1，但節點10至10m各自分別地包括CPU 12至12m以外的多個CPU。節點10至10m中的CPU分別地係透過本地XB 16至16m而連結至通用縱橫式交換器6。

於節點10至10m中的CPU輸出記憶體存取請求亦即請求存取記憶體給個別本地XB 16至16m，藉此CPU轉移記憶體存取請求給其它節點。然後，於節點10至10m中的CPU獲得一記憶體存取請求，該等CPU係以類似CPU存取連結至本身CPU的記憶體時所執行的類似方式執行記憶體存取。

後文中，將參考圖2說明節點10至10m之功能組態。於後文說明中，將描述節點10的功能組態。節點10a至10m執行與藉節點10執行的相同處理；因此將刪除其說明。

圖2為示意圖例示說明依據第一實施例一個節點之功能組態。於圖2例示說明之實例中，節點10包括該服務處理器11、CPU 12至15、本地XB 16、記憶體17至24、及一周邊組件互連體快速(PCIe)交換器25。又復，節點10包括連結至該服務LAN之一LAN配接器26、連結至該管理LAN之一

LAN配接器27、一串列附件SCSI (SAS)28、及一硬碟驅動裝置(HDD)29。

記憶體17及記憶體18係連結至CPU 13且係為儲存裝置接收由CPU 13所執行的記憶體存取。記憶體19及記憶體20係連結至CPU 12且係為儲存裝置接收由CPU 12所執行的記憶體存取。記憶體21及記憶體22係連結至CPU 14且係為儲存裝置接收由CPU 14所執行的記憶體存取。

記憶體23及記憶體24係連結至CPU 15且係為儲存裝置接收由CPU 15所執行的記憶體存取。於圖2例示說明之實例中，節點10包括HDD 29；但實施例並非圍限於此。舉例言之，HDD 29也可藉使用諸如儲存區域網路(SAN)技術等技術而配置在節點10外部。

服務處理器11執行節點10的管理控制。更明確言之，服務處理器11包括一處理器及執行處理的一可現場程式規劃陣列(FPGA)，且透過管理LAN而從管理終端4接收執行管理控制的指令。然後，服務處理器11依據接收自管理終端4的指令而從事各項管理控制。

PCIe交換器25為一交換器控制存取至例如透過PCIe卡槽而連結至各個CPU 12至15的一I/O裝置。LAN配接器26為連結服務LAN至節點10的LAN配接器。LAN配接器27為連結管理LAN至節點10的LAN配接器。SAS 28為LAN或安裝在PCIe卡槽之SAS的一配接器，且中繼HDD 29與CPU 12至15各自間的連結。

後文中，將描述CPU 12至15。於後文說明中，將描述

CPU 12。CPU 13至15具有與由CPU 12所執行的相同功能，故將刪除其說明。

CPU 12與其它CPU 13至15全部皆彼此連結。此外，CPU 12包括快取由記憶體19或記憶體20所儲存的資料之一快取記憶體，且係藉使用所快取的資料來執行算術處理。又復，CPU 12具有一記憶體介面，執行針對記憶體19及記憶體20各自的記憶體存取。此外，CPU 12具有PCIe介面功能，其係藉使用HDD 29、內部LAN、或管理LAN透過PCIe交換器25而控制存取。

又，CPU 12透過本地XB 16及通用縱橫式交換器6而藉互連來連結至包括在其它節點10a至10m的CPU，且具有一互連路由器功能，其發射與接收介於各個CPU間的記憶體存取請求。當例如藉欲執行的處理而簽發一記憶體存取請求給記憶體19或記憶體20時，具有此種組態的CPU 12執行下列處理。更明確言之，CPU 12存取記憶體19或記憶體20且快取欲存取的資料。

此外，當藉欲執行的處理而簽發一記憶體存取請求給記憶體19及記憶體20以外之一記憶體時，CPU 12識別連結至屬於記憶體存取目標的一記憶體的該CPU，且發送該記憶體存取請求給該經識別的CPU。更明確言之，CPU 12具有一節點對映圖，其中一欲存取的記憶體位址係與獨特地指示存取由該記憶體位址所指示的一儲存區之該CPU的資訊相連結。

然後，若一處理簽發一記憶體存取請求，則該CPU 12

參考該節點對映圖，且識別連結至屬於記憶體存取目標的一記憶體的該CPU。隨後，CPU 12產生一封包，該封包的目的地是所識別的CPU，將該記憶體存取請求存取於該所產生的封包，且輸出該封包給彼此連結的CPU 12至15或輸出給本地XB 16。若該封包係輸出給本地XB 16，則本地XB 16透過通用縱橫式交換器6發送該封包給與該封包的目的地相對應的一CPU。

此時，除了前述處理外，CPU 12執行下列處理。首先，CPU 12事先該儲存區設定稱作為一記憶體符記的一值，該值係與在記憶體19與記憶體20間的其它節點共享。然後，CPU 12通知CPU簽發一記憶體存取請求給許可存取的一儲存取的記憶體19或記憶體20，且通知該CPU與該被通知區相連結的一記憶體符記作為一存取符記。

此外，CPU 12先前從CPU 12對其發送一記憶體存取請求的一節點獲得一存取符記且保有所得的存取符記。然後，若CPU 12簽發一記憶體存取請求，則CPU 12儲存所保有的存取符記連同該記憶體存取請求於一封包且發送該封包。

又復，若CPU 12從在另一個節點的一CPU接收一記憶體存取請求，則CPU 12決定連同該記憶體存取請求所接收的存取符記是否匹配與欲存取的該儲存區相連結的的該記憶體符記。若該存取符記係匹配該記憶體符記，則CPU 12執行該記憶體存取；但若該存取符記係不匹配該記憶體符記，則CPU 12拒絕該記憶體存取的執行。

若CPU 12在記憶體19或記憶體20上執行記憶體存取，則CPU 12執行保有藉記憶體19或記憶體20所快取的資料之同調處理。舉例言之，CPU 12發送一監聽給已經從記憶體19或記憶體20快取資料的該CPU且執行所快取資料的回寫。

本地XB 16為一交換器，其係將在含括於節點10的該等CPU 12至15與含括在其它節點10a至10m中的該等CPU間交換的一封包傳送至規定目的地。舉例言之，本地XB 16將由CPU 12所簽發且欲傳輸給CPU 12a的該封包透過通用縱橫式交換器6而發送至含括在節點10a的該CPU 12a。

記憶體17至記憶體24為其中儲存由各個OS或應用程式所使用的資料之記憶體，且為例如雙排式記憶體模組(DIMM)。又復，記憶體17至24對映相同的記憶體位址空間。

此外，記憶體17至24各自具有一區其只能由同一個節點內的CPU 12至15存取，亦即一本地區其係給含括CPU 12至15的該節點所專用的一區。此外，記憶體17至24各自具有一區可由含括在資訊處理系統1中的任意CPU所存取，亦即具有一共用記憶體區其可由全部節點所使用。又，針對記憶體17至24的該區係劃分成多個節段。各個節段係為一保護區，其中存取控制係藉一記憶體符記進行，或為一未經保護區其中存取控制係不藉一記憶體符記進行。

後文中，將參考圖3及4說明對映於記憶體17至24的記憶體位址。圖3為示意圖例示說明依據第一實施例之記憶體配置。例如，在含括於各個CPU的兩個記憶體間，資訊處

理系統1使用一個記憶體作為只用於本地區的一記憶體，而使用另一個記憶體作為只用於共用記憶體區的一記憶體。更明確言之，如圖3之(A)例示說明，記憶體17、記憶體19、記憶體21、及記憶體23係用作為該等節點專用的記憶體；及如圖3之(B)例示說明，記憶體18、記憶體20、記憶體22、及記憶體24係用作為共用記憶體。

圖4為示意圖例示說明一種記憶體對映圖，其中依據第一實施例的資訊處理系統係對映至各個記憶體。圖4例示說明一個實例，其中共製備128-兆位元組記憶體位址空間，而128兆位元組中的64兆位元組係分派給本地區，而128兆位元組中的64兆位元組係分派給共用記憶體區。

舉例言之，資訊處理系統1分派該區中的「0」至「 $2^{46}-1$ 」記憶體位址用作為含括在節點10至10m的記憶體內部的本地區。又復，資訊處理系統1分派該區中的「 2^{46} 」至「 $2^{47}-1$ 」記憶體位址用作為含括在節點10至10m的記憶體內部的共用記憶體區。

更明確言之，於圖3及4例示說明之實例中，資訊處理系統1將如圖4之(C)例示說明的該區中的記憶體位址，對映至如圖3之(A)例示說明的記憶體17、記憶體19、記憶體21、及記憶體23上。此外，資訊處理系統1將如圖4之(D)例示說明的該區中的記憶體位址，對映至如圖3之(B)例示說明的記憶體18、記憶體20、記憶體22、及記憶體24上。圖4例示說明的「 2^{47} 」至「 $2^{48}-1$ 」位址乃用作為無法快取(UC)區的位址範圍區，且係為例如由I/O裝置諸如HDD 29所使用的

I/O空間之位址範圍。

如前文說明，資訊處理系統1將單塊記憶體位址空間對映至含括在節點10至10m的記憶體上。據此，任意的CPU可直接地存取在任意節點的共用記憶體區。圖4例示說明的記憶體對映圖僅為其中一例；因此也可執行任意地分派。

更明確言之，無需分派在該記憶體對映圖上的全部記憶體位址給該等記憶體。舉例言之，也可存在有記憶體孔。此外，該記憶體對映圖也可經設定使得由OS所使用的本地區係存在於含括於資訊處理系統1的各個節點，而共用記憶體區係存在於一或多個節點。

若資訊處理系統1將連續記憶體位址對映至含括在單一節點的該等記憶體(例如記憶體17及記憶體18)，則資訊處理系統1將可刪除控制資源。此外，資訊處理系統1係分派記憶體位址使得節段區至少為對齊。

後文中，將參考圖5說明於各個節點中分派給該等記憶體的共用記憶體區。圖5為示意圖例示說明依據第一實施例由資訊處理系統所分派的共用記憶體區。於圖5例示說明之實例中，將描述一種情況其中資訊處理系統1包括節點#0至#n，且分配共用記憶體區給節點#0至#n。

舉例言之，如圖5之(1)例示說明，資訊處理系統1分派一64-兆位元組本地區及一64-兆位元組共用記憶體區給節點#0至#n中之各者。更明確言之，如圖5之(2)例示說明，資訊處理系統1均等地分配64-兆位元組共用記憶體區給節點#0至#n中之各者。

此時，節點#0至#n中之各者將所分配的共用記憶體區劃分成一保護區及一未經保護區。此處所述保護區乃一區，當一記憶體存取請求係接收自另一個節點時藉由決定存取符記是否匹配記憶體符記而來保護存取。未經保護區乃一區其中另一個節點可自由地從事記憶體存取。

舉例言之，如圖5之(3)例示說明，針對分派給節點#1的共用記憶體區，節點#1可使用至多4兆位元組作為一保護區或一未經保護區。此外，如圖5之(4)例示說明，節點#1將該共用記憶體區劃分成具有2個十億位元組的相對應於節段#0至#2047的多個節段。然後如圖5之(5)例示說明，節點#1針對各個節段設定一個記憶體符記。

又復，於圖5例示說明之實例中，節點#1使用節段#0至#2作為保護區，且設定一記憶體符記於各個節段。記憶體符記例如為13-位元件。於圖5例示說明之實例中，節點#1設定節段#3至#2047作為未經保護區，且設定「a110」作為記憶體符記。更明確言之，當節點#0至#n中之各者存取相對應的未經保護區時，節點可藉使用「a110」的存取符記來存取未經保護區。

此外，節點#m也將其本身的共用記憶體區設定為節段#0至#2047，且在節段#0至#2047中之各者設定一記憶體符記。如圖5之(6)例示說明，藉節點#m設定在節段#0至#2047中之各者的記憶體符記係由節點#1用作為存取符記。據此，當節點#1存取節點#m中的共用記憶體區時，節點#1獲得由節點#m所設定的記憶體符記，且使用所得記憶體符記

作為存取符記，藉此節點#1存取節點#m中的共用記憶體區。

節點#0至#n中之各者在該本地區包括一共用記憶體節段管理表，該管理表係用來管理在各個共用記憶體區中的各個節段。又復，為了防止相同記憶體符記被設定在多個節段，節點#0至#n中之各者在該本地區包括一記憶體符記管理表，該管理表係用來管理已被設定的記憶體符記。後文中，將參考圖6及7說明說共用記憶體節段管理表及記憶體符記管理表。

首先，將參考圖6說明共用記憶體節段管理表之一個實例。圖6為示意圖例示說明依據第一實施例之一種共用記憶體節段管理表。圖6例示說明含括於節點#2的該共用記憶體節段管理表。如圖6之例示說明，該共用記憶體節段管理表於其中儲存一節段號碼、一使用旗標、一應用程式ID、一使用節點旗標、及一符記。

節段號碼為指示共用記憶體區中各個節段的號碼。使用旗標為指示各個節段是否使用的旗標。應用程式ID為獨特地指示使用各個節段的一項處理的ID。使用節點旗標為一旗標指示使用各個節段之一節點。符記為設定在各個節段之一記憶體符記。

於圖6例示說明之實例指示於節點#2中的節段#0及節段#1係由節點#1執行的各項處理中藉應用程式ID「100」指示之一處理使用，且指示記憶體符記分別地為「3015」及「2156」。此外，實例指示不使用節點#2中的節段#2，節段#3係由節點#3執行的各項處理中藉應用程式ID「150」指示

之一處理使用，且指示記憶體符記分別地為「a110」，亦即節段#3為未經保護區。

節點#0至#n各自藉使用圖6例示說明的共用記憶體節段管理表而執行接收自其它節點的記憶體存取。例如，節點#2連同得自節點#1中由應用程式ID「100」指示的該處理之存取符記「3015」，就節段#0獲得一記憶體存取請求。於此種情況下，由於所得存取符記匹配儲存在共用記憶體節段管理表的記憶體符記，故節點#2執行記憶體存取。

相反地，節點#2連同得自節點#1中由應用程式ID「100」指示的該處理之存取符記「3000」，就節段#0獲得一記憶體存取請求，節點#2決定記憶體符記不匹配該存取符記。因此節點#不執行該記憶體存取請求。

又復，節點#0至#n各自改變於該等節段中之一者內的記憶體符記，節點#0至#n各自係指該共用記憶體節段管理表，且係識別正在使用與該欲改變的記憶體符記相聯結的該節段之一節點。然後，節點#0至#n各自通知所識別的節點一個新的記憶體符記。

共用記憶體節段管理表也可藉一共用記憶體驅動程式設定，該共用記憶體驅動程式為任意地管理該共用記憶體的驅動程式軟體。舉例言之，當資訊處理系統1被啟動時，或當首次存取共用記憶體區時，共用記憶體驅動程式產生一共用記憶體節段管理表。

後文中，記憶體符記管理表將參考圖7作說明。圖7為示意圖例示說明依據第一實施例之一種記憶體符記管理

表。於圖7例示說明之實例中，記憶體符記管理表以相聯結的方式在其中儲存一記憶體符記之一值、一使用旗標、及一應用程式ID。圖7例示說明由節點#1所儲存的記憶體符記管理表。

舉例言之，於圖7例示說明之記憶體符記管理表實例指示記憶體符記「0」及記憶體符記「1」係分配給由應用程式ID「100」所指示的該等處理所使用的節段。此外，記憶體符記管理表指示記憶體符記「2」係不分配給該等節段。此外，記憶體符記管理表指示記憶體符記「3」係分配給由應用程式ID「150」所指示的該等處理所使用的節段。類似於共用記憶體節段管理表，共用記憶體驅動程式也可任意地產生記憶體符記管理表。

於後文中，將參考圖8描述CPU 12的特定功能組態之一實例。圖8為示意圖例示說明依據第一實施例之一種CPU實例。於圖8例示說明之實例中，假設CPU 12為含括多個核心之一多核心處理器，且為含有一層級1(L1)快取記憶體、一L2快取記憶體、及一記憶體存取控制器(MAC)的一處理器。

如圖8之例示說明，CPU 12包括多個核心30a及30d、L1快取記憶體30b及30e、及多個轉譯後備緩衝器(TLB)30c及30f。此外，CPU 12包括分別地針對核心30a及30d的存取符記表31及31a。此外，CPU 12包括一層級2(L2)快取標籤32、一L2快取記憶體32a、一快取監聽控制器33、一記憶體請求狀態機34、一位址對映圖35、一記憶體存取埠36、及一目標位址檢查器37。

此外，CPU 12包括一記憶體符記檢查器38、一原位狀態機39、一記憶體存取控制器(MAC)40、一非快取請求狀態機41、一PCIe介面42、及一錯誤暫存器43。此外，CPU 12包括一暫存器控制器44及一路由器46。該暫存器控制器44包括一請求暫存器45。核心30a及核心30d以外的核心並未例示說明於圖8；但CPU 12可包括任意數目的核心。

核心30a及30d藉使用儲存在L1快取記憶體30b及30e的資料而執行算術處理。L1快取記憶體30b及30e為可以高速存取的儲存裝置，且係為快取儲存在記憶體19及20的資料之快取記憶體。TLB 30c及30f將由核心30a及30d所輸出的虛擬位址轉換成實體位址。

存取符記表31為一暫存器，其中儲存一存取符記用來得藉該核心30a執行的處理存取該共用記憶體區。舉例言之，由另一個節點使用由核心30a所執行的應用程式通知的一存取符記係設定在存取符記表31。

若欲存取的資料並不保有在L1快取記憶體30b，則核心30a輸出該目標存取的一實體位址及指示一記憶體存取請求的一操作代碼。然後，存取符記表31輸出所設定的存取符記及記憶體存取請求給L2快取標籤32。

存取符記表31為一暫存器，當藉該核心30a所執行的處理改變時，該暫存器執行脈絡切換。據此，當CPU 12將藉該核心30a執行的處理改變成另一項處理時，CPU 12容易地改變設定在存取符記表31中的存取符記。更明確言之，當CPU 12改變藉該核心30a執行的處理時，CPU 12藉由只執

行脈絡切換而未自欲存取的節點再度接收到該存取符記表的通知，CPU 12可容易地且快速地改變該存取符記。

L2快取標籤32係針對L2快取記憶體32a的一標籤，且係用來存取L2快取記憶體32a的一管線。此外，就由核心30a及30d所簽發的相同位址而言，L2快取標籤32具有在記憶體存取請求上執行序列化處理的機轉。

快取監聽控制器33控制L2快取記憶體32a的監聽。更明確言之，快取監聽控制器33接收由原位狀態機39或其它節點所簽發的一監聽。然後，藉簽發的一監聽給L2快取標籤32，快取監聽控制器33檢查欲藉該L2快取記憶體32a快取的該資料之狀態。隨後，快取監聽控制器33發送L2快取記憶體32a的監聽結果給屬於該監聽之簽發源的原位狀態機39或發送給其它節點。

若欲存取的資料係不儲存在L2快取記憶體32a，亦即在L2快取記憶體32a出現快取失誤時，記憶體請求狀態機34執行下列處理。更明確言之，記憶體請求狀態機34簽發一記憶體存取請求給原位狀態機39，且管理一項處理直到獲得回應為止。

此外，在從記憶體存取請求簽發時直到獲得回應的時間週期期間，若記憶體請求狀態機34獲得靶定於相同記憶體位址的該記憶體存取請求，則記憶體請求狀態機34讓接續的記憶體存取請求等候。然後，若簽發對前一個記憶體存取請求的回應，則記憶體請求狀態機34簽發變成等候中的一記憶體存取請求。此外，記憶體請求狀態機34包括多

個分錄，將針對各個分錄的一識別符8-位元分錄ID附接至一記憶體請求，及簽發該記憶體請求。

此外，藉由輸出相對應於針對該記憶體存取請求之一目標的一實體位址給位址對映圖35，記憶體請求狀態機34獲得CPU的識別符，存取由該實體位址所指示之一儲存區。然後，記憶體請求狀態機34藉使用該CPU的識別符作為目的地ID(DID)亦即目的地而簽發一記憶體存取請求。

位址對映圖35以相聯結的方式於其中儲存一實體位址及一CPU的識別符，該CPU的識別符存取由該實體位址所指示之一儲存區，且將該實體位址轉換成該CPU的識別符。舉例言之，當位址對映圖35從記憶體請求狀態機34獲得一實體位址時，位址對映圖35簽發該CPU的識別符，其係儲存在所得實體位址且與該實體位址相聯結，給記憶體請求狀態機34作為目的地ID(DID)。

記憶體存取埠36為就記憶體19或記憶體20接收一記憶體存取請求的一埠，亦即就一原位節點(其本身的節點)接收一記憶體存取請求。舉例言之，記憶體存取埠36為一FIFO且就一原位節點保有一記憶體存取請求。若原位狀態機39變成其中可執行一記憶體存取請求的狀態，則記憶體存取埠36簽發所保有的記憶體存取請求給目標位址檢查器37、記憶體符記檢查器38、及原位狀態機39。

目標位址檢查器37從記憶體存取埠36決定相對應於針對該記憶體存取請求的目標存取之一記憶體位址是否為其本身節點中的該記憶體的記憶體位址，亦即記憶體19或記

憶體20的記憶體位址。然後，目標位址檢查器37通知原位狀態機39有關該決定結果。

記憶體符記檢查器38決定設定在相對應於針對該記憶體存取請求的目標存取之一記憶體位址的一記憶體符記是否匹配連同該記憶體存取請求所簽發的一存取符記。若該記憶體符記係匹配該存取符記，則記憶體符記檢查器38簽發指示核准該存取之一信號給該原位狀態機39。此外，若該記憶體符記係不匹配該存取符記，則記憶體符記檢查器38簽發指示不許可該存取之一信號給該原位狀態機39。

記憶體符記檢查器38包括一記憶體符記暫存器，其中係儲存一記憶體符記。暫存器控制器44設定或更新該記憶體符記。

取決於連同該記憶體存取請求所簽發的一存取符記是否匹配一記憶體符記，原位狀態機39執行記憶體存取。更明確言之，原位狀態機39從記憶體存取埠36獲得一記憶體存取請求。此外，原位狀態機39從目標位址檢查器37獲得一決定結果，指示該記憶體存取請求的目標是否為其本身節點。此外，原位狀態機39從該記憶體符記檢查器38獲得一信號指示核准存取或一信號指示不許可存取。

然後，若原位狀態機39獲得一決定結果指示該記憶體存取請求的目標係為其本身節點且獲得指示核准存取之一信號，則原位狀態機39執行該記憶體存取。相反地，若原位狀態機39獲得一決定結果指示該記憶體存取請求的目標並非為其本身節點或獲得指示不核准存取之一信號，則原

位狀態機39不執行該記憶體存取而執行存取例外處置。此外，若原位狀態機39獲得一信號指示不許可存取，則原位狀態機39在錯誤暫存器43內儲存符記不匹配錯誤，指示該記憶體符記係不匹配該存取符記。

此外，若原位狀態機39執行記憶體存取，則原位狀態機39執行下列處理。首先，原位狀態機39簽發給記憶體存取控制器40靶定用於記憶體存取請求的一實體位址，及從記憶體19或記憶體20獲得靶定用於記憶體存取請求的資料。然後，原位狀態機39藉使用目錄資訊而決定所得資料的快取狀態。

然後，原位狀態機39依據該決定結果而簽發一監聽等，且執行處理使得所得資料係匹配該快取資料。保有同調之習知方法係用於所得資料匹配該快取資料之處理。然後，原位狀態機39執行記憶體存取，諸如讀或寫，及然後，傳輸有關該記憶體存取為完成的回應給該簽發源。

記憶體存取控制器40執行有關記憶體19及記憶體20的記憶體存取。舉例言之，記憶體存取控制器40從該原位狀態機39獲得一指令來讀取一實體位址及資料。然後，記憶體存取控制器40將儲存在由該實體位址指示的儲存區之該資料發送給原位狀態機39。

此外，當記憶體存取控制器40從該原位狀態機39獲得一實體位址、一寫指令、及資料時，記憶體存取控制器40將所得資料寫在由該實體位址所指示的儲存區。記憶體存取控制器40也可具有巡查功能或執行處理諸如正確寫入功

能。

非快取請求狀態機41從包括於CPU 12的核心30a及30d或包括於其它CPU中之一者的一核心接收得自一無快取(UC)區的一記憶體位址的一指令，且透過PCIe執行一處理之設定。

舉例言之，若核心30a及30d各自含有就一I/O裝置諸如HDD 29的一讀或寫請求，則非快取請求狀態機41簽發所得請求給PCIe介面42。然後，若非快取請求狀態機41接收到對來自PCIe介面42的該請求的回應，則非快取請求狀態機41將所得回應發送給核心30a及30d。

又復，非快取請求狀態機41從核心30a及30d中之各者獲得有關暫存器控制器44之一請求。於此種情況下，非快取請求狀態機41簽發所得請求給暫存器控制器44。此外，若非快取請求狀態機41獲得對來自該暫存器控制器44之該請求的一回應，則非快取請求狀態機41發送所得回應給核心30a及30d。

此外，若非快取請求狀態機41透過該暫存器控制器44，從服務處理器11獲得針對該位址對映圖之一設定請求，則非快取請求狀態機41具有依據所得請求設定位址對映圖35的功能。

PCIe介面42係連結至PCIe交換器25且係為PCIe的根複合體。舉例言之，當PCIe介面42從非快取請求狀態機41獲得有關I/O裝置等的一請求，PCIe介面42透過PCIe交換器25簽發所得請求給I/O裝置等。然後，當PCIe介面42透過PCIe

交換器25獲得一回應時，PCIe介面42發送所得回應給該非快取請求狀態機41。

此外，PCIe介面42具有直接記憶體存取(DMA)功能且可簽發一DMA請求給一任意CPU。此外，當PCIe介面42從其它節點獲得一DMA請求時，該PCIe介面42執行該DMA請求，及然後簽發一DMA回應給該請求源。

錯誤暫存器43乃一暫存器，其保有出現在CPU 12的錯誤資訊。暫存器控制器44為控制器，其寫或讀該CPU 12內的暫存器。舉例言之，錯誤暫存器43具有用以將一實體位址轉成一暫存器位址的功能。

然後，暫存器控制器44從該非快取請求狀態機41等，獲得有關由該實體位址所載明之該暫存器的一讀請求或一寫請求。其後，暫存器控制器44將所載明的實體位址轉換成該暫存器位址，且在由該暫存器位址所載明的暫存器上執行寫或讀。

此外，暫存器控制器44包括請求暫存器45，且透過內置積體電路(I2C)或聯合測試行動小組(JTAG)，而從服務處理器11獲得有關該請求暫存器45的一讀請求或一寫請求。於此種情況下，暫存器控制器44執行所得請求，及輸出一回應給服務處理器11。例如當暫存器控制器44接到指令透過非快取請求狀態機41，藉由該核心30a或30d所執行之一處理而改變該記憶體符記時，暫存器控制器44更新一記憶體符記暫存器，容後詳述。

路由器46乃互連體的路由器，及發送一封包給由目的

地ID(DID)所載明的該目的地CPU。此外，當路由器46傳輸一封包時，路由器46附上一循環冗餘檢查(CRC)值來保護欲傳輸的封包。

於圖8例示說明之實例中，監聽係以「監聽」表示，回應係以「回應」或「Rsp」表示，實體位址係以「位址」表示，及記憶體符記或存取符記係以「符記」表示。此外，於圖8例示說明之實例中，操作代碼係以「op code」表示。此外，由記憶體請求狀態機34所簽發之一記憶體存取請求，亦即來自其本身節點的一請求係以「原位請求」表示。

此外，記憶體存取請求係以「MEM Req」表示，而對該記憶體存取請求之一回應係以「MEM Rsp」表示。更明確言之，接收自其本身節點中之一記憶體的對該記憶體存取請求之一回應係以「來自本身的MEM Rsp」表示，而接收自其它節點中的一記憶體之一回應係以「來自其它的MEM Rsp」表示。此外，對來自其它節點的一記憶體存取請求之一回應係以「對其它的MEM Rsp」表示。

此外，來自其它節點的監聽係以「來自其它的監聽」表示，欲傳輸給其它節點的監聽係以「對其它的監聽」表示。此外，簽發給在其本身節點的L2快取記憶體之監聽係以「對本身的監聽」表示，而對簽發給在其本身節點的L2快取記憶體的監聽之一回應係以「對本身的監聽回應」表示。此外，對已經傳輸給其它節點的監聽之一回應係以「來自其它的監聽回應」表示，而對欲傳輸給其它節點的監聽之一回應係以「對其它的監聽回應」表示。

又復，由非快取請求狀態機41傳輸之一請求及一回應係以「NC(非同調)請求/NC回應」表示。有關一暫存器的讀請求及寫請求及由該暫存器控制器44所發送與接收的回應係以「REG R/W REQ及回應」表示。此外，由該暫存器控制器44所執行的一暫存器之讀或寫係以「暫存器R/W及回應」表示。透過PCIe介面42的請求及回應係以「PCI CNFG Req及回應」表示。此外，DMA請求係以「DMA請求」表示，及DMA回應係以「DMA回應」表示。

後文中，將參考圖9說明含括於CPU 12中作為暫存器的存取符記表31。圖9為示意圖例示說明依據第一實施例針對一存取符記表之一暫存器的組態。如圖9之例示說明，CPU 12包括多個位址空間識別符(ASI)暫存器47至47c，於其中資料可從核心30a直接寫入。此外，CPU 12運用ASI暫存器中之一者作為該存取符記表31。

圖10為示意圖例示說明於該存取符記表中的暫存器欄位。於圖10例示說明之實例中，存取符記表31具有含「0」至「31」位元的32-位元欄位，且使用從「31」至「13」的位元作為反相欄位。此外，存取符記表31使用從「12」至「0」位元作為含有存取符記的欄位。

此時，存取符記表31使用相對應於從「12」至「0」位元的13位元作為存取符記。更明確言之，當存取符記表31設定存取符記具有相等數目的節段時，存取符記表31並不設定一存取符記為無存取。據此，存取符記表31設定存取符記，其數目係大於節段的數目。

此處將回頭參考圖9作說明。對ASI暫存器47至47c中之各者及存取符記表31，核心30a簽發一「ASI Reg R/W指令」、一「位址[11:2]」、及一「位元組致能[3:0]」。此處所述「ASI Reg R/W指令」乃讀或寫ASI暫存器47至47c中之各者及存取符記表31的一指令。

此處所述「位址[11:2]」乃一虛擬位址，其為記憶體存取請求的目標。又復，此處所述「位元組致能[3:0]」表示指示致能態的位元。當核心30a指示ASI暫存器47至47c中之各者及存取符記表31作寫入時，核心30a同時簽發該「ASI Reg寫入資料[31:0]」。

當ASI暫存器47至47c各自獲得一讀取指令時，ASI暫存器47至47c各自簽發「讀取資料Reg A [31:0]」至「讀取資料Reg X [31:0]」。然後，存取符記表31輸出「讀取資料Reg AKT [0:31]」。隨後，核心30a基於「讀取資料Reg A [31:0]」、「讀取資料Reg X [31:0]」、及「讀取資料Reg AKT [0:31]」而獲得「讀取資料 [31:0]」。

此時，當存取符記表31獲得一讀取指令時，存取符記表31輸出該存取符記[12:0]。當L1快取記憶體30b從核心30a接收到「MEM REQ」其為記憶體存取請求時，L1快取記憶體30b決定是否出現快取命中。若出現快取命中，則L1快取記憶體30b發送其中已經出現快取命中的資料給核心30a作為「MEM REQ L1 HIT Rsp」。

相反地，若已經發生快取失誤，則L1快取記憶體30b發送一記憶體存取請求給TLB 30c。TLB 30c將屬於該記憶

體存取請求的目標的一虛擬位址轉換成一實體位址，及簽發一記憶體存取請求給該L2快取標籤32。L2快取標籤32決定在相對應於該記憶體存取請求的目標的該實體位址是否出現快取命中。若已經出現快取命中，則L2快取標籤32透過該TLB 30c傳輸該資料給L1快取記憶體30b及核心30a。

相反地，若已經出現快取失誤，則L2快取標籤32發送一記憶體存取請求給該記憶體請求狀態機34。此時，由該存取符記表31所簽發的存取符記[12:0]係附上在該記憶體存取請求。然後，該記憶體存取請求係發送給記憶體請求狀態機34作為「原位請求」。由核心30a所簽發的「NC請求」係藉非快取請求狀態機41透過TLB 30c執行，及「NC回應」係簽發給核心30a。

於後文中，將參考圖11敘述記憶體符記檢查器38之組態實例。圖11為示意圖例示說明依據第一實施例一種記憶體符記檢查器之實例。於圖11例示說明之實例中，記憶體符記檢查器38包括一自身節點ID暫存器38a、一記憶體符記暫存器38b、一操作代碼符記解碼器38c、及一存取決定控制器38d。於圖11例示說明之實例中，將對其中單一節點包括四個CPU之情況作說明。此外，假設記憶體符記暫存器38b具有512分錄且具有與圖10例示說明的存取符記暫存器相同組態。

舉例言之，記憶體符記檢查器38從暫存器控制器44獲得「Reg R/W指令」、「位址[11:2]」、「位元組致能[3:0]」、及「寫入資料[31:0]」。存取符記表31含有來自暫存器控制器

44的「發送方ID(SID)[7:3]」。此處所述「SID [7:3]」為指示相對應於該記憶體存取請求的請求源之一CPU的識別符。

又復，記憶體符記檢查器38從記憶體存取埠36獲得一記憶體存取請求。圖12為示意圖例示說明一種記憶體存取請求之實例。如圖12之例示說明，該記憶體存取請求之一封包含有一格式、一目的地ID、一操作代碼、一傳輸源ID、一分錄ID、一符記、及一目標位址。該格式含有4-位元資訊指示該封包格式的类型別。

目的地ID含有指示一目的地CPU的一ID，亦即8-位元DID。操作代碼於其中儲存4-位元資訊指示該記憶體存取請求的內容。該符記於其中儲存16-位元存取符記。該目標位址含有屬於該記憶體存取請求的目標之一實體位址。除了圖12例示說明之封包外，用以寫位元組致能的資料係傳輸給該記憶體存取請求，該請求為資料之寫入請求。

此外，針對不需要存取符記的一記憶體存取請求，例如一回閃指令及一回寫指令，傳送其中使用保留作為符記之一封包。替代一符記，監聽於其中儲存指示請求源CPU之一8-位元CPU ID及獨特地指示一請求源處理之一8-位元分錄ID。

此外，具有與圖12例示說明的封包類似格式之一封包係發送給該記憶體存取請求作為一回應。舉例言之，替代一符記，回應於該監聽係儲存8-位元監聽回應資訊。此外，其中使用保留作為符記之一封包係傳輸作為對該記憶體存取請求之一完成回應，及該讀取資料係附上對一讀取請求

的一回應。

此處將回頭參考圖11作說明。記憶體符記檢查器38從所得記憶體存取請求獲得SID [7:0]，且決定所得SID是否匹配包括於一自身節點的一CPU的ID相對應的一自身節點ID。於圖11例示說明之實例中，記憶體符記檢查器38決定四個CPU的ID，亦即CPU ID0至CPU ID3是否匹配SID。然後，記憶體符記檢查器38將該決定結果輸入存取決定控制器38d作為「我的節點命中」。

又復，記憶體符記檢查器38從該記憶體存取請求獲得操作代碼(op code [3:0])，且將該操作代碼輸入操作代碼符記解碼器38c。於此種情況下，操作代碼符記解碼器38c分析所得操作代碼，及決定處理是否需要存取符記與記憶體符記間的匹配。然後，操作代碼符記解碼器38c輸入該決定結果給存取決定控制器38d作為一符記致能。

圖13為示意圖例示說明由操作代碼所指示之處理之一實例。圖13例示說明當執行使用MESI協定(伊利諾協定)的快取同調時簽發的記憶體存取請求之操作代碼。

舉例言之，含有操作代碼「0000」的該記憶體存取請求乃執行稱作為「READ IF」之一處理的請求。「READ IF」乃一處理用以當讀取自一核心的指令區與快取態為「共享(S)」時獲得快取之一處理。含有操作代碼「0001」的該記憶體存取請求乃執行稱作為「READ OP」之一處理的請求。「READ OP」乃一處理用以當讀取自一核心的資料區與快取態為「共享(S)」或「互斥(E)」時獲得快取之一處理。

含有操作代碼「0010」的該記憶體存取請求乃執行稱作為「READ EX」之一處理的請求。「READ EX」乃一處理用以從一核心獲得針對儲存資料之一互斥-正確獲得讀取，及當一快取態為「共享(S)」時獲得快取之一處理。含有操作代碼「0100」的該記憶體存取請求乃執行稱作為「DMA讀」之一處理的請求。「DMA讀」指示讀取自一I/O的DMA且為不獲得快取之一處理。

含有操作代碼「0110」的該記憶體存取請求乃執行稱作為「DMA寫」之一處理的請求。「DMA寫」指示自一I/O的DMA寫且為不獲得快取之一處理。含有操作代碼「0111」的該記憶體存取請求乃執行稱作為「部分DMA寫」之一處理的請求。「部分DMA寫」乃自一I/O部分地執行DMA寫的處理且為不獲得快取之一處理。

含有操作代碼「1000」的該記憶體存取請求乃執行稱作為「回寫」之一處理的請求。「回寫」為當執行CPU快取驅離時，指示快取態為「修正(M)」的寫入資料之一處理。含有操作代碼「1001」的該記憶體存取請求乃執行稱作為「回閃」之一處理的請求。

「回閃」為當執行CPU快取驅離時，改變「E」或「S」之快取態成為「I無效(I)」之一處理。含有操作代碼「1010」的該記憶體存取請求乃執行稱作為「快取刷新」之一處理的請求。「快取刷新」乃藉一核心執行快取刷新之一請求。

於圖13例示說明之實例中，若操作代碼為「0000」、「0001」、或「0010」，則操作代碼符記解碼器38c決定存取

符記須匹配記憶體符記且輸出符記致能「1」。若操作代碼係非為「0000」、「0001」、或「0010」，則操作代碼符記解碼器38c決定存取符記無須匹配記憶體符記且輸出符記致能「0」。與圖13例示說明之處理相聯結的操作代碼僅供舉例說明；因此，任意操作代碼也可與任意處理相聯結。

此處將回頭參考圖11作說明。記憶體符記檢查器38藉使用包括欲存取的一實體位址之一節段位址而搜尋記憶體符記暫存器38b有關一記憶體符記。然後，記憶體符記檢查器38將一「符記匹配」輸入存取決定控制器38d，該「符記匹配」指示所搜尋的記憶體符記是否匹配該記憶體存取請求記憶體存取請求之一存取符記。

於圖11例示說明之實例中，連續的記憶體位址係分派給單一節點內的一節段。若係分派非連續的記憶體位址，則改變存取符記的分錄號碼之一電路係在先一個階段配置。於圖11例示說明之實例中，若該位址符記為「0000」，則係指示有關未經保護區的記憶體存取請求。據此，記憶體符記檢查器38輸入指示「無存取」存取決定控制器38d之一信號。

存取決定控制器38d通知原位狀態機39，依據輸入「我的節點命中」、「符記致能」、「符記匹配」、及「無存取」，指示是否許可執行一記憶體存取請求。圖14為示意圖例示說明藉一存取決定控制器進行決定之一實例。

舉例言之，若「我的節點命中」為「否(0)」，符記致能為「是(1)」，符記匹配為「否(0)」，及無存取為「否(0)」，

則存取決定控制器38d不許可執行記憶體存取請求。於此種情況以外的情況下，存取決定控制器38d許可執行記憶體存取請求。更明確言之，針對用以保有快取同調的處理，諸如回寫，存取決定控制器38d許可執行記憶體存取請求而與該符記匹配之值獨立無關。

此處將回頭參考圖11作說明。依據接收自存取決定控制器38d及目標位址檢查器37的通知，原位狀態機39執行來自記憶體存取埠36的記憶體存取請求。圖15為示意圖例示說明由原位狀態機執行的處理之內容。例如，若原位狀態機39被存取決定控制器38d通知指示不許可記憶體存取請求的執行，則原位狀態機39執行當該存取符記係不匹配該記憶體符記時所從事的存取例外處理。

此外，若原位狀態機39被存取決定控制器38d通知指示許可記憶體存取請求的執行，且被目標位址檢查器37通知指示一記憶體係不在其本身的節點內(失誤)，則原位狀態機39執行下列處理。換言之，原位狀態機39執行當目標位址不相匹配時所從事的存取例外操作。

又復，若原位狀態機39被存取決定控制器38d通知指示許可記憶體存取請求的執行，且被目標位址檢查器37通知指示一記憶體係在其本身的節點內(命中)，則原位狀態機39執行記憶體存取請求。若由本身節點所簽發的記憶體存取請求係不被許可，則CPU 12至15及12a至12m各自也可藉發送同一個記憶體存取請求而重新嘗試。

後文中，將參考圖16描述藉資訊處理系統1執行節點間

的記憶體存取之處理流程。圖16為示意圖例示說明節點間的記憶體存取之處理流程。圖16例示說明一個實例其中節點#1執行在節點#2中之該記憶體上的記憶體存取。

舉例言之，如圖16之(E)例示說明，藉使用記憶體符記管理表，由節點#2所執行的應用程式選擇未曾使用的一記憶體符記。然後，如圖16之(F)例示說明，由節點#2所執行的應用程式通知由節點#1所執行的應用程式欲存取之一位址區及相對應於該記憶體符記之一存取符記。

於此種情況下，如圖16之(G)例示說明，由節點#1所執行的應用程式設定於存取符記表31中的該存取符記。如圖16之(H)例示說明，由節點#2所執行的應用程式在記憶體符記暫存器38b設定與通知給節點#1的該存取符記相對應的該記憶體符記。

此時，如圖16之(I)例示說明，當核心30a簽發一記憶體存取請求時，節點#1將設定在存取符記表31的一存取符記附接至該記憶體存取請求，且藉使用一互連體而發送該記憶體存取請求給節點#2。然後，如圖16之(J)例示說明，節點#2決定附接至該記憶體存取請求的該存取符記是否匹配設定在記憶體符記暫存器38b的該記憶體符記。

然後，若該存取符記係匹配該記憶體符記，則節點#2執行有關欲存取區的該記憶體存取請求；而若該存取符記並不匹配該記憶體符記，則節點#2不執行該記憶體存取請求。

如前述，在資訊處理系統1中的各個節點通知一節點存

取在一存取符記的主題節點中之一記憶體，且儲存所通知的該存取符記於一記憶體符記暫存器作為一記憶體符記。若該等節點各自存取含括在另一個節點中之一記憶體，則各個節點將由一欲存取的節點通知的一存取符記附接至一記憶體存取請求，且發送該記憶體存取請求。然後，節點各自決定附接至該記憶體存取請求的該存取符記是否匹配該記憶體符記。若該等節點決定存取符記係匹配記憶體符記，則該等節點各自執行記憶體存取。

於是，即便節點中之一者故障且即便不與另一個節點通訊，資訊處理系統1可藉欲存取之一節點改變一記憶體符記而停止從故障節點執行記憶體存取。結果，資訊處理系統1正確地執行故障切換。

後文中，將描述因欲存取之一節點改變一記憶體符記而停止從故障節點執行記憶體存取之處理程序。圖17為示意圖例示說明一種停止從故障節點執行記憶體存取之處理程序。圖17例示說明其中節點#1存取節點#2中的記憶體之實例。舉例言之，節點#1在存取符記表中儲存由含括在節點#2中之應用程式所通知的存取符記，及藉附上該存取符記而發送一記憶體存取請求給該節點#2。

此時，若在由節點#1所執行的應用程式中出現故障，且不與節點#2進行通訊，則節點#2中之應用程式不會改變儲存在節點#1中的存取符記。但由節點#2所執行的應用程式能改變儲存在節點#2中的記憶體符記。於是，由節點#2所執行的應用程式改變儲存在節點#2的記憶體符記暫存器

中的記憶體符記。由於藉故障節點#1傳輸的存取符記並不匹配儲存在節點#2的記憶體符記，故節點#2可防止節點#1執行記憶體存取。

此時，節點#2可於任意時間改變記憶體符記。更明確言之，節點#2改變記憶體符記且發送改變的記憶體符記給節點#1作為一存取符記。然後，當節點#1可與節點#2通訊時，節點#1正確地改變存取符記，如此可繼續存取節點#2。

相反地，若在節點#1出現故障而節點#1無法與節點#2通訊，因節點#1不改變存取符記，故節點#1無法存取節點#2。結果，當資訊處理系統1偏移由節點#1所執行的處理至另一個節點時，資訊處理系統1可防止腦分裂狀態的發生。於是，資訊處理系統1正確地執行故障切換。

後文中，將參考圖18及19描述中斷其中已經出現故障的一節點的連結之處理。圖18為第一示意圖例示說明一種防止其中已經出現故障的一節點執行記憶體存取之處理。圖19為第二示意圖例示說明一種防止其中已經出現故障的一節點執行記憶體存取之處理。圖18及19各自例示說明一個實例其中，當節點#1至#3各自簽發一記憶體存取請求給該節點#4時，節點#1中的應用程式已經故障，因此節點#1中的應用程式無法與其它節點#2至#4通訊。

舉例言之，如圖18之(K)例示說明，由節點#4執行的監視器應用程式將儲存在節點#4的記憶體符記「777」在任意時間改變成「888」。此外，節點#4分配新記憶體符記「888」給節點#1至#3作為存取符記。

此時，由於節點#1中的應用程式已經故障，因此無法與其它節點#2至#4通訊，故節點#1中的應用程式不會將存取符記「777」更新為新的存取符記「888」，如圖18之(L)例示說明。相反地，如圖18之(M)例示說明，節點#2及節點#3能將存取符記「777」改變成新的存取符記「888」。

結果，如圖19之(N)例示說明，具有故障應用程式的節點#1不存取節點#4；但節點#2及節點#3可繼續存取節點#4。因此節點#1至#4各自容易地中斷故障節點的連結。

此外，節點#1至#4各自無需決定因一處理，諸如移交而在一應用程式是否出現故障。更明確言之，節點#1至#4各自可中斷一故障節點的連結而未識別該節點是否故障。因此節點#1至#4各自可即刻中斷一故障節點的連結。

後文中，將參考圖20至25描述藉使用一記憶體符記及一存取符記執行記憶體存取處理之流程。此外，將對一實例作說明，其中包括核心30a及記憶體19的節點#1發送一記憶體存取請求給節點#2，包括一核心30g、一記憶體19a、及一記憶體19b。記憶體19a是個只包括一本地區的記憶體，記憶體19b是個只包括一共用記憶體區的記憶體。

後文說明中，應用程式為由節點#1及#2各自執行的應，OS為由節點#1及#2各自獨立地操作的OS，及共用記憶體驅動程式為控制共用記憶體區之一驅動程式。任一種方法皆可用來請求一符記，或體現節點間之通訊，只要可執行節點中應用程式間的通訊即可。此點可藉使用例如透過LAN的訊息通訊、使用一共用記憶體透過一互連體的訊息

通訊、及使用一共用儲存裝置之通訊而予體現。

首先，將參考圖20描述一種藉節點#1執行獲得一存取符記之處理流程。圖20為一序列圖例示說明一種獲得一存取符記之處理流程。舉例言之，藉核心30a執行的應用程式從由核心30g執行的應用程式請求一存取符記(步驟S101)。然後，由核心30g執行的應用程式請求該共用記憶體驅動程式分配一記憶體符記(步驟S102)。

然後，由核心30g執行的共用記憶體驅動程式參照儲存在記憶體19a的記憶體符記管理表。又復，由核心30g執行的共用記憶體驅動程式登錄未曾用在該記憶體符記管理表的一記憶體符記(步驟S103)且獲得該記憶體符記(步驟S104)。此時，由核心30g執行的共用記憶體驅動程式也登錄由該記憶體符記所使用之一處理的應用程式ID。

又復，由核心30g執行的共用記憶體驅動程式登錄所得記憶體符記在該記憶體符記管理表及該記憶體符記暫存器(步驟S105)。此時，由核心30g所執行的共用記憶體驅動程式也登錄例如一欲存取的節段號碼、一使用旗標、及一應用程式ID。此外，由核心30g執行的共用記憶體驅動程式通知由核心30a所執行的應用程式該記憶體符記用作為一存取符記(步驟S106)。

相反地，當由核心30a執行的應用程式獲得該存取符記(步驟S107)時，由核心30a執行的應用程式請求由核心30a執行的共用記憶體驅動程式設定所得存取符記(步驟S108)。然後，由核心30a執行的共用記憶體驅動程式設定

所得存取符記於存取符記表31(步驟S109)。

接著，由核心30a執行的共用記憶體驅動程式通知由核心30a執行的應用程式欲存取的一虛擬位址(VA)(步驟S110)。然後，由核心30a執行的應用程式簽發一資料讀取請求(步驟S111)。此時，設定在存取符記表31的該存取符記係設定在資料讀取請求(步驟S112)。

然後，節點#2決定設定在資料讀取請求的該存取符記是否匹配登錄在記憶體符記暫存器的該記憶體符記(步驟S113)。然後，若該存取符記匹配該記憶體符記，則節點#2許可存取該節段區(步驟S114)。

後文中，將參考圖21描述其中節點#1存取節點#中的多個節段之一實例。圖21為一序列圖例示說明一種存取多個節段之處理流程。舉例言之，藉核心30a執行的應用程式請求由核心30a執行的共用記憶體驅動程式設定一存取符記#A，用來存取在記憶體19b中的節段#A(步驟S201)。然後，由核心30a執行的共用記憶體驅動程式設定該存取符記#A於存取符記表31(步驟S202)。

此外，由核心30a執行的應用程式簽發有關節段#A的一資料讀取請求(步驟S203)。此時，存取符記#A係設定在該資料讀取請求(步驟S204)。然後，節點#2決定設定在該資料讀取請求的存取符記#A是否匹配於節段#A中的該記憶體符記(步驟S205)。若該存取符記#A匹配於節段#A中的該記憶體符記，則節點#2許可有關節段#A的記憶體存取(步驟S206)。

又復，由核心30a執行的應用程式請求由核心30a執行的共用記憶體驅動程式設定一存取符記#B，用來存取在記憶體19b中的節段#B(步驟S207)。然後，由核心30a執行的共用記憶體驅動程式設定該存取符記#B於存取符記表31(步驟S208)。

此外，由核心30a執行的應用程式簽發有關節段#B的一資料讀取請求(步驟S209)。此時，存取符記#B係設定在該資料讀取請求(步驟S210)。然後，節點#2決定設定在該資料讀取請求的存取符記#B是否匹配於節段#B中的該記憶體符記(步驟S211)。若該存取符記#B匹配於節段#B中的該記憶體符記，則節點#2許可有關節段#B的記憶體存取(步驟S212)。

後文中，將參考圖22描述當發生脈絡切換時其中在存取符記表31中的資料堆疊的一處理流程。圖22為一序列圖例示說明一種脈絡切換之流程。於圖22例示說明之實例中，將對欲執行的應用程式切換成應用程式#B之處理流程作說明，原因在於當應用程式#A正在由核心30a執行時發生中斷處理。

舉例言之，當核心30a正在跑應用程式#A時出現計時器中斷(步驟S301)。於此種情況下，由核心30a所執行的OS執行包括在存取符記表31中的資料之脈絡切換(步驟S302)，及堆疊暫存器值於記憶體19(步驟S303)。然後，由核心30a所執行的OS獲得當應用程式#B 19正在執行時所得的在記憶體中堆疊的暫存器值(步驟S304)。然後，由核心30a所執行

的OS設定所得值於該存取符記表31(步驟S305)。

隨後，由核心30a所執行的應用程式#B簽發一資料讀取請求(步驟S306)。然後，設定在存取符記表31的存取符記，亦即由應用程式#B所使用的存取符記係設定在該資料讀取請求(步驟S307)。然後，節點#2決定由應用程式#B所使用的節段的記憶體符記是否匹配設定在該資料讀取請求中的存取符記(步驟S308)。若該記憶體符記匹配該存取符記，節點#2許可存取該共用記憶體區(步驟S309)。

後文中，將參考圖23說明藉節點#2執行改變記憶體符記之一處理流程。圖23為一序列圖例示說明一種改變一記憶體符記之處理流程。於圖23例示說明之實例中，將描述一個實例其中於節點#1未出現故障，因而可正確地存取節點#2。假設圖23例示說明的步驟S411至S414執行的處理係與圖22例示說明的步驟S306至S309執行的處理相同；因此在此刪除其說明。

舉例言之，由核心30g執行的應用程式請求由核心30g執行的共用記憶體驅動程式在任意時間更新一符記(步驟S401)。然後，由核心30g執行的共用記憶體驅動程式搜尋記憶體符記管理表尋找新的可用的記憶體符記(步驟S402)，且通知由核心30g執行的應用程式欲使用的新記憶體符記。隨後，由核心30g執行的應用程式簽發一存取停止請求給由核心30a執行的應用程式(步驟S403)及等候預定時間週期(步驟S404)。

然後，由核心30g執行的應用程式通知由核心30g執行

的共用記憶體驅動程式欲使用的新記憶體符記(步驟S405)。於此種情況下，由核心30g執行的共用記憶體驅動程式設定新記憶體符記於該記憶體符記暫存器(步驟S406)。此外，由核心30g執行的共用記憶體驅動程式更新該記憶體符記管理表(步驟S407)。

然後，由核心30g執行的應用程式通知由核心30a執行的應用程式該新穎存取符記及該記憶體存取的恢復執行(步驟S408)。於此種情況下，由核心30a執行的應用程式通知由核心30a執行的共用記憶體驅動程式新的存取符記(步驟S409)。然後，由核心30a執行的共用記憶體驅動程式設定該新的存取符記於存取符記表31(步驟S410)。

當許可重新嘗試記憶體存取時，無需等候何時執行記憶體符記的改變處理。於是，將參考圖24描述由節點#2執行記憶體符記的改變處理之變化例。圖24為一序列圖例示說明一種改變該記憶體符記之處理的變化例。於圖24例示說明之實例中，將對一個實例作說明，其中節點#1尚未發生故障，因此在記憶體符記已經改變後節點#2正確地存取。

舉例言之，由核心30g執行的應用程式請求由核心30g執行的共用記憶體驅動程式在任意時間更新一符記(步驟S501)。然後，由核心30g執行的共用記憶體驅動程式搜尋該記憶體符記管理表尋找新的可用的記憶體符記(步驟S502)，且通知由核心30g執行的應用程式欲使用的新記憶體符記(步驟S503)。

然後，由核心30g執行的應用程式通知由核心30g執行

的共用記憶體驅動程式欲使用的新記憶體符記(步驟S504)。於此種情況下，由核心30g執行的共用記憶體驅動程式設定新記憶體符記於該記憶體符記暫存器(步驟S505)。此外，由核心30g執行的共用記憶體驅動程式登錄該新的記憶體符記於該記憶體符記管理表(步驟S506)，且通知由核心30g執行的應用程式新記憶體符記已經登錄(步驟S507)。

此外，由核心30g執行的應用程式將新存取符記傳輸給由核心30a執行的應用程式(步驟S508)。此時，由核心30a執行的應用程式簽發一資料讀取請求(步驟S509)。然後，由核心30a執行的共用記憶體驅動程式設定該新存取符記於存取符記表31(步驟S510)。

但於步驟S510設定的該存取符記並不匹配新記憶體符記，因此該記憶體存取被拒絕(步驟S511)。於是節點#2發送一回應指示存取被拒絕給節點#1(步驟S512)。此時，因由核心30a執行的應用程式從由核心30g執行的應用程式接收新存取符記，故由核心30a執行的應用程式通知由核心30a執行的共用記憶體驅動程式有關該符記(步驟S513)。然後，由核心30a執行的共用記憶體驅動程式設定該新存取符記於存取符記表31(步驟S514)。

隨後，由核心30a執行的應用程式發送含有該資料讀取請求之一重新嘗試(步驟S515)。然後，該新存取符記係被設定在含有該資料讀取請求之該重新嘗試(步驟S516)。結果，節點#2決定由應用程式#B所使用的該節段內的該記憶體符

記是否匹配設定在該資料讀取請求中的該存取符記(步驟S517)。

若該記憶體符記匹配該存取符記，則該節點#2許可存取該共用記憶體區(步驟S518)。此外，若設定在接收自節點#1的該重新嘗試中的存取符記並不匹配該記憶體符記，則節點#2判定已經發生記憶體存取錯誤，因此，節點#2可執行中斷節點#1之連結的處理。

後文中，將參考圖25對處理流程作說明，其中當節點#2改變該記憶體符記時，節點#1中發生故障，因而中斷節點#1的連結，原因在於未正確地執行存取。圖25為一序列圖例示說明一種中斷一故障節點的連結之處理流程。

於圖25例示說明之實例中，將對如下實例作說明，其中於節點#1已經發生故障，故不進行與節點#2的通訊。此外，於圖25例示說明之實例中，假設節點#1在預定時間間隔簽發心搏通知給欲存取的節點#2。

首先，由核心30a執行的應用程式簽發心搏通知(步驟S601)。但因節點#1發生故障，故心搏通知並不遞送給節點#2。於是，由核心30g執行的應用程式檢測得節點#1的異常(步驟S602)。然後，由核心30g執行的應用程式請求由核心30g執行的共用記憶體驅動程式更新一記憶體符記(步驟S603)。然後，由核心30g執行的共用記憶體驅動程式搜尋該記憶體符記管理表，找出一新記憶體符記(步驟S604)且通知由核心30g執行的應用程式該新的記憶體符記。

然後，由核心30g執行的應用程式傳輸該新記憶體符記

給節點#1作為存取符記；但因節點#1已經發生故障，故新存取符記並未到達節點#1。然後，由核心30g執行的應用程式通知由核心30g執行的共用記憶體驅動程式有關該欲使用的新記憶體符記(步驟S605)。

於此種情況下，由核心30g執行的共用記憶體驅動程式設定該新記憶體符記於該記憶體符記暫存器(步驟S606)。此外，由核心30g執行的共用記憶體驅動程式登錄該新記憶體符記於該記憶體符記管理表(步驟S607)，且通知由核心30g執行的應用程式該新記憶體符記已經被登錄。

此外，由核心30g執行的應用程式通知節點#1恢復執行通訊的通知(步驟S608)；但因節點#1已經發生故障，故恢復執行通訊的通知並未到達節點#1。據此，由核心30g執行的應用程式簽發一中斷節點#1之連結的請求給由核心30g執行的共用記憶體驅動程式(步驟S609)。然後，由核心30g執行的共用記憶體驅動程式搜尋該共用記憶體節段管理表，找出由欲中斷連結的該節點#1所使用的一共用記憶體區(步驟S610)。

此時，當由核心30a執行的應用程式簽發一資料讀取請求時(步驟S611)，節點#1設定該存取符記於該資料讀取請求(步驟S612)且傳輸該存取符記給節點#2。但因該存取符記不匹配該新記憶體符記(步驟S613)，故節點#2拒絕該記憶體存取且發送一位址例外給節點#1(步驟S614)。

於此種情況下，節點#1執行回寫處理，其中儲存於L1快取記憶體30b、L1快取記憶體30e、及L2快取記憶體32a

中的資料係發送給節點#2(步驟S615)。當節點#2藉此方式而從節點#1獲得回寫資料時，節點#2執行回寫至該共用記憶體區而不決定是否可執行記憶體存取(步驟S616)。

然後，藉由使用快取刷新及暗示回寫，節點#1及節點#2許可保留在L1快取記憶體30b及30e中之各者及L2快取記憶體32a的資料匹配保留在記憶體19b的資料。然後，節點#2將在節點#1中的由核心30a所執行的處理工作移至由另一個CPU所執行的處理，且執行中斷連結處理用以中斷節點#1之連結(步驟S617及S618)。然後，由核心30g執行的共用記憶體驅動程式通知由核心30g執行的應用程式，指示已經完成中斷節點#1之連結(步驟S619)。

後文中，將參考圖26說明一種改變一記憶體符記之處理實例。圖26為流程圖例示說明一種改變記憶體符記之處理流程。於圖26例示說明之實例中，假設CPU 12計數針對各項處理一記憶體存取請求的符記錯誤數目，係以「N」表示。此外，在CPU 12執行圖26例示說明之處理之前，CPU 12於各項處理的「N」中設定一個初值「0」。此外，圖26例示說明其中10次或以上重新嘗試失敗的節點藉使用記憶體錯誤處置而中斷連結之實例。

例如，CPU 12執行一記憶體存取請求作為正常處理，該項請求係依據記憶體符記及存取符記二者執行(步驟S701)。此時，CPU 12決定是否產生一錯誤信號(步驟S702)。該錯誤信號為通知何時出現錯誤之一信號，且為例如通知何時錯誤係儲存在錯誤暫存器43之一信號。

此時，若CPU 12決定產生一錯誤信號(步驟S702)，則CPU 12決定由該錯誤信號所指示的錯誤內容是否為符記錯誤，指示該記憶體符記並不匹配該存取符記(步驟S703)。若CPU 12決定由該錯誤信號所指示的錯誤內容為符記錯誤(步驟S703為是)，則CPU 12執行下列處理。換言之，CPU 12決定請求該記憶體存取結果導致符記錯誤的該項處理之「N」值是否小於「10」(步驟S704)。

若CPU 12決定請求該記憶體存取的該項處理之「N」值係小於「10」(步驟S704為是)，則CPU 12加「1」至請求該記憶體存取的該項處理的「N」值(步驟S705)。然後，CPU 12再度執行正常處理(步驟S701)。

又復，若CPU 12決定請求該記憶體存取的該項處理之「N」值係等於或大於「10」(步驟S704為否)，則CPU 12執行下述處理。換言之，CPU 12執行記憶體錯誤處置(步驟S706)，中斷請求該記憶體存取的該項處理的該節點之連結且結束處理程序。

後文中，將對下述處理作說明，當CPU 12中斷一節點之連結時，更新共用記憶體節段管理表及記憶體符記管理表二者之處理。圖27為流程圖例示說明一種更新共用記憶體節段管理表及記憶體符記管理表之處理流程。針對該共用記憶體節段管理表中的各個分錄，CPU 12執行圖27例示說明之處理。此外，CPU 12執行當一節段釋放請求係得自由與一記憶體存取請求的該簽發源相對應的一節點所執行之一處理程序時觸發的圖27例示說明之處理。

舉例言之，CPU 12參照一使用旗標，決定該使用旗標是否為「1」(步驟S801)。若該使用旗標係為「1」(步驟S801為是)，則CPU 12決定簽發該節段釋放請求的該處理之應用程式ID是否匹配一分錄的應用程式ID(步驟S802)。

若CPU 12決定該等應用程式ID確實匹配(步驟S802為是)，則CPU 12決定執行簽發一節段釋放請求的該處理之該節點的使用節點旗標是否為「1」(步驟S803)。若CPU 12決定執行簽發該節段釋放請求的該處理之該節點的使用節點旗標為「1」(步驟S803為是)，則CPU 12設定執行簽發該節段釋放請求的該處理之該節點的使用節點旗標為「0」(步驟S804)。

然後，CPU 12決定另一個使用節點旗標是否為「0」(步驟S805)。若另一個使用節點旗標係為「0」(步驟S805為是)，則CPU 12設定該使用節點旗標為「0」(步驟S806)。然後，CPU 12識別該分錄的該記憶體符記之值，且搜尋記憶體符記管理表中的分錄，找出其中儲存有一記憶體符記具有與所識別值相同值的一分錄。然後，CPU 12設定所搜尋的分錄之使用旗標為「0」(步驟S807)。

然後，CPU 12簽發一完成回應，指示該節點的中斷連結已經完成(步驟S808)及結束處理。又復，若該使用旗標係非為「1」(步驟S801為否)，應用程式ID不匹配(步驟S802為否)，或使用節點旗標為「0」(步驟S803為否)，則CPU 12簽發一完成回應(步驟S808)。此外，若使用節點旗標皆非為「0」(步驟S805為否)，則CPU 12簽發一完成回應(步驟

S808)。

圖28為示意圖例示說明該共用記憶體節段管理表之更新。舉例言之，如圖28中藉影線區例示說明，藉由執行圖27例示說明之於步驟S804及S806的處理，CPU 12設定使用旗標及使用節點旗標二者為「0」。

圖29為示意圖例示說明該記憶體符記管理表之更新。舉例言之，如圖29中藉影線區例示說明，藉由執行圖27例示說明之於步驟S807的處理，CPU 12設定使用旗標為「0」。藉此方式，藉由執行圖27例示說明之處理，CPU 12可藉妥當更新該共用記憶體節段管理表及該記憶體符記管理表而中斷一節點的連結。

第一實施例之優點

如前述，CPU 12至15及12a至12m各自具有該存取符記表31及31a。此外，CPU 12至15及12a至12m各自將一存取符記附接至欲發送給另一個節點之一記憶體存取請求且傳輸該記憶體存取請求。此外，CPU 12至15及12a至12m各自包括一記憶體符記暫存器，該記憶體符記暫存器中儲存一記憶體存取符記，該記憶體符記係與發送給一CPU的該存取符記值具有相同值，而該CPU簽發一記憶體存取請求給該相對應CPU。然後，CPU 12至15及12a至12m各自基於所接收的記憶體存取請求之該存取符記及儲存在該記憶體符記暫存器的該記憶體符記二者而控制該記憶體存取請求之執行。

於是，藉由改變儲存在該相對應CPU的記憶體符記暫

存器中的該記憶體符記，CPU 12至15及12a至12m各自防止由一故障節點簽發該記憶體存取請求的執行。結果，資訊處理系統1可避免腦分裂的發生，如此正確地執行故障切換。

此外，當變更該記憶體符記時，CPU 12至15及12a至12m各自通知相對應於一記憶體存取請求的簽發源之一節點有關一個新記憶體符記作為一存取符記。然後，若該記憶體符記匹配所接收的記憶體存取請求之一存取符記，則CPU 12至15及12a至12m各自許可執行該記憶體存取請求。

因此，CPU 12至15及12a至12m各自可防止由故障節點執行記憶體存取而無需決定一存取來源節點是否已經出現故障。結果，當藉使用移交而與一種方法比較決定在存取來源節點是否已經出現故障時，CPU 12至15及12a至12m各自可即刻地且容易地中斷一節點的連結。

此外，當CPU 12至15及12a至12m各自拒絕一記憶體存取請求時，CPU 12至15及12a至12m各自更新一記憶體符記，及然後將由在相對應於該記憶體存取請求的該簽發源之一節點中的一CPU所快取的資料回寫至由該CPU所存取之一記憶體。因此，即便CPU 12至15及12a至12m各自包括一快取記憶體，CPU 12至15及12a至12m各自能夠正確地中斷一節點的連結，同時維持快取同調。

此外，CPU 12至15及12a至12m各自連同停止存取該記憶體存取請求的該簽發源的一指令，發送一新存取符記。然後，CPU 12至15及12a至12m各自在預定時間之後更新該

記憶體符記，及指示該記憶體存取請求的該簽發源恢復執行該存取。

於是，CPU 12至15及12a至12m各自可拒絕來自一故障節點的記憶體存取，而無需接收到來自該記憶體存取請求的該簽發源的一重新嘗試。結果，當CPU 12至15及12a至12m各自不允許記憶體存取請求時，CPU 12至15及12a至12m各自可妥當地中斷一故障節點的連結而無需決定在該節點內發生的錯誤型別。

此外，若不許可有關另一個節點的記憶體存取，則CPU 12至15及12a至12m各自可執行重新嘗試。更明確言之，CPU 12至15及12a至12m各自具有若不許可該記憶體存取即重新嘗試相同記憶體存取請求的功能。

於此種情況下，CPU 12至15及12a至12m各自可正確地中斷一故障節點的連結而無需等候何時通知一新存取符記。更明確言之，由於在當存取來源節點藉使用一改變的存取符記執行記憶體存取時的時間與當一存取目的地節點改變該記憶體符記時的時間之間有一時間落後，故一存取可能被拒絕。

舉例言之，於某些情況下，存取可能被拒，原因在於一存取來源節點附上一尚未經改變的存取符記，但一存取目的地節點決定是否可藉使用一已經改變的存取符記來執行存取。同理，於某些情況下，存取可能被拒，原因在於一存取來源節點附上一已經改變的存取符記，但一存取目的地節點決定是否可藉使用一尚未經改變的存取符記來執

行存取。

此時，若存取被拒，則存取來源節點發送一重新嘗試。然後，正常操作的該存取來源節點附上當該記憶體存取被重新嘗試時所得的一存取符記。然後，存取目的地節點決定是否可藉使用該已經改變的存取符記來執行存取。結果，CPU 12至15及12a至12m各自許可由正常操作的該存取來源節點所決定的該記憶體存取。

相反地，不正常操作的該存取來源節點附上尚未經改變的一存取符記至該記憶體存取的重新嘗試，存取目的地節點決定是否可藉使用該已經改變的存取符記來執行存取。結果，CPU 12至15及12a至12m各自決定是否許可由不正常操作的該存取來源節點執行該記憶體存取的重新嘗試，及中斷相對應於該重新嘗試的簽發源之一節點的連結。

若CPU 12至15及12a至12m各自拒絕該記憶體存取的重新嘗試，則相對應於該記憶體存取請求之重新嘗試的簽發源之一節點中的一CPU將該被快取的資料回寫至由該CPU由存取的記憶體。因此，CPU 12至15及12a至12m各自可正確地中斷該節點的連結，同時維持快取同調。

此外，CPU 12至15及12a至12m各自使用一暫存器，其中當執行一脈絡切換時資料被堆疊作為一存取符記表。因此，即便當使用多重執行緒來執行一處理時，CPU 12至15及12a至12m各自可防止一故障節點執行記憶體存取，且正確地執行一故障切換而未使得OS或應用程式變複雜化。

[b]第二實施例

於前文解說中，已經對依據本發明之實施例作說明；但實施例並非圍限於此而可以前述實施例以外的多個實施例體現。因此後文中將敘述含括於本發明之另一個實施例作為第二實施例。

(1)存取符記表

前述CPU 12分別地含括於核心30a及30d中的存取符記表31及31a；但實施例並非圍限於此。舉例言之，若核心30a為一多執行緒能核心，則CPU 12也可針對各個硬體執行緒使用一存取符記表。

舉例言之，圖30為示意圖例示說明一多執行緒能存取符記表之實例。於圖30例示說明之實例中，核心30a具有兩個硬體執行緒。於此種情況下，CPU 12包括由硬體執行緒各自所使用的兩個存取符記表31b及31c。此外，CPU 12在一封包內儲存一存取符記，該存取符記係由簽發該記憶體存取請求的該執行緒所使用且係由該存取符記表所輸出。此外，存取符記表的數目並不限於圖30例示說明之實例；因此，CPU 12也可具有任意數目的存取符記表。

如前述，若CPU 12針對各個硬體執行緒使用一存取符記表，則CPU 12防止由一故障節點執行記憶體存取，而不使得OS或應用程式變複雜化，即便當藉使用多重執行緒執行一處理時亦復如此。結果，資訊處理系統1可正確地執行一故障切換。

(2)未經保護區

當前述CPU 12存取一未經保護區時，CPU 12使用

「a110」作為一存取符記及一記憶體符記；但實施例並非囿限於此。舉例言之，藉設定一有效位元在由該記憶體符記暫存器所儲存之一記憶體符記內及藉使用該有效位元，也可能指示一目標存取係在保護區或在未經保護區。

舉例言之，圖31為示意圖例示說明該記憶體符記檢查器之變化例。於圖31例示說明之實例中，一記憶體符記檢查器38e包括一記憶體符記暫存器38f。記憶體符記暫存器38f使用含括在記憶體符記暫存器38b的1-位元保留作為一有效位元，且將該有效位元「0」儲存在與未經保護區相聯結的一個欄位。於此種情況下，若一目標存取係在一未經保護區，指示無存取之一信號係輸入至存取決定控制器38d；因此，可妥當體現記憶體存取而不設定一記憶體符記於該未經保護區。

(3) 因未授權存取所致的記憶體符記之匹配

前述CPU 12取決於一記憶體符記是否匹配一存取符記而決定是否執行一記憶體存取請求；但並非囿限於此等實施例。舉例言之，除了決定一記憶體符記是否匹配一存取符記之外，也可能決定簽發一記憶體存取請求的一節點是否為先前許可的節點。

舉例言之，圖32為示意圖例示說明一記憶體符記檢查器，其只允許先前許可的節點來執行記憶體存取。於圖32例示說明之實例中，除了記憶體符記暫存器38b外，一記憶體符記檢查器38g包括一節點決定電路38h，及該節點決定電路38h包括一節點列表暫存器38i。此時，被許可存取之

一節點的節點ID係儲存在針對各個節段區的節點列表暫存器38i中。

從與該記憶體存取請求之目標相對應的該節段位址，節點列表暫存器38i輸出被允許存取的該節點的節點ID。然後，節點決定電路38h將一節點ID匹配輸出給存取決定控制器38d，該節點ID匹配指示已由該節點列表暫存器38i輸出的該節點ID是否匹配屬該記憶體存取請求的該簽發源的該節點之節點ID。

然後，若符記致能於該音訊信號訊框中包含該等事件之多個槽位的指示「是(1)」且若「我的節點命中」，符記匹配、節點ID匹配、及無存取全部皆為「0」，則存取決定控制器38d不允許執行記憶體存取請求。除此之外，存取決定控制器38d許可執行記憶體存取請求。如前文說明，CPU 12藉檢查一節點ID可防止未授權存取。

依據實施例之一個面向，於一共用記憶體組態正確地執行故障切換。

【圖式簡單說明】

圖1為示意圖例示說明依據第一實施例之一種資訊處理系統；

圖2為示意圖例示說明依據第一實施例一個節點之功能組態；

圖3為示意圖例示說明依據第一實施例記憶體之一種配置；

圖4為示意圖例示說明一種記憶體對映圖其中依據第

一實施例的資訊處理系統係對映至各個記憶體；

圖5為示意圖例示說明依據第一實施例藉該資訊處理系統所配置的共用記憶體區；

圖6為示意圖例示說明依據第一實施例一種共用記憶體區管理表；

圖7為示意圖例示說明依據第一實施例一種記憶體符記管理表；

圖8為示意圖例示說明依據第一實施例一種CPU之實例；

圖9為示意圖例示說明依據第一實施例針對一存取符記表之一暫存器的組態；

圖10為示意圖例示說明於該存取符記表中的暫存器欄位；

圖11為示意圖例示說明依據第一實施例一種記憶體符記檢查器之實例；

圖12為示意圖例示說明一種記憶體存取請求之實例；

圖13為示意圖例示說明一種藉操作代碼指示之處理實例；

圖14為示意圖例示說明藉一存取決定控制器進行決定之一實例；

圖15為示意圖例示說明由原位狀態機執行的處理之內容；

圖16為示意圖例示說明節點間的記憶體存取之處理流程；

圖17為示意圖例示說明一種停止從故障節點執行記憶體存取之處理程序；

圖18為第一示意圖例示說明一種防止其中已經出現故障的一節點執行記憶體存取之處理；

圖19為第二示意圖例示說明一種防止其中已經出現故障的一節點執行記憶體存取之處理；

圖20為一序列圖例示說明一種獲得一存取符記之處理流程；

圖21為一序列圖例示說明一種存取多個節段之處理流程；

圖22為一序列圖例示說明一種脈絡切換之流程；

圖23為一序列圖例示說明一種改變一記憶體符記之處理流程；

圖24為一序列圖例示說明一種改變該記憶體符記之處理之變化例；

圖25為一序列圖例示說明一種中斷一故障節點的連結之處理流程；

圖26為流程圖例示說明一種改變記憶體符記之處理流程；

圖27為流程圖例示說明一種更新共用記憶體節段管理表及記憶體符記管理表之處理流程；

圖28為示意圖例示說明該共用記憶體節段管理表之更新；

圖29為示意圖例示說明該記憶體符記管理表之更新；

圖30為示意圖例示說明一多執行緒能存取符記表之實例；

圖31為示意圖例示說明該記憶體符記檢查器之變化例；

圖32為示意圖例示說明一記憶體符記檢查器，其只允許先前許可的節點來執行記憶體存取；

圖33為示意圖例示說明一種具有叢集組態之資訊處理系統；

圖34為示意圖例示說明節點間之記憶體存取；

圖35為示意圖例示說明一種使用共用記憶體組態技術之資訊處理系統；

圖36為示意圖例示說明一種未授權記憶體存取之防止功能；

圖37A為第一示意圖例示說明其中故障切換並非正常地執行的一個實例；及

圖37B為第二示意圖例示說明其中故障切換並非正常地執行的一個實例。

【主要元件符號說明】

- 1...資訊處理系統
- 2、2a...用戶終端
- 3、5...區域網路(LAN)交換器
- 4...管理終端
- 6...通用縱橫式交換器
- 7、11、11a-m...服務處理器

- 10、10a-m、50-57...節點
- 12-15、12a-m、50c、51c、52c...中央處理單元(CPU)
- 16、16a-m...本地縱橫式交換器(XB)
- 17-24、19a、19b、50a、51a、52a...記憶體
- 25...周邊組件互連體快速(PCIe)交換器
- 26、27...LAN配接器
- 28...串列附件SCSI (SAS)
- 29...硬碟驅動裝置(HDD)
- 30a、30d、30g...核心
- 30b、30e...L1快取記憶體
- 30c、30f...轉譯後備緩衝器(TLB)
- 31、31a...存取符記表
- 31b-c...存取符記表
- 32...L2快取標籤
- 32a...L2快取記憶體
- 33...快取監聽控制器
- 34...記憶體請求狀態機
- 35...位址對映圖
- 36...記憶體存取埠
- 37...目標位址檢查器
- 38、38e、38g...記憶體符記檢查器
- 38a...自身節點ID暫存器
- 38b、38f...記憶體符記暫存器
- 38c...操作代碼符記解碼器

- 38d...存取決定控制器
- 38h...節點決定電路
- 38i...節點列表暫存器
- 39...原位狀態機
- 40...記憶體存取控制器
- 41...非快取請求狀態機
- 42...PCIe介面
- 43...錯誤暫存器
- 44...暫存器控制器
- 45...請求暫存器
- 46...路由器
- 47-47c...位址空間識別符(ASI)暫存器
- 50b、51b、52b...作業系統(OS)
- S101-114、S201-212、S301-309、S401-414、S501-518、S601-619、
S701-706、S801-808...步驟

七、申請專利範圍：

1. 一種資訊處理裝置，其係包含：

多個節點具有一第一節點及一第二節點，該等節點各自包括一處理器及一記憶體，其中至少一部分區係設定作為一共用記憶體區；及

連結該等節點之一互連體，其中

該第一節點包括：

一第一儲存單元，於該儲存單元中儲存用以存取在該第二節點之一記憶體的識別資訊；及

一發送單元，其係藉附上儲存在該第一儲存單元的該識別資訊而發送欲發送的通訊資料給該第二節點，及

該第二節點包括：

一第二儲存單元，於該儲存單元中儲存識別資訊，該識別資訊係用以控制許可從另一個節點存取在該第二節點中之一記憶體內的該共用記憶體區；及

一控制單元，其係基於附上至從該第一節點所發送的該通訊資料的該識別資訊及儲存在該第二儲存單元之該識別資訊而決定是否許可存取在該第二節點中之一記憶體內的該共用記憶體區。

2. 如申請專利範圍第1項之資訊處理裝置，其中

該第一儲存單元儲存由該第二節點所通知的該識別資訊，

該第二節點進一步係包括：

一通知單元，其通知該第一節點一塊新的識別資訊；及

一改變單元，其係將儲存在該第二儲存單元之該識別資訊改變成由該通知單元所通知的該識別資訊，

當附上至從該第一節點所發送的該通訊資料的該識別資訊係匹配儲存在該第二儲存單元之該識別資訊時，該控制單元許可存取至在該第二節點中之該記憶體內的該共用記憶體區，及

當附上至從該第一節點所發送的該通訊資料的該識別資訊係並不匹配儲存在該第二儲存單元之該識別資訊時，該控制單元不許可存取至在該第二節點中之該記憶體內的該共用記憶體區。

3. 如申請專利範圍第2項之資訊處理裝置，其中

該第一節點係進一步包括一快取記憶體其保有得自該第二節點中之該記憶體的資料，

當該控制單元不許可存取至該第一節點中的該快取記憶體用以保有得自該第二節點中之該記憶體的資料時，該控制單元將由含括在該第一節點之該快取記憶體所保有的資料回寫至該第二節點中之該記憶體，及

在該控制單元將由含括在該第一節點之該快取記憶體所保有的資料回寫至該第二節點中之該記憶體之前，該改變單元將儲存在該第二儲存單元之該識別資訊

改變成該塊新的識別資訊。

4. 如申請專利範圍第2或3項之資訊處理裝置，其中
該通知單元連同停止存取之一指令，通知該第一節點該塊新的識別資訊，及
從該通知單元通知該第一節點停止該存取之該指令開始算起已經經歷一預定時間後，該改變單元將儲存在該第二儲存單元之該識別資訊改變成由該通知單元所通知的該塊新識別資訊。
5. 如申請專利範圍第2或3項之資訊處理裝置，其中
該通知單元通知該第一節點該塊新的識別資訊，
該改變單元將儲存在該第二儲存單元之該識別資訊改變成該塊新識別資訊，及
當不許可存取至在該第二節點中之該記憶體內的該共用記憶體區時，該發送單元重新發送用以請求該相同存取的該通訊資料。
6. 如申請專利範圍第5項之資訊處理裝置，其中當該控制單元不許可存取至在該第二節點中之該記憶體內的該共用記憶體區時，該控制單元將含括在該第一節點的由該快取記憶體所保有的資料回寫至該第二節點的該記憶體。
7. 如申請專利範圍第1至3項中任一項之資訊處理裝置，其中該第一節點使用當執行一脈絡切換時用以堆疊資料的一暫存器之一部分作為該第一儲存單元。
8. 如申請專利範圍第1至3項中任一項之資訊處理裝置，其

中該第一節點包括該第一儲存單元，其數目係與在該第一節點中並列執行的執行緒數目相同。

9. 一種由一資訊處理裝置執行的未授權存取防止方法，該資訊處理裝置係包括多個節點具有一第一節點及一第二節點，該等節點各自包括一處理器及一記憶體，其中至少一部分區係設定作為一共用記憶體區，及連結該等節點之一互連體，該未授權存取防止方法係包含：

藉該第一節點發送用以存取在該第二節點中之一記憶體的識別資訊且附接至欲發送給該第二節點之通訊資料；

藉該第二節點，從另一個節點獲得用以控制許可存取在該第二節點中之一記憶體內的該共用記憶體區的識別資訊；及

基於附上至從該第一節點所發送的該通訊資料之該識別資訊及於該獲得時所獲得的該識別資訊，藉該第二節點決定是否許可一存取至在該第二節點中之該記憶體內的該共用記憶體區。

八、圖式：

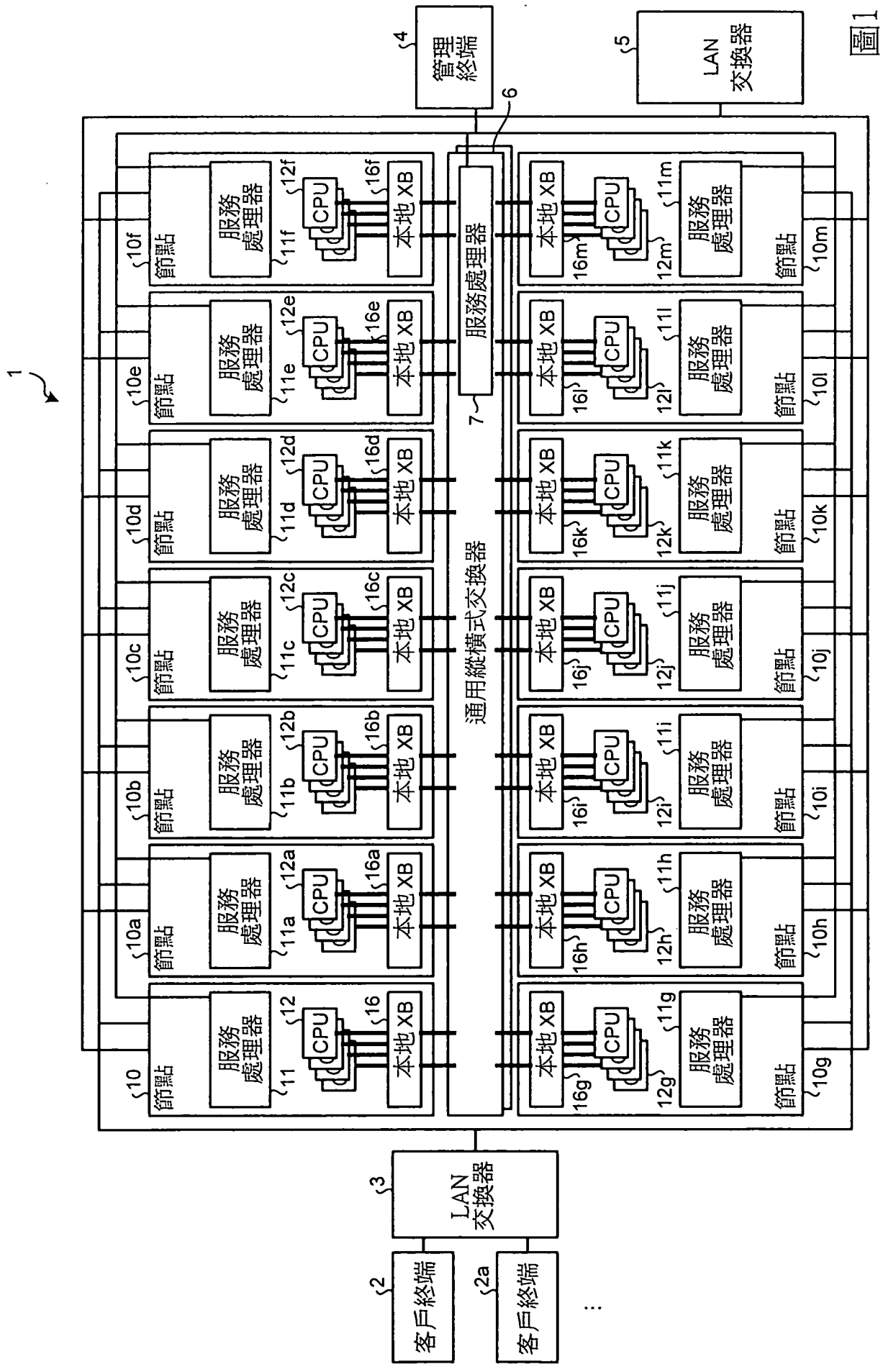


圖1

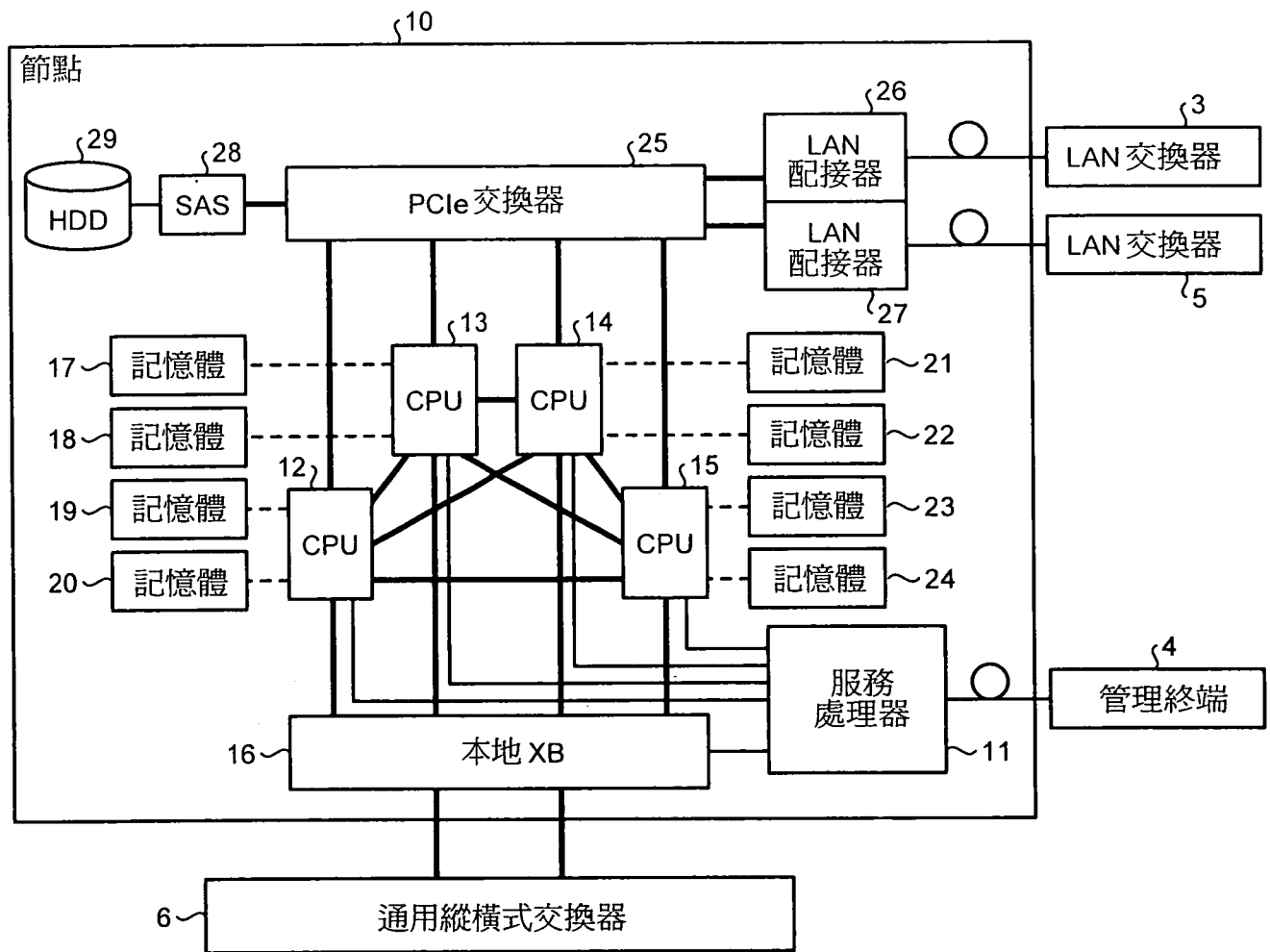


圖2

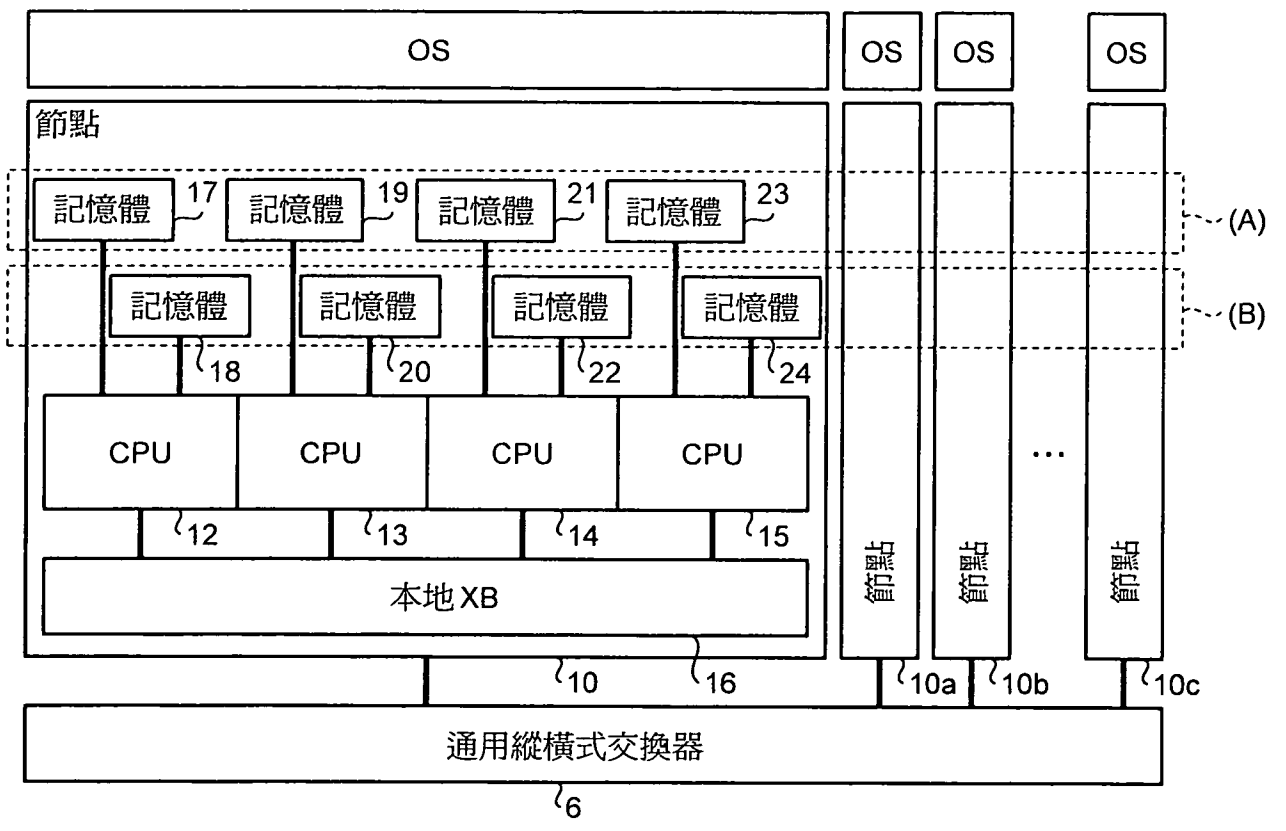


圖3

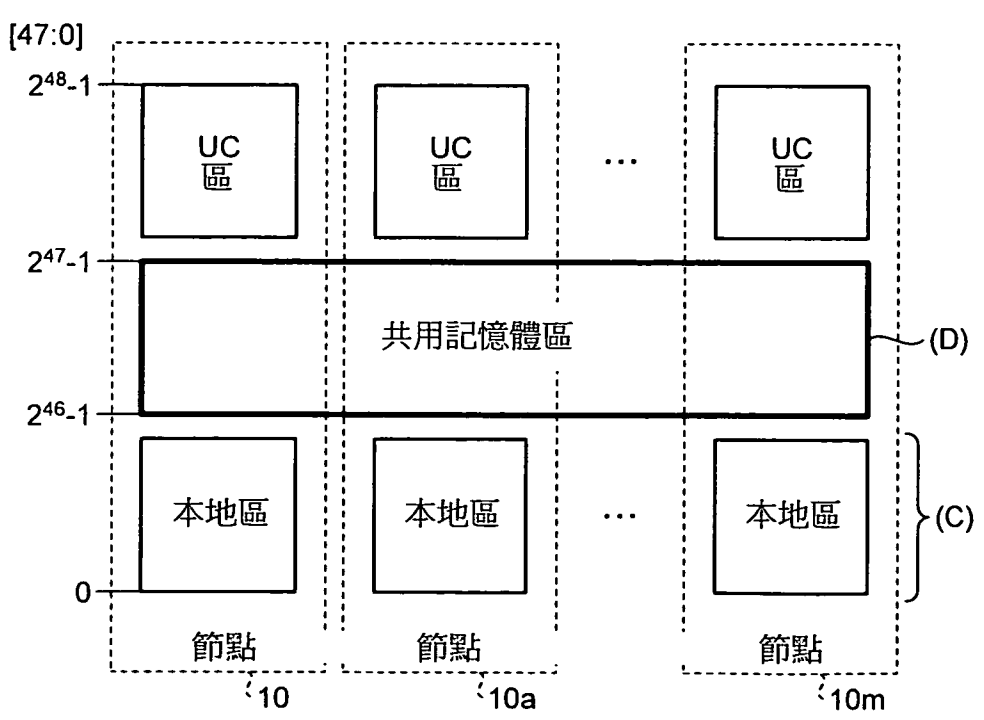


圖4

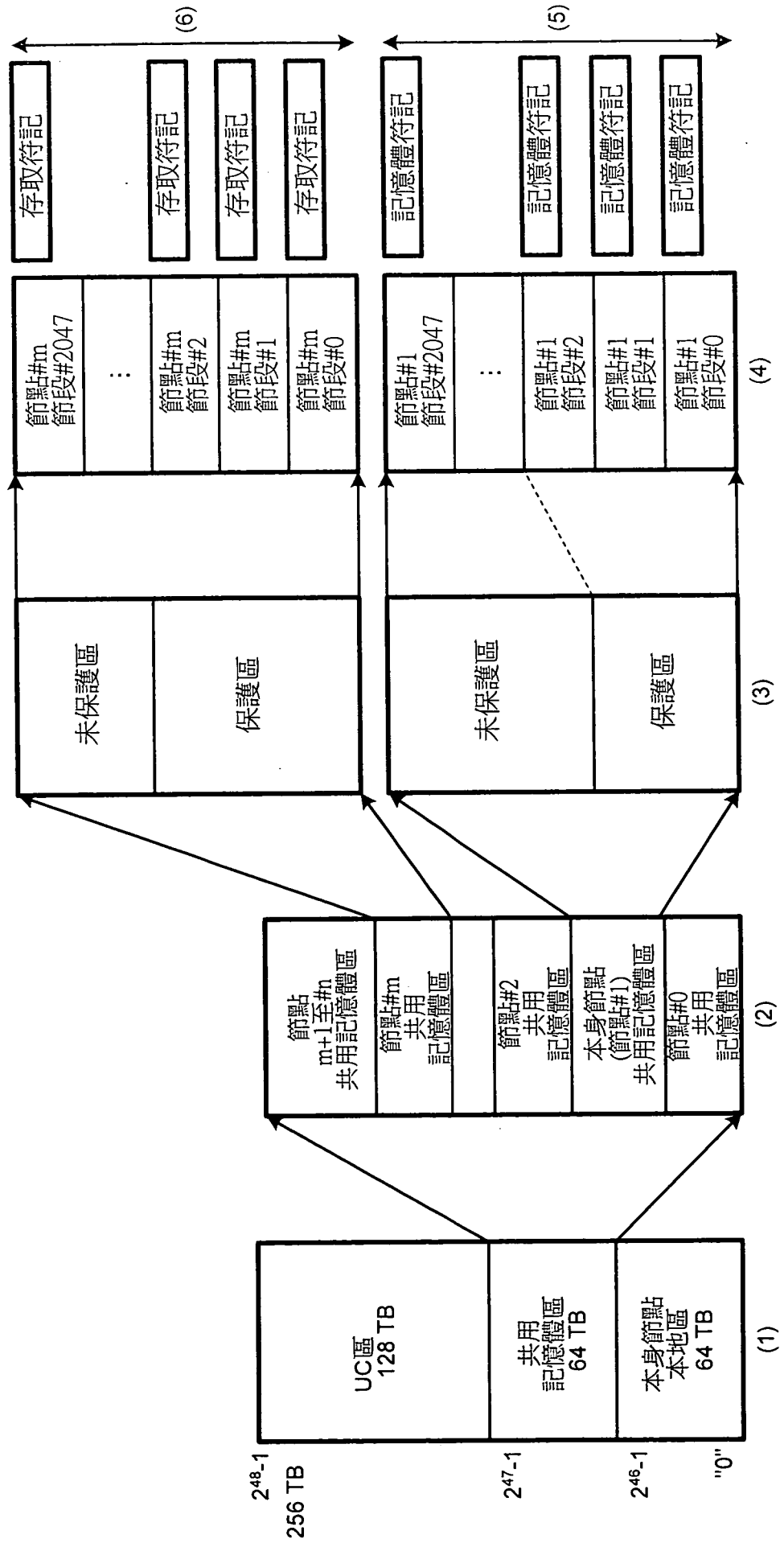


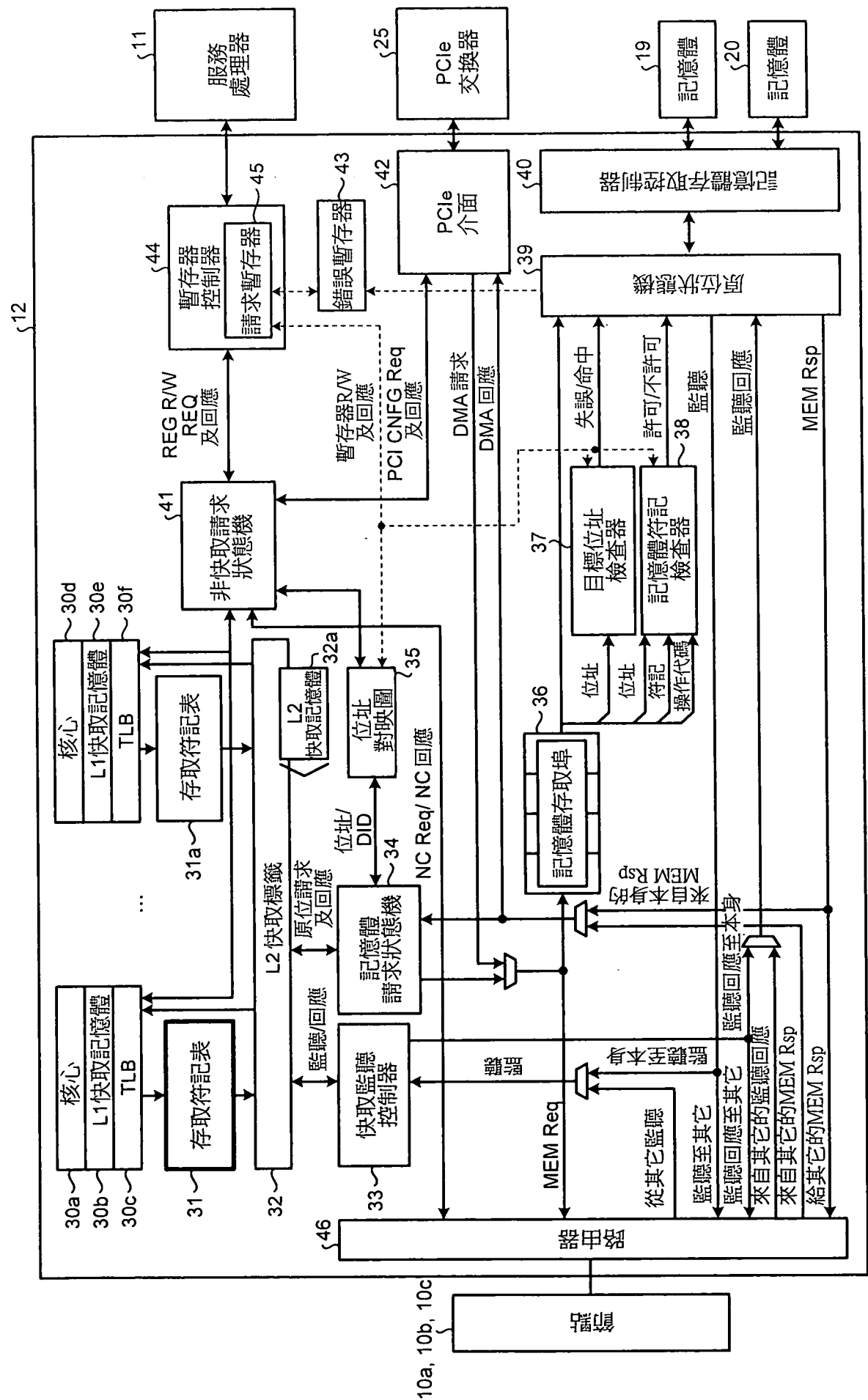
圖5

節段號碼	使用旗標	應用程式 ID	使用節點旗標					符記
			1	2	3	...	m	
0	1	100	1	x	0	...	0	3015
1	1	100	1		0	...	0	2156
2	0	x	x		x	...	x	x
3	1	150	0		1	...	1	全部皆0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2047	0	x	x	x	x	...	x	全部皆0

圖6

記憶體符記值	使用旗標	應用程式 ID
0	1	100
1	1	100
2	0	x
3	1	150
⋮	⋮	⋮
4095	0	x

圖7



10a, 10b, 10c

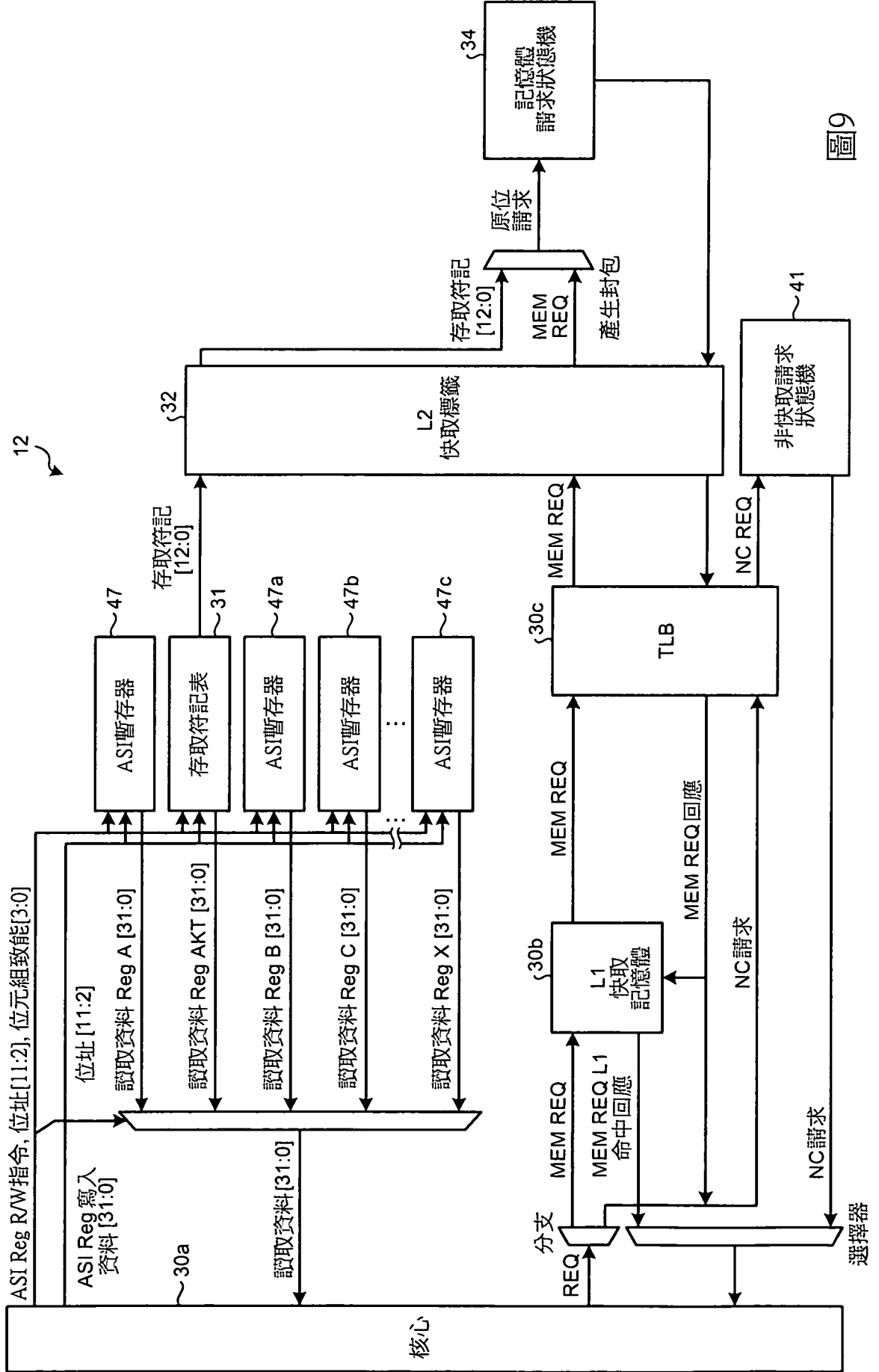


圖9

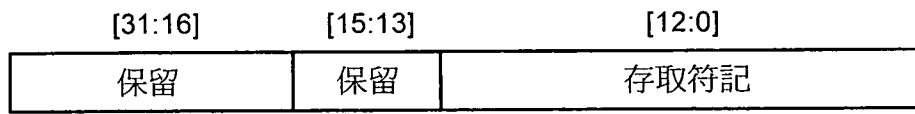


圖10

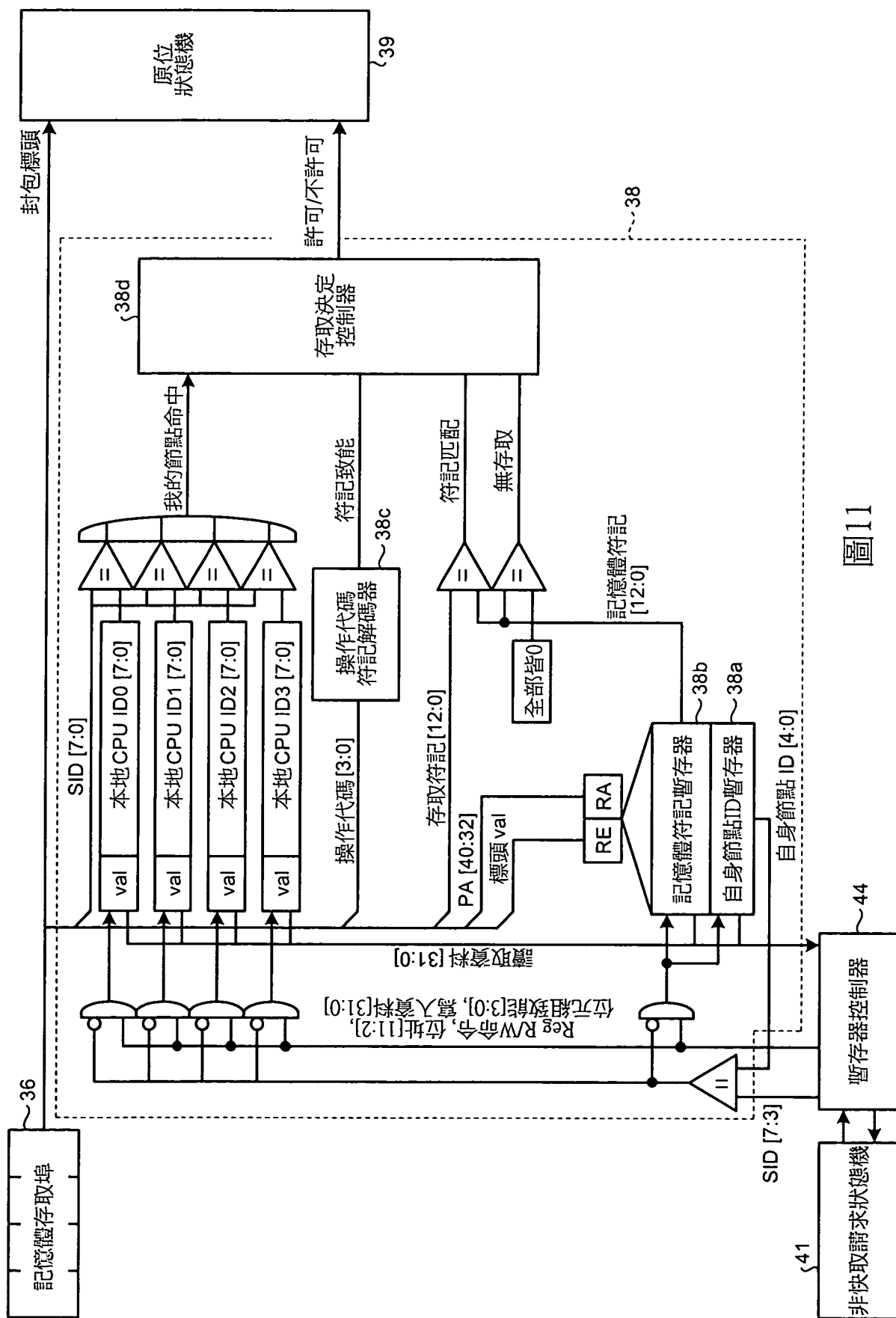


圖11

格式	目的地ID	操作代碼	傳輸來源ID	分錄ID	符記	目標位址
----	-------	------	--------	------	----	------

圖12

操作代碼	操作代碼名稱	符記致能	操作代碼之意義
0000	READ IF	1	從核心讀取指令區，獲取快取記憶體於S態
0001	READ OP		從核心讀取資料區，獲取快取記憶體於S/E態
0010	READ EX		從核心讀取互斥權獲取以儲存，獲取快取記憶體於S/E態
0100	DMA讀取	0	從IO讀取DMA，不獲取快取
0110	DMA寫入		從IO寫入DMA(128B完整)，不獲取快取
0111	DMA部分寫入		從IO寫入DMA(部分)，不獲取快取
1000	回寫		在CPU快取驅離時回寫M資料
1001	回快閃		在CPU快取驅離時將得自S/E態之通知改變成I
1010	快取刷新		得自核心的快取刷新

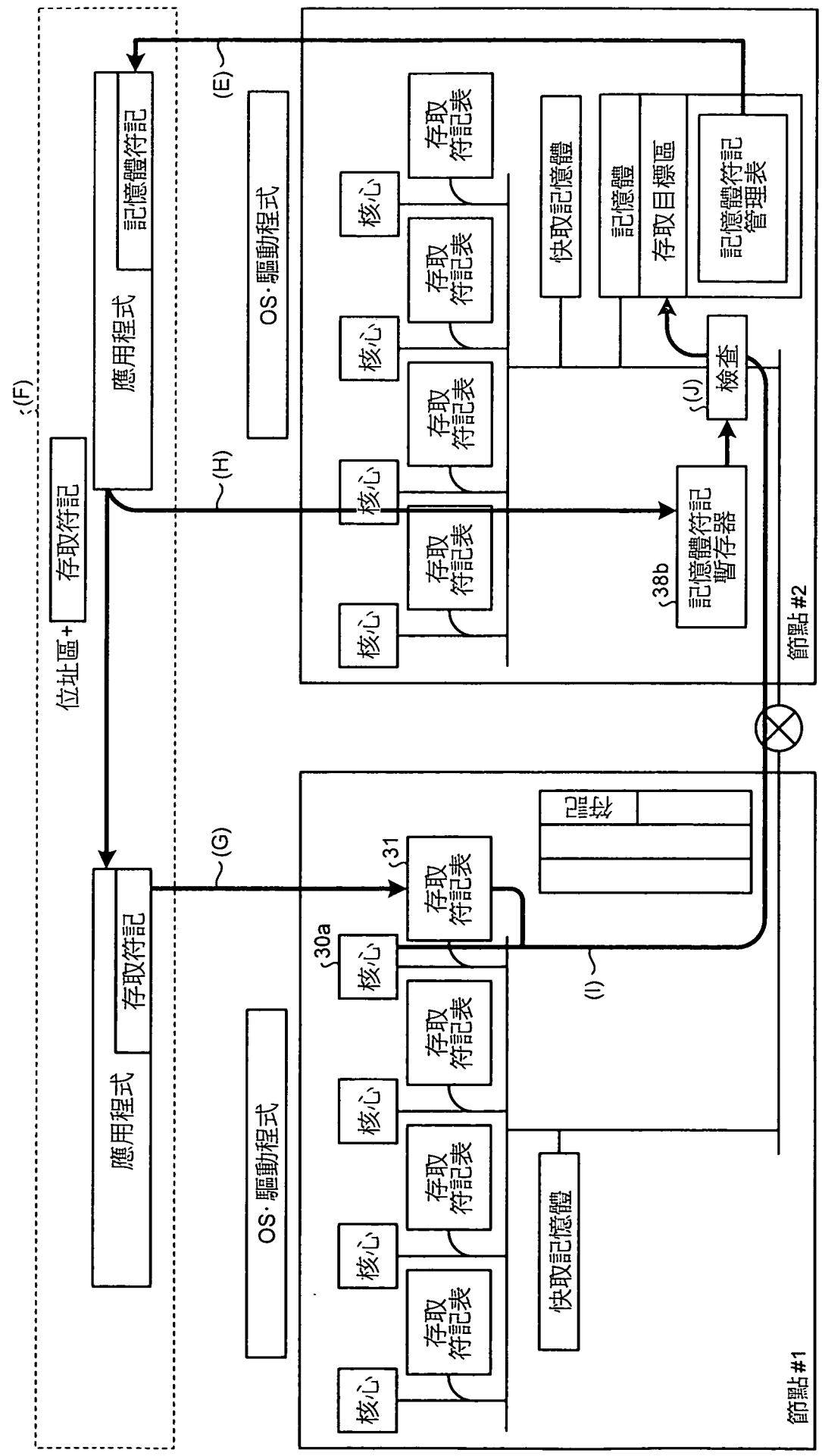
圖13

我的節點命中	符記致能	符記匹配	無存取	許可/不許可
否	是	否	否	不許可
其他				許可

圖14

許可/不許可	目標位址檢查器	原位機
不許可	選項	執行存取例外操作
許可	失誤	執行存取例外操作
	命中	執行存取

圖15



節點#1

節點#2

圖16

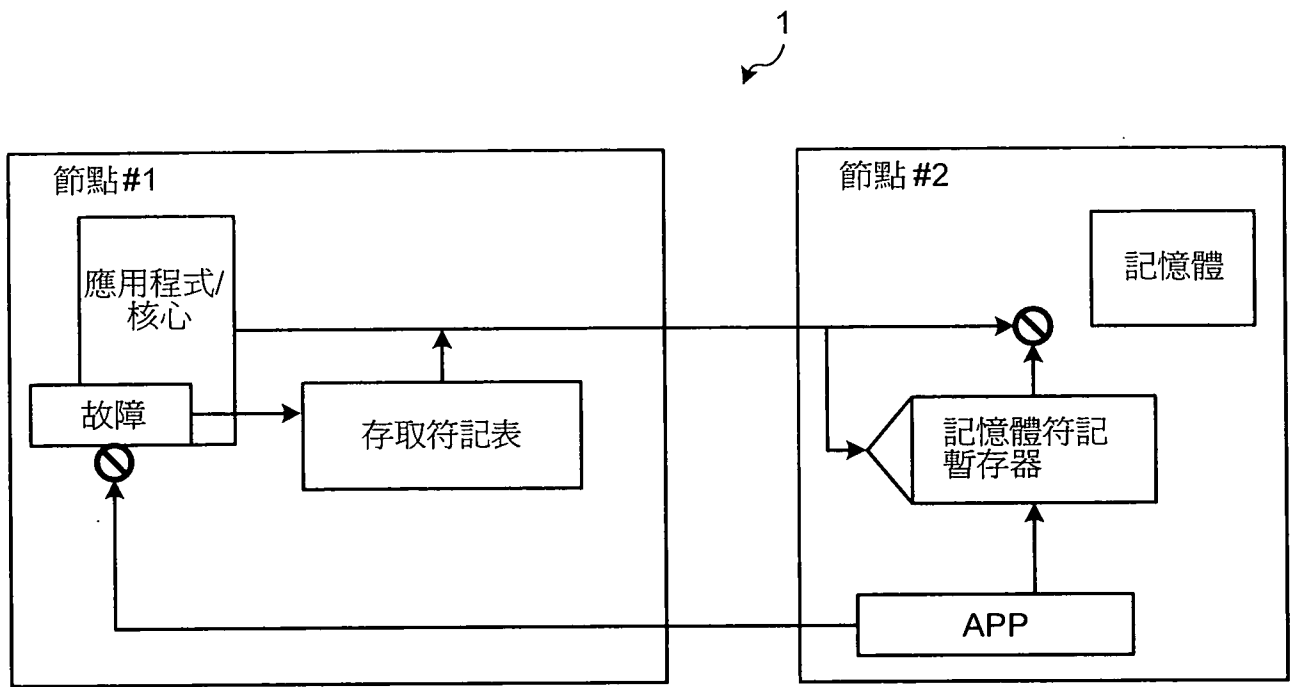


圖17

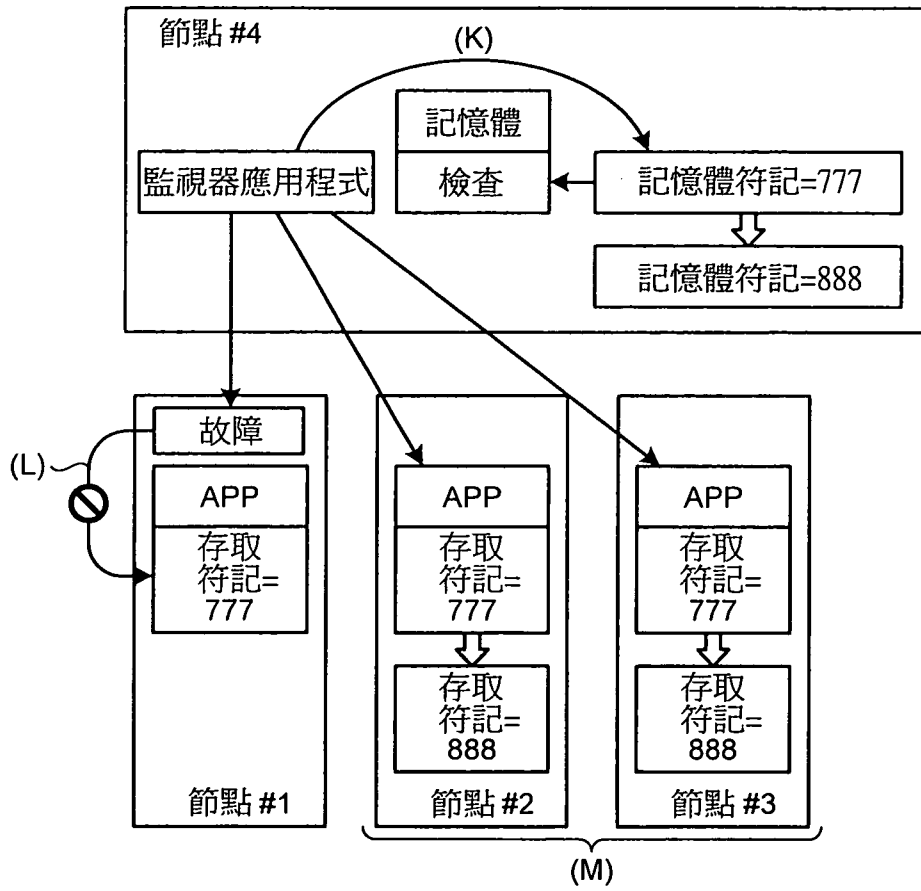


圖18

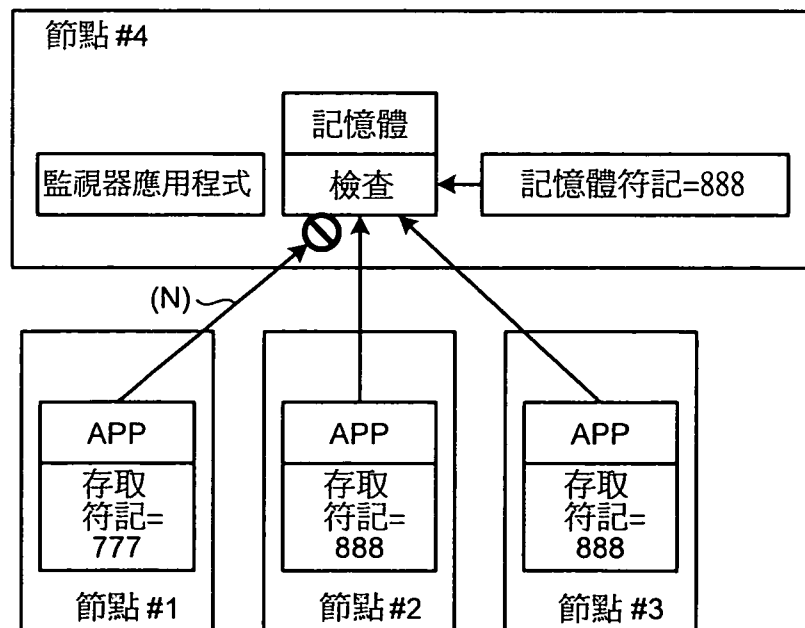


圖19

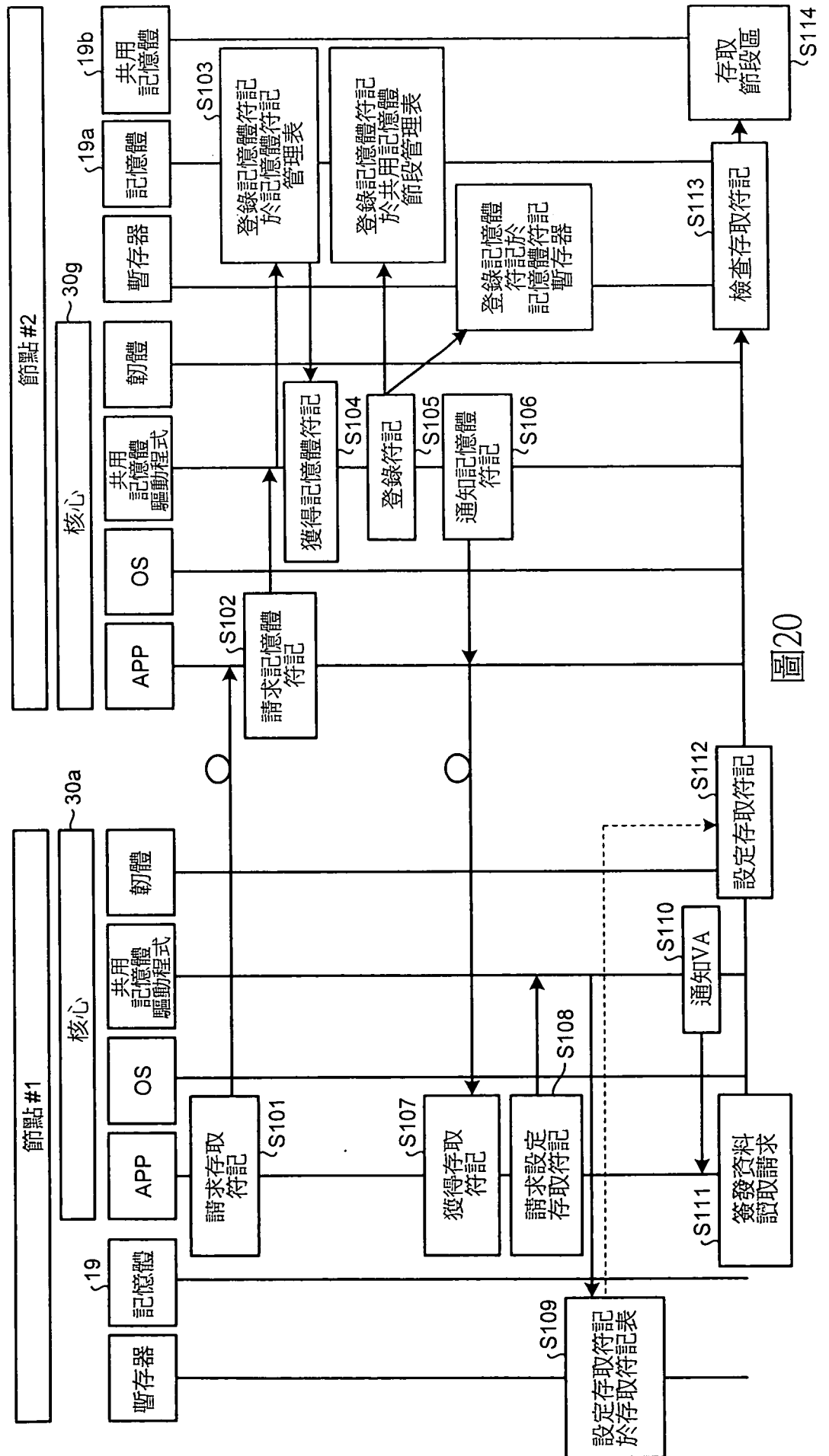


圖 20

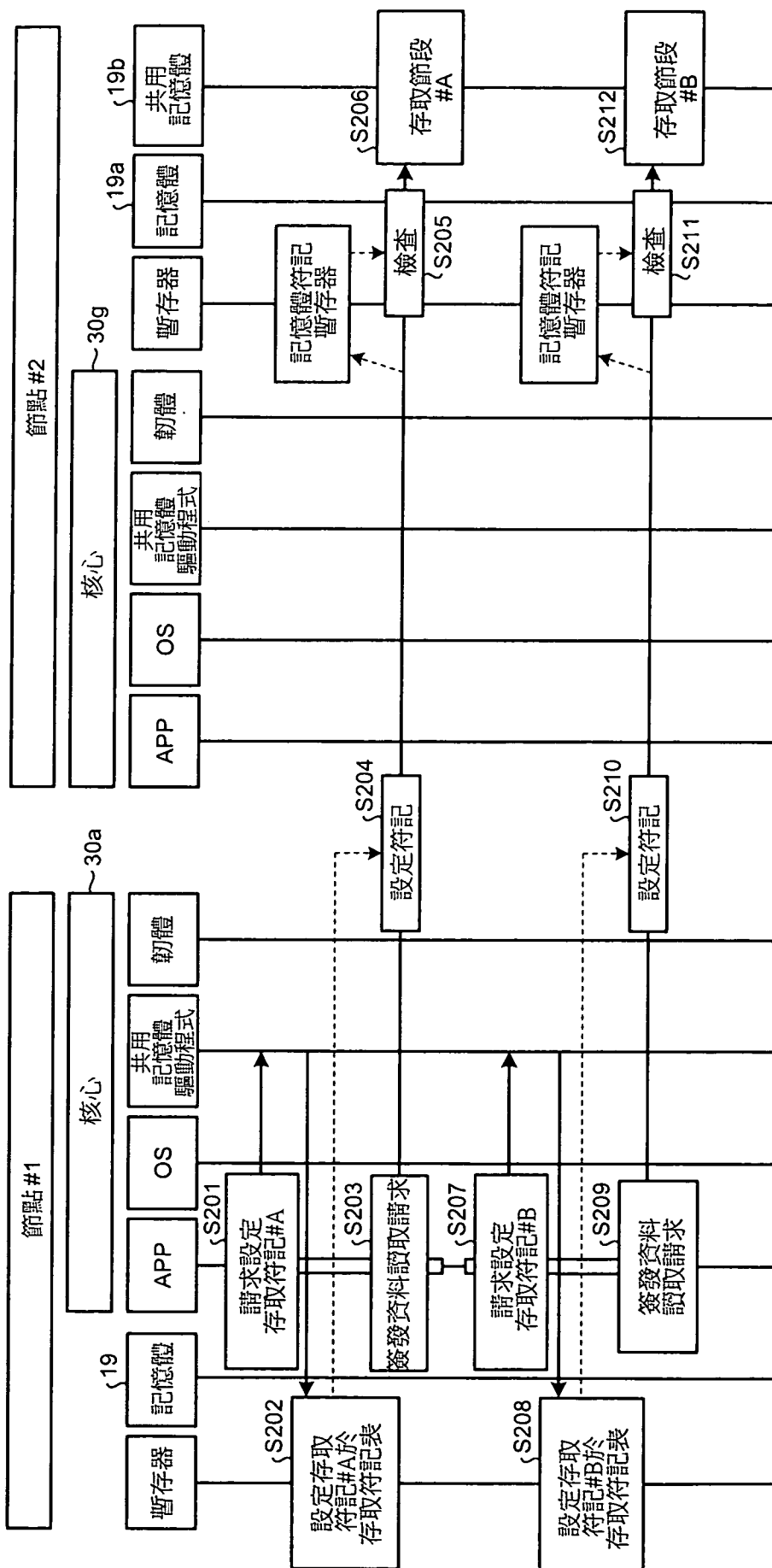


圖21

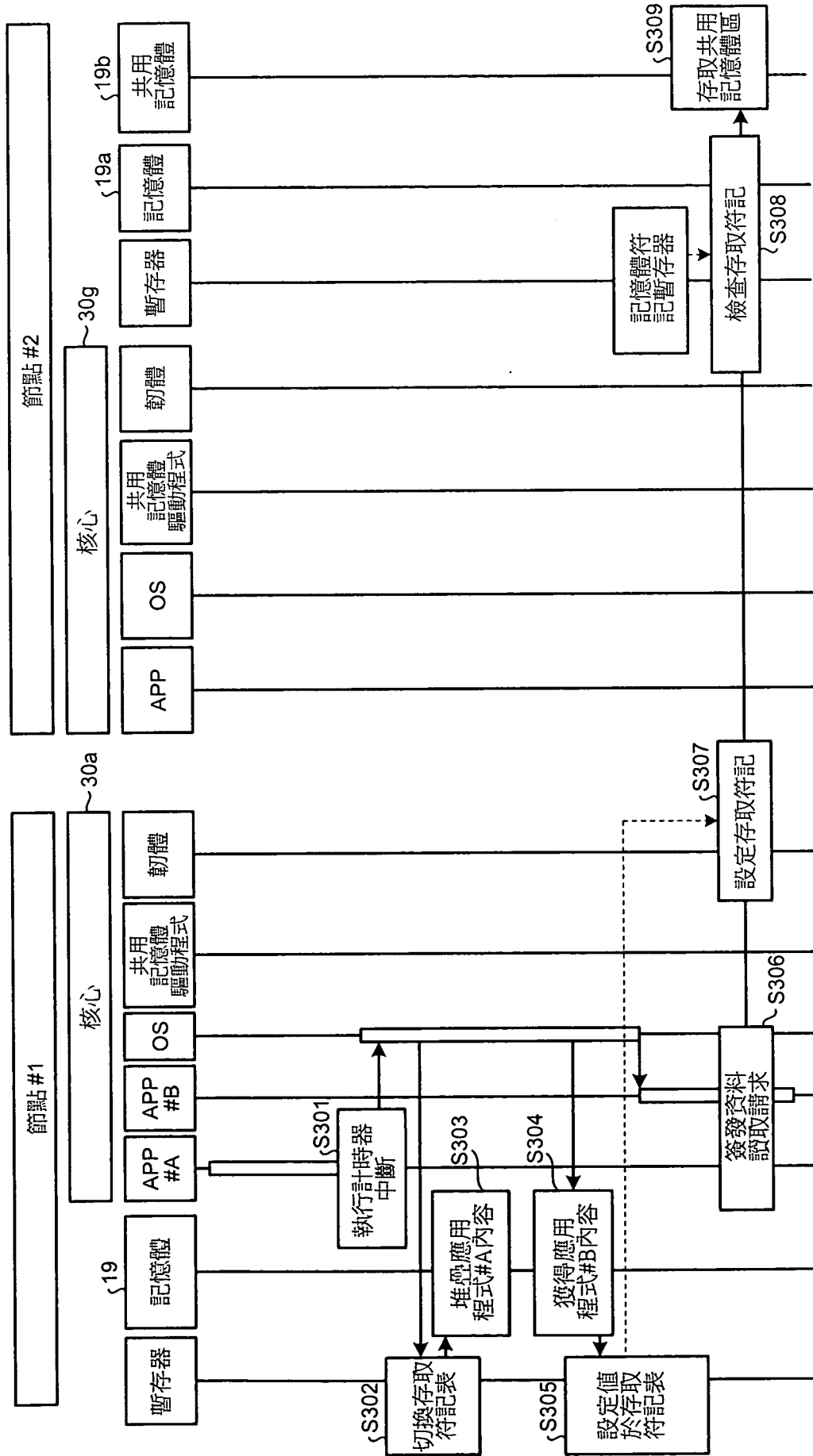


圖22

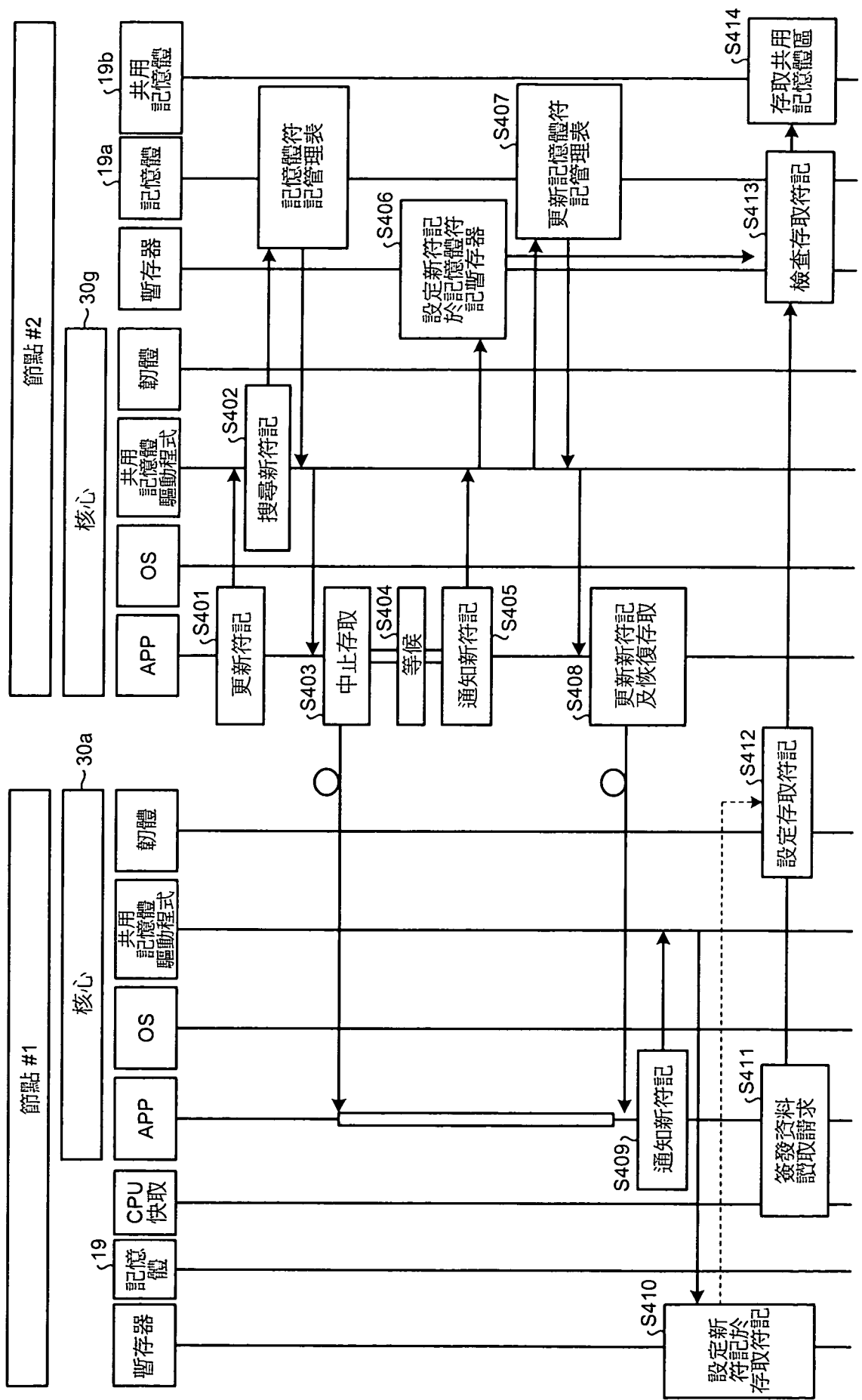


圖 23

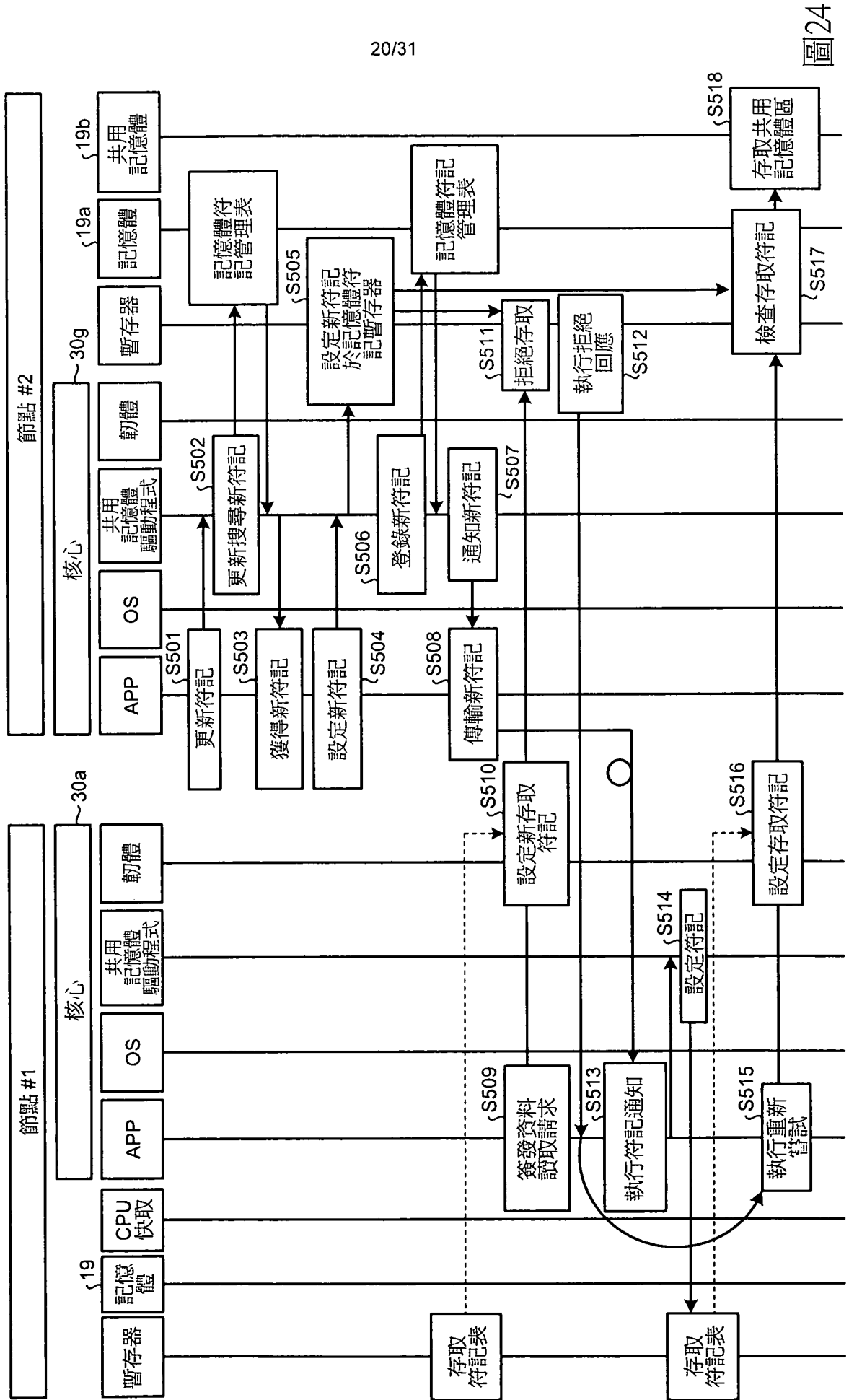
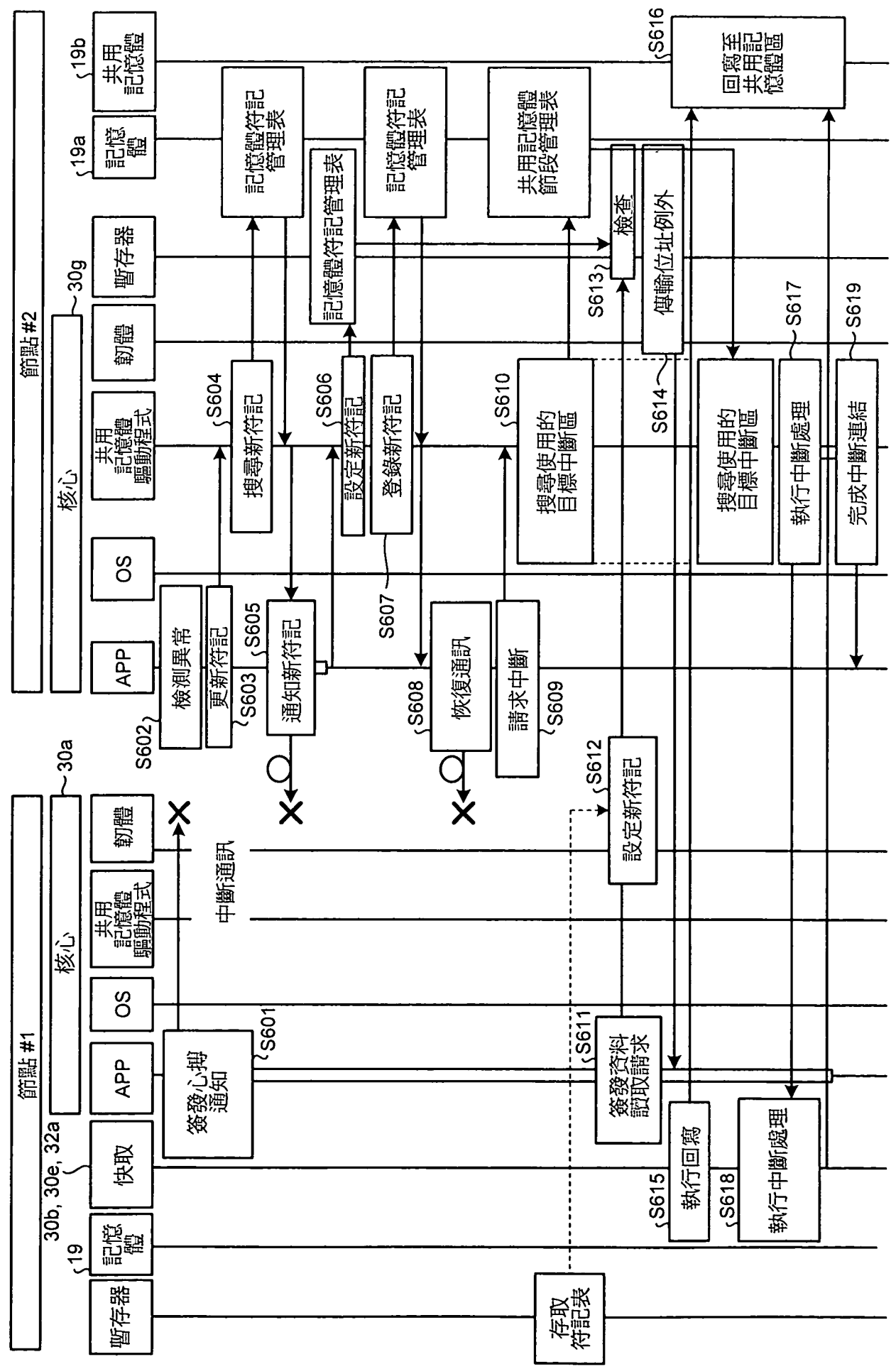


圖24



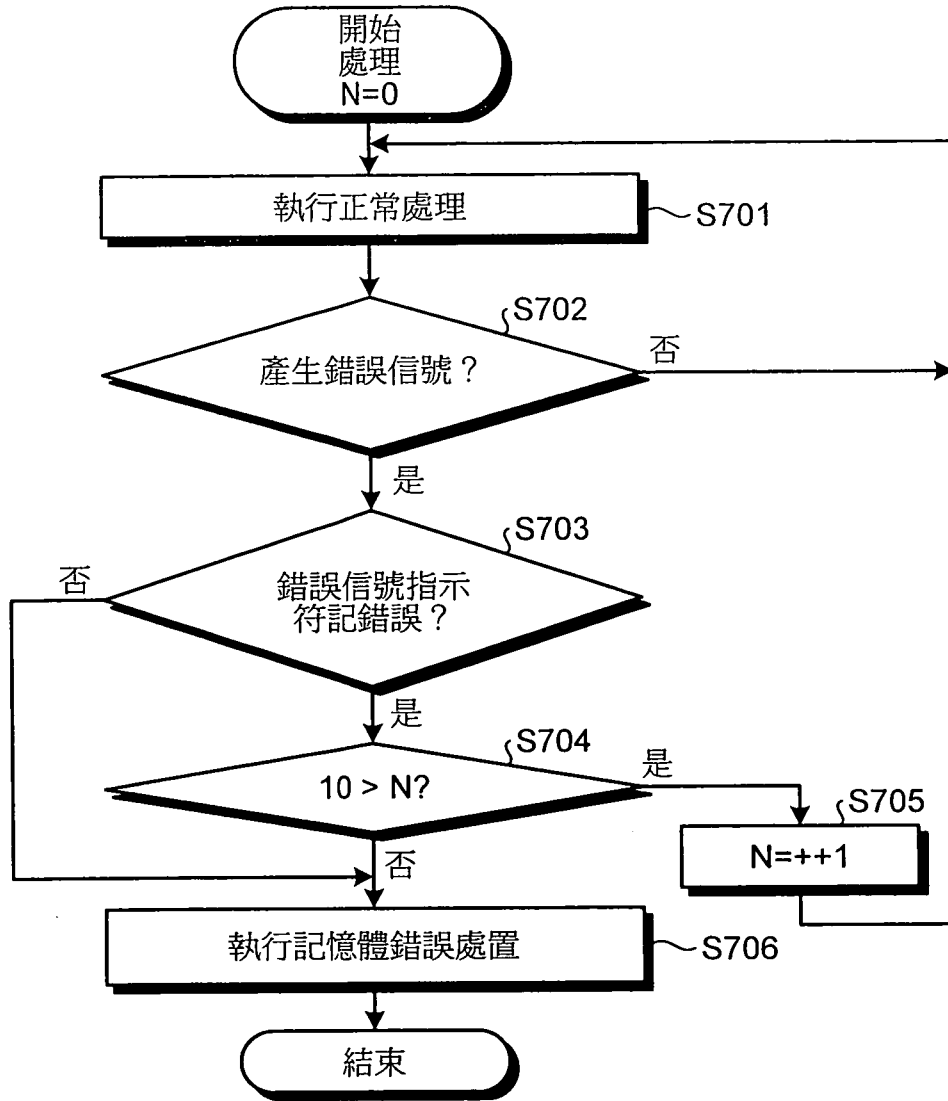


圖26

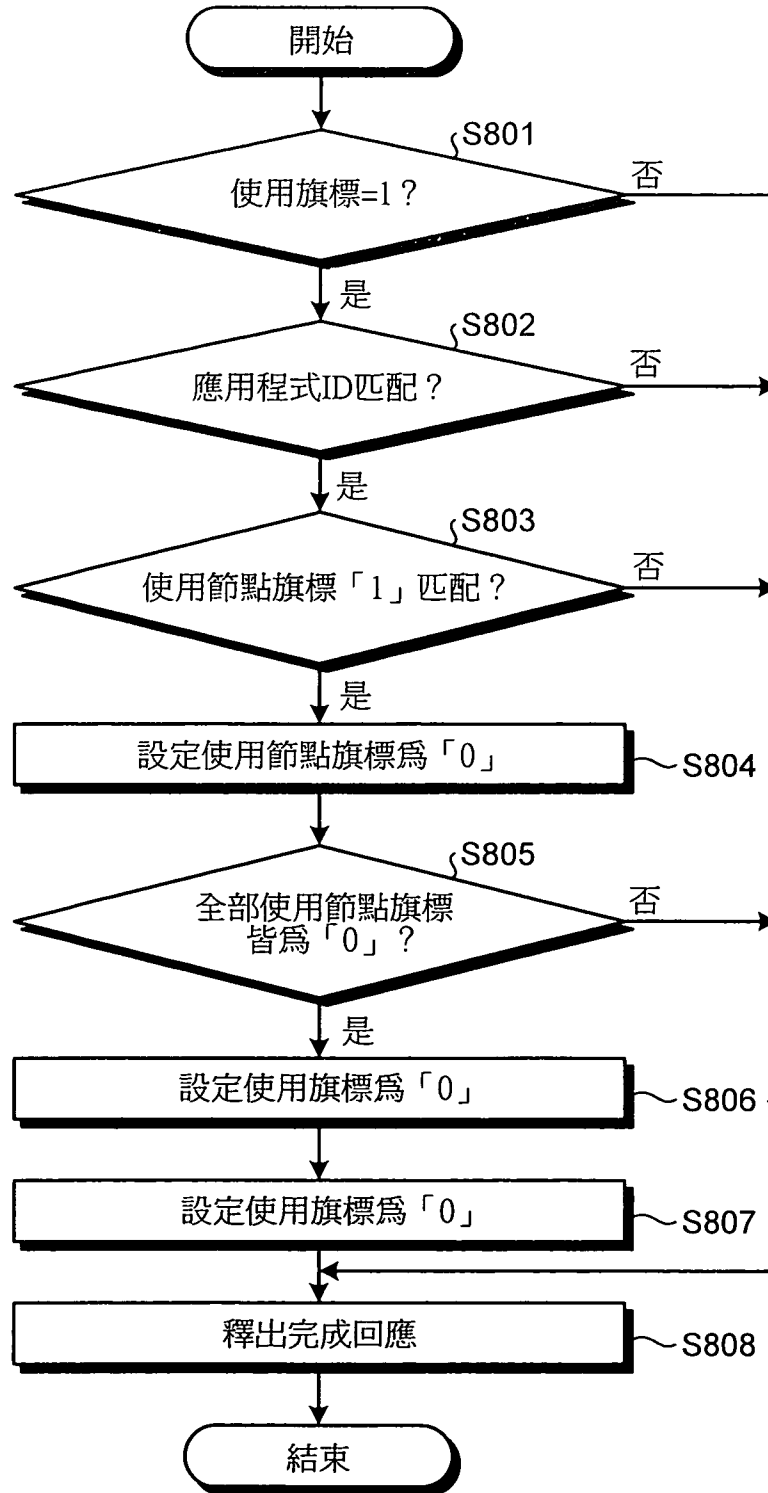


圖27

節段號碼	使用旗標	應用程式	使用節點旗標					符記
			1	1	2	...	m	
0	1	100	1	x	0	...	0	0
1	1=>0	100	1=>0		0	...	0	1
2	0	x	x		x	...	x	x
3	1	150	0		1	...	1	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2047	0	x	x	x	x	...	x	x

圖28

記憶體符記值	使用旗標	應用程式
0	1	100
1	1=>0	100
2	0	x
3	1	150
⋮	⋮	⋮
4095	0	x

圖29

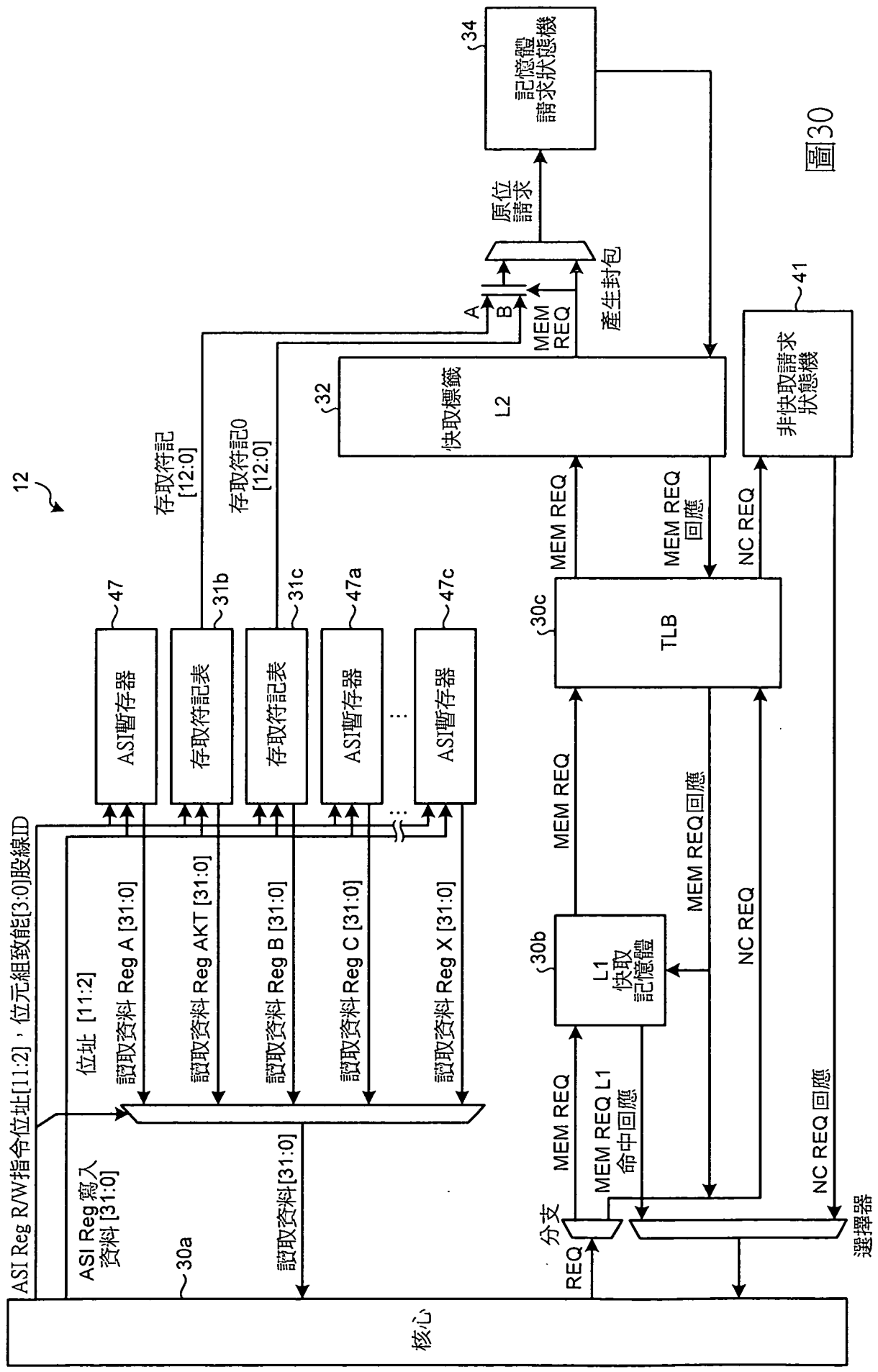


圖30

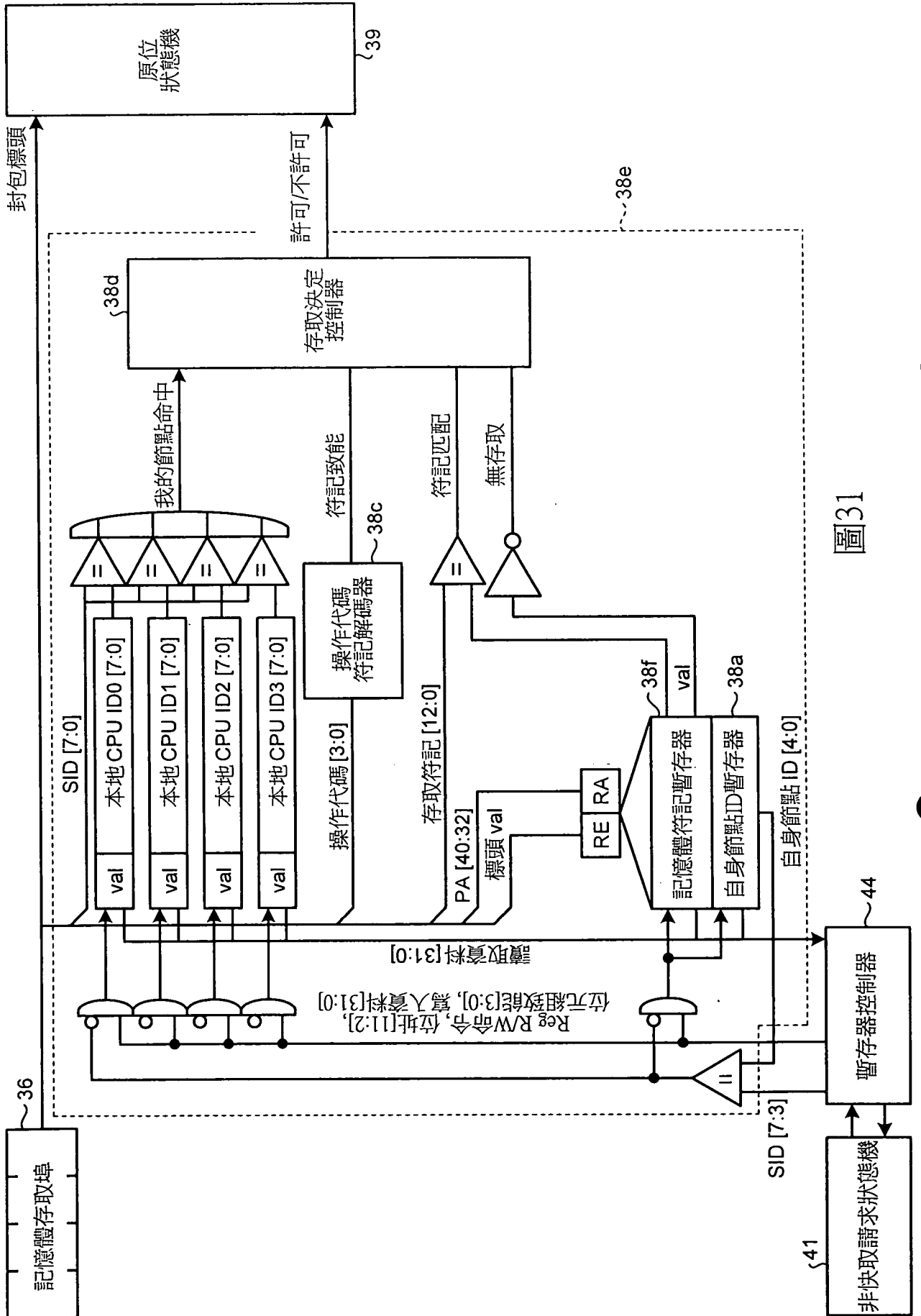


圖31

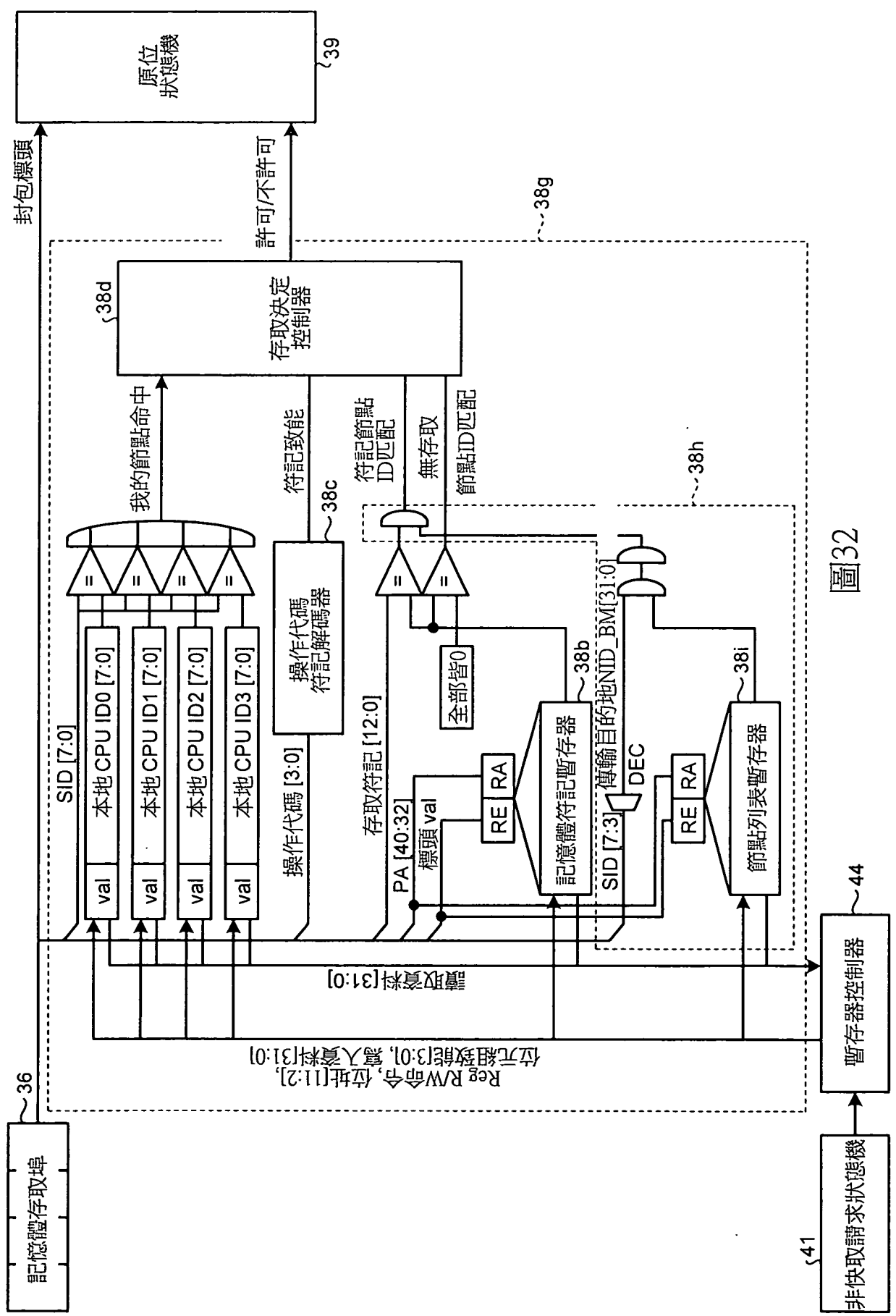


圖32

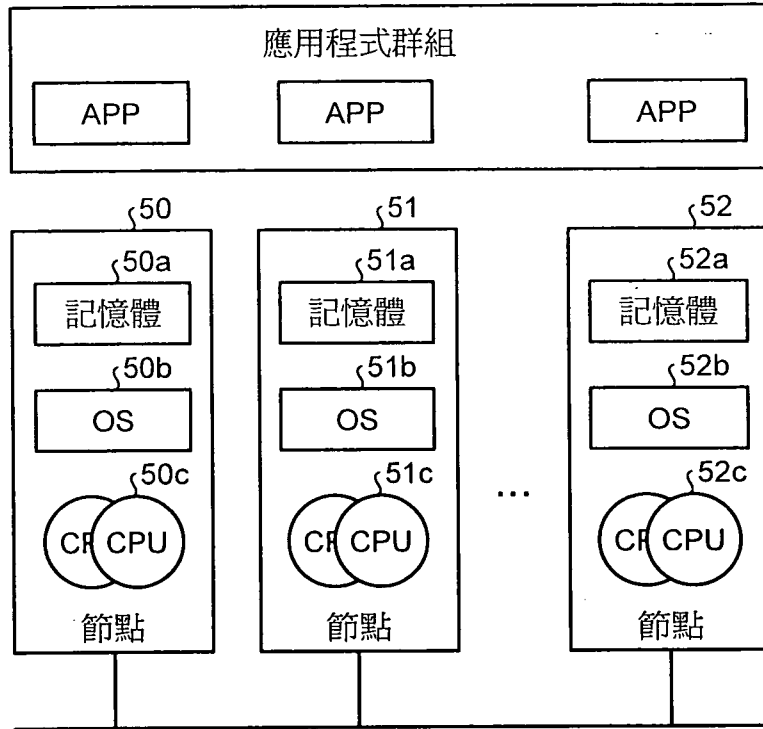


圖33

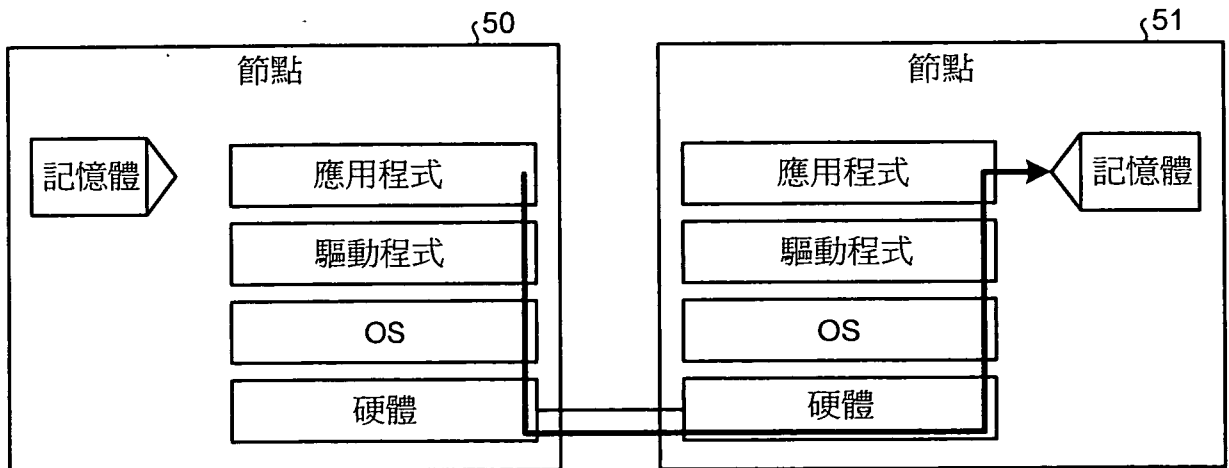


圖34

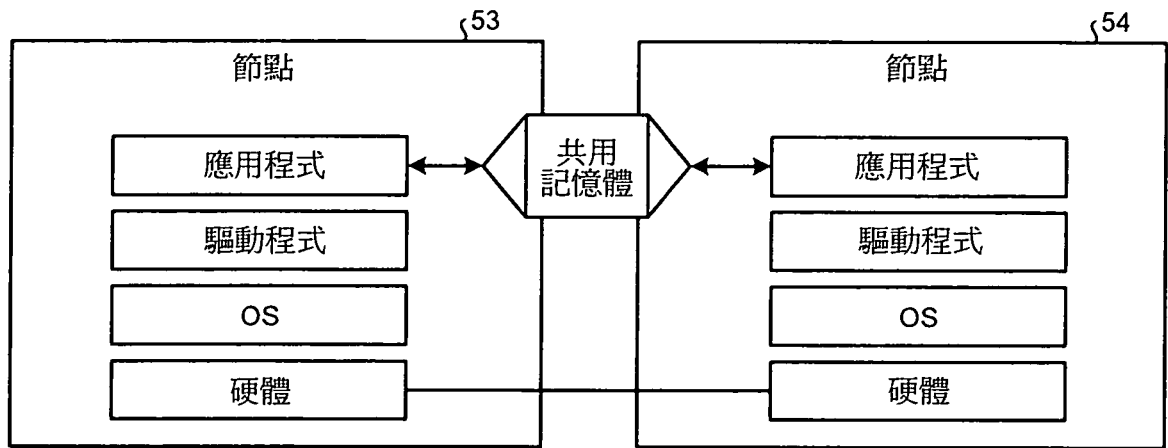


圖35

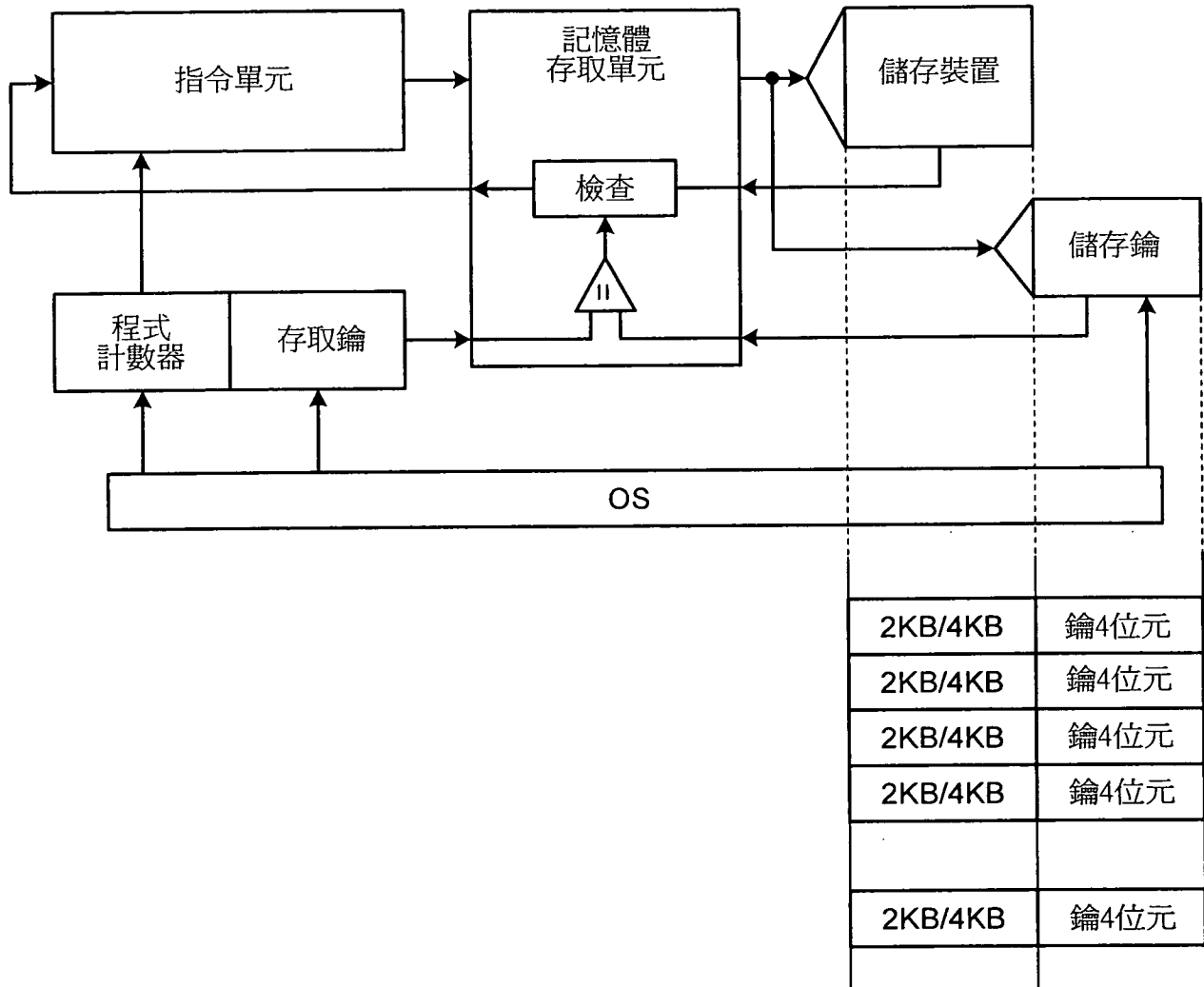


圖36

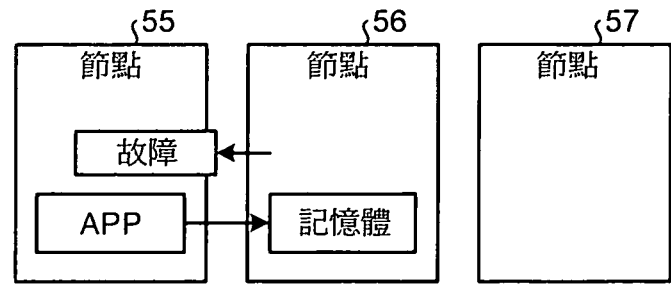


圖37A

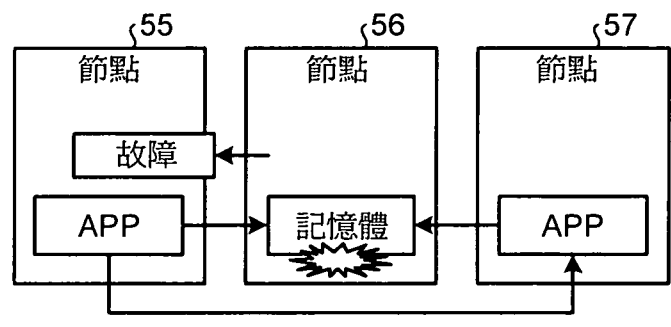


圖37B