

發明專利說明書

(本申請書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※申請案號：94124585

※申請日期：94年07月20日

※IPC分類：G06F15/16(2006.01)

一、發明名稱：

(中) 資訊處理裝置及資訊處理方法、以及記錄有程式之電腦可讀取之媒體
(英)

二、申請人：(共 1 人)

1. 姓 名：(中) 新力股份有限公司
(英) SONY CORPORATION

代表人：(中) 1. 中鉢良治
(英) 1. CHUBACHI, RYOJI

地 址：(中) 日本國東京都品川區北品川六丁目七番三五號
(英) 7-35, Kitashinagawa 6-Chome, Shinagawa-Ku, Tokyo, Japan

國籍：(中英) 日本 JAPAN

三、發明人：(共 2 人)

1. 姓 名：(中) 五十崎正明
(英) ISOZAKI, MASAOKI

國 籍：(中) 日本
(英) JAPAN

2. 姓 名：(中) 荒木祐一
(英) ARAKI, YUICHI

國 籍：(中) 日本
(英) JAPAN

四、聲明事項：

◎本案申請前已向下列國家(地區)申請專利 ☐ 主張國際優先權：

【格式請依：受理國家(地區)；申請日；申請案號數 順序註記】

1. 日本 ; 2004/07/20 ; 2004-211420 ☒ 有主張優先權

I289766

(此處由本局於收
文時黏貼條碼)

758174本
公告

發明專利說明書

(本申請書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※申請案號：94124585

※申請日期：94年07月20日

※IPC分類：G06F15/16(2006.01)

一、發明名稱：

(中) 資訊處理裝置及資訊處理方法、以及記錄有程式之電腦可讀取之媒體
(英)

二、申請人：(共 1 人)

1. 姓 名：(中) 新力股份有限公司
(英) SONY CORPORATION

代表人：(中) 1. 中鉢良治
(英) 1. CHUBACHI, RYOJI

地 址：(中) 日本國東京都品川區北品川六丁目七番三五號
(英) 7-35, Kitashinagawa 6-Chome, Shinagawa-Ku, Tokyo, Japan

國籍：(中英) 日本 JAPAN

三、發明人：(共 2 人)

1. 姓 名：(中) 五十崎正明
(英) ISOZAKI, MASAOKI

國 籍：(中) 日本
(英) JAPAN

2. 姓 名：(中) 荒木祐一
(英) ARAKI, YUICHI

國 籍：(中) 日本
(英) JAPAN

四、聲明事項：

◎本案申請前已向下列國家(地區)申請專利 ☐ 主張國際優先權：

【格式請依：受理國家(地區)；申請日；申請案號數 順序註記】

1. 日本 ; 2004/07/20 ; 2004-211420 ☒ 有主張優先權

(1)

九、發明說明

【發明所屬之技術領域】

本發明係有關於資訊處理裝置及資訊處理方法以及程式，尤其是適用於複數處理器所致之分散處理被執行時的資訊處理裝置及資訊處理方法以及程式。

【先前技術】

近年來，使用複數處理器或電腦，讓處理被分散執行的分散處理受到矚目。作為進行分散處理的方法有，令透過通訊網而連接之複數電腦來執行處理之方法，或令被設於1台電腦上的複數處理器來執行處理之方法，以及將上述2種方法加以組合來執行處理之方法。

要求（指示）執行分散處理的裝置或處理器，係將用來執行分散處理所必需的資料及程式，送訊至被要求執行之其他裝置或其他處理器。收到分散處理所必需的資料及程式的裝置或處理器，會執行所要求的處理，並將施行過所要求之處理的資料，送訊至分散處理之要求來源的裝置或處理器。

身為分散處理要求來源的裝置或處理器，係將從被要求執行之其他裝置或其他處理器所送訊過來的資料加以接收，並根據所收到的資料，執行所定的處理，或將收到的資料予以記錄。

先前，如上述，已有藉由使用均等的模組構造、共通的運算模組，以及均等的軟體來執行分散處理，來實現高

(2)

速處理用電腦架構的技術（例如申請專利範圍第 1 至專利文獻 5）。

又，近年來，資訊處理裝置的時脈速度之提升和積體度的提升，可將非常高速處理器聚集在同一晶片上，因此複數處理器所致之分散處理，可以不必藉助大型的裝置，就能加以實現。

〔專利文獻 1〕日本特開 2002-342165 號公報

〔專利文獻 2〕日本特開 2002-351850 號公報

〔專利文獻 3〕日本特開 2002-358289 號公報

〔專利文獻 4〕日本特開 2002-366533 號公報

〔專利文獻 5〕日本特開 2002-366534 號公報

上述專利文獻中，作為基本的處理用模組係為處理器元素（PE）。PE，係具備有處理單元（PU）、直接記憶體存取控制器（DMAC）及複數之附加處理單元（APU），亦即相對於主處理器的複數子處理器。

【發明內容】

〔發明所欲解決之課題〕

近年來，隨著資訊處理裝置的時脈速度之提升與積體度之提升，雖然使得處理上所必須之消費電力已較先前大幅提高，但在先前的分散處理系統中，都沒有為了讓其執行所被要求的處理且抑制處理器（或處理器群）的消費電力，而考慮到被分散之處理的分配目標處理器之選擇方法。

(3)

本發明係有鑑於此種狀況而研發，爲了能夠令其執行被要求的處理，且可抑制處理器（或處理器群）之消費電力，而適切地選擇被分散之處理的分配目標，並使以邏輯執行緒方式而動作中之機能保持在正常狀態，而可實現高信賴性之資訊處理。

〔用以解決課題之手段〕

本發明之資訊處理裝置，其特徵爲，具備：第 1 資訊處理手段；和複數之第 2 資訊處理手段；第 1 資訊處理手段係具備：應用程式執行控制手段，其係控制應用程式之執行；和動作資訊取得手段，其係取得第 2 資訊處理手段之動作的相關資訊；和分散處理控制手段，藉由其執行是受到應用程式執行控制手段加以執行控制的應用程式，而將第 1 資訊處理手段及複數之第 2 資訊處理手段所被分配之複數處理總結成一個提供機能的處理單位，並根據動作資訊取得手段所取得到的動作資訊，而控制爲了提供處理單位所對應之機能所需之分散處理；分散處理控制手段，係控制著處理單位所對應之分散處理，使得第 1 資訊處理手段及第 2 資訊處理手段中，相對於全處理能力之使用中的處理能力的比率，不會超過 1 以上。

第 1 資訊處理手段中，係可更具備：時脈數比設定手段，其係根據分散處理控制手段所致之控制，而將相對於第 1 資訊處理手段及第 2 資訊處理手段所能動作之最大動作時脈數的、現在可能動作之時脈數的比，加以設定。

(4)

時脈數比設定手段，係可設定時脈數的比，使得相對於第 1 資訊處理手段及第 2 資訊處理手段所能動作之最大動作時脈數的、現在可能動作之時脈數的比，成為共通。

時脈數比設定手段，係可將相對於第 1 資訊處理手段及第 2 資訊處理手段所能動作之最大動作時脈數的、現在可能動作之時脈數的比，以第 1 資訊處理手段及第 2 資訊處理手段為個別獨立的方式，而加以設定。

分散處理控制手段，係對可能動作之最大動作時脈數的設定，可針對相對於全處理能力之使用中的處理能力之比率為低的第 2 資訊處理手段，讓其優先地被分配處理單位所對應之分散處理。

本發明之資訊處理方法，係屬於具備第 1 資訊處理手段和複數之第 2 資訊處理手段之資訊處理裝置的資訊處理方法，其特徵為，含有：分散處理開始要求步驟，其係藉由第 1 資訊處理手段所致之應用程式的執行，而將複數之第 2 資訊處理手段所被分配之複數處理總結成一個提供機能的處理單位，並要求為了提供處理單位所對應之機能所需之分散處理之開始；和分配控制步驟，其係根據第 1 資訊處理手段及第 2 資訊處理手段之動作相關的動作資訊，來控制前記處理單位所對應之前記分散處理的分配，使得第 1 資訊處理手段及第 2 資訊處理手段中，相對於全處理能力之使用中的處理能力的比率，不會超過 1 以上。

本發明之資訊處理程式，係屬於用來令電腦執行有使用第 1 資訊處理手段與複數第 2 資訊處理手段的分散處理

(5)

所需之程式，其特徵為，令電腦執行含有以下步驟之處理：

：分散處理開始要求步驟，其係藉由第 1 資訊處理手段所致之應用程式的執行，而將複數之第 2 資訊處理手段所被分配之複數處理總結成一個提供機能的處理單位，並要求為了提供處理單位所對應之機能所需之分散處理之開始；

和分配控制步驟，其係根據第 1 資訊處理手段及第 2 資訊處理手段之動作相關的動作資訊，來控制前記處理單位所對應之前記分散處理的分配，使得第 1 資訊處理手段及第 2 資訊處理手段中，相對於全處理能力之使用中的處理能力的比率，不會超過 1 以上。

本發明之資訊處理裝置及資訊處理方法以及程式中，係藉由第 1 資訊處理手段所致之應用程式之執行，而執行分散處理；並根據第 1 資訊處理手段及第 2 資訊處理手段之動作相關的動作資訊，以使得第 1 資訊處理手段及第 2 資訊處理手段中，相對於全處理能力的使用中之處理能力的比率不會超過 1 以上的方式，來控制處理單位所對應之分散處理的分配。

〔發明效果〕

若根據本發明，則可執行分散處理。尤其是，因為是使得第 1 資訊處理手段及第 2 資訊處理手段中，相對於全處理能力的使用中之處理能力的比率不會超過 1 以上的方式，來控制處理單位所對應之分散處理的分配，所以能夠使得以邏輯執行緒方式而動作中之機能保持在正常狀態

(6)

，而可實現高信賴性之資訊處理。

【實施方式】

以下將說明本發明的實施形態，但若將本說明書所記載之發明，和發明實施形態的對應關係加以例示，則如下所示。本記載，係用來確認支持本說明書中所記載之發明的實施形態，是有記載於本說明書中。因此，雖然發明的實施形態中有所記載，但作為對應於發明者，就算是這裡所未有記載的實施形態，其並非意味著該實施形態並非本發明所對應者。反之，就算是實施形態是以對應於發明的方式而記載於此，這也並非意味著該實施形態，並無對應於本發明以外之發明。

再者，本記載並非代表本說明書中所記載之發明的全體。換言之，本記載並不能夠否定，屬於本說明書所記載之發明且該申請中尚未請求之發明的存在，亦即，不能否定將來可能分割申請、藉由修正而出現、追加之發明的存在。

申請專利範圍第 1 項之資訊處理裝置，其特徵為，具備：第 1 資訊處理手段（例如圖 1 的主處理器 42）；和複數之第 2 資訊處理手段（例如圖 1 的子處理器 43）；第 1 資訊處理手段係具備：應用程式執行控制手段（例如圖 8 的應用程式執行控制部 101），其係控制應用程式之執行；和動作資訊取得手段（例如圖 8 的動作資訊取得 104），其係取得第 2 資訊處理手段之動作的相關資訊；

(7)

和分散處理控制手段（例如圖 8 的邏輯執行緒執行控制部 102），藉由其執行是受到應用程式執行控制手段加以執行控制的應用程式，而將第 1 資訊處理手段及複數之第 2 資訊處理手段所被分配之複數處理總結成一個提供機能的處理單位，並根據動作資訊取得手段所取得到的動作資訊，而控制爲了提供處理單位所對應之機能所需之分散處理；分散處理控制手段，係控制著處理單位所對應之分散處理，使得第 1 資訊處理手段及第 2 資訊處理手段中，相對於全處理能力之使用中的處理能力的比率，不會超過 1 以上。

第 1 資訊處理手段中，係可更具備：時脈數比設定手段（例如圖 8 的動作率設定部 103），其係根據分散處理控制手段所致之控制，而將相對於第 1 資訊處理手段及第 2 資訊處理手段所能動作之最大動作時脈數的、現在可能動作之時脈數的比（例如處理器動作率 $SCPU_RATE[k]$ ），加以設定。

申請專利範圍第 6 項所記載之資訊處理方法，係屬於具備第 1 資訊處理手段（例如主處理器 42）和複數之第 2 資訊處理手段（例如子處理器 43）之資訊處理裝置的資訊處理方法，其特徵爲，含有：分散處理開始要求步驟（例如圖 11 的步驟 S1 或圖 15 的步驟 S31 之處理），其係藉由第 1 資訊處理手段所致之應用程式的執行，而將複數之第 2 資訊處理手段所被分配之複數處理總結成一個提供機能的處理單位，並要求爲了提供處理單位所對應之機能

(8)

所需之分散處理之開始；和分配控制步驟（例如圖 11 的步驟 S2 至步驟 S13 之處理或使用圖 16 及圖 17 所說明之處理），其係根據第 1 資訊處理手段及第 2 資訊處理手段之動作相關的動作資訊，來控制前記處理單位所對應之前記分散處理的分配，使得第 1 資訊處理手段及第 2 資訊處理手段中，相對於全處理能力之使用中的處理能力的比率，不會超過 1 以上。

又，即便是申請專利範圍第 7 項所記載之程式中，各步驟所對應之實施形態（但僅為一例），係相同於申請專利範圍第 6 項所記載之資訊處理方法。

以下將參照圖面，說明本發明的實施形態。

圖 1 係能夠執行分散處理的通訊系統之一實施形態的圖示。該通訊系統中，例如，係透過對應於家用網路、LAN（Local Area Network）、WAN（Wide Area Network）或網際網路等廣域網路等之網路 2，而使資訊處理裝置 1-1 至資訊處理裝置 1-n 彼此連接。

資訊處理裝置 1-1，係一旦被指示要執行由複數資訊處理裝置所致之分散處理，便生成出含有用來執行所要求之處理所需之資料及程式的軟體胞，並將所生成的軟體胞，透過網路 2，送訊至資訊處理裝置 1-2 至資訊處理裝置 1-n 之中的任何裝置。

資訊處理裝置 1-2 至資訊處理裝置 1-n，係分別將從資訊處理裝置 1-1 所送來的軟體胞加以接收，並根據所收到的軟體胞，執行被要求的處理。資訊處理裝置 1-2 至資

(9)

訊處理裝置 1-n，係在執行所被要求的處理後，將所被要求之處理結果所得到的資料，透過網路 2 送訊至資訊處理裝置 1-1。以下，當無必要一一區別資訊處理裝置 1-1 至資訊處理裝置 1-n 的時候，就簡稱為資訊處理裝置 1。

資訊處理裝置 1-1，係將從資訊處理裝置 1-1 至資訊處理裝置 1-n 之中的任一者所送來的資料加以接收，並根據所收到的資料，來執行所定的處理，或將收到的資料加以記錄。

資訊處理裝置 1-1，係被構成爲含有：至少一個資訊處理控制器 11、主記憶體 12、記錄部 13-1、記錄部 13-2、匯流排 14、操作輸入部 15、通訊部 16、顯示部 17，及碟驅動機 18。

資訊處理控制器 11，係將主記憶體 12 中所記錄的各種程式加以執行，並控制資訊處理裝置 1-1 全體。資訊處理控制器 11，係生成軟體胞，並將所生成的軟體胞，透過匯流排 14 而供給至通訊部 16。資訊處理控制器 11，係將從通訊部 16 所供給來的資料，供給至記錄部 13-1 或記錄部 13-2。資訊處理控制器 11，係根據從操作輸入部 15 所輸入之使用者指令，而將所被指定的資料，從主記憶體 12、記錄部 13-2 或記錄部 13-2 中加以取得，並已取得的資料，透過匯流排 14 而供給至通訊部 16。

又，資訊處理控制器 11 上，係被分配有能夠橫跨網路 2 全體而唯一特定出資訊處理裝置 1-1 的資訊處理裝置 ID。

(10)

資訊處理控制器 11，係具備有：匯流排 41、主處理器 42、子處理器 43-1 至子處理器 43-m、DMAC (Direct Memory Access Controller，直接記憶體存取控制器) 44、金鑰管理表記錄部 45，以及 DC (Disk Controller，碟控制器) 46。

主處理器 42，係和子處理器 43-1 至子處理器 43-m、DMAC44、金鑰管理表記錄部 45 及 DC46，透過匯流排 41 而彼此連接。又，主處理器 42 上，是被分配有當成識別子的用來特定出主處理器 42 所需之主處理器 ID。同樣地，每一個子處理器 43-1 至子處理器 43-m 上，是被一一分配有當成識別子的用來特定每一個子處理器 43-1 至子處理器 43-m 所需之子處理器 ID。

主處理器 42，係當要令透過網路 2 所連接之資訊處理裝置 1-2 至資訊處理裝置 1-n 來執行分散處理的時候，便生成軟體胞，並將所生成的軟體胞透過匯流排 41 及匯流排 14，供給至通訊部 16。又，主處理器 42，係亦可構成爲，會執行管理所需之程式以外的程式。此時，主處理器 42 便充當子處理器而發揮機能。

主處理器 42，係可對子處理器 43-1 至子處理器 43-m，令每一子處理器獨立執行程式，並以邏輯性的 1 個機能（邏輯執行緒）的方式來求出結果。亦即，主處理器 42，係進行子處理器 43-1 至子處理器 43-m 所致之程式執行的排程管理，和資訊處理控制器 11（資訊處理裝置 1-1）之整體管理。

(11)

主處理器 42，係具備近端儲存區 51-1，會將從主記憶體 12 所載入的資料及程式，暫時記憶在近端儲存區 51-1。主處理器 42，係從近端儲存區 51-1 讀取資料及程式，並根據所讀取到的資料及程式，執行各種處理。

子處理器 43-1 至子處理器 43-m，係根據主處理器 42 的控制，平行且獨立地執行程式、處理資料。甚至，亦可能構成爲，因應需要，而由主處理器 42 所執行的程式，和每一子處理器 43-1 至子處理器 43-m 所執行的每一程式，協同動作。

每一子處理器 43-1 至子處理器 43-m，係分別具備近端儲存區 51-2 至近端儲存區 51- (m+1)。每一子處理器 43-1 至子處理器 43-m，係在每一近端儲存區 51-2 至近端儲存區 51- (m+1) 中，因應需要，而將資料及程式予以暫時記憶。每一子處理器 43-1 至子處理器 43-m，係從每一近端儲存區 51-2 至近端儲存區 51- (m+1) 讀取資料及程式，並根據所讀取到的資料及程式，執行各種處理。

以下，當無必要一一區別子處理器 43-1 至子處理器 43-m 的時候，就簡稱爲子處理器 43。同樣地，當無必要一一區別近端儲存區 51-2 至近端儲存區 51- (m+1) 的時候，就簡稱爲近端儲存區 51。

DMAC44，係根據金鑰管理表記錄部 45 中所記錄的主處理器金鑰、子處理器金鑰及存取金鑰，而管理從主處理器 42 及子處理器 43 往主記憶體 12 所記憶之程式及資料的存取。

(12)

金鑰管理表記錄部 45，係記錄著主處理器金鑰、子處理器金鑰及存取金鑰。此外，關於主處理器金鑰、子處理器金鑰及存取金鑰的細節將於後述。

DC46，係管理從主處理器 42 及子處理器 43 往記錄部 13-1 及記錄部 13-2 的存取。

此外，此處雖然說明了在資訊處理裝置 1-1 中僅具備 1 個資訊處理控制器 11 的例子，但資訊處理裝置 1-1 亦可為具備有複數個資訊處理控制器 11。

主記憶體 12，例如，係由 RAM 所構成。主記憶體 12，係將主處理器 42 及子處理器 43 所執行的各種程式及資料予以暫時記憶。

記錄部 13-1 及記錄部 13-2，例如，都是由硬碟等所構成。記錄部 13-1 及記錄部 13-2，係將主處理器 42 及子處理器 43 所執行的各種程式及資料予以記錄。又，記錄部 13-及記錄部 13-2，係將從資訊處理控制器 11 所供給而來的資料予以記錄。以下，當無必要一一區別記錄部 13-及記錄部 13-2 的時候，就簡稱為記錄部 13。

又，資訊處理控制器 11，係透過匯流排 14，連接著操作輸入部 15、通訊部 16、顯示部 17 及碟驅動機 18。操作輸入部 15，例如，係由鍵盤、按鈕、觸控面板、滑鼠等所成，會接受使用者所致之操作輸入，並將操作輸入所對應之資訊，透過匯流排 14，供給至資訊處理控制器 11。

通訊部 16，係將從資訊處理控制器 11 所供給而來的

(13)

軟體胞，透過網路 2 而送訊至資訊處理裝置 1-2 至資訊處理裝置 1-n。又，通訊部 16，係將從資訊處理裝置 1-2 至資訊處理裝置 1-n 所送來的資料，透過匯流排 14 而供給至資訊處理控制器 11。

顯示部 17，例如，係由 CRT (cathode ray tube) 或 LCD (Liquid Crystal Display) 等所構成，會將透過匯流排 14 所供給而來之資訊處理控制器 11 之處理所產生的資訊（例如包含應用程式執行所產生的資料、應用程式執行所必須要通知給使用者等的資訊）加以顯示。

碟驅動機 18，係當裝著了磁碟片 61、光碟 62、光磁碟 63、或半導體記憶體 64 等時，便將它們予以驅動，以取得其中所記錄的程式或資料。所取得之程式或資料，係因應需要，而透過匯流排 14，轉送至資訊處理控制器 11，藉由資訊處理控制器 11 而記錄至記錄部 13。

又，資訊處理裝置 1 中，爲了對資訊處理裝置 1 全體進行電源供給，而設有電源部 19。

此外，資訊處理裝置 1-2 至資訊處理裝置 1-n，由於係和資訊處理裝置 1-1 的構成相同，因此省略其說明。資訊處理裝置 1-2 至資訊處理裝置 1-n 係不限於上述構成，可因應需要，而追加或刪減機能，可具有對應其機能的構成。

其次，參照圖 2 至圖 4，說明子處理器 43 在對主記憶體 12 進行存取時的處理。

如圖 2 所示，主記憶體 12 中，配置著能夠指定複數

(14)

位址的記憶體位置 (memory location)。對各記憶體位置，係分配有用來存放表示資料狀態之資訊所需的追加區段。追加區段，係含有 F/E 位元、子處理器 ID 及 LS 位址 (Local Storage Address)。又，各記憶體位置內，還如後述般地分配有存取金鑰。

值為“0”的 F/E 位元，係為子處理器 43 所讀取之處理中的資料、或因為呈空白狀態而為非最新資料的無效資料，是代表不可從該記憶體位置讀取。又，值為“0”的 F/E 位元，係表示該當記憶體位置內是可寫入資料，且一旦寫入資料後，F/E 位元便被設定成 1。

值為“1”的 F/E 位元，係表示該當記憶體位置的資料是未被子處理器 43 讀取，表示是未處理的最新資料。F/E 位元為“1”的記憶體位置的資料是可被讀取，在被子處理器 43 讀取後，F/E 位元便被設定成“0”。又，值為“1”的 F/E 位元，係表示記憶體位置是不可寫入資料。

再者，上記 F/E 位元為“0”（不可讀取/可寫入）之狀態下，可針對該當記憶體位置設定讀取預約。對 F/E 位元為“0”的記憶體位置進行讀取預約的時候，子處理器 43 係將子處理器 43 的子處理器 ID 及 LS 位址做為讀取預約資訊而寫入至讀取預約的記憶體位置的追加區段內。然後，藉由資料讀取側的子處理器 43，將資料寫入至已讀取預約的記憶體位置內，並且 F/E 位元為“1”（可讀取/不可寫入）被設定時，事先當作讀取預約資訊而被寫入至



(15)

追加區段的子處理器 ID 及 LS 位址會被讀出。

當有必要藉由複數子處理器而將資料進行多階段處理時，藉由如此地控制各記憶體位置之資料的讀取/寫入，進行前階段處理的子處理器 43，就可將處理完的資料寫入至主記憶體 12 中的所定位置後，立即地，由進行後階段處理的其他子處理器 43 來將前處理後的資料予以讀出。

如圖 3 所示，各子處理器 43 內的近端儲存區 51，是由可指定複數位址之記憶體位置所構成。對各記憶體位置，係同樣地分配有追加區段。追加區段，係含有忙碌位元。

子處理器 43 是在將主記憶體 12 內的資料讀取至子處理器 43 的近端儲存區 51 的記憶體位置時，將對應之忙碌位元設定成“1”而進行預約。忙碌位元被設定為 1 的記憶體位置中，是無法存放其他的資料。在讀取至近端儲存區 51 的記憶體位置後，忙碌位元便被設定成“0”，就變成可以存放其他資料。

再者，如圖 2 所示，在和資訊處理控制器 11 連接的主記憶體 12 內，含有複數的沙箱（sandbox）。沙箱係用來將主記憶體 12 內的領域予以固定，各沙箱係被分配給各子處理器 43，可由所分配的子處理器 43 做排他性的使用。亦即，各子處理器 43 雖然能夠使用被分配給自己的沙箱，但無法對超過該領域的資料進行存取。

主記憶體 12，雖然是由複數的記憶體位置所構成，

(16)

但沙箱係為這些記憶體位置的集合。

再者，為了實現主記憶體 12 的排他性控制，使用了如圖 4 所示的金鑰管理表。金鑰管理表，係被記錄在金鑰管理表記錄部 45 內，並和 DMAC44 建立關連。金鑰管理表內的各項目（entry）中，含有子處理器 ID、子處理器金鑰及金鑰遮罩。

當子處理器 43 對主記憶體 12 進行存取時，子處理器 43 係向 DMAC44，輸出讀取或寫入的指令。該指令內，係含有用來特定子處理器 43 的子處理器 ID 及存取要求目的的地亦即主記憶體 12 的位址。

DMAC44 係在要執行從子處理器 43 所供給而來的指令時，先參照金鑰管理表，調查存取要求來源之子處理器 43 的子處理器金鑰。然後，DMAC44 會將調查到的存取要求來源之子處理器金鑰，和身為存取要求目的地的主記憶體 12 的被分配給記憶體位置之存取金鑰做一比較，只有 2 兩把金鑰一致的時候，才執行從子處理器 43 供給而來的指令。

圖 4 所示的金鑰管理表中所記錄的金鑰遮罩，係可藉由其任意的位元變成“1”，而使被賦予關連至該金鑰遮罩的子處理器金鑰之相對應位元可為“0”或“1”。

例如，假設子處理器金鑰為“1010”。通常，藉由該子處理器金鑰只可以存取到帶有“1010”之存取金鑰的沙箱。可是，當和該子處理器金鑰賦予關連到的金鑰遮罩被設定為“0001”時，只有金鑰遮罩被設定為“1”的位

(17)

數，子處理器金鑰和存取金鑰的一致判定會被遮蔽（針對金鑰遮罩的位元被設定成“1”的位數，不會被進行判定），藉由該子處理器金鑰“1010”，就可以存取到帶有“1010”或“1011”之任一種存取金鑰的沙箱。

如以上，就可實現主記憶體 12 的沙箱排他性。亦即，當需要藉由複數的子處理器 43 來將資料做多階段處理的時候，藉由以上這種構成，就可只讓進行前階段處理的子處理器 43 和進行後階段處理的子處理器 43，能夠存取主記憶體 12 的所定位址，而可保護資料。

例如，金鑰遮罩的值，可考慮變更如下。首先，在資訊處理裝置 1-1 剛剛啟動後的階段中，金鑰遮罩的值全部“0”。假設主處理器 42 內的程式被執行，並和載入至子處理器 43 內的程式進行協同動作。由子處理器 43-1 所輸出的處理結果資料，會被記憶在主記憶體 12，當想要將主記憶體 12 中所記憶的處理結果資料，輸入至子處理器 43-2 時，則記憶著由子處理器 43-1 所輸出之處理結果資料的主記憶體 12 的領域，必須要能夠從子處理器 43-1 及子處理器 43-2 來進行存取。此種情況下，主處理器 42，係會將金鑰遮罩的值做適切的變更，設置能夠讓來自複數之子處理器 43 存取的主記憶體領域，藉此使得子處理器 43 所致之多階段處理成為可能。

更具體而言，例如，根據從資訊處理裝置 1-2 至資訊處理裝置 1-n 所送來的資料，由子處理器 43-1 執行所定處理，並將處理過的資料，記憶至主記憶體 12 的第 1 領

(18)

域。然後，子處理器 43-2，係從主記憶體 12 的第 1 領域中，讀取出所記憶的資料，並根據所讀取之資料，執行所定之處理，將處理過的資料，記憶至不同於主記憶體 12 之第 1 領域的第 2 領域。

此處，若當子處理器 43-1 的子處理器金鑰係為“0100”、主記憶體 12 的第 1 領域的存取金鑰係為“0100”、子處理器 43-2 的子處理器金鑰係為“0101”、主記憶體 12 的第 2 領域的存取金鑰係為“0101”時，則子處理器 43-2 係無法存取主記憶體 12 的第 1 領域。於是，藉由將子處理器 43-2 的金鑰遮罩改成“0001”，子處理器 43-2 就可對主記憶體 12 的第 1 領域進行存取。

其次，參照圖 5 及圖 6，說明由資訊處理裝置 1-1 生成軟體胞，並根據所生成的軟體胞，而令資訊處理裝置 1-2 至資訊處理裝置 1-n 執行分散處理時的處理。

資訊處理裝置 1-1 的主處理器 42，係生成出含有執行處理所必需指令、程式及資料的軟體胞，透過網路 2 而送訊至資訊處理裝置 1-2 至資訊處理裝置 1-n。

圖 5 係圖示了軟體胞之構成的例子。

軟體胞，是由送訊源 ID、送訊目標 ID、回應目標 ID、胞介面、DMA 指令、程式及資料所構成。

送訊源 ID 中，含有身為軟體胞之送訊源之資訊處理裝置 1-1 的網路位址及資訊處理裝置 1-1 的資訊處理控制器 11 的資訊處理裝置 ID，再者，還含有資訊處理裝置 1-1 之資訊處理控制器 11 所具備之主處理器 42 及子處理器

(19)

43 的識別子（主處理器 ID 及子處理器 ID）。

送訊目標 ID 內，含有身為軟體胞送訊目標的資訊處理裝置 1-2 至資訊處理裝置 1-n 的網路位址、資訊處理裝置 1-2 至資訊處理裝置 1-n 的資訊處理控制器之資訊處理裝置 ID，及資訊處理裝置 1-2 至資訊處理裝置 1-n 的資訊處理控制器所具備的主處理器及子處理器的識別子。

又，回應目標 ID 內，含有身為軟體胞之執行結果之回應目標的資訊處理裝置 1-1 的網路位址、資訊處理裝置 1-1 的資訊處理控制器 11 之資訊處理裝置 ID，及資訊處理裝置 1-1 的資訊處理控制器 11 所具備的主處理器 42 及子處理器 43 的識別子。

胞介面，係為軟體胞利用上必須之資訊，是由全域 ID、必要之子處理器資訊、沙箱大小、以及前次軟體胞 ID 所構成。

全域 ID，係可以橫跨整個網路 2 而將該當軟體胞予以唯一識別的 ID，是根據送訊源 ID 及軟體胞生成或送訊的時日（日期及時刻）而產生的。

必須之子處理器資訊，係設定有該當軟體胞在執行上所必須之子處理器的數目。沙箱大小，係設定有該當軟體胞在執行上所必須之主記憶體及子處理器的近端儲存區的記憶體量。

前次軟體胞 ID，係串流資料等之要求序列性執行的 1 群軟體胞內的，上一個軟體胞的識別子。

軟體胞的執行會期（session），係由 DMA 指令、程

(20)

式及資料所構成。DMA 指令內，含有程式啟動上所必須之一連串的 DMA 指令；程式內則含有被子處理器所執行的子處理器程式。此處的資料，係為含有該子處理器程式的程式所處理的資料。

再者，DMA 指令內還含有載入指令、踢除指令、機能程式執行指令、狀態要求指令、以及狀態回送指令。

載入指令，係將主記憶體 12 內的資訊載入至子處理器之近端儲存區的指令，除了載入指令本身以外，還含有主記憶體位址、子處理器 ID 及 LS 位址。主記憶體位址，係表示資訊的載入源亦即主記憶體之所定領域的位址。子處理器 ID 及 LS 位址則是表示資訊的載入目標亦即子處理器的識別子及近端儲存區的位址。

踢動指令，係程式執行開始指令，除了踢動指令本身以外，還含有子處理器 ID 及程式計數器。子處理器 ID 係識別踢動對象之子處理器，程式計數器，係賦予程式執行用程式計數器所需之位址。

機能程式執行指令，係某一裝置（例如資訊處理裝置 1-1）對其他裝置（例如資訊處理裝置 1-2 至資訊處理裝置 1-n），要求機能程式之執行的指令。收到機能程式執行指令的資訊處理裝置（例如資訊處理裝置 1-2 至資訊處理裝置 1-n）的資訊處理控制器，係藉由機能程式 ID，來識別應要啟動之機能程式。

狀態要求指令，係將有關於送訊目標 ID 所示之資訊處理裝置 1-2 至資訊處理裝置 1-n 之現在的動作狀態（狀

(21)

況) 的裝置資訊，發送至回應目標 ID 所示之資訊處理裝置 1-1 的要求送訊之指令。

狀態回送指令，係收到上記狀態要求指令的資訊處理裝置 1-2 至資訊處理裝置 1-n，將裝置資訊，回應給該當狀態要求指令內所含之回應目標 ID 所示之資訊處理裝置 101 的指令。狀態回送指令，係將裝置資訊存放至執行會期的資料領域。

圖 6 中，圖示了 DMA 指令為狀態回送指令時的軟體胞的資料領域之構造。

資訊處理裝置 ID，係用來識別具備資訊處理控制器之資訊處理裝置 1 的識別子，是表示發送狀態回送指令的資訊處理裝置 1-2 至資訊處理裝置 1-n 的 ID。資訊處理裝置 ID，係例如，當每一台資訊處理裝置 1-2 至資訊處理裝置 1-n 的電源打開時，藉由資訊處理裝置 1-2 至資訊處理裝置 1-n 之資訊處理控制器 11 所含之主處理器 42，根據電源打開時的日時、資訊處理裝置 1-2 至資訊處理裝置 1-n 的網路位址及資訊處理裝置 1-2 至資訊處理裝置 1-n 的資訊處理控制器 11 中所含之子處理器 43 的數目等而產生的。

資訊處理裝置種別 ID，係含有代表資訊處理裝置 1-2 至資訊處理裝置 1-n 之特徵的值。所謂資訊處理裝置 1-2 至資訊處理裝置 1-n 的特徵，係代表資訊處理裝置 1-2 至資訊處理裝置 1-n 是何種裝置之資訊，例如，係代表資訊處理裝置 1-2 至資訊處理裝置 1-n，是硬碟錄影機、PDA

(22)

(Personal Digital Assistants) 、可攜式 CD (Compact Disc) 播放器等之資訊。又，資訊處理裝置種別 ID，亦可為代表映像聲音記錄、映像聲音再生等，表示資訊處理裝置 1-2 至資訊處理裝置 1-n 所具有之機能。代表資訊處理裝置 1-2 至資訊處理裝置 1-n 的特徵或機能的值是預先決定，收到狀態回送指令的資訊處理裝置 1-1，係藉由讀出資訊處理裝置種別 ID，就可掌握資訊處理裝置 1-2 至資訊處理裝置 1-n 的特徵或機能。

MS (Master/Slave，主/從) 狀態，係代表著資訊處理裝置 1-2 至資訊處理裝置 1-n 是在主機裝置或是從機裝置的哪種狀態下動作中，因此這裡被設為 0 時表示是以主機裝置的身分在動作著，被設為 1 時表示是以從機裝置的身分在動作著。

主處理器動作頻率，係代表資訊處理控制器 11 的主處理器 42 的動作頻率。主處理器使用率，係表示關於主處理器 42 上現在正在動作中的所有程式的，在主處理器 42 上的使用率。主處理器使用率，係代表著相對於對象主處理器 42 之全處理能力的使用中之處理能力的比率的值，例如是以用來評估處理器處理能力之單位 MIPS 的單位來算出，或著根據每單位時間內之處理器使用時間而算出。

子處理器數，係代表資訊處理控制器 11 所具備的子處理器 43 的數量。子處理器 ID，係用來識別資訊處理控制器 11 之各子處理器 43 所需之識別子。

(23)

子處理器狀態，係代表各子處理器 43 的狀態，有 “unused”、“reserved”、“busy” 等狀態。“unused” 係表示該當子處理器現在並未使用，也沒有使用的預約。“reserved” 則表示現在雖然未使用，但已經有預約之狀態。“busy” 則是表示現在正在使用中。

子處理器使用率，係有關於該當子處理器上現在正在執行的，或該當子處理器上已被預約執行的程式，在子處理器 43 上的使用率。亦即，子處理器使用率，係當子處理器狀態為 “busy” 時，則表示現在的使用率；當子處理器狀態為 “reserved” 時，則表示之後所使用的預定之推定使用率。

子處理器 ID、子處理器狀態及子處理器使用率，係針對 1 個子處理器 43 設定一組，而設定有對應於 1 個資訊處理控制器 11 之子處理器 43 的組數。

主記憶體總容量及主記憶體使用量，係分別代表著資訊處理控制器 11 上所連接之主記憶體 12 的總容量及現在使用中的容量。

記錄部數，係代表連接在資訊處理控制器 11 上的記錄部 13 的數目。記錄部 ID，係用來將連接在資訊處理控制器 11 上的記錄部 13 予以唯一識別的資訊。記錄部種別 ID，係代表記錄部 13 的種類（例如：硬碟、CD±RW、DVD±RW、記憶體碟、SRAM、ROM 等）之資訊。

記錄部總容量及記錄部使用量，分別代表著記錄部 ID 所識別的記錄部 13 的總容量及現在使用中的容量。

(24)

記錄部 ID、記錄部種別 ID、記錄部總容量及記錄部使用量，係針對 1 個記錄部 13 設定一組，並只設定有對應於資訊處理控制器 11 上所連接之記錄部 13 之數量的組數。亦即，當 1 個資訊處理控制器 11 是連接著複數之記錄部 13 的時候，各個記錄部 13 是被分配到不同的記錄部 ID，且記錄部種別 ID、記錄部總容量及記錄部使用量亦被個別地予以管理。

如此一來，資訊處理裝置 1-1，係當要令資訊處理裝置 1-2 至資訊處理裝置 1-n 執行分散處理時，便生成軟體胞，並將所生成的軟體胞，透過網路 2 而發送至資訊處理裝置 1-2 至資訊處理裝置 1-n。

此外，以下，從資訊處理裝置 1-1 送往資訊處理裝置 1-2 至資訊處理裝置 1-n 的各種資料，因為是先存放至軟體胞內才被發送，而為重複，因此省略其說明。

資訊處理裝置 1-2 至資訊處理裝置 1-n 之中的任一資訊處理控制器 11 所含之主處理器 42，會生成以上這種構成的軟體胞，透過網路 2 而向其他資訊處理裝置 1 的資訊處理控制器 11，發送所生成的軟體胞。送訊源的資訊處理裝置 1、送訊目標之資訊處理裝置 1、回應目標之資訊處理裝置 1、以及設於各裝置內的資訊處理控制器 11，係分別藉由上記的送訊源 ID、送訊目標 ID 及回應目標 ID 而予以識別。

收到軟體胞的資訊處理裝置 1 的資訊處理控制器 11 所含之主處理器 42，係將該軟體胞存放在主記憶體 12 內

(25)

。然後，送訊目標的主處理器 42，會將軟體胞讀出，處理其所含有之 DMA 指令。

具體而言，送訊目標之主處理器 42，首先會執行載入指令。藉此，根據載入指令所指示的主記憶體 12 的位址，載入指令所含之子處理器 ID 及 LS 位址所特定之子處理器內之近端儲存區 51 的所定領域內，便會被載入資訊。此處所載入的資訊，係為所收到之軟體胞內所含之子處理器程式或資料，或者其他被指示的資料。

其次，主處理器 42 會將踢動指令，同樣地連同踢動指令中所含有之程式計數器而輸出至其所含有之子處理器 ID 所指示的子處理器 43。

被指示的子處理器，係依照該踢動指令及程式計數器，執行子處理器程式。然後，在將執行結果儲存至主記憶體 12 後，向主處理器 42 通知執行完畢的事件。

此外，送訊目標之資訊處理裝置 1 之資訊處理控制器 11 中，執行軟體胞的處理器並非侷限於子處理器 43，而是亦可指定由主處理器 42 來執行軟體胞所含之機能程式等之主記憶體用程式。

此時，送訊源之資訊處理裝置 1，係向送訊目標之資訊處理裝置 1，不是發送子處理器程式，而是改以發出含有主記憶體用程式及要被主記憶體用程式所處理的資料且 DMA 指令為載入指令的軟體胞，令主記憶體 12 記憶著主記憶體用程式及要被其所處理之資料。其次，送訊源資訊處理裝置 1，係向送訊目標資訊處理裝置 1，發出出含

(26)

有關於送訊目標之資訊處理裝置內之資訊處理控制器 11 的主處理器 ID、主記憶體位址、用來識別主記憶體用程式所需之後述之機能程式 ID 等之識別子、及程式計數器，且 DMA 指令為踢動指令或機能程式執行指令的軟體胞，令主處理器 42 執行該當主記憶體用程式。

如以上，適用了本發明的網路系統中，軟體胞之送訊源的資訊處理裝置 1，係除了將子處理器程式或主記憶體用程式藉由軟體胞而發送至送訊目標資訊處理裝置 1，還令該當子處理器程式載入至送訊目標資訊處理裝置 1 之資訊處理控制器 11 所設之子處理器 43，而可使該當子處理器程式或該當主記憶體用程式，在送訊目標資訊處理裝置 1 上執行。

軟體胞之送訊目標資訊處理裝置 1 之資訊處理控制器 11 中，當收到的軟體胞中所含之程式為子處理器程式時，令該當子處理器程式載入至被指定的子處理器 43 內。然後，在所指定的子處理器 43 上，軟體胞所含之子處理器程式或主記憶體用程式便會執行。

因此，使用者即使不操作軟體胞之送訊目標的資訊處理裝置 1，也能自動地，令子處理器程式或主記憶體用程式，從軟體胞之送訊源的資訊處理裝置 1，分配至送訊目標的資訊處理裝置 1，並藉由軟體胞之送訊目標的資訊處理裝置 1 的資訊處理控制器 11，促使它們執行。又，在各子處理器 43 間，是以 DMA 方式進行資料傳送，且藉由使用上述的沙箱，即使需要在 1 個資訊處理控制器 11 內

(27)

，將資料予以多階段處理的時候，也能高速且高安全性地執行處理。

圖 7 中係圖示了資訊處理控制器 11 的主記憶體 12 所記憶之軟體之構成的軟體堆疊。這些軟體（程式），係在資訊處理裝置 1 打開電源前，就被記錄在資訊處理控制器 11 所連接之記錄部 13 內。

驅動程式軟體 81-1 至 81-p，係用來促使硬體動作所需之專用軟體。驅動程式軟體 81-1 至 81-p 中，例如有，促使連接在操作輸入部 15 上的滑鼠動作所需的滑鼠驅動程式、促使用來讓顯示部 17 顯示出影像所需之顯示卡動作所需的顯示卡驅動程式、促使透過通訊部 16 而連接之未圖示的印表機動作所需的印表機驅動程式等等。

核心（OS）82，係控制著資訊處理控制器 11 之基本動作，管理各種資源，例如，將應用程式 87 所發生的命令，傳達至驅動程式軟體 81-1 至 81-p。又，核心 82 係也會進行應用程式 87 之執行上下文（某應用程式 87 所正在利用之暫存器集或主記憶體影像、檔案處理等）或 GUI 之零件等軟體資源的管理。

裝置驅動程式 83，係用來執行往資訊處理控制器 11 或資訊處理裝置 1 的資料之輸出入（收送訊）機能所需之驅動程式，例如，播送接收、螢幕輸出、位元串流輸出入、網路輸出入用程式等，隨著資訊處理裝置 1 所具有的機能，而讓每一資訊處理控制器 11 具備之。

函式庫 84，亦稱 DLL（Dynamic Link Library），是

(28)

將數個函數像是書庫般予以集結，是將軟體執行上所必須之機能獨立成檔案的方式而予以共通化而成者，通常係將檔案本身稱為 DLL。次常式係屬於程式內部的，但函式庫 84 係和程式不為一體而是執行時會被載入，兩者有所區別。由於以函式庫 84 的方式所提供的機能係不須重新開發即可達成，因此可提高應用程式 87 的開發效率，更可將相同程式零件讓複數應用程式 87 來共享，因此可節省磁碟或記憶體之容量。

電源管理器 85，係判斷裝置驅動程式 83 所管理之每一主處理器 42 及子處理器 43 的動作狀況，和被要求執行之應用程式 87 的執行所需之條件，來管理每一處理器的消費電力，當決定分散處理的分配，或分散處理之分配是不可能的時候，便將所要求之處理之執行的取消，通知給系統服務/開發框架 86。亦即，電源管理器 85 係用來控制資訊處理控制器 11 的主處理器 42 及子處理器 43 所執行之分散處理，會管理每一主處理器 42 及子處理器 43 的動作狀況，決定受應用程式 87 所要求之邏輯執行緒執行所需之分散處理的分配，而將程式及資料，分別載入至對應之主處理器 42 及子處理器 43，而令其執行。

具體而言，電源管理器 85，係控制主處理器 42 及子處理器 43 的動作時脈數成最佳的值，以使得受應用程式 87 所要求之處理為可以執行，且資訊處理控制器 11 的消費電力可以降低。此處，將現在設定之時脈數相對於每一處理器的最大動作時脈數的比，定義為處理器動作率。處

(29)

理器動作率，係可在每一主處理器 42 及子處理器 43 中呈獨立控制，也可是為統一而加以控制。亦即，主處理器 42 及子處理器 43，雖然係潛在地能夠以最大動作時脈數來動作，但實際上是以根據電源管理器 85 所設定的處理器動作率的上限動作時脈數來進行動作。

亦即，處理器動作率 $SCPU_RATE[k]$ (k 係為了能夠區別受到電源管理器 85 控制之處理器而分配的編號)，係以上限動作時脈數 / 最大動作時脈數 $SCPU_CK[k]$ 來表示；當處理器動作率，係在每一主處理器 42 及子處理器 43 中，是為同一而被控制的時候，處理器動作率 $SCPU_RATE[k] = SYSTEM_RATE$ 。

又，處理器使用率 $SCPU_STAT[k]$ ，在各處理器中，是以在規定時間內應用程式之執行所使用的週期數（其中 1 週期 (1 cycle) 係為 CPU 最大動作時脈數的倒數）/ 規定時間 / 最大動作時脈數 $SCPU_CK[k]$ 來表示。

例如，假設主處理器 42 及子處理器 43，是可以潛在的最大動作時脈數 1.2MHz 來動作。此時，作為處理器動作率，例如，若可設定 1.0、0.5、0.25，則上限動作時脈數便為 1.2MHz、600Hz、300Hz。具體而言，當處理器動作率被設定成 0.5 時，上限動作時脈數，係為 $1.2\text{MHz} \times 0.5 = 600\text{Hz}$ 。然後，在規定時間 1sec 內，應用程式的執行是使用 300 週期的時候，則處理器使用率 $SCPU_STAT[k]$ ，係為 $300 \text{ 週期} / 1\text{sec} / 1.2\text{MHz} = 0.25$ 。

亦即，處理器使用率和處理器動作率，恆成立下式（

(30)

1) 的關係。

$$0 \leq \text{處理器使用率 (SCPU_STAT[k])} \leq \text{處理器動作率 (SCPU_RATE[k])} \leq 1.0 \quad \cdot \cdot \cdot (1)$$

又，電源管理器 85，係對未被分派處理的子處理器 43 設定為待命 (Suspend) 模式，使對應的子處理器不會被分配到處理，而可降低資訊處理控制器 11 的消費電力。

系統服務/開發框架 86，係將藉由應用程式 87 所執行的處理，分派給資訊處理裝置 1 的複數子處理器 43 的每一個而執行分散處理，或者，提供藉由複數的資訊處理裝置 1 而執行之分散處理的系統服務的軟體模組群。

應用程式 87，係為使用資訊處理控制器 11，以實現例如播送波的收訊控制、內容的記錄或再生、3 次元圖形演算處理、電子郵件的收授、網頁的瀏覽等等目的所需之軟體程式。例如，當 3 次元圖形所需的應用程式 87 被執行時，便將 1 畫面份之處理按照像素加以分割，而可基於同一程式、同一模組資料，以複數的處理器進行平行演算。亦即，當使用資訊處理控制器 11，而 3 次元圖形所需的應用程式 87 被執行時，藉由系統服務/開發框架 86，被按照像素而分割後的處理，會分別被分派給複數的子處理器 43，而可使其進行平行演算。

圖 8 係用來說明由資訊處理控制器 11 去管理每一處理器的消費電力，並控制分散處理之分配之機能的機能方塊圖。

(31)

圖 8 終，應用程式執行控制部 101、邏輯執行緒執行控制部 102、動作率設定部 103、動作資訊取得 104 及動作資訊記憶體 105，係藉由資訊處理裝置 1 的主處理器 42 所實現的機能。

此外，圖 8 中，當實現這些機能的硬體並非上述的時候，例如，藉由主處理器 42 所實現之機能當中的一部份，是藉由不同硬體來實現的時候，本發明當然仍是可以適用。又，這些構成，係可全部構成在單 1 晶片內，或可構成爲個別之晶片。

應用程式執行控制部 101，係執行用來執行所定應用程式所需之演算處理。又，應用程式執行控制部 101，係當要令子處理器 43 當中的任一者執行分散處理時，便向邏輯執行緒執行控制部 102，供給圖 9 所示的 REQUEST_information（應用程式要求條件）以當作是邏輯執行緒的設定資訊，而要求邏輯執行緒的生成。所謂邏輯執行緒，係定義爲，將因應複數子處理器 43 的每一個之需要而被分配的複數程式（這些程式係可爲對應於同一應用程式者，亦可爲對應於不同之應用程式），總結成 1 個機能而加以提供之處理單位。換言之，因應複數子處理器 43 的每一個的需要而被分配的複數個程式，係被記憶在主記憶體 12 的所定記憶領域之程式，藉由來自應用程式執行控制部 101 之執行受到控制之應用程式的要求，而從主記憶體 12 載入至所對應之子處理器 43 的近端儲存區 51 而加以執行。

(32)

圖 9 所示的從應用程式執行控制部 101 供給至邏輯執行緒執行控制部 102 的 REQUEST_information (A 處理器要求條件) 中，開頭是存放有：用來識別應用程式的 16 位元組之 ID (應用程式 ID)、表示有無需要在規定時間內處理完畢的 1 位元組之 REQ_REAL 資訊、表示在決定子處理器之分配時希望能較為優先的子處理器之子處理器 ID 的 1 位元組之 PRIORITY_ID、表示所要求之處理所需要的子處理器數的 REQ_NUM 資訊，其後則存放有：表示被分配處理的子處理器所被要求的因為要求之處理而被佔用的記憶體容量的 REQ_MEM 資訊、表示被分配處理的子處理器所被要求的因為要求之處理而被佔用的規定時間內處理週期數的 REQ_PC 資訊，是數量相同於主處理器 42 所被分配的處理、和 REQ_NUM 資訊所代表之處理所需之子處理器數而被存放。具體而言，作為 REQ_MEM0 及 REQ_PC0，是存放有主處理器 42 所被要求的記憶體容量和週期 (cycle) 數；接著，作為 REQ_MEM1 及 REQ_PC1、REQ_MEM2 及 REQ_PC2...，則分別存放著各子處理器所被要求的記憶體容量和週期 (cycle) 數。對應於子處理器的 REQ_MEM 及 REQ_PC 的各資訊的存放順序，係需要的週期數由大而小依序排列，而當週期數相同時則記憶體量越大者越優先排列，較為理想。

此處，REQ_REAL 時係代表需要即時處理；所謂處理週期 (cycle) 數，係對應於應用程式執行時所需要的處理量之值。又，應用程式的執行要求及

(33)

REQUEST_information 的供給，有時也會透過網路 2，而從異於自身的其他資訊處理裝置 1 的資訊處理控制器 11 來供給之。

回到圖 8 的說明。

邏輯執行緒執行控制部 102，係在內部具有暫存器及記憶體，會進行應用程式執行控制部 101 所執行之應用程式所對應之邏輯執行緒的生成及刪除。具體而言，邏輯執行緒執行控制部 102，係根據從應用程式執行控制部 101 供給過來的，圖 9 所示的 REQUEST_information，和動作資訊記憶體 105 中所記憶之子處理器 43 的各自動作狀態，而對子處理器 43，適切地分派分散處理，藉此除了可抑制消費電力的增加，還可進行所生成之邏輯執行緒的狀態監視與動作控制。邏輯執行緒執行控制部 102，係會產生出記載有應用程式執行控制部 101 之全體之管理資訊的表格，和按照所執行的邏輯執行緒別，而記載有邏輯執行緒之執行所需之資訊的表格，因應需要，而參照表格的資訊，進行更新，藉此而控制邏輯執行緒的執行。

作為邏輯執行緒執行控制部 102 中所管理的資訊，除了上述的 REQUEST_information 以外，還有：表示資訊處理控制器 11 所擁有之子處理器 43 數的 NUM_SCPU、例如 1sec 等之表示進行分散處理之分派時的作為基準時間所用的規定時間的 TIME_DEF、表示相對於記憶體使用率的可用份量 (margine) 的 MEM_MARGIN、表示以第 k 處理器切換執行緒 (應用程式) 時的贅算 (overhead) 量

(34)

的 $\text{THREAD_OVERHEAD}[k]$ 、表示第 k 處理器的處理器動作率之設定值（ $0 \sim 1$ 的值）的 $\text{SCPU_RATE}[k]$ 、表示相對於第 k 處理器之處理器使用率的處理器動作率的可用份量的 RATE_MARGIN ，以及在未被要求即時處理的應用程式中，當分配處理不成功時，表示打斷分配處理回圈的次數之 N_GIVEUP 。

此處，所謂的第 k 處理器，係第 0 代表主處理器 42，第 1 以後則代表子處理器 43 之中的任一者。又， MEM_MARGIN 中，例如，使用 10Kbytes 的值；在 RATE_MARGIN 中，例如使用 0.1 等之值；它們都是在進行分散處理之分配之判別時會被使用。又，所謂「切換執行緒時的贅算量」，具體而言，係將現在處理器中所正在執行的處理的上下文（context），從近端儲存區 51 中撤除，或回復成原本狀態之前所需的週期數。

此外，當在所有的處理器上將處理器動作率以共通的方式加以控制時，處理器動作率的設定值，係會配合 SYSTEM_RATE 。又，會成立下式（2）。

處理器上限動作時脈數 = 處理器最大動作時脈數 $\times$ 處理器動作率 . . . (2)

動作率設定部 103，係除了根據邏輯執行緒執行控制部 102 的控制，而設定主處理器 42 及子處理器 43 的處理器動作率 $\text{SCPU_RATE}[k]$ 或 SYSTEM_RATE ，同時還將已設定之處理器動作率 $\text{SCPU_RATE}[k]$ 或 SYSTEM_RATE 的值，供給至動作資訊記憶體 105 而登錄之。

(35)

動作資訊取得 104，係取得子處理器 43 的動作資訊，供給至動作資訊記憶體 105。

動作資訊記憶體 105，係對應於近端儲存區 51-1 之記憶領域之中的至少一部份，將動作資訊取得 104 之處理所取得到的子處理器 43 的動作資訊及主處理器 42 的動作資訊，和被動作率設定部 103 所設定之主處理器 42 及子處理器 43 的處理器動作率 $SCPU_RATE[k]$ 或 $SYSTEM_RATE$ ，加以記憶。

動作資訊記憶體 105 中所記錄之資訊中，例如有：表示主處理器 42 及子處理器 43 所能利用之記憶體容量的 $MEM_CPU[k]$ (kbytes)，資訊處理控制器 11 內所設之第 k 處理器 42 或子處理器 43 之最大動作時脈數的 $SCPU_CK[k]$ 、表示第 k 處理器 42 或子處理器 43 之處理器使用率的 $SCPU_STAT[k]$ 、表示第 k 處理器 42 或子處理器 43 上現在正在動作之應用程式數的 $SCPU_APPNUM[k]$ 、表示能否對第 k 處理器 42 或子處理器 43 分配程式的旗標 $SCPU_SUSPEND[k]$ 等。此處，當 $SCPU_SUSPEND[k] = 0$ 時，表示第 k 處理器 42 或子處理器 43 可以被分配程式，當 $SCPU_SUSPEND[k] = 1$ 時，表示第 k 處理器 42 或子處理器 43 禁止被分配程式。

此處，現在的第 k 處理器 42 或子處理器 43 的動作時脈數，係以 $SCPU_CK[k] \times SCPU_RATE[k]$ 來表示。又，處理器動作率與處理器使用率的關係，係滿足下式 (3)。亦即，當在所有的處理器上是將處理器動作率共通控制

(36)

的時候，處理器使用率與處理器使用率的關係，係滿足下式（4）。

$$0 \leq \text{SCPU_STAT}[k] \leq \text{SCPU_RATE}[k] \leq 1 \cdot \cdot \cdot (3)$$

$$0 \leq \text{SCPU_STAT}[k] \leq \text{SYSTEM_RATE} \leq 1 \cdot \cdot \cdot (4)$$

又，表示第 k 處理器 42 或子處理器 43 之處理器使用率的 $\text{SCPU_STAT}[k]$ ，係隨著每一新的應用程式所致之分散處理而增加。例如，第 1 個應用程式的時候， $\text{SCPU_STAT}[k]$ 的增加量，係以下式（5）來表示；第 2 個以降之應用程式時， $\text{SCPU_STAT}[k]$ 的增加量，係以下式（6）來表示。

$$\text{SCPU_STAT}[k] \text{ 的增加量} = (\text{REQ_PC}[k] \div \text{TIME_DEF}) \div \text{SCPU_CK}[k] \cdot \cdot \cdot (5)$$

$$\begin{aligned} &\text{SCPU_STAT}[k] \text{ 的增加量} \\ &= ((\text{REQ_PC}[k] + \text{THREAD_OVERHEAD}[k]) \div \text{TIME_DEF}) \div \text{SCPU_CK}[k] \cdot \cdot \cdot (6) \end{aligned}$$

其次，說明資訊處理控制器 11 內所含之主處理器 42 及子處理器 43 的消費電力。

資訊處理控制器 11 所含之主處理器 42 及子處理器 43 的每一個的消費電力，是隨著與現在處理器動作率成正比增加的消費電力量，和與現在處理器使用率成正比增加的消費電力量而決定。與現在處理器動作率成正比增加的消費電力量 PT-A ，係為即使主處理器 42 或子處理器 43 未被分散處理所使用時仍會消耗的電力量，係以下式（7）表示。此處， PA 係為動作時脈數比例係數，是隨著

(37)

硬體構成而預先決定的值。然後，與現在處理器使用率成正比增加的消費電力量 $PT-B$ ，則以下式（8）來表示。此處， K 係為使用時脈數比比例係數且是隨著硬體構成而預先決定的值。

$$PT-A = PA (W/Cycle) \times SCPU_RATE[k] \times SCPU_CK[k] \quad \dots (7)$$

$$PT-B = K (W/Cycle) \times SCPU_STAT[k] \times SCPU_CK[k] \quad \dots (8)$$

因此，資訊處理控制器 11 內所含之主處理器 42 及子處理器 43 的合計的消費電力 ΣPT ，係以下式（9）來表示。又，當處理器動作率 $SCPU_RATE[k]$ ，並非各處理器獨立設定，而是全部視為共通值（ $SYSTEM_RATE$ ）而加以操控的時候，資訊處理控制器 11 內所含之主處理器 42 及子處理器 43 的合計的消費電力 ΣPT ，係以下式（10）來表示。

$$\Sigma PT = \Sigma (PA \times SCPU_RATE[k] \times SCPU_CK[k] + K \times SCPU_STAT[k] \times SCPU_CK[k]) \quad \dots (9)$$

$$\Sigma PT = \Sigma (PA \times SYSTEM_RATE \times SCPU_CK[k] + K \times SCPU_STAT[k] \times SCPU_CK[k]) \quad \dots (10)$$

圖 10 係為資訊處理控制器 11 內所含之主處理器 42 及子處理器 43 的每一個的消費電力 PT ，與動作時脈數的關係圖。

如圖 10 所示，雖然處理器的使用率越大，消費電力 PT 越大，但即使處理器的使用率為 0，只要仍對處理器供

(38)

給時脈，就仍會發生爲了處理器及匯流排之控制而消耗的電力量等、並非依存於應用程式處理的固定消費電力。供給至主處理器 42 及子處理器 43 的時脈，係以 $SYSTEM_RATE \times SCPU_CK$ 來表示。亦即，即使是同一處理器使用率（ $SCPU_STAT$ ），處理器動作率（ $SYSTEM_RATE$ 或 $SCPU_RATE[k]$ ）越大，則固定份量的消費電力越會增加。

其次，參照圖 11 及圖 12 的流程圖，說明適用了本發明之資訊處理裝置 1 上所執行的分散處理分配決定處理 1。

步驟 S1 中，邏輯執行緒執行控制部 102 係接受來自應用程式執行控制部 101 的 $REQUEST_information$ 之供給，判斷是否有被指示新的分散處理之分配。步驟 S1 中，當判斷爲沒有被指示新的分散處理之分配時，則一直重複步驟 S1 的處理直到判斷爲有新的分散處理之分配指示爲止。

步驟 S1 中，當判斷爲有新分散處理之分配的指示時，則在步驟 S2 中，邏輯執行緒執行控制部 102 係從所供給而來的 $REQUEST_information$ 中所含的資訊中，取得有指示要分配之分散處理的執行所需要的子處理器數 j 。

步驟 S3 中，邏輯執行緒執行控制部 102，作爲對應於存放往各處理器之處理分配計算過程中的暫定值的內部暫存器的值，而定義了處理器使用率 $TEMP_STAT[k]$ 及處理器動作率 $TEMP_RATE[k]$ 。

(39)

步驟 S4 中，邏輯執行緒執行控制部 102，係將動作資訊記憶體 105 所記憶之分散處理之分配前，亦即，現在在各處理器的處理器使用率 $SCPU_STAT[k]$ 及處理器動作率 $SCPU_RATE[k]$ 的值，加以取得，而代入至處理器使用率 $TEMP_STAT[k]$ 及處理器動作率 $TEMP_RATE[k]$ 。

此處，針對在複數設置子處理器 43 的資訊處理控制器 11 中，已經在執行某些應用程式的狀態下，從透過網路 2 而連接之其他資訊處理裝置 1，又送來新的其他應用程式之執行要求時的情形加以說明。又，邏輯執行緒執行控制部 102 或動作資訊記憶體 105 現在正在管理的設定值，係設定為 $REQ_REAL = 1$ （要求即時處理）、 $NUM_SCPU = 3$ 、 $TIME_DEF = 1$ （1 秒）、 $RATE_MARGIN = 0.1$ 、 $MEM_MARGIN = 10$ 。

使用圖 13，說明已經執行某些應用程式之狀態的主處理器 42 及子處理器 43-1 至子處理器 43-3 的處理器動作率 $SYSTEM_RATE$ （在此處，是將所有處理器上的處理器動作率視為共通而進行說明），和處理器使用率 $SCPU_STAT[k]$ 的關係。此處，為了簡化說明，係假設 $THREAD_OVERHEAD$ 為 0 而進行說明。

此處，主處理器 42 及子處理器子處理器 43-1 至子處理器 43-3 的處理器動作率 $SYSTEM_RATE$ 係為 0.5。然後，因為 $k = 0$ 所代表之主處理器 42 的處理器使用率 $SCPU_STAT[0]$ 係為 0.3，所以處理器使用率 $SCPU_STAT[0]/$ 處理器動作率 $SYSTEM_RATE$ 係為 $0.3/0.5$

(40)

，主處理器 42 所能利用的記憶體容量 MEM_CPU[0]係為 5000 (kbytes)，現在使用中的記憶體容量係為 1000 (kbytes)，代表最大動作時脈數的 SCPU_CK[0]係為 1000。

因為 $k = 1$ 所代表之子處理器 43-1 的處理器使用率 SCPU_STAT[1] 係為 0.4，所以處理器使用率 SCPU_STAT[1]/處理器動作率 SYSTEM_RATE 係為 $0.4/0.5$ ，子處理器 43-1 所能利用的記憶體容量 MEM_CPU[1]係為 100 (kbytes)，現在使用中的記憶體容量係為 20 (kbytes)，代表最大動作時脈數的 SCPU_CK[1]係為 1000。

又， $k = 2$ 所代表之子處理器 43-2 的處理器使用率 SCPU_STAT[2] 係為 0.2，所以處理器使用率 SCPU_STAT[2]/處理器動作率 SYSTEM_RATE 係為 $0.2/0.5$ ，子處理器 43-2 所能利用的記憶體容量 MEM_CPU[2]係為 100 (kbytes)，現在使用中的記憶體容量係為 10 (kbytes)，代表最大動作時脈數的 SCPU_CK[2]係為 1000。

然後， $k = 3$ 所代表之子處理器 43-3 的處理器使用率 SCPU_STAT[3] 係為 0.1，所以處理器使用率 SCPU_STAT[3]/處理器動作率 SYSTEM_RATE 係為 $0.1/0.5$ ，子處理器 43-3 所能利用的記憶體容量 MEM_CPU[3]係為 100 (kbytes)，現在使用中的記憶體容量係為 5 (kbytes)，代表最大動作時脈數的 SCPU_CK[3]係為 1000。

然後，假設 REQUEST_information 中所記載的被新要求執行之應用程式的動作條件係必須之子處理器數

(41)

REQ_NUM 為 2，主處理器 42 所被要求的處理週期數 REQ_PC[0]係為 100，所要求之記憶體容量 REQ_MEM[0]為 100 (kbytes)，第 1 子處理器 43 所被要求的處理週期數 REQ_PC[1]係為 400，所要求之記憶體容量 REQ_MEM[1]為 50 (kbytes)，第 2 子處理器 43 所被要求的處理週期數 REQ_PC[2]係為 200，所要求之記憶體容量 REQ_MEM[2]為 30 (kbytes)。

根據以上的條件，在以下的處理中，會進行滿足條件的分散處理之分配。

步驟 S5 中，邏輯執行緒執行控制部 102，係對代表要求之處理所需要的子處理器數 REQ_NUM，將一表示要進行第幾號之處理器的分配決定處理的暫存器的值 t ，設為 $t=0$ 。此處， $t=0$ 的處理器係代表主處理器 42， $t \geq 1$ 的處理器係代表子處理器 43 當中的任一者。

步驟 S6 中，邏輯執行緒執行控制部 102，係將表示資訊處理控制器 11 中所設的處理器當中，能否對第幾號處理器進行分配處理的暫存器值 s ，設為 $s=0$ 。

步驟 S7 中，邏輯執行緒執行控制部 102，係判斷 REQ_NUM(t) 之處理，是能否藉由第 s 號處理器來執行。

具體而言，邏輯執行緒執行控制部 102，係除了將 REQ_NUM(t) 之處理所致之處理器使用率 SCPU_STAT[s]的增加量予以加算，還會以處理器使用率 SCPU_STAT[s] + RATE_MARGIN 不會超過 1，以及針對被

(42)

REQ_NUM (t) 之處理所要求之記憶體容量要成立 (REQ_MEM[s] + MEM_MARGIN) < (MEM[s]_CPU-現在的記憶體使用量) 為條件，來判斷 REQ_NUM (t) 之處理，是能否藉由第 s 號處理器來執行。

步驟 S7 中，當判斷為 REQ_NUM (t) 的處理係不可能以第 s 號處理器來執行時，則在步驟 S8 中，邏輯執行緒執行控制部 102 便將暫存器的值 s，設為 $s = s + 1$ 。

步驟 S9 中，邏輯執行緒執行控制部 102 係會判斷是否 $s > (\text{資訊處理控制器 11 內所設置之子處理器 43 的數目})$ 。步驟 S9 中，若判斷為不是 $s > (\text{資訊處理控制器 11 內所設置之子處理器 43 的數目})$ 時，則處理便返回步驟 S7，重複其以降的處理。步驟 S9 中，若判斷是 $s > (\text{資訊處理控制器 11 內所設置之子處理器 43 的數目})$ 時，則處理便進入後述之步驟 S19。

步驟 S7 中，若判斷為 REQ_NUM (t) 的處理係可能以第 s 號處理器來執行時，則在步驟 S10 中，邏輯執行緒執行控制部 102，便在表示滿足必要條件之處理器之識別編號的內部暫存器所對應的 SET_CPUID[t] 中，記載入已分配之第 s 號處理器的處理器 ID (主處理器 ID 或子處理器 ID)。

步驟 S11 中，邏輯執行緒執行控制部 102，係會更新存放分配計算過程中之暫定值的內部暫存器值所對應之處理器使用率 TEMP_STAT[s]，以使得已分配之第 s 號處理器的處理量的處理器使用率會被加算。

(43)

步驟 S12 中，邏輯執行緒執行控制部 102 係將暫存器 t 的值，設為 $t = t + 1$ 。

步驟 S13 中，邏輯執行緒執行控制部 102 係參照暫存器 t 的值，判斷是否 $t > j$ (j 係分散處理所必須之子處理器數)。步驟 S13 中，若判斷為非 $t > j$ 時，則由於所被要求之處理分配尚未結束，因此處理會返回步驟 S6，重複其以降之處理。

步驟 S13 中，若判斷是 $t > j$ ，則由於所被要求之處理分配已經結束，因此在步驟 S14 中，邏輯執行緒執行控制部 102 係將代表分配已經成功的旗標加以豎起，亦即設 $SET_NG = 0$ 。

步驟 S15 中，邏輯執行緒執行控制部 102 係將所有的處理器的處理器動作率 $TEMP_RATE[k]$ ，及最新的處理器使用率 $TEMP_STAT[k]$ ，加以比較，藉此而判斷是否有需要隨著新的分散處理之分配，而變更處理器動作率的設定。

步驟 S15 中，當判斷為有需要變更處理器動作率的設定時，亦即，無論在哪個處理器上，皆為處理器動作率 $TEMP_RATE[k] < 處理器使用率 TEMP_STAT[k]$ 的時候，則在步驟 S16 中，邏輯執行緒執行控制部 102 係會算出新的處理器動作率，並將算出結果，供給至動作率設定部 103。動作率設定部 103，係對主處理器 42 或子處理器 43 當中的對應者，進行處理器動作率的設定變更。

步驟 S17 中，邏輯執行緒執行控制部 102，係將動作

(44)

資訊記憶體 105 所記憶之分散處理之分配前，亦即，現在
的各處理器的處理器使用率 $SCPU_STAT[k]$ 及處理器動作
率 $SCPU_RATE[k]$ 的值，分別更新成，藉由分配計算所得
到的處理器使用率 $TEMP_STAT[k]$ 及 $TEMP_RATE[k]$ 。

使用圖 14，說明新分散處理的分配結果，與處理器
動作率之設定。

在用圖 13 說明過的狀態下，如上述，當：必須之子
處理器數 REQ_NUM 係為 2，主處理器 42 所被要求的處
理週期數 $REQ_PC[0]$ 係為 100，所要求之記憶體容量
 $REQ_MEM[0]$ 係為 100 (kbytes)，第 1 子處理器 43 所被
要求的處理週期數 $REQ_PC[1]$ 係為 400，所要求之記憶體
容量 $REQ_MEM[1]$ 係為 50 (kbytes)，第 2 子處理器 43
所被要求的處理週期數 $REQ_PC[2]$ 係為 200，所要求之記
憶體容量 $REQ_MEM[2]$ 係為 30 (kbytes) 的分散處理之分
配被要求時，如圖 14 所示， $k=0$ 所代表的主處理器 42
的處理器使用率 $SCPU_STAT[0]$ 係變為 0.4， $k=1$ 所代表
之子處理器 43-1 的處理器使用率 $SCPU_STAT[1]$ 係變為
0.8， $k=2$ 所代表之子處理器 43-2 的處理器使用率
 $SCPU_STAT[2]$ 係變為 0.4， $k=3$ 所代表之子處理器 43-3
的處理器使用率 $SCPU_STAT[3]$ 維持 0.1 不變。

由於 $RATE_MARGIN = 0.1$ ，因此當新的分散處理的
分配後，處理器動作率是受到個別控制的時候，則只有子
處理器 43-1 的處理器動作率 $SCPU_RATE[1]$ 會被更新成
0.9；而當新的分散處理的分配後，處理器動作率仍是共

(45)

通控制的時候，則所有處理器的處理器動作率會因為 $RATE_MARGIN = 0.1$ ，而使 $SYSTEM_RATE$ 被變更成 0.9 。又，在每一處理器上，藉由新的分散處理被分配，主處理器 42 所能利用的記憶體容量 ($MEM_CPU[0]$ -現在的記憶體使用量) 係變為 3900 (kbytes)，子處理器 43-1 所能利用的記憶體容量 ($MEM_CPU[1]$ -現在的記憶體使用量) 係變為 30 (kbytes)，子處理器 43-2 所能利用的記憶體容量 ($MEM_CPU[2]$ -現在的記憶體使用量) 係變為 60 (kbytes)，子處理器 43-3 所能利用的記憶體容量 ($MEM_CPU[3]$ -現在的記憶體使用量) 則為 95 (kbytes) 不變。

步驟 S15 中，若判斷為沒有必要變更處理器動作率之設定時，或步驟 S17 的處理結束後，則在步驟 S18 中，邏輯執行緒執行控制部 102 係根據已設定之分配，將程式及資料載入至對應的處理器，控制邏輯執行緒之處理。主處理器 42 及子處理器 43，便會執行所被分配的執行緒之處理，而結束處理。

步驟 S9 中，若判斷為 $s > (\text{資訊處理控制器 11 內所設置之子處理器 43 的數目})$ 時，則在步驟 S19 中，邏輯執行緒執行控制部 102 係設 $SET_NG = 1$ ，並向使用者及分散處理要求來源的資訊處理裝置 1，通知無法進行分散處理之分配，而結束處理。

藉由如此處理，適用了本發明的資訊處理裝置 1 上，就可抑制處理器 (或處理器群) 的消費電力，而適切地選

(46)

擇被分散之處理的分配目標。

藉由上述處理而分配了分散處理的狀態，係為上述圖 14 中的狀態，當處理器動作率是被共通控制時的消費電力，係藉由下式 (11) 求出。

$$\begin{aligned}\Sigma PT &= \Sigma (PA \times \text{SYSTEM_RATE}[k] + K \times \text{SCPU_STAT}[k]) \times \text{SCPU_CK}[k] \\ &= (0.9 \times 4 \times PA + (0.4 + 0.8 + 0.4 + 0.1) \times K) \times 1000 \\ &= (3.6 \times PA + 1.7 \times K) \times 1000 \quad \cdot \cdot \cdot (11)\end{aligned}$$

順便一提，特別是，當處理器動作率被共通控制的情況下，圖 14 所示的子處理器 43-3 上，無論處理器使用率是否為低，因為處理器動作率都是很高，所以消費電力上會有浪費分配的情形。

於是，為了能更加抑制消費電力，亦可根據新的分散處理被分配前的處理器動作率 $\text{SCPU_RATE}[k]$ 或 SYSTEM_RATE ，與處理器使用率 $\text{SCPU_STAT}[k]$ 之間的關係，來決定分散處理的分配目標。

其次，參照圖 15 的流程圖，說明根據新的分散處理被分配前的處理器動作率與處理器使用率之間的關係，來決定分散處理的分配目標的分散處理分配決定處理 2。

步驟 S31 中，邏輯執行緒執行控制部 102 係接受來自應用程式執行控制部 101 的 $\text{REQUEST_information}$ 之供給，判斷是否有新的分散處理之分配的指示。步驟 S31 中，當判斷為沒有被指示新的分散處理之分配時，則一直重複步驟 S31 的處理直到判斷為有新的分散處理之分配指示

(47)

爲止。

此處，在複數設置子處理器 43 的資訊處理控制器 11 中，假設已經正在執行某些應用程式之狀態的主處理器 42 及子處理器 43-1 至子處理器 43-3 的狀態，是和用圖 13 說明過的情形一樣。

亦即，處理器動作率 $SYSTEM_RATE$ 爲 0.5， $k=0$ 所代表之主處理器 42 的處理器使用率 $SCPU_STAT[0]$ 係爲 0.3，因此，處理器使用率 $SCPU_STAT[0]$ /處理器動作率 $SYSTEM_RATE[0]$ 係爲 $0.3/0.5$ ，主處理器 42 所能利用之記憶體容量 $MEM_CPU[0]$ 係爲 5000 (kbytes)，代表最大動作時脈數的 $SCPU_CK[0]$ 係爲 1000。 $k=1$ 所代表之子處理器 43-1 的處理器使用率 $SCPU_STAT[1]$ 係爲 0.4，所以處理器使用率 $SCPU_STAT[1]$ /處理器動作率 $SYSTEM_RATE[1]$ 係爲 $0.4/0.5$ ，子處理器 43-1 所能利用的記憶體容量 $MEM_CPU[1]$ 係爲 100 (kbytes)，代表最大動作時脈數的 $SCPU_CK[1]$ 係爲 1000。又， $k=2$ 所代表之子處理器 43-2 的處理器使用率 $SCPU_STAT[2]$ 係爲 0.2，所以處理器使用率 $SCPU_STAT[2]$ /處理器動作率 $SYSTEM_RATE[2]$ 係爲 $0.2/0.5$ ，子處理器 43-2 所能利用的記憶體容量 $MEM_CPU[2]$ 係爲 100 (kbytes)，代表最大動作時脈數的 $SCPU_CK[2]$ 係爲 1000。再者， $k=3$ 所代表之子處理器 43-3 的處理器使用率 $SCPU_STAT[3]$ 係爲 0.1，所以處理器使用率 $SCPU_STAT[3]$ /處理器動作率 $SYSTEM_RATE[3]$ 係爲 $0.1/0.5$ ，子處理器 43-3 所能利用

(48)

的記憶體容量 MEM_CPU[3]係為 100 (kbytes)，代表最大動作時脈數的 SCPU_CK[3]係為 1000，假設以上記來進行說明。又，為了簡化說明，係假設 THREAD_OVERHEAD 為 0 而進行說明。

在此種狀態下，針對從透過網路 2 連接的其他資訊處理裝置 1，又有新的其他應用程式之執行要求的情況來加以說明。又，邏輯執行緒執行控制部 102 或動作資訊記憶體 105 現在正在管理的設定值，係設定為 REQ_REAL = 1 (要求即時處理)、NUM_SCPU = 3、TIME_DEF = 1 (1 秒)、RATE_MARGIN = 0.1、MEM_MARGIN = 10。

然後，假設 REQUEST_information 中所記載的被新要求執行之應用程式的動作條件係必須之子處理器數 REQ_NUM 為 2，主處理器 42 所被要求的處理週期數 REQ_PC[0]係為 100，所要求之記憶體容量 REQ_MEM[0]為 100 (kbytes)，第 1 子處理器 43 所被要求的處理週期數 REQ_PC[1]係為 400，所要求之記憶體容量 REQ_MEM[1]為 50 (kbytes)，第 2 子處理器 43 所被要求的處理週期數 REQ_PC[2]係為 200，所要求之記憶體容量 REQ_MEM[2]為 30 (kbytes)。

步驟 S31 中，當判斷有被指示新的分散處理之分配時，則在步驟 S32 中，邏輯執行緒執行控制部 102，係將表示是否滿足分配條件的旗標 SET_NG 予以初期化，亦即，設 SET_NG = 0。

步驟 S33 中，邏輯執行緒執行控制部 102，係取得現

(49)

在各處理器的處理器使用率 $SCPU_STAT[k]$ ，代入至 $TEMP_STAT[k]$ 。

步驟 S34 中，邏輯執行緒執行控制部 102，係取得現在各處理器的處理器動作率 $SCPU_RATE[k]$ ，代入至 $TEMP_RATE[k]$ 。

步驟 S35 中，邏輯執行緒執行控制部 102，係將表示處理之分配已決定之處理器的旗標 $SET_CPUID[t]$ 予以初期化，亦即，設 $SET_CPUID[t] = 0$ 。

步驟 S36 中，使用圖 16 及圖 17 的流程圖而執行後述的處理器分配決定處理。

步驟 S37 中，邏輯執行緒執行控制部 102 係使用圖 16 及圖 17 的流程圖而根據後述之處理器分配決定處理的處理結果也就是 SET_NG 的值，來判斷新的分散處理之分配是否成功。

步驟 S37 中，若判斷為新的分散處理之分配是成功時，則步驟 S38 中，邏輯執行緒執行控制部 102 係將所有的處理器的處理器動作率 $TEMP_RATE[k]$ ，及最新的處理器使用率 $TEMP_STAT[k]$ ，加以比較，藉此而判斷是否有需要隨著新的分散處理之分配，而變更處理器動作率的設定。

步驟 S38 中，當判斷為有需要變更處理器動作率的設定時，亦即，無論在哪個處理器上，皆為處理器動作率 $TEMP_RATE[k] < 處理器使用率 TEMP_STAT[k]$ 的時候，則在步驟 S39 中，邏輯執行緒執行控制部 102 係會算出新

(50)

的處理器動作率，並將算出結果，供給至動作率設定部 103。動作率設定部 103，係對主處理器 42 或子處理器 43 當中的對應者，進行處理器動作率的設定。

步驟 S40 中，邏輯執行緒執行控制部 102，係將動作資訊記憶體 105 所記錄之分散處理之分配前，亦即，現在各處理器的處理器使用率 $SCPU_STAT[k]$ 及處理器動作率 $SCPU_RATE[k]$ 的值，分別更新成，藉由分配計算所得到的處理器使用率 $TEMP_STAT[k]$ 及 $TEMP_RATE[k]$ 。新的分散處理之分配結果，和動作率的設定，係使用圖 18 至圖 20 而說明如後。

步驟 S38 中，若判斷為沒有必要變更處理器動作率之設定時，或步驟 S39 的處理結束後，則在步驟 S41 中，邏輯執行緒執行控制部 102 係根據已設定之分配，將程式及資料載入至對應的處理器，控制邏輯執行緒之處理。主處理器 42 及子處理器 43，便會執行所被分配的執行緒之處理，而結束處理。

步驟 S37 中，若判斷為分散處理之分配並未成功時，則在步驟 S42 中，邏輯執行緒執行控制部 102 係向使用者及分散處理要求來源的資訊處理裝置 1，通知分散處理之分配為不可進行，並結束處理。

藉由如此處理，適用了本發明的資訊處理裝置 1 上，相較於使用圖 11 及圖 12 所說明過的分散處理分配決定處理 1 執行的情形，可更加抑制處理器（或處理器群）的消費電力，而適切地選擇被分散之處理的分配目標。

(51)

其次，參照圖 16 及圖 17 的流程圖，說明圖 15 的步驟 S36 中所執行的處理器分配決定處理。

步驟 S61 中，邏輯執行緒執行控制部 102，係從所被供給之 REQUEST_information 中所含有的資訊中，取得出受到分配指示之分散處理之執行上所必需的子處理器數 j。

步驟 S62 中，邏輯執行緒執行控制部 102，係為了判斷分散處理之分配可否，作為用來區別每一處理器所需的內部暫存器值的對應值，而定義了 SCPU_NB。

步驟 S63 中，邏輯執行緒執行控制部 102，係將 SCPU_NB[0]視為是主處理器 42。

步驟 S64 中，邏輯執行緒執行控制部 102，係根據所被供給的 REQUEST_information 中所含的資訊，來判斷已接受執行要求的處理中，是否已有指定 PRIORITY_ID。步驟 S64 中，若判斷為沒有指定 PRIORITY_ID 時，則處理便進入後述的步驟 S67。

步驟 S64 中，若判斷為已有指定 PRIORITY_ID 時，則於步驟 S65 中，邏輯執行緒執行控制部 102 便將 SCPU_NB[1]，視為 PRIORITY_ID 所指定之子處理器 43。

步驟 S66 中，邏輯執行緒執行控制部 102，係參照所被供給之 REQUEST_information 中所含之資訊，在 $2 \leq i \leq (\text{資訊處理控制器 11 所含之子處理器 43 數})$ 的 SCPU_NB[i]中，略過已經指定待命 (suspend) 的子處理器 43，將相對於現在處理器動作率 SCPU_RATE[k]或

(52)

SYSTEM_RATE 之處理器使用率 SCPU_STAT[k]的可用度由高往低排序後的子處理器 43 的子處理器 ID，賦予對應關聯，然後處理就進入後述的步驟 S68。

步驟 S64 中，若判斷為 PRIORITY_ID 沒有被指定時，則於步驟 S67 中，邏輯執行緒執行控制部 102，係參照所被供給之 REQUEST_information 中所含之資訊，在 $1 \leq i \leq$ (資訊處理控制器 11 所含之子處理器 43 數) 的 SCPU_NB[i]中，略過已經指定待命 (suspend) 的子處理器 43，將相對於現在處理器動作率 SCPU_RATE[k]或 SYSTEM_RATE 之處理器使用率 SCPU_STAT[k]的可用度由高往低排序後的子處理器 43 的子處理器 ID，賦予對應關聯。

亦即，藉由步驟 S63 至步驟 S67 之處理，除了 SCPU_NB[0]上被賦予對應了主處理器 42，SCPU_NB[1]，SCPU_NB[2]，SCPU_NB[3]... SCPU_NB[k]上，當已有指定 PRIORITY_ID 時、則以被指定的子處理器 43 為優先、已被指定待命之子處理器 43 被略過的狀態，而被賦予對應至相對於現在處理器動作率 SCPU_RATE[k]或 SYSTEM_RATE 之處理器使用率 SCPU_STAT[k]的可用度由高往低排序後的子處理器 43。

亦即，在使用圖 13 所說明過的狀況中，當沒有 PRIORITY_ID 及待命之指定時，除了 SCPU_NB[0]被賦予對應至主處理器 42，同時，SCPU_NB[1]被賦予對應至子處理器 43-3，SCPU_NB[2]被賦予對應至子處理器 43-2，

(53)

SCPU_NB[3]被賦予對應至子處理器 43-1。

步驟 S66 或步驟 S67 的處理結束後，於步驟 S68 中，邏輯執行緒執行控制部 102，係對代表要求之處理所需要的子處理器數 REQ_NUM，將一表示要進行第幾號之處理器的分配決定處理的暫存器的值 t ，設為 $t=0$ 。此處， $t=0$ 的處理器係代表主處理器 42， $t \geq 1$ 的處理器係代表子處理器 43 當中的任一者。

步驟 S69 中，邏輯執行緒執行控制部 102，係將 SCPU_NB[i]中的變數 i ，設為 $i=0$ 。

步驟 S70 中，邏輯執行緒執行控制部 102，係參照 REQUEST_information 中所含有之資訊，判斷被要求分散至子處理器 43 當中任一者而進行分配的 REQ_NUM(t)之處理，是否能藉由 SCPU_NB[i]所代表的子處理器 43 來執行。

具體而言，邏輯執行緒執行控制部 102，係除了將 REQ_NUM(t)之處理所致之，SCPU_NB[i]所代表之處理器上的處理器使用率 SCPU_STAT[k]的增加量予以加算，還會以處理器使用率 $\text{SCPU_STAT}[k] + \text{RATE_MARGIN}$ 不會超過 1，以及針對被 REQ_NUM(t)之處理所要求之 SCPU_NB[i]所代表之處理器中的記憶體容量要成立 ($\text{REQ_MEM}[s] + \text{MEM_MARGIN}$) < ($\text{MEM}[s]_\text{CPU}$ -現在的記憶體使用量) 為條件，來判斷 REQ_NUM(t)之處理，是能否藉由 SCPU_NB[i]所代表之處理器來執行。

步驟 S70 中，當判斷為 REQ_NUM(t)的處理係不可

(54)

能以 SCPU_NB[i]所代表之處理器來執行時，則在步驟 S71 中，邏輯執行緒執行控制部 102 便將 SCPU_NB[i]中的變數 i，設為 $i = i + 1$ 。

步驟 S72 中，邏輯執行緒執行控制部 102，係判斷是否為 $i >$ （除了待命以外，實質上能夠分配處理的子處理器 43 的數目）。步驟 S72 中，若判斷為並非 $i >$ （除了待命以外，實質上能夠分配處理的子處理器 43 的數目）時，則處理便返回步驟 S70，並重複其以降的處理。步驟 S72 中，若判斷為 $i >$ （除了待命以外，實質上能夠分配處理的子處理器 43 的數目）時，則處理便進入後述的步驟 S78。

步驟 S70 中，若判斷為 REQ_NUM(t) 之處理是可藉由 SCPU_NB[i]所代表之處理器來執行時，則於步驟 S73 中，邏輯執行緒執行控制部 102 係將代表處理之分配已決定之處理器的暫存器值 SET_CPUID[t]，設為 SCPU_NB[i]所代表之處理器的處理器 ID。

步驟 S74 中，邏輯執行緒執行控制部 102，係在 SCPU_NB[i]所代表之處理器的 TEMP_RATE[k]上，加算了新分配之處理所對應之處理器使用率，而更新之。

步驟 S75 中，邏輯執行緒執行控制部 102，係將表示目前正在進行第幾號處理器之分配決定處理的暫存器值 t，設為 $t = t + 1$ 。

亦即，對於和用圖 13 說明過之情形相同之狀態的主處理器 42 及子處理器 43-1 至子處理器 43-3，必要之子處

(55)

理器 REQ_NUM 係為 2，主處理器 42 所被要求之處理週期數 REQ_PC[0] 係為 100，所要求之記憶體容量 REQ_MEM[0] 係為 100 (kbytes) 的新的分散處理被要求時，i = 0 的情況中，邏輯執行緒執行控制部 102，係如下式 (12) 所示般，對主處理器 42 進行新的分散處理的分配時，由於處理器使用率 SCPU_STAT[0] + RATE_MARGIN 不超過 1，且如下式 (13) 所示， $(REQ_MEM[0] + MEM_MARGIN) < (MEM_CPU[0] - \text{現在記憶體使用量})$ 為成立，因此判斷為，REQ_NUM (0) 之處理，係可藉由 SCPU_NB[0] 所代表之主處理器 42 來執行。

$$\begin{aligned}
 & \text{TEMP_STAT}[0] \text{ (分配後)} \\
 &= (\text{REQ_PC}[0] \div \text{TIME_DEF}) \div \text{SCPU_CK}[0] + \text{TEMP_STAT}[0] \text{ (分配前)} + \text{RATE_MARGIN} \\
 &= (100 \div 1) \div 1000 + 0.3 + 0.1 \\
 &= 0.5 \quad \cdot \cdot \cdot (12)
 \end{aligned}$$

$$\text{MEM_CPU}[0] - \text{現在記憶體使用量} - \text{MEM_MARGIN} (= 4890) > \text{REQ_MEM}[0] (= 100) \quad \cdot \cdot \cdot (13)$$

因此，邏輯執行緒執行控制部 102，係在 SET_CPUID[0] 中記載入主處理器 42 的處理器 ID，並令 TEMP_STAT[0] = 0.4，將現在記憶體使用量設為 1000 + 100 = 1100，同時，還將表示目前正在進行第幾號處理器之分配決定處理的暫存器值 t，設為 t = 1。

步驟 S76 中，邏輯執行緒執行控制部 102，係會判斷

(56)

是否 $t > j$ (j 係分散處理所必須之子處理器數)。步驟 S76 中，若判斷為並非 $t > j$ 時，則處理便返回步驟 S69，並重複其以降之處理。

例如，當 $t = 1$ 時，由於判斷為並非 $t > j$ ，所以和上述情況相同地，在所被要求之新的分散處理中，必須之子處理器數 REQ_NUM 為 2，第 1 子處理器 43 所被要求之處理週期數 REQ_PC[1] 為 400，所要求之記憶體容量 REQ_MEM[1] 為 50 (kbytes) 時， $i = 1$ 的情況下，邏輯執行緒執行控制部 102，係如下式 (14) 所示般地藉由 SCPU_NB[1] 所代表之子處理器 43-3，處理器使用率 SCPU_STAT[3] + RATE_MARGIN 係不超過 1，或如下式 (15) 所示般地、 $(\text{REQ_MEM}[1] + \text{MEM_MARGIN}) < (\text{MEM_CPU}[3] - \text{現在記憶體使用量})$ 是成立，因此判斷為，REQ_NUM (1) 的處理，是可藉由 SCPU_NB[1] 所代表之子處理器 43-3 來執行。

$$\begin{aligned}
 & \text{TEMP_STAT}[3] \text{ (分配後)} \\
 &= (\text{REQ_PC}[1] \div \text{TIME_DEF}) \div \text{SCPU_CK}[3] + \text{TEMP_STAT}[3] \text{ (分配前)} + \text{RATE_MARGIN} \\
 &= (400 \div 1) \div 1000 + 0.1 + 0.1 \\
 &= 0.6 \quad \cdot \cdot \cdot (14)
 \end{aligned}$$

$$\begin{aligned}
 & \text{MEM_CPU}[3] - \text{現在記憶體使用量} - \text{MEM_MARGIN} (= 85) > \text{REQ_MEM}[1] (= 50) \quad \cdot \cdot \cdot (15)
 \end{aligned}$$

因此，邏輯執行緒執行控制部 102，係在 SET_CPUID[1] 中記載入子處理器 43-3 的處理器 ID，並令

(57)

TEMP_STAT[1] = 0.5，將現在記憶體使用量設為 55 (kbytes)，同時，還將表示目前正在進行第幾號處理器之分配決定處理的暫存器值 t，設為 t = 2。

然後，同樣地，在所被要求之新的分散處理中，第 2 子處理器 43 所被要求之處理週期數 REQ_PC[2] 為 200，所要求之記憶體容量 REQ_MEM[2] 為 30 (kbytes) 時，i = 2 的情況下，邏輯執行緒執行控制部 102，係如下式 (16) 所示般地藉由 SCPU_NB[2] 所代表之子處理器 43-2，處理器使用率 SCPU_STAT[2] + RATE_MARGIN 係不超過 1，或如下式 (17) 所示般地、(REQ_MEM[2] + MEM_MARGIN) < (MEM_CPU[2] - 現在記憶體使用量) 是成立，因此判斷為，REQ_NUM(2) 的處理，是可藉由 SCPU_NB[2] 所代表之子處理器 43-2 來執行。

TEMP_STAT[2] (分配後)

$$\begin{aligned}
 &= (REQ_PC[2] \div TIME_DEF) \div SCPU_CK[2] + \\
 &TEMP_STAT[2] (分配前) + RATE_MARGIN \\
 &= (200 \div 1) \div 1000 + 0.2 + 0.1 \\
 &= 0.5 \quad \cdot \cdot \cdot (16)
 \end{aligned}$$

$$\begin{aligned}
 &MEM_CPU[2] - \text{現在記憶體使用量} - MEM_MARGIN (= \\
 &80) > REQ_MEM[2] (= 30) \quad \cdot \cdot \cdot (17)
 \end{aligned}$$

因此，邏輯執行緒執行控制部 102，係在 SET_CPUID[2] 中記載入子處理器 43-2 的處理器 ID，並令 TEMP_STAT[2] = 0.4，將現在記憶體使用量設為 40 (kbytes)，同時，還將表示目前正在進行第幾號處理器之

(58)

分配決定處理的暫存器值 t ，設為 $t = 3$ 。

步驟 S76 中，若判斷為 $t > j$ 時，則於步驟 S77 中，邏輯執行緒執行控制部 102 會將表示分配已成功的旗標豎立起來，而設 $SET_NG = 0$ ，處理便返回圖 15 的步驟 S36，進入步驟 S37。

步驟 S72 中，若判斷為 $i > (\text{除了待命等以外，實質上能夠分配處理的子處理器 } 43 \text{ 的數目})$ 時，則於步驟 S78 中，邏輯執行緒執行控制部 102 係為了表示分配失敗而將旗標撤下，設 $SET_NG = 1$ ，處理便返回圖 15 的步驟 S36，進入步驟 S37。

藉由如此處理，適用了本發明的資訊處理裝置 1 上，相較於使用圖 11 及圖 12 所說明過的分散處理分配決定處理 1 執行的情形，可更加抑制處理器（或處理器群）的消費電力，而適切地選擇被分散之處理的分配目標。

和使用圖 13 所說明過之情形相同之狀態的主處理器 42 及子處理器 43-1 至子處理器 43-3 上，當處理器動作率 $SYSTEM_RATE$ 是被共通控制時的分配結果，示於圖 18。

$k = 0$ 所代表的主處理器 42 的處理器使用率 $SCPU_STAT[0]$ 係變為 0.4， $k = 1$ 所代表的子處理器 43-1 的處理器使用率 $SCPU_STAT[1]$ 係維持 0.4 不變， $k = 2$ 所代表的子處理器 43-2 的處理器使用率 $SCPU_STAT[2]$ 係變為 0.4， $k = 3$ 所代表的子處理器 43-3 的處理器使用率 $SCPU_STAT[3]$ 係變為 0.5。此處，由於 $RATE_MARGIN$ 係為 0.1，因此處理器動作率 $SYSTEM_RATE$ 係被控制成

(59)

0.6。

藉由上述處理而分配了分散處理的狀態，係為上述圖 18 中的狀態，當處理器動作率是被共通控制時的消費電力，係藉由下式 (18) 求出。

$$\begin{aligned}\Sigma PT &= \Sigma (PA \times \text{SYSTEM_RATE} + K \times \text{SCPU_STAT}[k]) \times \text{SCPU_CK}[k] \\ &= (0.6 \times 4 \times PA + (0.4 + 0.4 + 0.4 + 0.5) \times K) \times 1000 \\ &= (2.4 \times PA + 1.7 \times K) \times 1000 \quad \cdot \cdot \cdot (18)\end{aligned}$$

亦即，相較於使用圖 14 所說明過的分配處理後，可減少 $1200 \times PA$ 份的消費電力。

圖 18 雖然是圖示了處理器動作率 SYSTEM_RATE 是被共通控制時的分配結果，但使用圖 15 至圖 17 之流程圖所說明過的處理，係處理器動作率 $\text{SCPU_RATE}[k]$ 為個別設定時，則能特別有效率地抑制消費電力。

例如，在已經執行某些應用程式之狀態的主處理器 42 及子處理器 43-1 至子處理器 43-3 的處理器動作率 $\text{SCPU_RATE}[k]$ 和處理器使用率 $\text{SCPU_STAT}[k]$ 之間的關係，係如圖 19 所示， $k=0$ 所代表之主處理器 42 的處理器使用率 $\text{SCPU_STAT}[0]$ 為 0.3、處理器使用率 $\text{SCPU_STAT}[0]/$ 處理器動作率 $\text{SCPU_RATE}[0]$ 為 $0.3/0.5$ ， $k=1$ 所代表之子處理器 43-1 的處理器使用率 $\text{SCPU_STAT}[1]$ 為 0.4、處理器使用率 $\text{SCPU_STAT}[1]/$ 處理器動作率 $\text{SCPU_RATE}[1]$ 為 $0.4/0.9$ ， $k=2$ 所代表之子處理器 43-2 的處理器使用率 $\text{SCPU_STAT}[2]$ 為 0.2、處理器

(60)

使用率 $SCPU_STAT[2]$ /處理器動作率 $SCPU_RATE[2]$ 爲 $0.2/0.3$ ，然後， $k=3$ 所代表之子處理器 43-3 的處理器使用率 $SCPU_STAT[3]$ 爲 0.1 、處理器使用率 $SCPU_STAT[3]$ /處理器動作率 $SCPU_RATE[3]$ 爲 $0.1/0.5$ 的狀態下，和上述情況相同地，考慮 $REQUEST_information$ 中所記載之新的執行被要求之應用程式的動作條件爲：必須之子處理器數 REQ_NUM 爲 2 ，主處理器 42 所被要求之處理週期數 $REQ_PC[0]$ 爲 100 ，第 1 子處理器 43 所被要求之處理週期數 $REQ_PC[1]$ 爲 400 ，第 2 子處理器 43 所被要求之處理週期數 $REQ_PC[2]$ 爲 200 ，的分散處理被要求時的分配。

$SCPU_NB[0]$ 所代表之主處理器 42 所被要求之處理的分配，雖然是相同於用式 (12) 所說明過的情形，但被分配給 $SCPU_NB[1]$ 的，會是相對於現在處理器動作率 $SCPU_RATE[k]$ 或 $SYSTEM_RATE$ 的處理器使用率 $SCPU_STAT[k]$ 之可用度最高的子處理器 43-1；被分配給 $SCPU_NB[2]$ 的，會是相對於現在處理器動作率 $SCPU_RATE[k]$ 或 $SYSTEM_RATE$ 的處理器使用率 $SCPU_STAT[k]$ 之可用度次高的子處理器 43-3。亦即， $SCPU_NB[1]$ 的子處理器 43-1 會被分配 $REQ_NUM(1)$ 之處理， $SCPU_NB[2]$ 的子處理器 43-3 會被分配 $REQ_NUM(2)$ 之處理。

因此，分配結果爲， $k=0$ 所代表的主處理器 42 的處理器使用率 $SCPU_STAT[0]$ 係變爲 0.4 ， $k=1$ 所代表的子

(61)

處理器 43-1 的處理器使用率 $SCPU_STAT[1]$ 係維持 0.8 不變， $k = 2$ 所代表的子處理器 43-2 的處理器使用率 $SCPU_STAT[2]$ 係變為 0.4， $k = 3$ 所代表的子處理器 43-3 的處理器使用率 $SCPU_STAT[3]$ 係變為 0.3。此處，由於 $RATE_MARGIN$ 為 0.1，因此處理器動作率 $SCPU_RATE[i]$ ，係維持以圖 19 說明過之處理的分配前的狀態不變，而不需要變更，因此可將新的處理分配所致之消費電力量的增加予以大幅抑制。

此外， REQ_REAL 被設定為 0（不須即時處理）的應用程式所對應之邏輯執行緒，係亦可將應用程式執行的工作優先度，排程成低於需要即時處理之工作，令其在有空閒時間時才被處理。例如，將 REQ_REAL 被設定為 0（不須即時處理）的應用程式所對應之邏輯執行緒，最初是以和需要即時處理之邏輯執行緒的相同條件而進行分配處理，但亦可為，當分配失敗時，則將處理所需的處理週期數乘以 $1/M$ （令規定時間成 M 倍），來再度執行分配處理。

藉由此種處理，可一邊抑制消費電力量，同時實現能提供所被要求之機能的資訊處理裝置。尤其是，在 CE（Consumer electronics，消費性電子）機器中，消費電力量的削減，是作為商品競爭力的一大賣點，因此適用本發明的有利點是非常大的。

上述一連串之處理，係可藉由軟體來執行。該軟體，係為構成該軟體的程式，從記錄媒體，安裝至組裝有專用

硬體的電腦、或藉由安裝各種程式而可執行各種機能的例如泛用之個人電腦等。

該記錄媒體，係如圖 1 所示，是爲了和電腦分開來，將程式另外提供給使用者所需而發佈的，記錄有程式的磁碟片 61（包含軟碟）、光碟 62（包含 CD-ROM（Compact Disk-Read Only Memory）、DVD（Digital Versatile Disk））、光磁碟 63（包含 MD（Mini-Disk）（商標）），或著半導體記憶體 64 等所成的封裝媒體等所構成。

又，本說明書中，將被記錄在記錄媒體中之程式加以描述的步驟，除了包含按照記載順序而在時間序列上進行的處理以外，還包含了並不一定要按照時間序列進行處理而是可平行地或個別地執行之處理。

此外，在本說明書中，所謂系統，係代表著由複數裝置所構成之裝置全體。

【圖式簡單說明】

〔圖 1〕適用了本發明之通訊系統之一實施形態的圖示。

〔圖 2〕主記憶體的說明圖。

〔圖 3〕子處理器的近端儲存區之說明圖。

〔圖 4〕金鑰管理表的說明圖。

〔圖 5〕軟體胞之構成例的圖示。

〔圖 6〕DMA 指令爲狀態回送指令時的軟體胞的軟體胞之資料領域之構造圖。

(63)

〔圖 7〕圖 1 之資訊處理控制器的主記憶體所記憶的軟體胞之構成的圖示。

〔圖 8〕資訊處理控制器可執行之機能的說明用之機能方塊圖。

〔圖 9〕REQUEST_information (應用程式要求條件) 的說明圖。

〔圖 10〕消費電力 PT，與動作時脈數的關係圖。

〔圖 11〕分散處理分配決定處理 1 的說明用流程圖。

〔圖 12〕分散處理分配決定處理 1 的說明用流程圖。

〔圖 13〕新處理之分配前的各處理器之狀態的說明圖。

〔圖 14〕處理分配後的各處理器之狀態的說明圖。

〔圖 15〕分散處理分配決定處理 2 的說明用流程圖。

〔圖 16〕處理器分配決定處理的說明用流程圖。

〔圖 17〕處理器分配決定處理的說明用流程圖。

〔圖 18〕處理分配後的各處理器之狀態的說明圖。

〔圖 19〕新處理之分配前的各處理器之狀態的說明圖。

〔圖 20〕處理分配後的各處理器之狀態的說明圖。

【主要元件符號說明】

(64)

- 1：資訊處理裝置
- 2：網路
- 3：資訊處理裝置
- 11：資訊處理控制器
- 12：主記憶體
- 15：操作輸入部
- 16：通訊部
- 19：電源部
- 42：主處理器
- 43：子處理器
- 51：近端儲存區
- 101：應用程式執行控制手段
- 102：邏輯執行緒執行控制部
- 103：動作率設定部
- 104：動作資訊取得
- 105：動作資訊記憶體

五、中文發明摘要

發明之名稱：資訊處理裝置及資訊處理方法，以及記錄有程式之電腦可讀取之媒體

根據處理器的動作率和使用率之關係來控制分散處理之分配。

步驟 S31 中若有指示新的分散處理時，則步驟 S32 中將表示是否滿足分配條件的旗標予以初期化，在步驟 S33、步驟 S34 中取得處理器使用率和處理器動作率，在步驟 S35 中將表示已決定分配之處理器的旗標予以初期化。步驟 S36 中執行處理器分配決定處理，步驟 S37 中若判斷為分配成功時，則步驟 S38 中會判斷是否有需要變更處理器動作率的設定，若有必要，則在步驟 S39 中算出新的處理器動作率，在步驟 S40 中更新處理器使用率和處理器動作率。步驟 S41 中則將程式與資料予以載入，而行邏輯執行緒的處理。本發明可適用於資訊處理裝置。

六、英文發明摘要

發明之名稱：

(1)

十、申請專利範圍

1. 一種資訊處理裝置，其特徵為，

具備：

第 1 處理器；和

複數之第 2 處理器；

前記第 1 處理器係

具備：

應用程式執行控制手段，其係控制應用程式之執行；和

動作資訊取得手段，其係取得前記第 2 處理器之動作的相關資訊；和

分散處理控制手段，當前記應用程式是受到前記應用程式執行控制手段加以執行控制之際，將前記應用程式之執行相關的複數處理總結成一個提供機能的處理單位，並為了提供前記處理單位所對應之前記機能，將處理分配給複數之前記第 2 處理器而令其加以執行的分散處理加以控制；和

時脈數比設定手段，將相對於前記第 2 處理器所能夠動作之最大動作時脈數的現在動作時脈數的比值，也就是時脈數比，加以設定；

前記動作資訊，係包含：前記最大動作時脈數、前記現在動作時脈數、在規定時間內前記第 2 處理器所被分配之處理所需之週期（cycle）數；

前記分散處理控制手段，係根據前記動作資訊，而控

(2)

制前記處理單位所對應之前記分散處理。

2.如申請專利範圍第 1 項所記載之資訊處理裝置，其中，

前記時脈數比設定手段，係對於所有之前記第 2 處理器設定時脈數，以使前記時脈數比變成共通的值。

3.如申請專利範圍第 1 項所記載之資訊處理裝置，其中，

前記時脈數比設定手段，係以複數之前記第 2 處理器呈個別獨立的方式，來設定前記時脈數比。

4.如申請專利範圍第 1 項所記載之資訊處理裝置，其中，

前記分散處理控制手段，係根據前記現在動作時脈數、前記在規定時間內之前記第 2 處理器所被分配之處理所需之週期數，而算出處理器的使用率，並進行控制以使得針對相對於前記時脈數比的前記處理器使用率之比率為低的前記第 2 處理器，優先地被分配前記處理單位所對應之前記分散處理。

5.如申請專利範圍第 4 項所記載之資訊處理裝置，其中

前記時脈數比設定手段，係根據前記處理器使用率，來更新前記時脈數比。

6.一種資訊處理方法，係屬於具備：

第 1 處理器；和

複數之第 2 處理器；

(3)

之資訊處理裝置的資訊處理方法，其特徵為，含有：

分散處理開始要求步驟，其係藉由前記第 1 處理器所致之應用程式之執行控制，而將複數之前記第 2 處理器所被分配的處理總結成一個提供機能的處理單位，並為了提供前記處理單位所對應之前記機能，而要求將處理分配給複數之前記第 2 處理器而令其加以執行的分散處理之開始；和

動作資訊取得步驟，其係取得前記第 2 處理器之動作相關的動作資訊，該動作資訊係包含：前記第 2 處理器所能動作之最大動作時脈數、前記第 2 處理器之現在動作時脈數、在規定時間內前記第 2 處理器所被分配之處理所需之週期（cycle）數；和

分配控制步驟，其係根據前記第 2 處理器之動作相關的動作資訊，來控制前記處理單位所對應之前記分散處理的分配；和

時脈數比設定步驟，將相對於前記第 2 處理器所能夠動作之最大動作時脈數的現在動作時脈數的比值，也就是時脈數比，加以設定。

7.一種記錄有電腦程式之電腦可讀取之媒體，該程式係屬於將使用了

第 1 處理器；和

複數之第 2 處理器；

之分散處理加以控制之電腦所能執行的程式，其特徵為，該程式係用來令電腦執行包含以下步驟之處理：

(4)

分散處理開始要求步驟，其係藉由前記第 1 處理器所致之應用程式之執行控制，而將複數之前記第 2 處理器所被分配的處理總結成一個提供機能的處理單位，並爲了提供前記處理單位所對應之前記機能，而要求將處理分配給複數之前記第 2 處理器而令其加以執行的分散處理之開始；和

動作資訊取得步驟，其係取得前記第 2 處理器之動作相關的動作資訊，該動作資訊係包含：前記第 2 處理器所能動作之最大動作時脈數、前記第 2 處理器之現在動作時脈數、在規定時間內前記第 2 處理器所被分配之處理所需之週期（cycle）數；和

分配控制步驟，其係根據前記第 2 處理器之動作相關的動作資訊，來控制前記處理單位所對應之前記分散處理的分配；和

時脈數比設定步驟，將相對於前記第 2 處理器所能夠動作之最大動作時脈數的現在動作時脈數的比值，也就是時脈數比，加以設定。

圖 1

758174

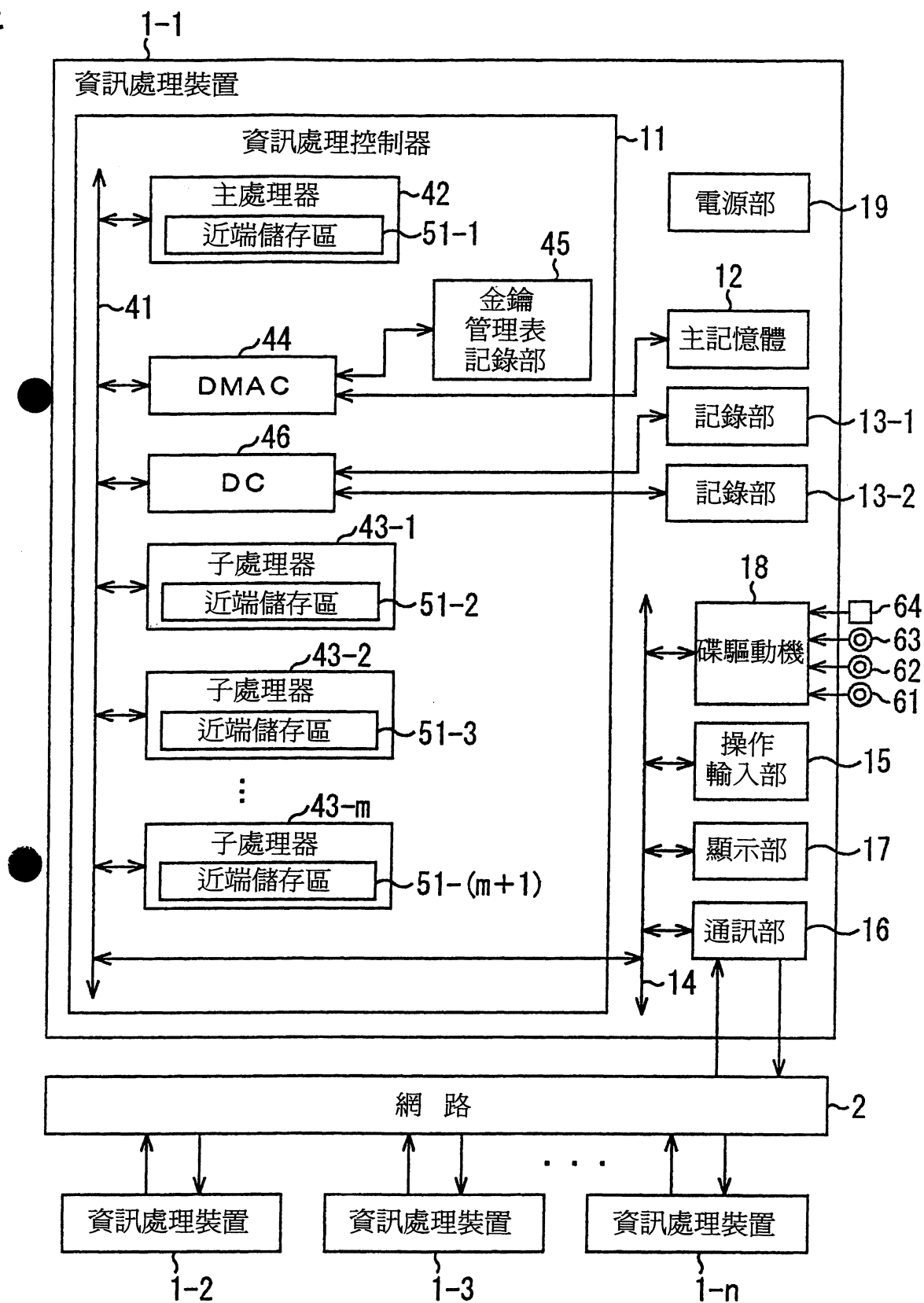


圖2

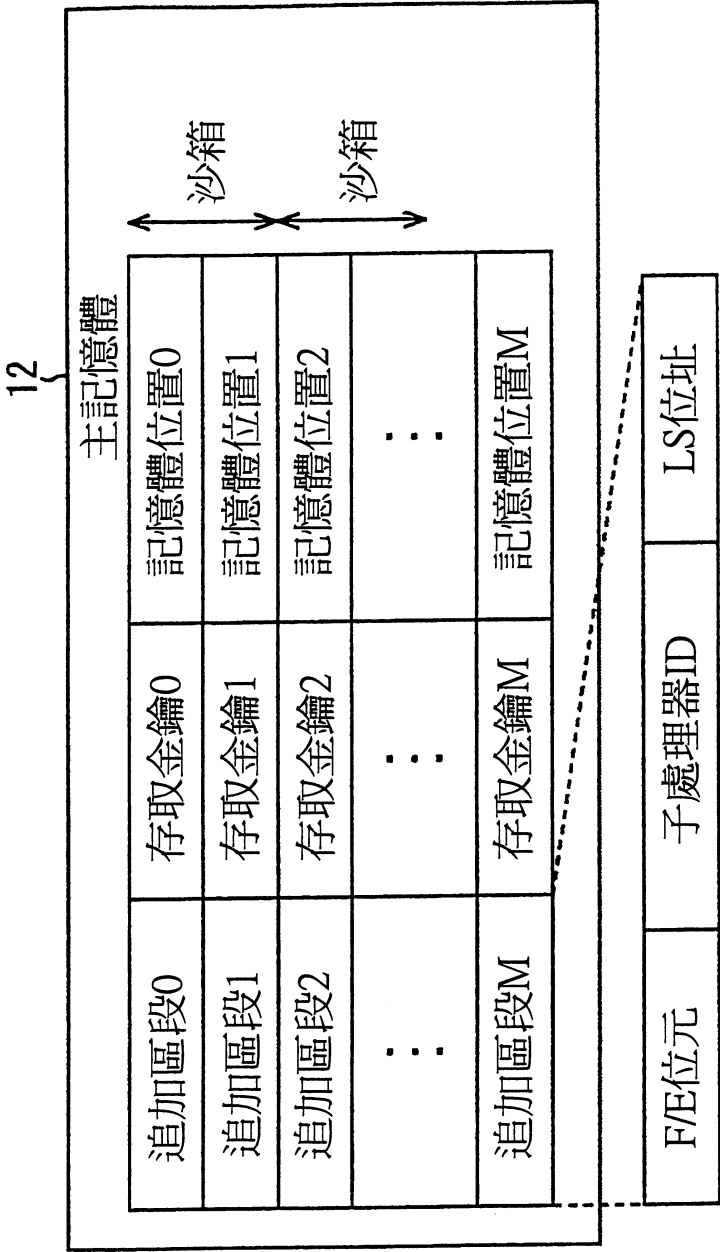


圖 3

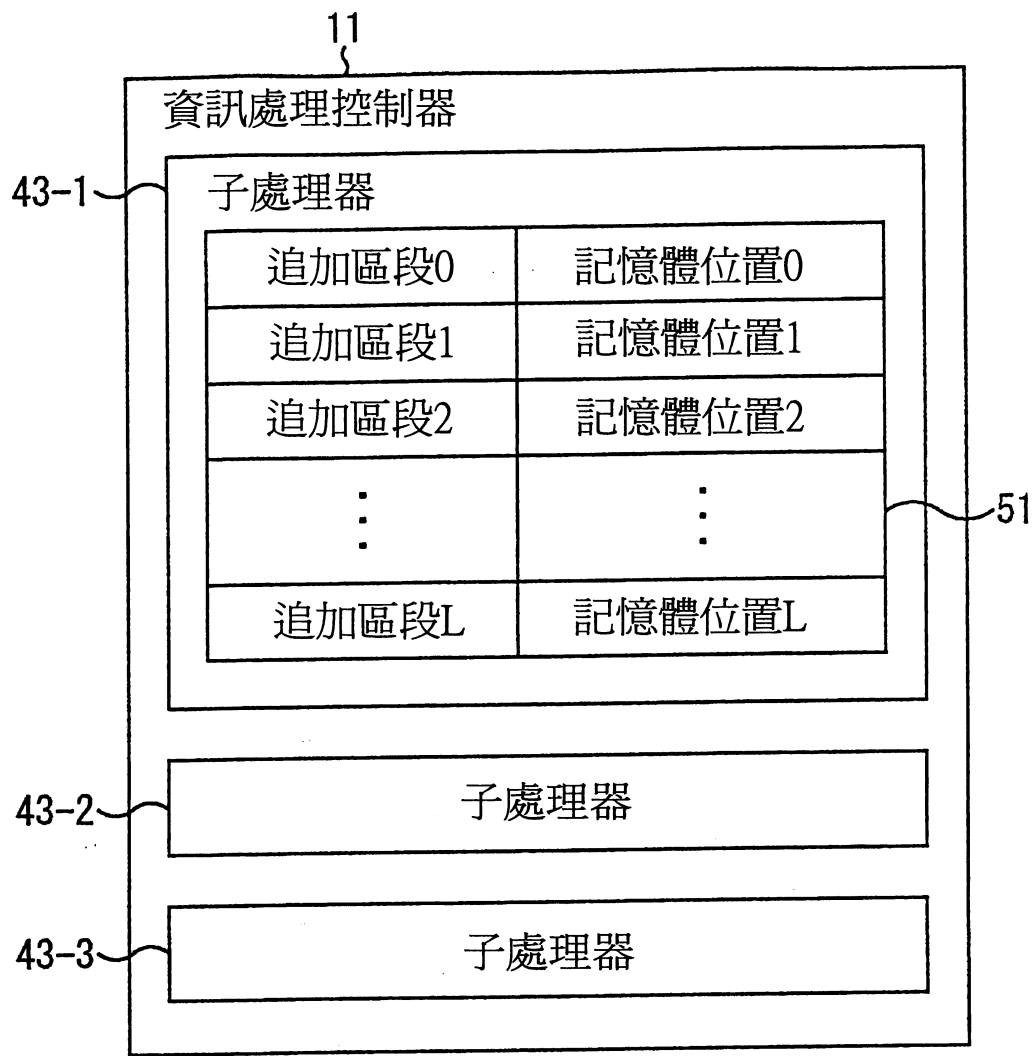


圖 4

金鑰管理表

子處理器ID		
0	子處理器金鑰0	金鑰遮罩0
1	子處理器金鑰1	金鑰遮罩1
2	子處理器金鑰2	金鑰遮罩2
⋮	⋮	⋮
N	子處理器金鑰N	金鑰遮罩N

圖5

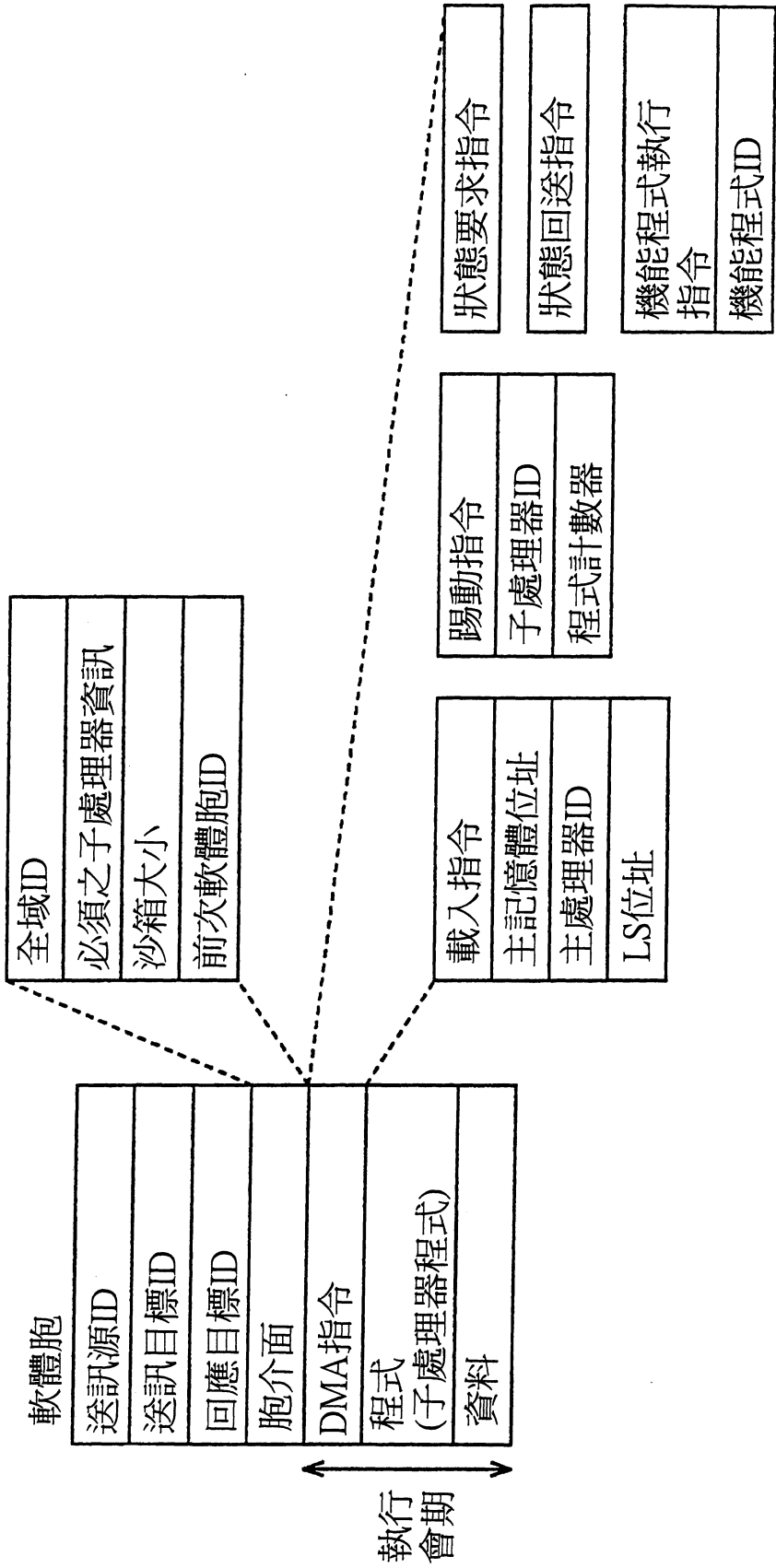


圖6

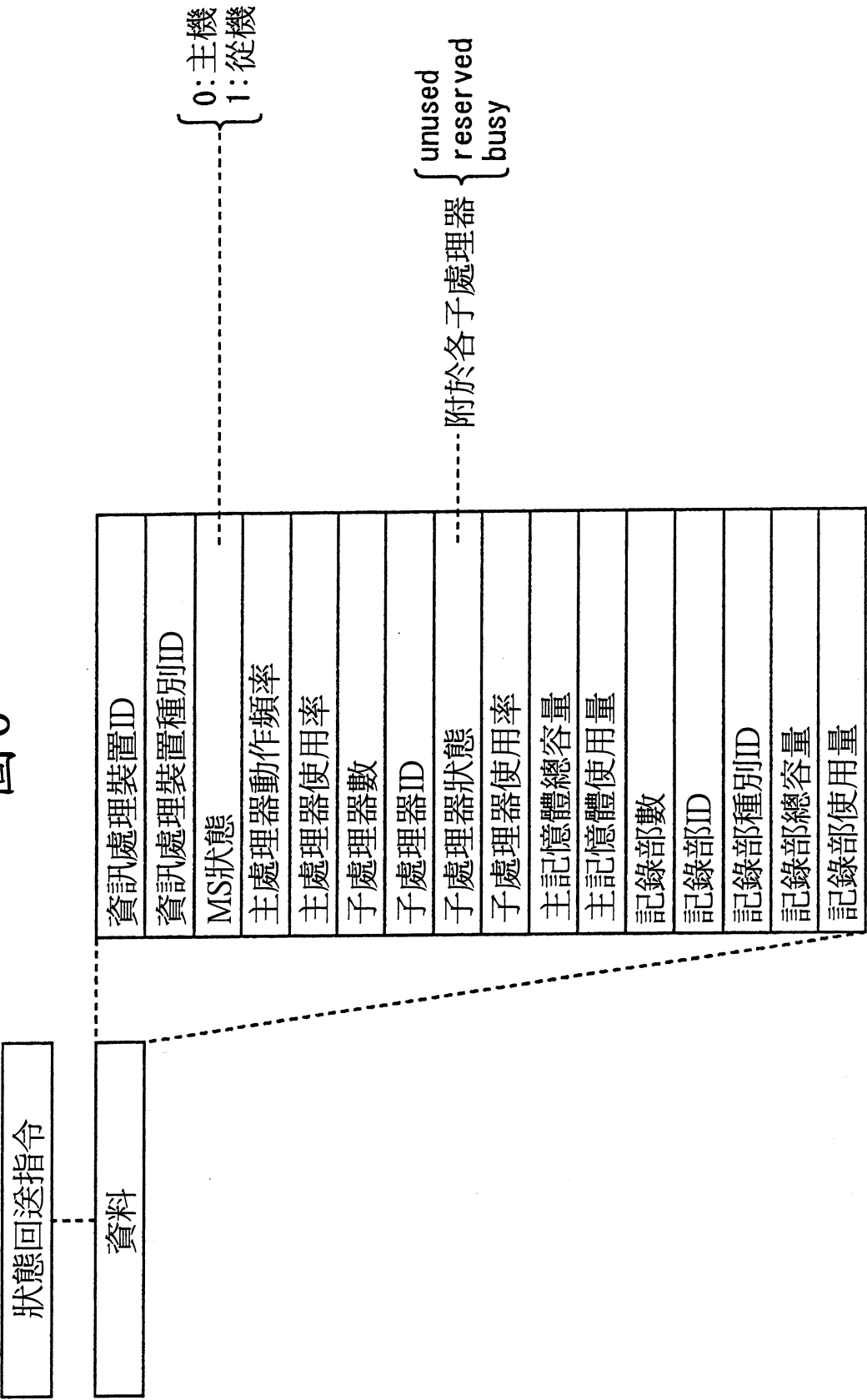


圖7

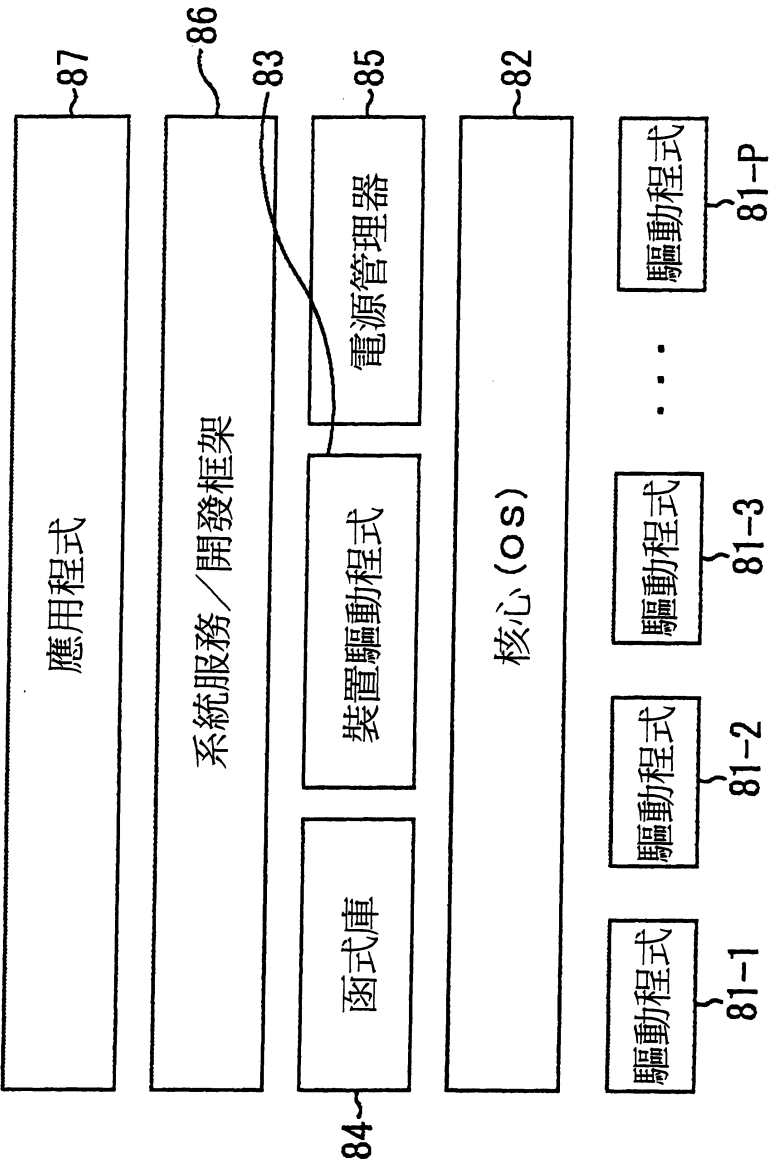


圖8

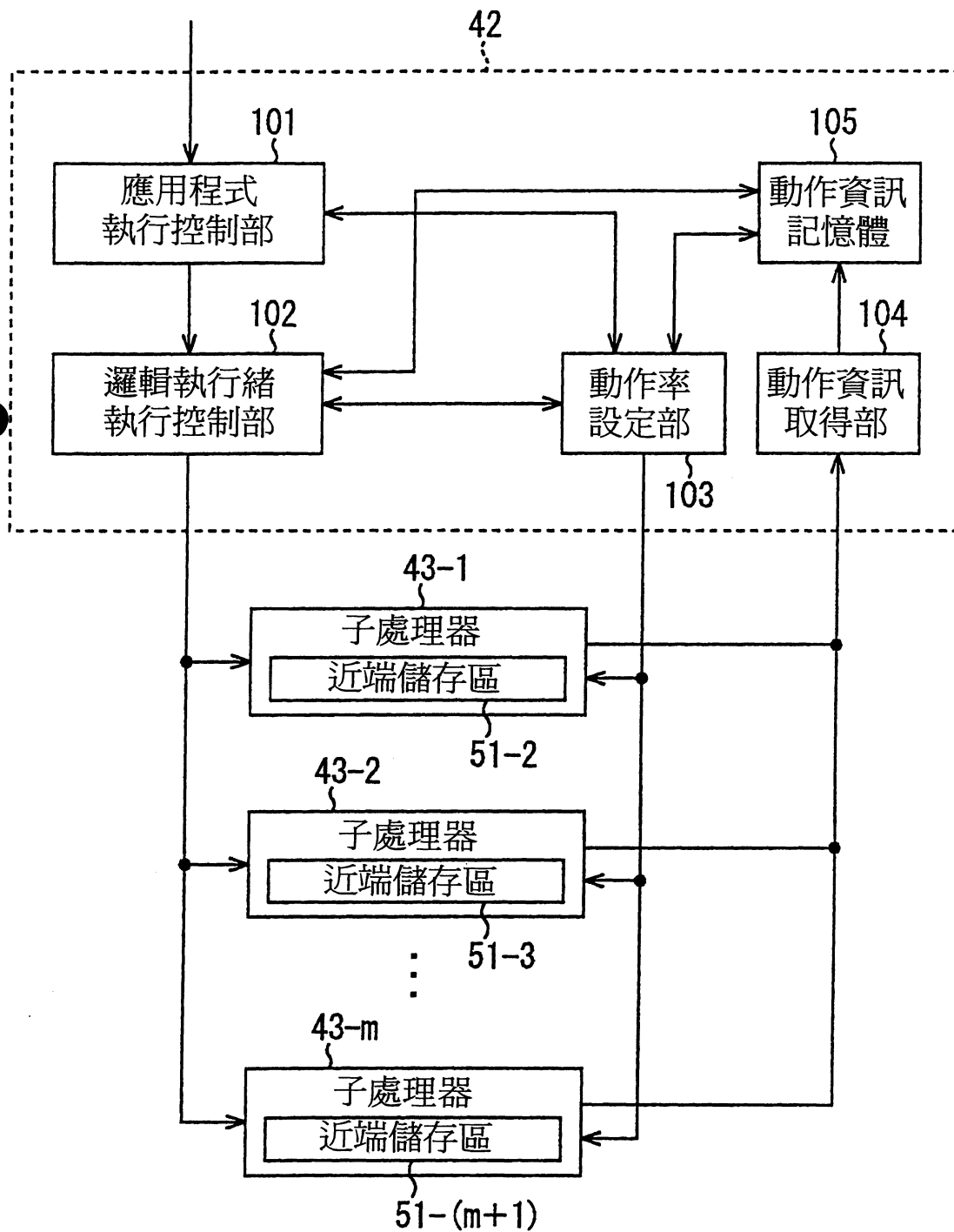


圖9

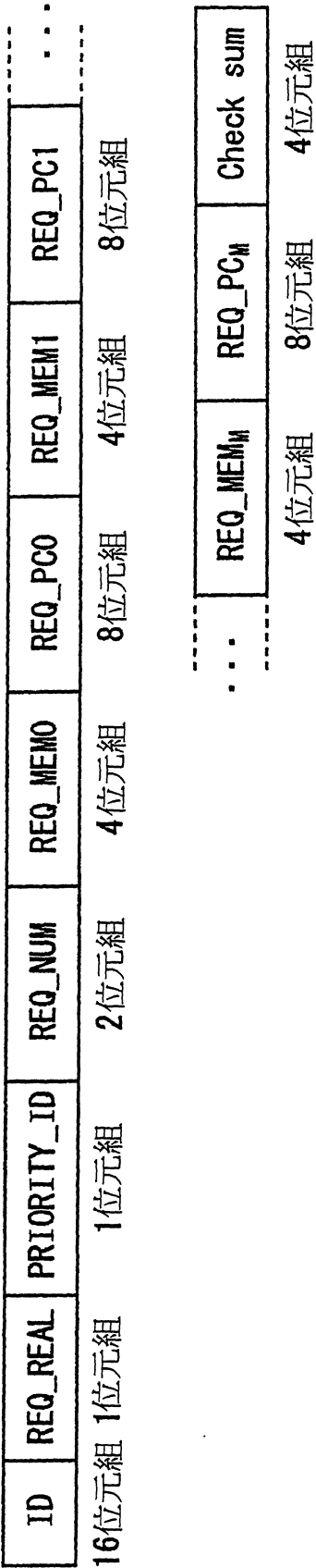


圖10

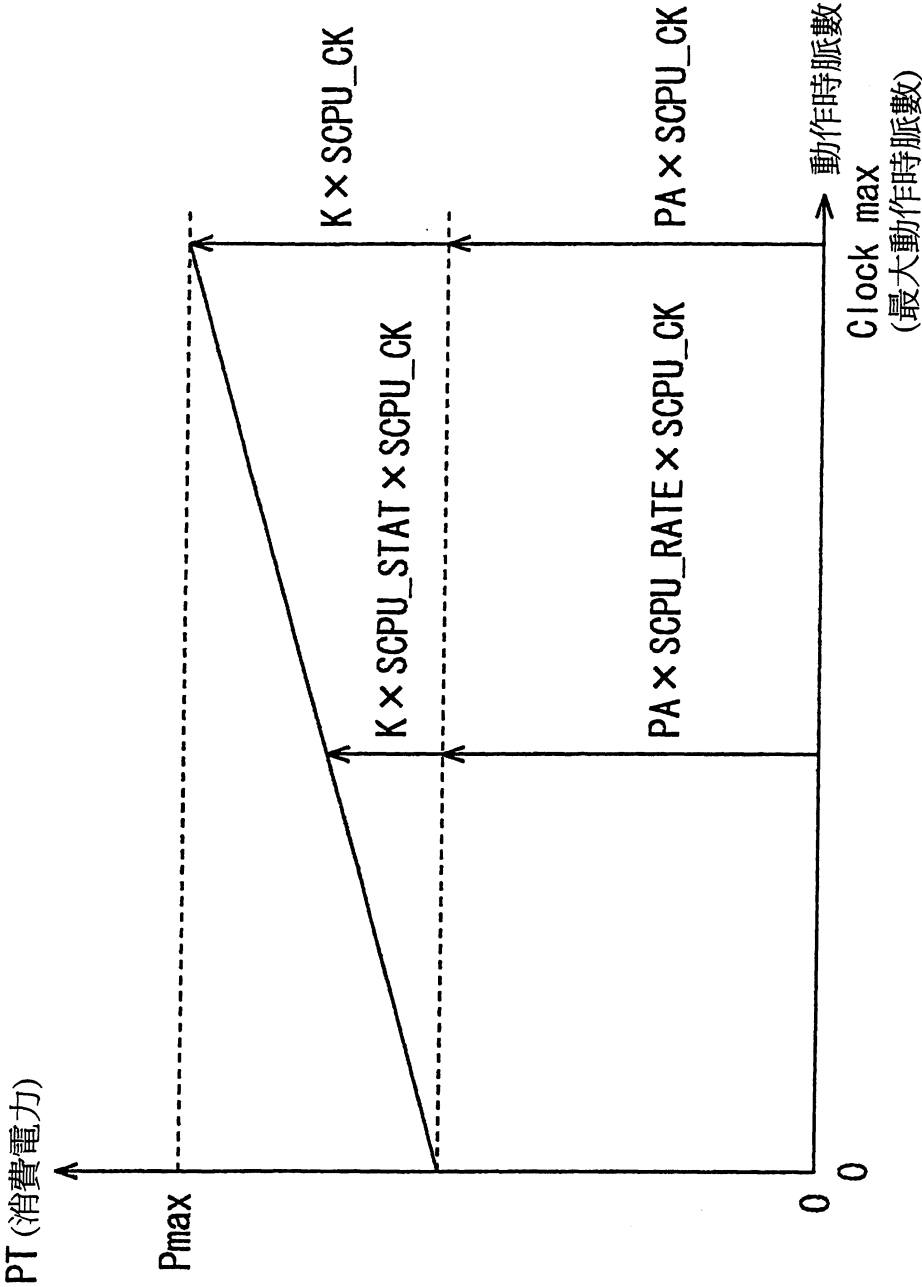


圖 11

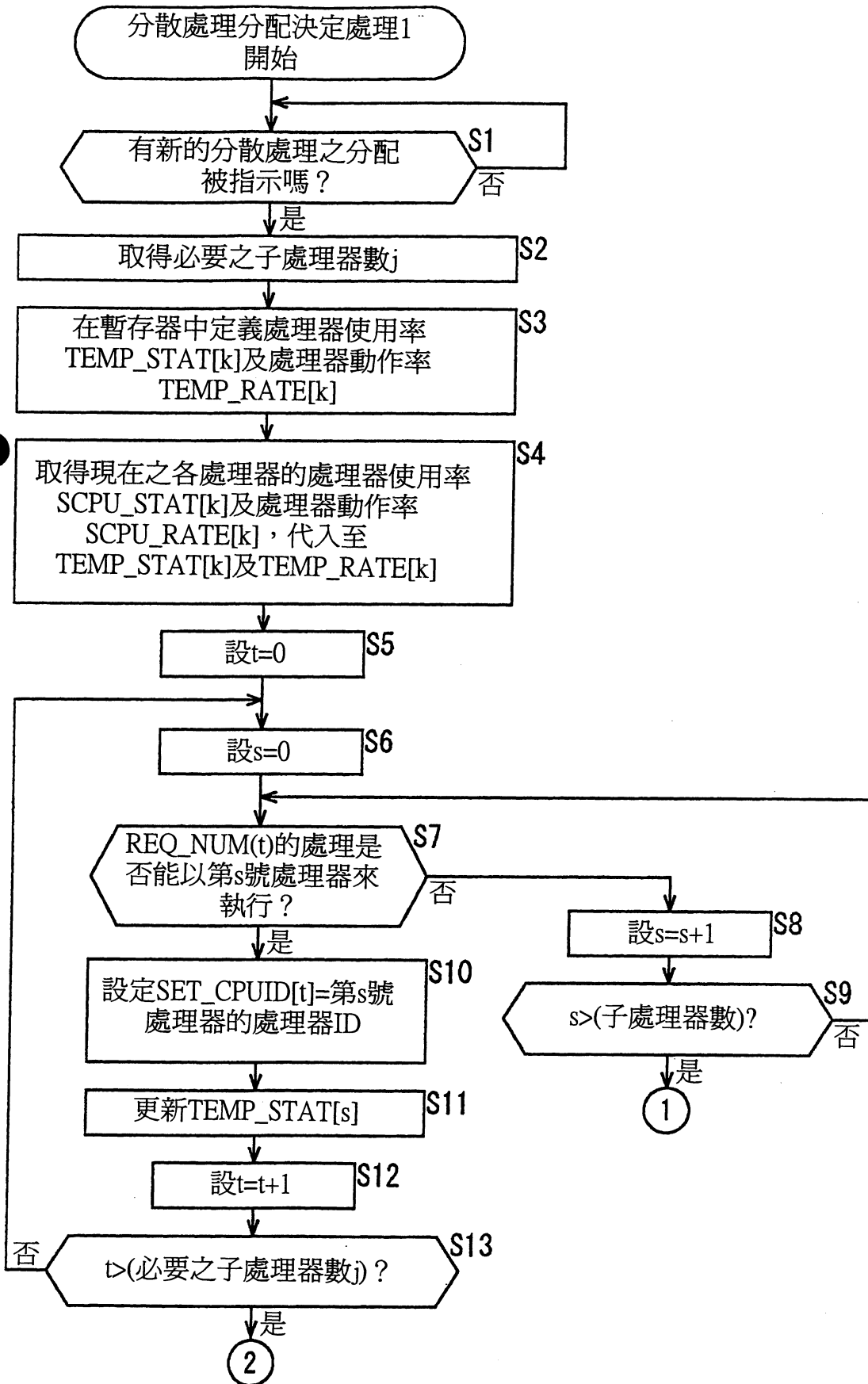


圖12

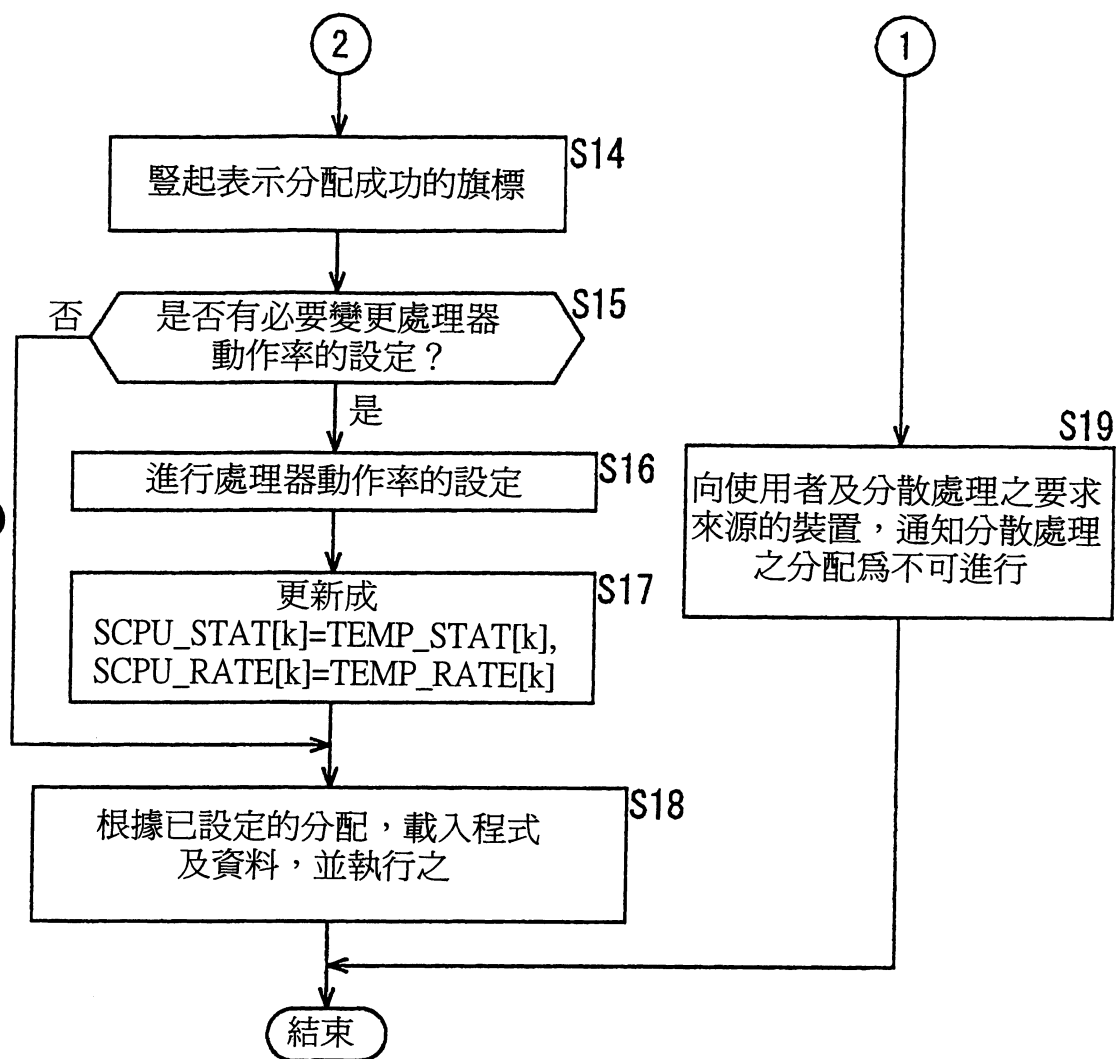


圖13

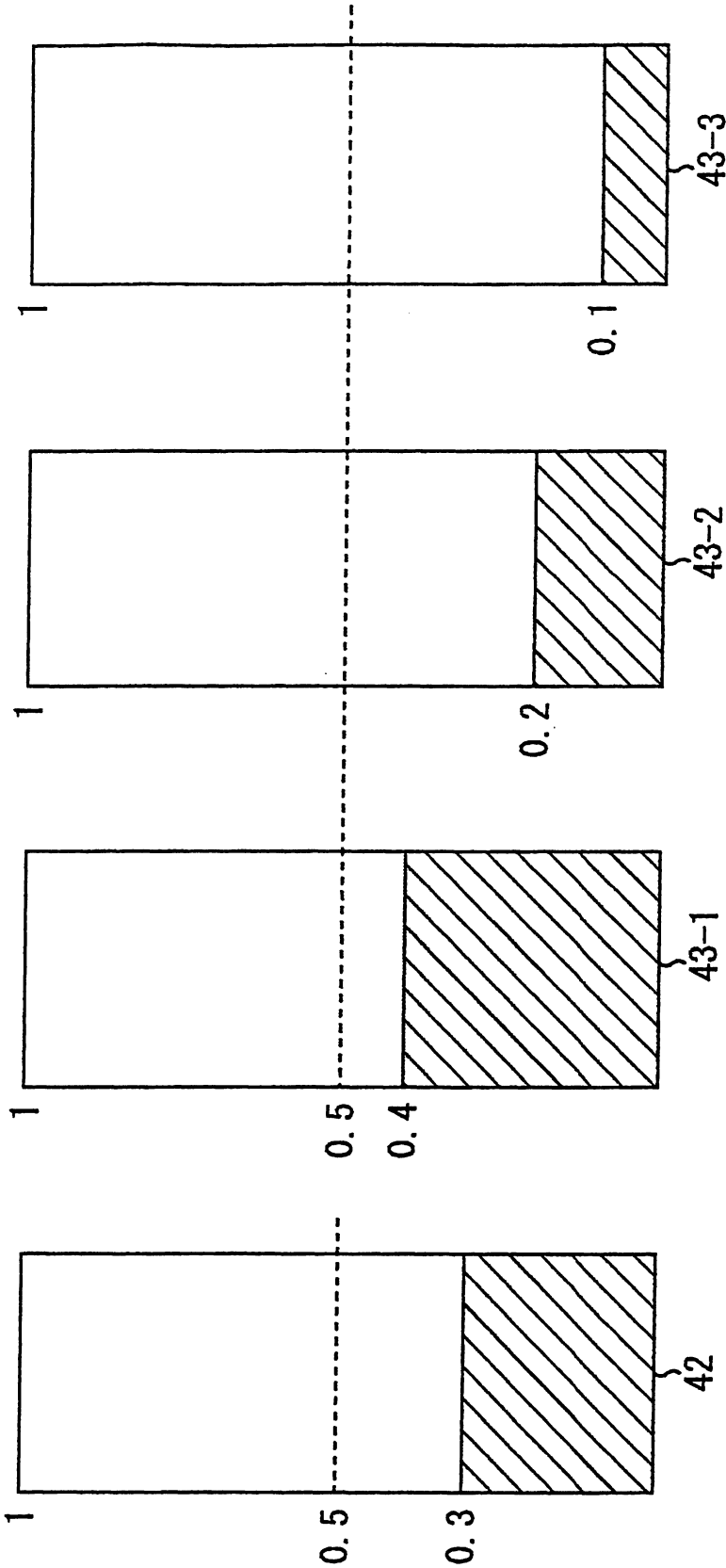


圖14

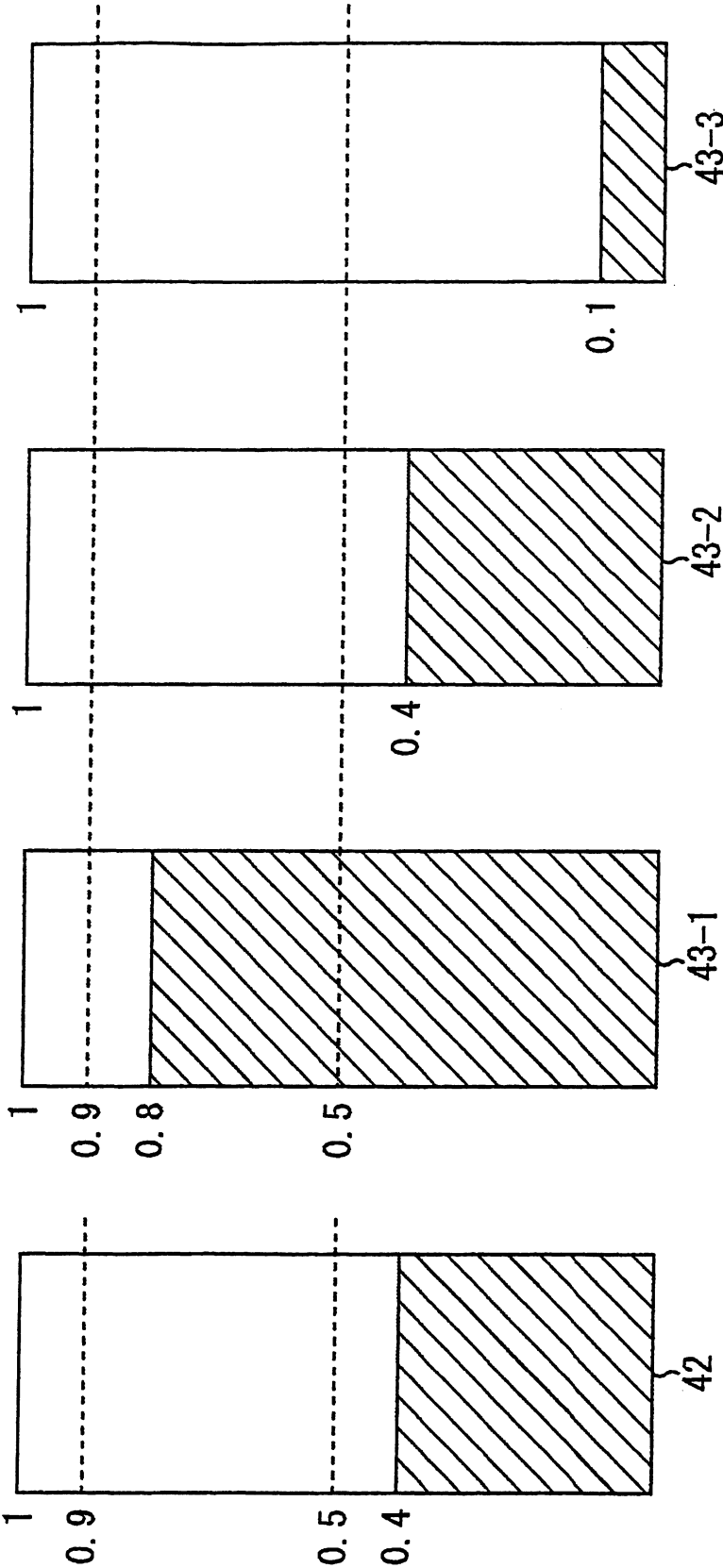


圖 15

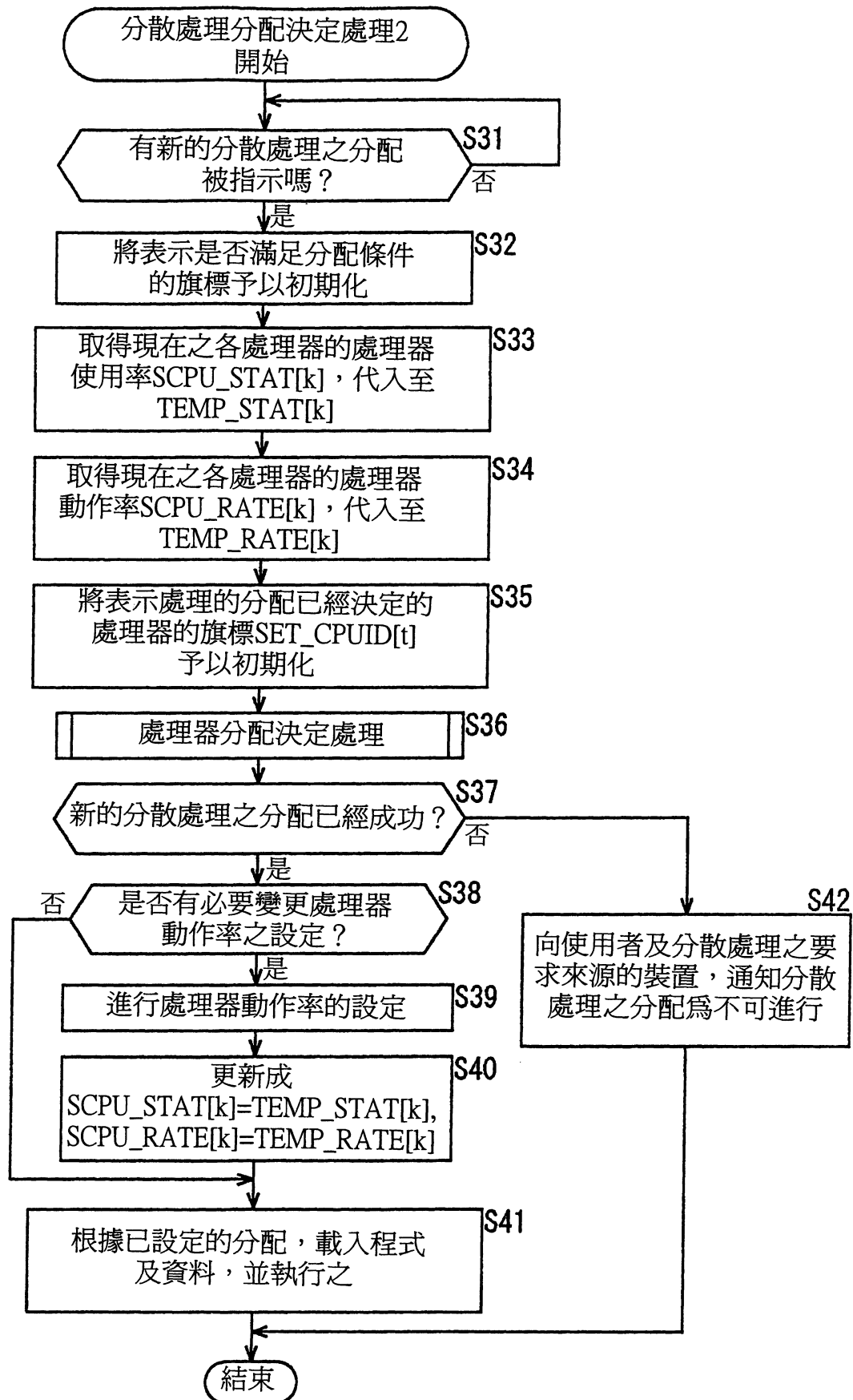


圖 16

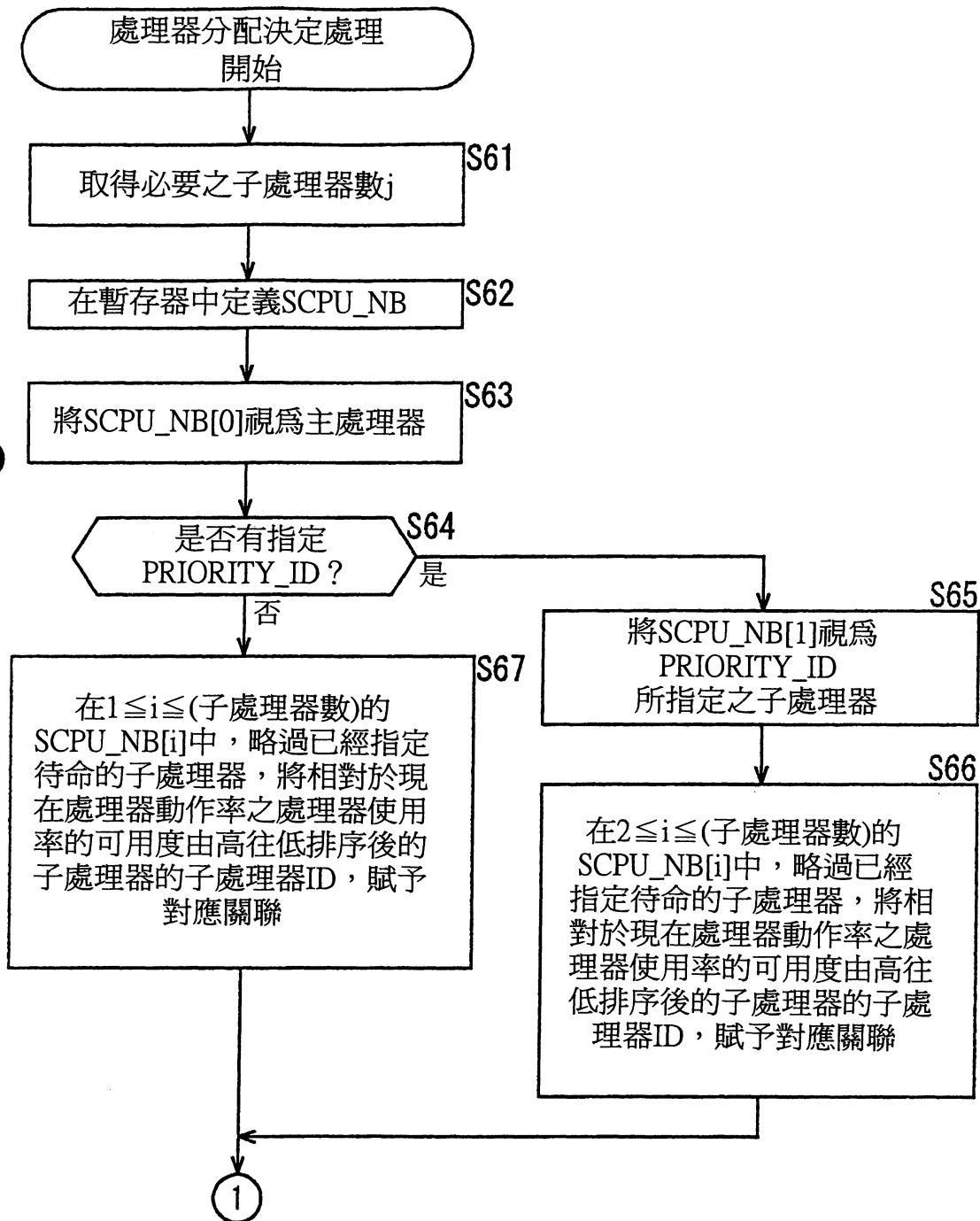


圖 17

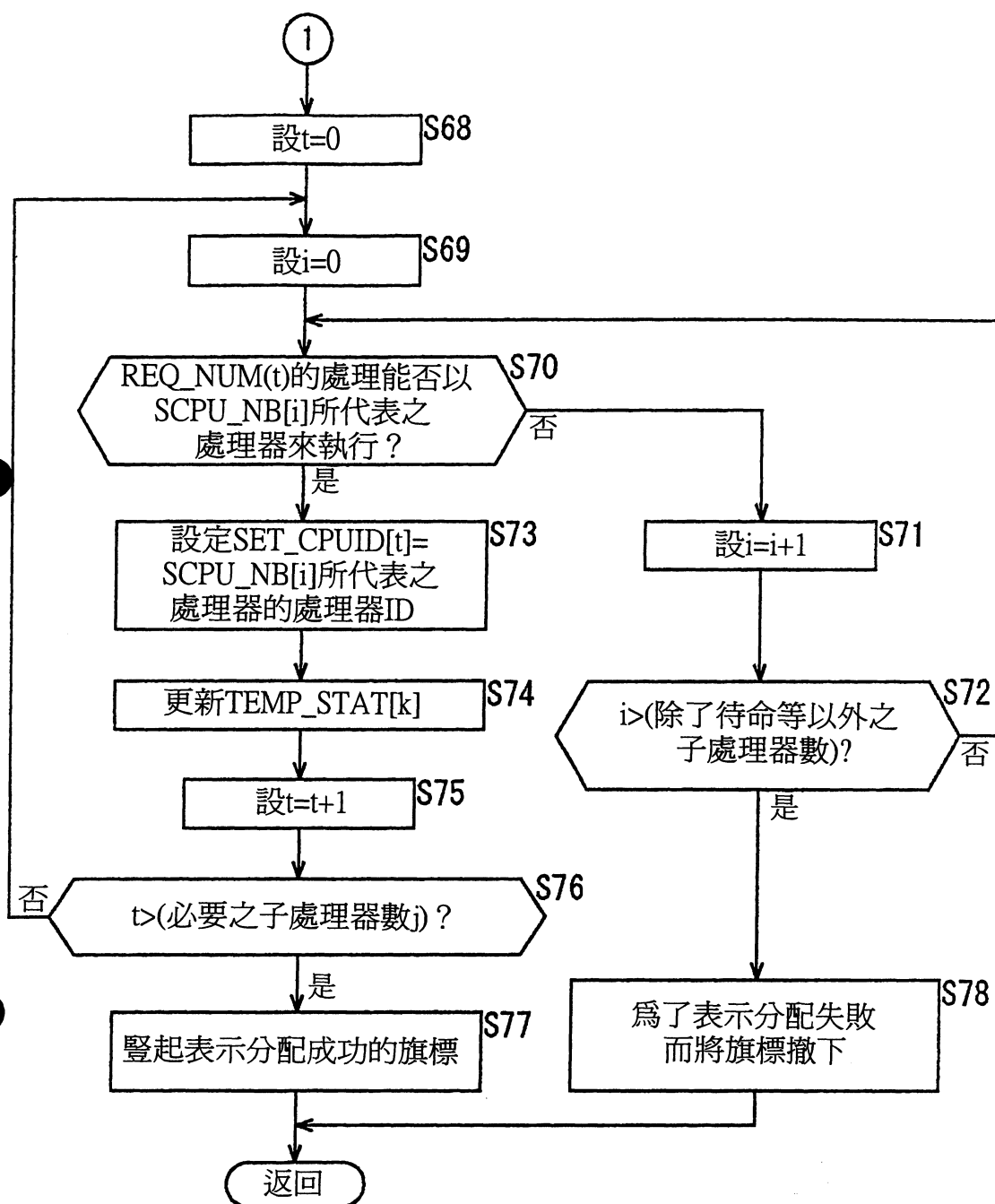


圖18

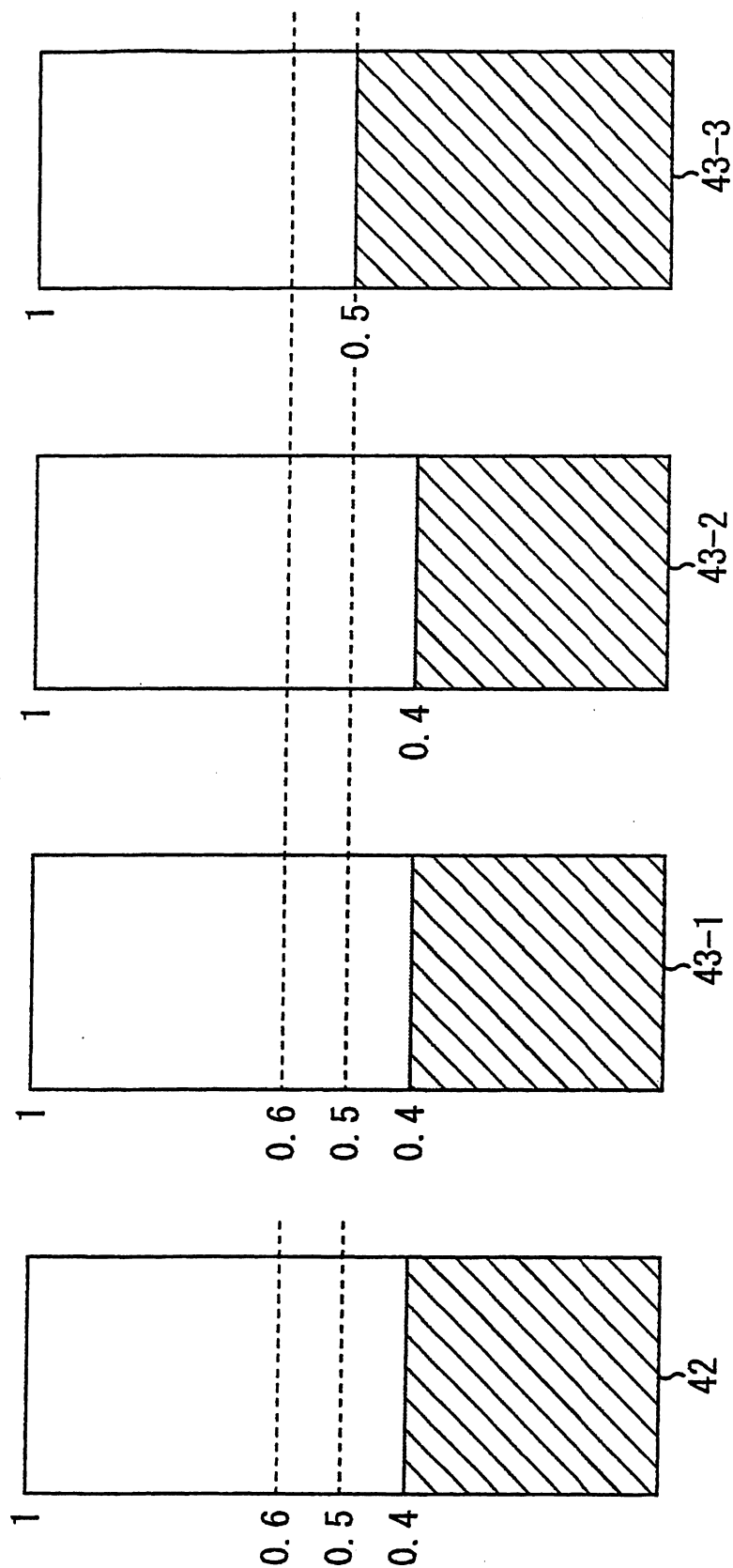


圖19

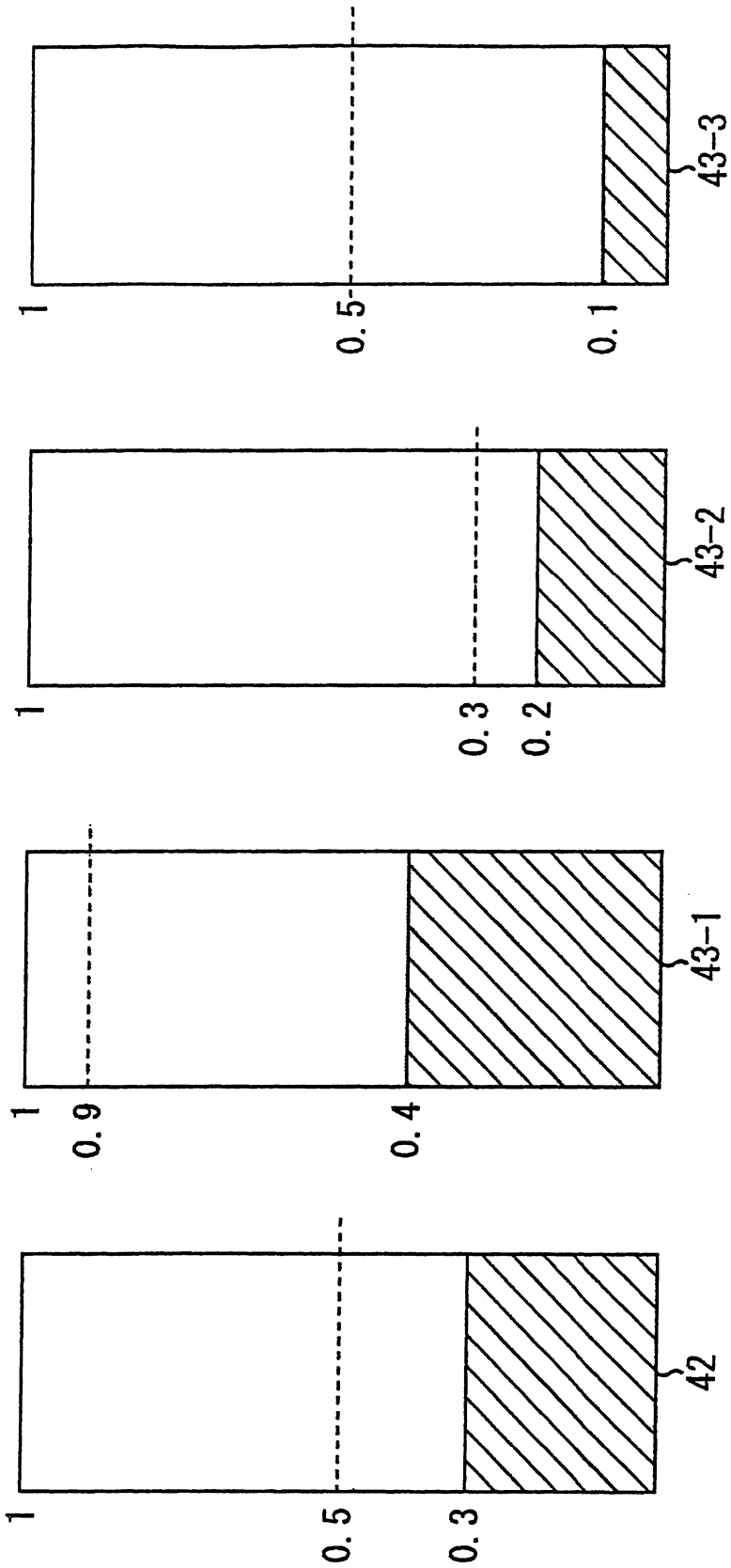
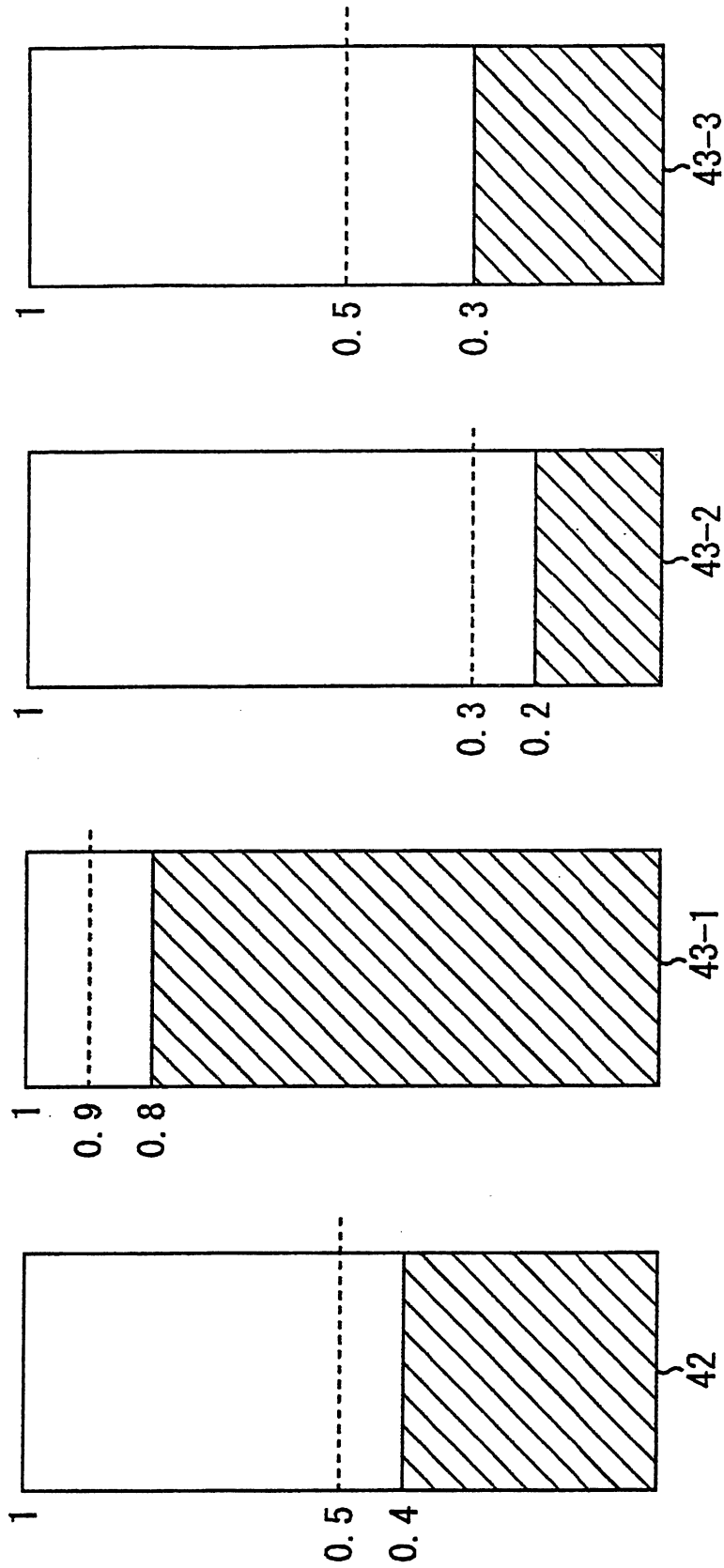


圖20



七、指定代表圖：

(一) 本案指定代表圖為：第 (15) 圖

(二) 本代表圖之元件符號簡單說明：無

八、本案若有化學式時，請揭示最能顯示發明特徵的化學式：無