



(19)中華民國智慧財產局

(12)發明說明書公開本

(11)公開編號：TW 201126414 A1

(43)公開日：中華民國 100 (2011) 年 08 月 01 日

(21)申請案號：099126236

(22)申請日：中華民國 99 (2010) 年 08 月 06 日

(51)Int. Cl. : **G06F9/30 (2006.01)**

(30)優先權：2010/01/22 美國 61/297,505

2010/03/29 美國 12/748,929

(71)申請人：威盛電子股份有限公司 (中華民國) VIA TECHNOLOGIES, INC. (TW)

新北市新店區中正路 535 號 8 樓

(72)發明人：亨利 G 葛蘭 HENRY, G. GLENN (US)；陳巨軒 CHEN, JUI-SHUAN (US)

(74)代理人：洪澄文；顏錦順

申請實體審查：有 申請專利範圍項數：13 項 圖式數：2 共 26 頁

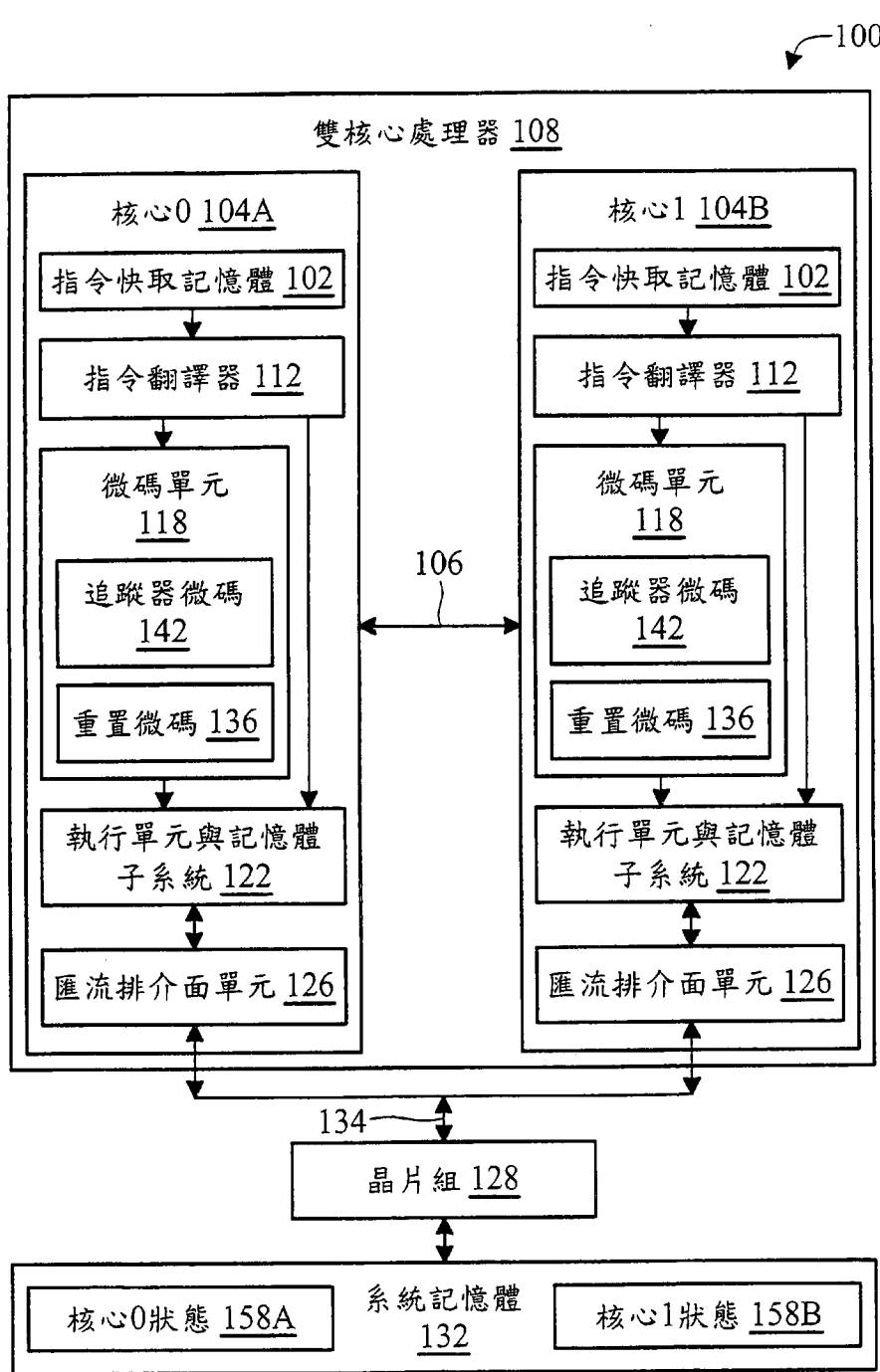
(54)名稱

多核心微處理器及其除錯方法

MULTI-CORE MICROPROCESSOR AND DEBUGGING METHOD THEREOF

(57)摘要

一種多核心微處理器包括第一處理核心、第二處理核心以及耦接於第一處理核心及第二處理核心之匯流排。匯流排於第一處理核心及第二處理核心間傳遞訊息。相應於所偵測到之既定事件，第一處理核心停止執行使用者程式指令，並經由匯流排中斷第二處理核心。相應於被第一處理核心所中斷，第二處理核心停止執行使用者程式指令。每一處理核心在停止執行使用者程式指令之後，輸出其狀態。每一處理核心等待以開始提取並執行使用者程式指令，直到經由匯流排接收到來自另一處理核心的通知，其中另一處理核心已準備開始提取並執行使用者程式指令。



- 100：系統
- 102：指令快取記憶體
- 104A：核心 0
- 104B：核心 1
- 106：核心之間的通信匯流排
- 108：雙核心處理器
- 112：指令翻譯器
- 118：微碼單元
- 122：執行單元與記憶體子系統
- 126：匯流排介面單元
- 128：晶片組
- 132：系統記憶體
- 134：處理器匯流排
- 136：重置微碼
- 142：追蹤器微碼
- 158A：核心 0 狀態
- 158B：核心 1 狀態



(19)中華民國智慧財產局

(12)發明說明書公開本

(11)公開編號：TW 201126414 A1

(43)公開日：中華民國 100 (2011) 年 08 月 01 日

(21)申請案號：099126236

(22)申請日：中華民國 99 (2010) 年 08 月 06 日

(51)Int. Cl. : **G06F9/30 (2006.01)**

(30)優先權：2010/01/22 美國 61/297,505

2010/03/29 美國 12/748,929

(71)申請人：威盛電子股份有限公司 (中華民國) VIA TECHNOLOGIES, INC. (TW)

新北市新店區中正路 535 號 8 樓

(72)發明人：亨利 G 葛蘭 HENRY, G. GLENN (US)；陳巨軒 CHEN, JUI-SHUAN (US)

(74)代理人：洪澄文；顏錦順

申請實體審查：有 申請專利範圍項數：13 項 圖式數：2 共 26 頁

(54)名稱

多核心微處理器及其除錯方法

MULTI-CORE MICROPROCESSOR AND DEBUGGING METHOD THEREOF

(57)摘要

一種多核心微處理器包括第一處理核心、第二處理核心以及耦接於第一處理核心及第二處理核心之匯流排。匯流排於第一處理核心及第二處理核心間傳遞訊息。相應於所偵測到之既定事件，第一處理核心停止執行使用者程式指令，並經由匯流排中斷第二處理核心。相應於被第一處理核心所中斷，第二處理核心停止執行使用者程式指令。每一處理核心在停止執行使用者程式指令之後，輸出其狀態。每一處理核心等待以開始提取並執行使用者程式指令，直到經由匯流排接收到來自另一處理核心的通知，其中另一處理核心已準備開始提取並執行使用者程式指令。

六、發明說明：

【發明所屬之技術領域】

本發明係有關於多核心處理器，特別是有關於多核心處理器的除錯（debug）。

【先前技術】

處理器包含一組微碼常式（microcode routine），其呈現休眠狀態（dormant）直到被軟體啟動而寫入至控制暫存器內（例如 WRMSR 指令）。微碼常式以下稱為追蹤器（tracer），其可當作一種進行處理器之除錯與效能調校（tune）的工具。一旦追蹤器被啟動，則有各種的事件可將追蹤器觸發，使其收集處理器狀態資訊並將處理器狀態資訊寫入至記憶體中的指定位址。使用追蹤器的方法之一係在規則性間隔中調用（invoke）追蹤器。例如，每次當追蹤器執行並引退（retire）N 個指令時（例如 100000 個指令，其數量係由使用者所設定），追蹤器會傳送（dump）處理器的狀態至記憶體。被傳送的處理器狀態以下將稱為檢查點（checkpoint）。在對處理器進行除錯時，工程師可從檢查點獲得處理器狀態，並將處理器狀態輸入至模擬器中進行模擬。

模擬器接收到來自檢查點的處理器狀態，以作為其輸入的一部份。模擬器的輸入為記憶體的狀態以及暫存器的狀態（或是處理器的快取記憶體），其包括處理器所執行的程式。模擬器為一理想處理器的功能性模型。也就是說，模擬器以處理器的初始輸入狀態開始執行，並引退在記憶

體中程式的指令，以便產生符合目標處理器結構（例如 x86 結構）的處理器的正確的輸出狀態。接著，模擬器的輸出狀態會與實際處理器所產生的輸出狀態進行比較，其將有助於對設計上的錯誤進行除錯。上述過程概括地描述如下：

1. 處理器會執行與引退 N 個指令，以及追蹤器會傳送檢查點的狀態至記憶體。
2. 追蹤器會重新開始執行被中止的處理器程序。（在一情況中，追蹤器會重置處理器，而重置微碼會從被傳送至記憶體之狀態檢查點來重新載入處理器的狀態。）
3. 步驟 1 與 2 會持續執行，直到使用者偵測到錯誤發生而停止循環，並將狀態檢查點儲存於一檔案中。
4. 從上述檔案回送第一狀態檢查點至模擬器。
5. 模擬器會執行與引退 N 個指令。
6. 將目前模擬之處理器狀態與下一個狀態檢查點進行比較。如果不符合的話，邏輯設計工程師會使用這些資訊對處理器進行除錯。
7. 否則，從上述檔案回送下一個狀態檢查點至模擬器，並重複步驟 5 與 6。

除了記憶體的覆蓋區（footprint）以及暫存器狀態之外，回送至模擬器的輸入亦包括關於由處理器外部之代理者（agent）所產生之事件出現的資訊。例如，中斷請求會發送至處理器。再者，系統內的其他代理者會對記憶體進行讀取以及寫入，其中記憶體係由處理器以及其他代理者共同使用。其他代理者可以是輸入裝置與輸出裝置或是其

他處理器。這些事件係發生在由不同代理者所共同使用之結構處理器匯流排上，因此可由連接至上述匯流排的邏輯分析儀進行擷取且與在匯流排上傳送至記憶體之狀態檢查點的時間有關聯。

在雙核心處理器的情況中，由一核心所執行的動作可能會影響到另一核心的功能。例如，一核心對記憶體進行存取時會影響到另一核心的操作。尤其是，部分錯誤只會發生在兩核心之間的交互作用（interaction）期間。

在使用模擬器來對雙核心處理器進行除錯的過程中，會偵測到一問題。具體地，實際處理器內的每一核心會部分且獨立地執行追蹤器的停止、傳送以及描述於前面步驟1與2之重新開始等程序。因此，在實際處理器之操作中由兩核心所產生的狀態檢查點並不需要彼此在時間上有關聯。此外，由於追蹤器的停止及重新開始等程序不能協調一致，因此部分核心交互作用相關的錯誤很可能無法被複製。

【發明內容】

本發明提供一種多核心微處理器。上述多核心微處理器包括：一第一處理核心以及一第二處理核心；以及，一匯流排，耦接於上述第一處理核心及上述第二處理核心，用以於上述第一處理核心以及上述第二處理核心之間傳遞訊息。上述第一處理核心以及上述第二處理核心被配置為：相應於所偵測到之一既定事件，上述第一處理核心停止執行使用者程式指令，並經由上述匯流排中斷上述第二

處理核心；相應於被上述第一處理核心所中斷，上述第二處理核心停止執行使用者程式指令；每一上述處理核心在停止執行使用者程式指令之後，輸出其狀態；以及，每一上述處理核心等待以開始提取並執行使用者程式指令，直到經由上述匯流排接收到來自另一上述處理核心的通知，其中另一上述處理核心已準備開始提取並執行使用者程式指令。

再者，本發明提供一種除錯方法，適用於一多核心微處理器，其中上述多核心微處理器包括一第一處理核心、一第二處理核心以及用以於上述第一處理核心以及上述第二處理核心之間傳遞訊息的一匯流排。上述除錯方法包括：藉由上述第一處理核心，偵測一既定事件；相應於所偵測到之上述既定事件，藉由上述第一處理核心停止執行使用者程式指令；經由上述匯流排，藉由上述第一處理核心中斷上述第二處理核心；相應於被上述第一處理核心所中斷，藉由上述第二處理核心停止執行使用者程式指令；在每一上述處理核心停止執行使用者程式指令之後，輸出其狀態；以及，藉由每一上述處理核心，等待以開始提取並執行使用者程式指令，直到經由上述匯流排接收到來自另一上述處理核心的通知，其中另一上述處理核心已準備開始提取並執行使用者程式指令。

【實施方式】

為讓本發明之上述和其他目的、特徵、和優點能更明顯易懂，下文特舉出較佳實施例，並配合所附圖式，作詳

細說明如下：

實施例：

為了解決上述問題，可修改雙核心處理器，使得當一核心的追蹤器觸發時，其能中斷另一處理器以便引起另一處理器內的追蹤器來傳送狀態檢查點。然後，兩核心能彼此進行通信 (communicate)，使得兩核心能同時重新啟動。

參考第 1 圖，第 1 圖係顯示根據本發明一實施例所述之具有雙核心處理器 108 之系統 100 的電路圖。系統 100 包括晶片組 128，其中晶片組 128 係經由處理器匯流排 134 耦接於雙核心處理器 108。系統 100 亦包括系統記憶體 132，其耦接於晶片組 128。雙核心處理器 108 包括兩核心，其可各自稱為核心 0 104A 以及核心 1 104B 或可統稱為核心 104。核心 0 104A 會將其狀態 158A 寫入至系統記憶體 132，而核心 1 104B 會將其狀態 158B 寫入至系統記憶體 132，其中核心 0 104A 的狀態 158A 以及核心 1 104B 的狀態 158B 可統稱為核心狀態 158。核心狀態 158 包含核心 104 之複數暫存器與複數快取記憶體的內容。在一實施例中，由於追蹤器可執行一寫回無效操作以強制快取資料至記憶體，使得記憶體狀態可以被擷取，於是快取記憶體的內容不會包含在核心狀態 158 內。

每一核心 104A、104B 包括指令快取記憶體 102 以及指令翻譯器 112，其中指令翻譯器 112 會將從指令快取記憶體 102 所提取之指令轉譯成微指令 (microinstruction) 以提供至每一核心 104A、104B 內的複數執行單元與記憶

體子系統 122。每一核心 104A、104B 亦包括匯流排介面單元 126，其耦接於執行單元與記憶體子系統 122 並介接（interface）於核心 104 與處理器匯流排 134 之間。

每一核心 104A、104B 亦包括微碼（microcode）單元 118。微碼單元 118 包括微定序器（microsequencer）（未顯示），其可從微碼唯讀記憶體（read only memory，ROM）（未顯示）中提取複數指令。微碼唯讀記憶體係用來儲存微碼指令。特別地，微碼指令包含重置微碼 136 以及追蹤器微碼 142。

在核心 104 被重置後，該核心 104 能立刻開始提取並執行重置微碼 136。重置微碼 136 會初始化核心 104，並在初始化程序的最後部分使得（cause）核心 104 開始從記憶體中提取使用者程式指令。具體而言，在追蹤器微碼 142 開始執行核心 104 的重置程序之前，追蹤器微碼 142 會設定一旗標給重置微碼 136，以指示上述重置程序係由追蹤器微碼 142 執行。上述旗標存在（reside）於非揮發、非使用者可定址之儲存元件。也就是說，在追蹤器執行的重置程序之後上述旗標的值仍會被保留住，然而在電力開啟重置（power-on reset）程序之後上述旗標的值將不會被保留，即上述旗標會相應於電力開啟重置程序而被預設至既定值。此外，當追蹤器微碼 142 傳送最新的檢查點至一記憶體時，追蹤器微碼 142 會儲存在上述記憶體對應位置之記憶體位址至非揮發儲存元件。反之，當重置微碼 136 被執行時，追蹤器微碼 142 會偵測到上述旗標被設定，並從儲存在非揮發儲存元件之對應記憶體位置中載入處理器之最

新傳送的狀態。已復原的狀態 (restored state) 包括程式之指令指標值，其中上述程式係在追蹤器微碼 142 被觸發時所執行。於是，當重置微碼 136 使得 (cause) 核心 104 開始來提取使用者程式指令時，核心 104 將會回復 (resume) 其在最新追蹤器微碼 142 狀態檢查點所停止的執行動作。根據本發明之實施例，重置微碼 136 會等待以使該核心 104 來回復提取使用者程式碼，直到偵測到另一核心 104 亦準備進行相同程序。根據本發明一實施例，匯流排介面單元 126 包括一控制暫存器，其可由追蹤器微碼 142 所程式化，以便請求上述重置程序。

核心 0 104A 以及核心 1 104B 係經由核心之間的通信匯流排 (inter-core communication bus, ICCB) 106 互相傳遞訊息。根據一實施例，核心之間的通信匯流排 106 為串列匯流排，然而其他種類的匯流排亦可被使用。尤其是，當離開重置程序時，核心 104 會經由通信匯流排 106 來進行通信以中斷彼此 (核心 0 104A、核心 1 104B)，而導致追蹤器微碼 142 被觸發並在同時回復提取使用者程式指令。通信匯流排 106 係不同於處理器匯流排 134。處理器匯流排 134 為雙核心處理器 102 的結構處理器匯流排，而通信匯流排 106 為非結構匯流排 (non-architectural bus)。也就是說，使用者程式指令無法指示核心 104 在通信匯流排 106 進行通信，而只有非使用者程式碼 (即在核心 104 內所執行的重置微碼 136、追蹤器微碼 142) 才能指示核心 104 在通信匯流排 106 進行通信。再者，只有在雙核心處理器 108 內的核 104 才能在通信匯流排 106 中進行通

信。也就是說，沒有其他位在雙核心處理器 108 外部的核心或處理器能透過通信匯流排 106 進行通信。通信匯流排 106 係設置在單一封裝內，其中上述單一封裝包括核心 104 以及通信匯流排 106。在一實施例中，雙核心處理器 102 為單一晶粒。在一實施例中，每一核心 104 係設置在不同晶粒中，而通信匯流排 106 會同時耦接於上述不同的晶粒。

參考第 2 圖，第 2 圖係顯示第 1 圖中系統 100 的操作流程圖。流程開始於步驟 202。

在步驟 202，追蹤器微碼 142 被核心 0 104A 所調用。例如，當核心 0 104A 偵測到從最後一個檢查點以來已經有 N 個指令被核心 0 104A 所引退時，核心 0 104A 對應地調用其追蹤器微碼 142。接著，流程進入步驟 204。

在步驟 204，於核心 0 104A 內所執行的追蹤器微碼 142 會經由通信匯流排 106 發送中斷訊息至核心 1 104B，以通知核心 1 104B 必須調用其追蹤器微碼 142。接著，核心 0 104A 會進入步驟 206，而核心 1 104B 會進入步驟 224。

在步驟 206，於核心 0 104A 內所執行的追蹤器微碼 142 會傳送核心 0 104A 的狀態至系統記憶體 132，以作為核心 0 的狀態 158A。接著，流程進入步驟 208。

在步驟 208，於核心 0 104A 內所執行的追蹤器微碼 142 會重置核心 0 104A。如先前所描述，在重置核心 0 104A 之前，追蹤器微碼 142 會設定旗標並將核心 0 的狀態 158A 之位址儲存在非揮發儲存元件。接著，流程進入步驟 212。

在步驟 212，核心 0 104A 被重置並開始執行其重置微碼 136。重置微碼 136 會偵測到旗標已被設定，並對應地

從系統記憶體 132 重新載入核心 0 的狀態 158A 至核心 0 104A，以作為其初始化功能的一部份。接著，流程進入步驟 214。

在步驟 214，於核心 0 104A 內所執行的重置微碼 136 會經由通信匯流排 106 發送訊息至核心 1 104B，以詢問核心 1 104B 是否已完成初始化程序，以及於核心 0 104A 內所執行的重置微碼 136 會準備開始來提取及執行使用者程式指令。接著，流程進入步驟 216。

在步驟 216，於核心 0 104A 內所執行的重置微碼 136 會判斷是否已在通信匯流排 106 上接收到來自核心 1 104B 的訊息，其指示核心 1 104B 已準備開始來提取及執行使用者程式指令。若是，則流程進入步驟 218。否則，流程返回步驟 214。在一實施例中，重置微碼 136 會在步驟 216 的判斷中循環 (loops) 一特定時間，以便在返回步驟 214 之前等待以接收來自核心 1 104B 的準備訊息，以便傳送其他準備訊息。在一實施例中，重置微碼 136 會在假設核心 1 104B 已無反應 (dead) 之前，維持其等待以接收來自核心 1 104B 之準備訊息的循環次數，然後流程進入步驟 218。

在步驟 218，於核心 0 104A 內所執行的重置微碼 136 會依據步驟 212 之核心 0 狀態 158A 所載入之指令指標值，使核心 0 104A 回復提取以及執行使用者程式指令。接著，流程結束於步驟 218。

在步驟 224，核心 1 104B 接收來自核心 0 104A 的中斷訊息，其係在步驟 204 所傳送。相應於所接收的中斷訊息，核心 1 104B 會調用其追蹤器微碼 142。接著，流程進入步

驟 226。

在步驟 226，於核心 1 104B 內所執行的追蹤器微碼 142 會傳送核心 1 104B 的狀態至系統記憶體 132，以作為核心 1 的狀態 158B。接著，流程進入步驟 228。

在步驟 228，於核心 1 104B 內所執行的追蹤器微碼 142 會重置核心 1 104B。如先前所描述，在重置核心 1 104B 之前，追蹤器微碼 142 會設定旗標並將核心 1 的狀態 158B 之位址儲存在非揮發儲存元件。接著，流程進入步驟 232。

在步驟 232，核心 1 104B 被重置並開始執行其重置微碼 136。重置微碼 136 會偵測到旗標已被設定，並對應地從系統記憶體 132 重新載入核心 1 的狀態 158B 至核心 1 104B，以作為其初始化功能的一部份。接著，流程進入步驟 234。

在步驟 234，於核心 1 104B 內所執行的重置微碼 136 會經由通信匯流排 106 發送訊息至核心 0 104A，以詢問核心 0 104A 是否已完成初始化程序，以及於核心 1 104B 內所執行的重置微碼 136 會準備開始來提取及執行使用者程式指令。接著，流程進入步驟 236。

在步驟 236，於核心 1 104B 內所執行的重置微碼 136 會判斷是否已在通信匯流排 106 上接收到來自核心 0 104A 的訊息，其指示核心 0 104A 已準備開始來提取及執行使用者程式指令。若是，則流程進入步驟 238。否則，流程返回步驟 234。在一實施例中，重置微碼 136 會在步驟 236 的判斷中循環一特定時間，以便在返回步驟 234 之前等待以接收來自核心 0 104A 的準備訊息，以便傳送其他準備訊

息。在一實施例中，重置微碼 136 會在假設核心 0 104A 已無反應之前，維持其等待以接收來自核心 0 104A 之準備訊息的循環次數，然後流程進入步驟 238。

在步驟 238，於核心 1 104B 內所執行的重置微碼 136 會依據步驟 232 之核心 1 的狀態 158B 所載入之指令指標值，使核心 1 104B 回復提取以及執行使用者程式指令。接著，流程結束於步驟 238。

因此，從第 2 圖可知，兩核心 104 能在相同時間傳送其狀態檢查點至記憶體，並同時回復使用者程式指令的執行。在一實施例中，兩核心 104 能達到在彼此的一處理器匯流排時脈週期內實現回復執行程序。

雖然雙核心處理器 108 係描述具有兩核心之處理器，然而具有多於兩核心之多核心處理器 108 的其他實施例亦可被考慮，其中多核心處理器 108 的每一核心能與其他核心進行通信，以決定是否全部的核心已準備好進行離開重置程序，使得全部的核心能離開重置程序並能同時開始提取使用者程式碼。

雖然本發明已以較佳實施例揭露如上，然其並非用以限定本發明，任何所屬技術領域中具有通常知識者，在不脫離本發明之精神和範圍內，當可作些許之更動與潤飾。例如，軟體可致能如本發明所述之裝置及方法的功能、製造、模型化、模擬、描述及/或測試，其能透過使用通用程式設計語言（例如 C、C++）、包括 Verilog、HDL、VHDL 等之硬體描述語言（HDL）或其他可用的程式來實現。上述軟體可設置在任何已知的電腦可用媒體中，例如磁帶、

半導體、磁碟、光碟（如 CD-ROM、DVD-ROM 等）、網路、有線連線、無線或其他通訊媒體。本發明之裝置及方法的實施例可包含在半導體智慧財產權核心內，例如微處理器核心（由 HDL 具體化），並轉換成積體電路的硬體產品。此外，本發明實施例所述之裝置及方法可以硬體與軟體之結合方式具體化。因此，本發明不應限定於已揭露的實施例，而應視後附之申請專利範圍所界定者為準。具體而言，本發明可實施於微處理器裝置中，其可被使用於通用電腦。最後，任何熟悉此項技藝者，可基於本發明所揭露的概念以及特定實施例，在不脫離本發明之精神和範圍內，可做些許更動與潤飾以達到本發明之相同目的。

【圖式簡單說明】

第 1 圖係顯示根據本發明一實施例所述之具有雙核心處理器之系統的電路圖；以及

第 2 圖係顯示第 1 圖中系統的操作流程圖。

【主要元件符號說明】

100～系統；

102～指令快取記憶體；

104A～核心 0；

104B～核心 1；

106～核心之間的通信匯流排；

108～雙核心處理器；

112～指令翻譯器；

- 118～微碼單元；
- 122～執行單元與記憶體子系統；
- 126～匯流排介面單元；
- 128～晶片組；
- 132～系統記憶體；
- 134～處理器匯流排；
- 136～重置微碼；
- 142～追蹤器微碼；
- 158A～核心 0 狀態；
- 158B～核心 1 狀態；以及
- 202-238～步驟。

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：99/26236

※申請日：

99.8.06

※IPC分類：

G06F9/30 (2006.01)

一、發明名稱：(中文/英文)

多核心微處理器及其除錯方法

MULTI-CORE MICROPROCESSOR AND
DEBUGGING METHOD THEREOF

二、中文發明摘要：

一種多核心微處理器包括第一處理核心、第二處理核心以及耦接於第一處理核心及第二處理核心之匯流排。匯流排於第一處理核心及第二處理核心間傳遞訊息。相應於所偵測到之既定事件，第一處理核心停止執行使用者程式指令，並經由匯流排中斷第二處理核心。相應於被第一處理核心所中斷，第二處理核心停止執行使用者程式指令。每一處理核心在停止執行使用者程式指令之後，輸出其狀態。每一處理核心等待以開始提取並執行使用者程式指令，直到經由匯流排接收到來自另一處理核心的通知，其中另一處理核心已準備開始提取並執行使用者程式指令。

三、英文發明摘要：

A multi-core microprocessor is provided. The multi-core microprocessor includes first and second processing cores and a bus coupled to the first and second

processing cores. The bus conveys messages between the first and second processing cores. The first processing core stops executing user instructions and interrupts the second processing core via the bus, in response to detecting a predetermined event. The second processing core stops executing user instructions, in response to being interrupted by the first processing core. Each processing core outputs its state after it stops executing user instructions. Each processing core waits to being fetching and executing user instructions until it receives a notification from the other processing core via the bus that the other processing core is ready to being fetching and executing user instructions.

七、申請專利範圍：

1.一種多核心微處理器，包括：

一第一處理核心以及一第二處理核心；以及

一匯流排，耦接於上述第一處理核心及上述第二處理核心，用以於上述第一處理核心以及上述第二處理核心之間傳遞訊息；

其中上述第一處理核心以及上述第二處理核心被配置為：

相應於所偵測到之一既定事件，上述第一處理核心停止執行使用者程式指令，並經由上述匯流排中斷上述第二處理核心；

相應於被上述第一處理核心所中斷，上述第二處理核心停止執行使用者程式指令；

每一上述處理核心在停止執行使用者程式指令之後，輸出其狀態；以及

每一上述處理核心等待以開始提取並執行使用者程式指令，直到經由上述匯流排接收到來自另一上述處理核心的通知，其中另一上述處理核心已準備開始提取並執行使用者程式指令。

2.如申請專利範圍第1項所述之多核心微處理器，其中上述既定事件包括以下之一者：

偵測到上述第一處理核心已引退一既定數量之指令；以及

由上述第一處理核心調用一追蹤器微碼。

3.如申請專利範圍第1項所述之多核心微處理器，其中

上述第一處理核心以及上述第二處理核心更被配置為：

每一上述處理核心在輸出其狀態之後進行重置，其中每一上述處理核心在被重置之後，重新載入其狀態。

4.如申請專利範圍第3項所述之多核心微處理器，其中上述第一處理核心以及上述第二處理核心被配置，使得每一上述處理核心輸出其狀態至上述多核心微處理器外部的一記憶體，以及在被重置之後每一上述處理核心從上述記憶體重新載入其狀態。

5.如申請專利範圍第1項所述之多核心微處理器，其中上述第一處理核心以及上述第二處理核心更被配置為：

每一上述處理核心經由上述匯流排詢問另一上述處理核心，以決定另一上述處理核心是否已準備開始提取並執行使用者程式指令。

6.如申請專利範圍第1項所述之多核心微處理器，其中每一上述處理核心內的一微碼單元被配置為等待以開始提取並執行使用者程式指令，直到經由上述匯流排接收到來自另一上述處理核心的通知，其中另一上述處理核心已準備開始提取並執行使用者程式指令。

7.如申請專利範圍第1項所述之多核心微處理器，其中每一上述處理核心內的一微碼單元被配置為輸出其狀態。

8.一種除錯方法，適用於一多核心微處理器，其中上述多核心微處理器包括一第一處理核心、一第二處理核心以及用於上述第一處理核心以及上述第二處理核心之間傳遞訊息的一匯流排，上述除錯方法包括：

藉由上述第一處理核心，偵測一既定事件；

相應於所偵測到之上述既定事件，藉由上述第一處理核心停止執行使用者程式指令；

經由上述匯流排，藉由上述第一處理核心中斷上述第二處理核心；

相應於被上述第一處理核心所中斷，藉由上述第二處理核心停止執行使用者程式指令；

在每一上述處理核心停止執行使用者程式指令之後，輸出其狀態；以及

藉由每一上述處理核心，等待以開始提取並執行使用者程式指令，直到經由上述匯流排接收到來自另一上述處理核心的通知，其中另一上述處理核心已準備開始提取並執行使用者程式指令。

9.如申請專利範圍第 8 項所述之除錯方法，其中上述既定事件包括以下之一者：

偵測到上述第一處理核心已引退一既定數量之指令；以及

由上述第一處理核心調用一追蹤器微碼。

10.如申請專利範圍第 8 項所述之除錯方法，更包括：在每一上述處理核心輸出其狀態之後進行重置，其中在每一上述處理核心被重置之後，重新載入其狀態。

11.如申請專利範圍第 8 項所述之除錯方法，更包括：藉由每一上述處理核心，經由上述匯流排詢問另一上述處理核心，以決定另一上述處理核心是否已準備開始提取並執行使用者程式指令。

12.如申請專利範圍第 8 項所述之除錯方法，其中每一

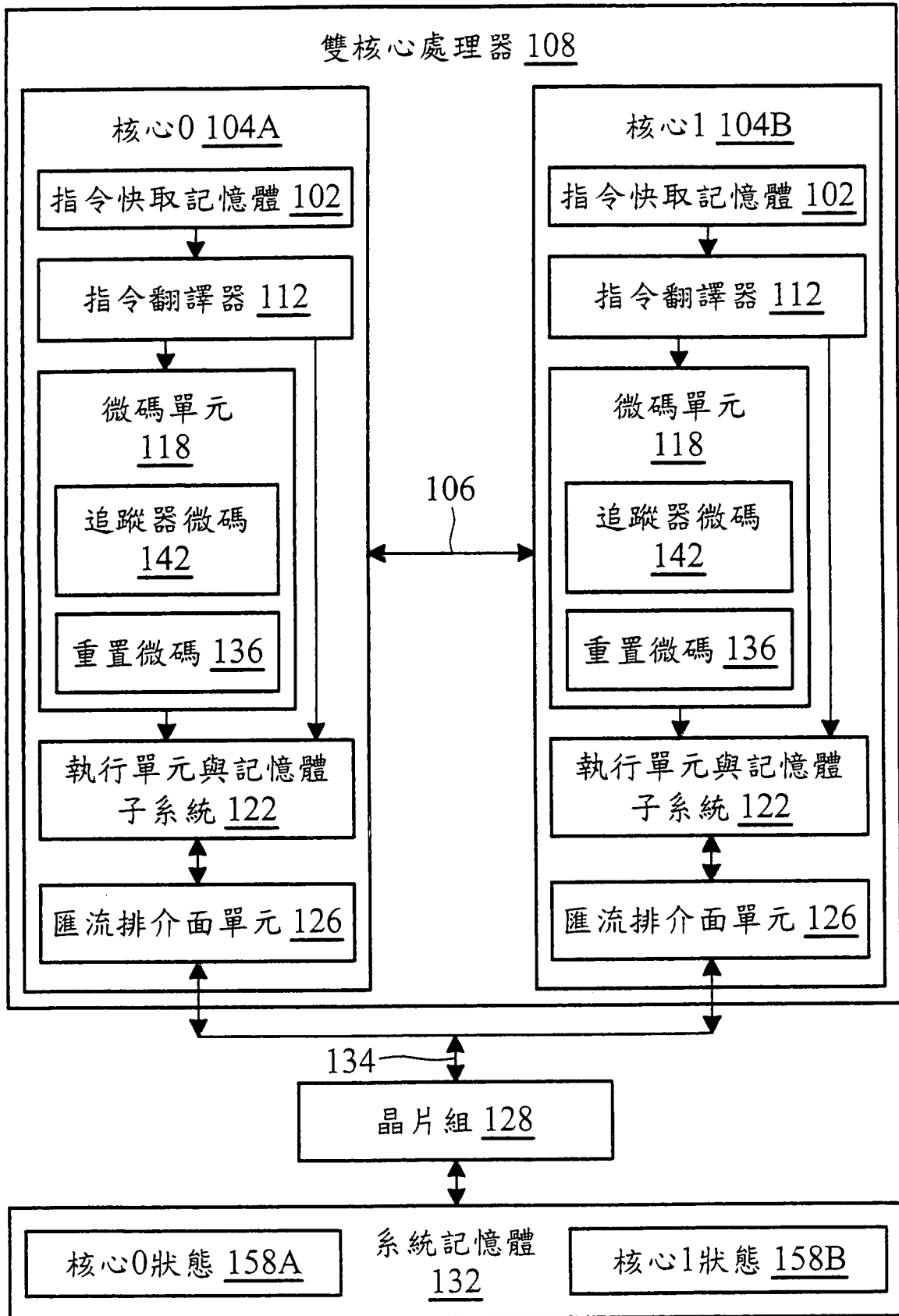
上述處理核心內的一微碼單元被配置為執行上述等待以開始提取並執行使用者程式指令之步驟，直到經由上述匯流排接收到來自另一上述處理核心的通知，其中另一上述處理核心已準備開始提取並執行使用者程式指令。

13.如申請專利範圍第 8 項所述之除錯方法，其中每一上述處理核心內的一微碼單元被配置為執行輸出其狀態。

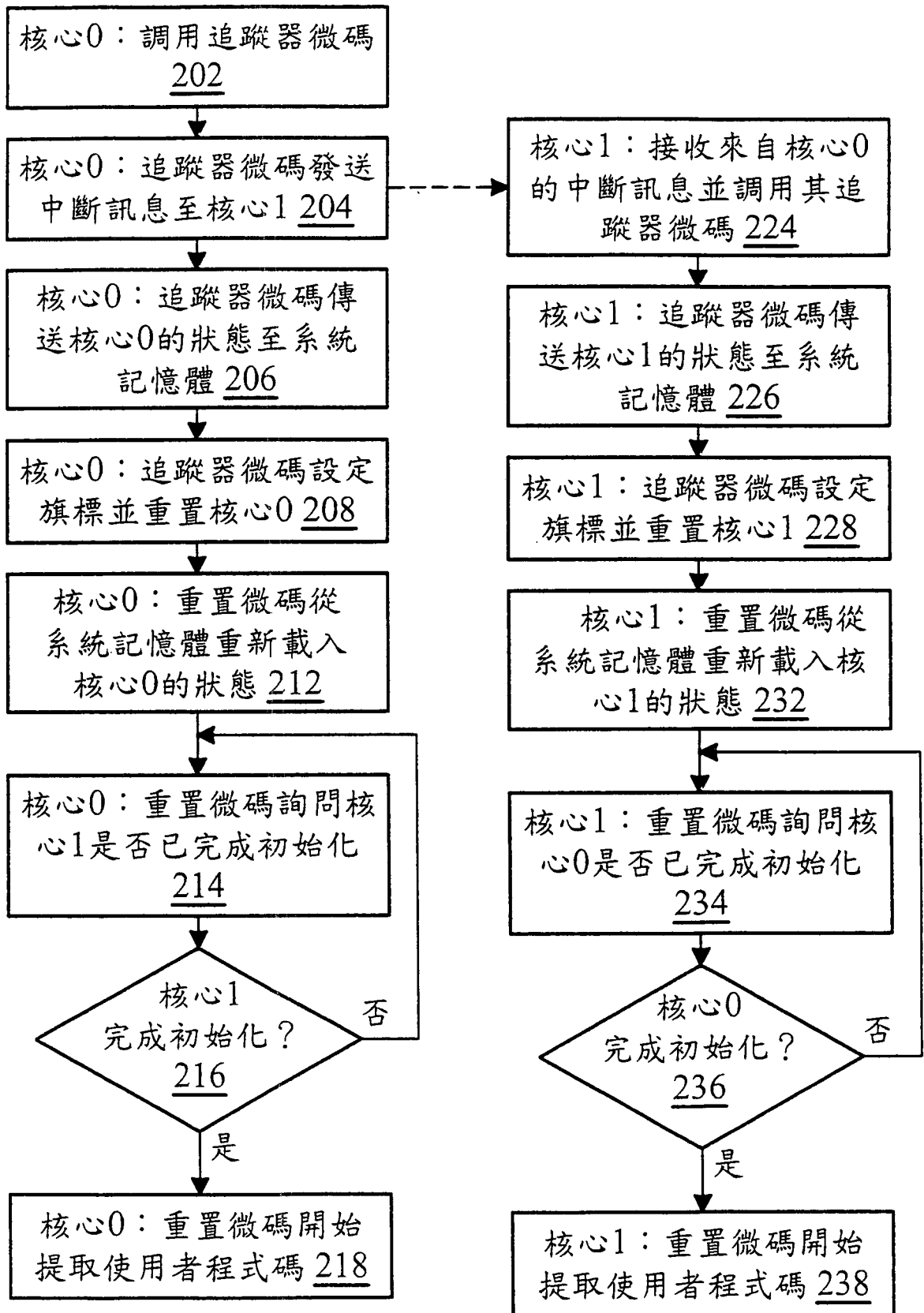
201126414

八、圖式：





第 1 圖



第 2 圖

四、指定代表圖：

(一)本案指定代表圖為：第(1)圖。

(二)本代表圖之元件符號簡單說明：

100～系統

102～指令快取記憶體

104A～核心 0

104B～核心 1

106～核心之間的通信匯流排

108～雙核心處理器

112～指令翻譯器

118～微碼單元

122～執行單元與記憶體子系統

126～匯流排介面單元

128～晶片組

132～系統記憶體

134～處理器匯流排

136～重置微碼

142～追蹤器微碼

158A～核心 0 狀態

158B～核心 1 狀態

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

略