



(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 I512498 B

(45)公告日：中華民國 104 (2015) 年 12 月 11 日

(21)申請案號：101102835 (22)申請日：中華民國 101 (2012) 年 01 月 30 日

(51)Int. Cl. : G06F15/82 (2006.01) G06F9/45 (2006.01)

(30)優先權：2011/01/27 美國 61/436,966

(71)申請人：軟體機器公司 (美國) SOFT MACHINES, INC. (US)
美國

(72)發明人：艾伯戴爾拉 摩翰麥德 ABDALLAH, MOHAMMAD (US)

(74)代理人：李宗德

(56)參考文獻：

TW I275102

TW I294086

US 6142682

US 7203932B1

審查人員：何旭智

申請專利範圍項數：22 項 圖式數：18 共 69 頁

(54)名稱

用於將客戶指令轉譯為本機指令之硬體加速組件

HARDWARE ACCELERATION COMPONENTS FOR TRANSLATING GUEST INSTRUCTIONS TO
NATIVE INSTRUCTIONS

(57)摘要

本發明揭示一種硬體為基轉譯加速器。該硬體包括：一客戶提取邏輯組件，用於存取客戶指令；一客戶提取緩衝器，其耦接至該客戶提取邏輯組件及一分支預測組件，用於將客戶指令組合成一客戶指令區塊；及轉換表，其耦接至該客戶提取緩衝器，用於將該客戶指令區塊轉譯為一對應的本機轉換區塊。該硬體另外包括：一本機快取，其耦接至該等轉換表，用於儲存該對應的本機轉換區塊；及一轉換後備緩衝器，其耦接至該本機快取，用於儲存該客戶指令區塊至對應的本機轉換區塊的一映射，其中在收到針對客戶指令的後續請求時，索引該轉換後備緩衝器以決定是否發生命中，其中該映射指示該客戶指令在該本機快取中具有一對應的轉換本機指令。

A hardware based translation accelerator. The hardware includes a guest fetch logic component for accessing guest instructions; a guest fetch buffer coupled to the guest fetch logic component and a branch prediction component for assembling guest instructions into a guest instruction block; and conversion tables coupled to the guest fetch buffer for translating the guest instruction block into a corresponding native conversion block. The hardware further includes a native cache coupled to the conversion tables for storing the corresponding native conversion block, and a conversion look aside buffer coupled to the native cache for storing a mapping of the guest instruction block to corresponding native conversion block, wherein upon a subsequent request for a guest instruction, the conversion look aside buffer is indexed to determine whether a hit occurred, wherein the mapping indicates the guest instruction has a corresponding converted native instruction in the native cache.

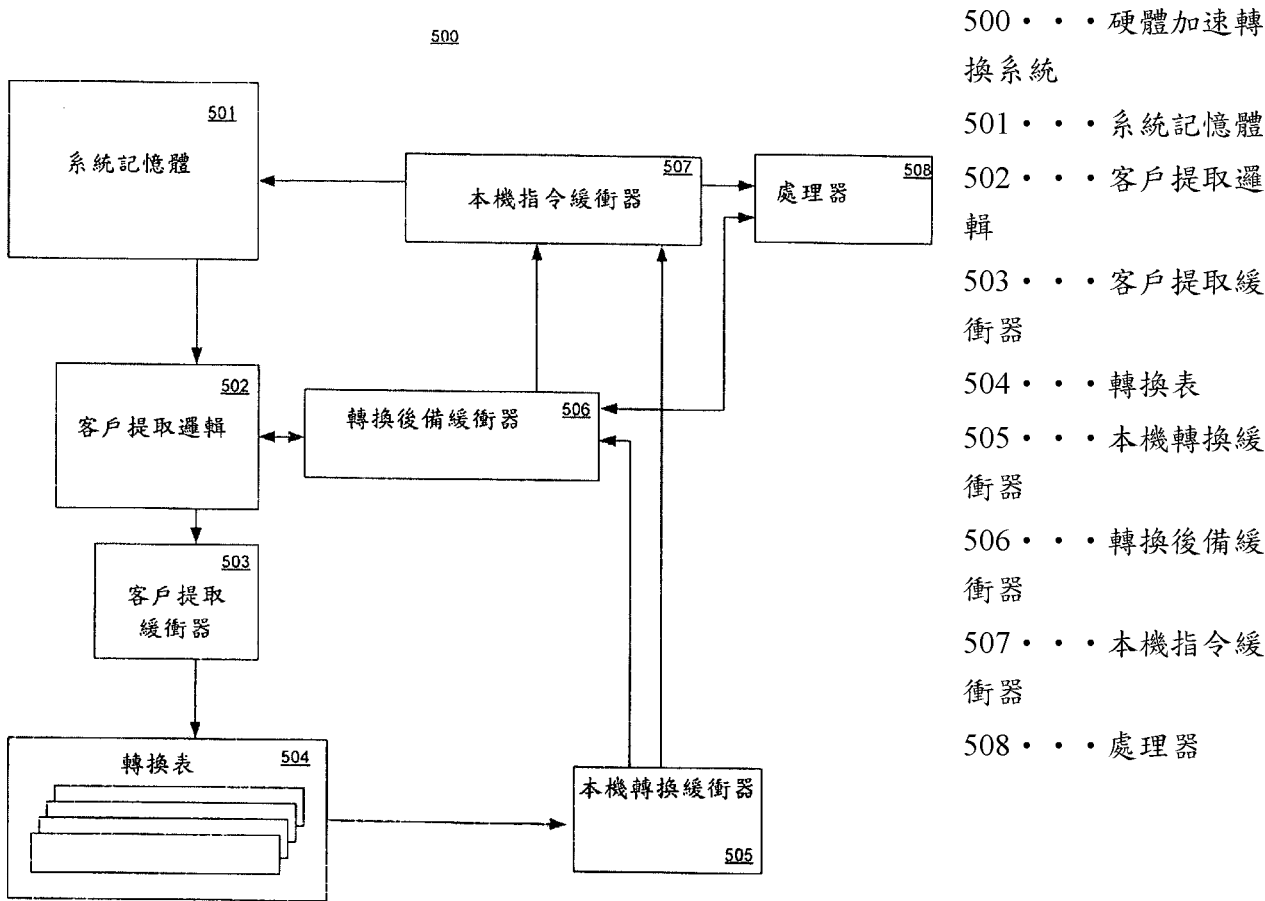


圖5

六、發明說明：

【相關申請案參照】

本申請案主張 2011 年 1 月 27 日由 Mohammad A. Abdallah 申請之同在申請中及共同讓與的美國臨時專利申請案第 61/436,966 號的權利，其標題為「用於將客戶指令轉譯為本機指令之硬體加速組件」，及其全文以引用的方式併入本文中。

【發明所屬之技術領域】

本發明一般有關數位電腦系統，尤其有關一種轉譯包含指令序列(instruction sequence)之指令的系統及方法。

【先前技術】

許多類型的數位電腦系統利用程式碼(code)變換(transformation)/轉譯(translation)或仿擬(emulation)，實施軟體功能性。一般而言，即使指令不在電腦系統的「本機」內，轉譯及仿擬均涉及檢查軟體指令的程式及履行軟體指令指定的功能及動作。在轉譯的情況中，非本機指令被轉譯為本機指令的形式，本機指令係設計可在電腦系統的硬體上執行。實例包括先前技術的轉譯軟體及/或硬體，其以工業標準 x86 應用程式操作，使這些應用程式可在非 x86 或替代的電腦架構上執行。一般而言，轉譯程序(translation process)利用大量的處理器週期(processor cycle)，因而造成不小的額外負擔。額外負擔所造成的效能損失可實質上抹滅轉譯程序所帶來的任何好處。

一個試圖解決此問題的做法涉及使用及時編譯(just-in-time compilation)。及時(JIT)編譯又稱為動態轉譯，是一種提高電腦

程式之運行時間效能(runtime performance)的方法。傳統上，電腦程式具有兩個運行時間變換模式：解譯模式(interpretation mode)或 JIT(及時)編譯/轉譯模式。解譯是一種解碼程序(decoding process)，其涉及一個指令一個指令地解碼，以低於 JIT 編譯的額外負擔將客戶碼變換為本機碼，但其產生表現較差的變換碼。另外，解譯係以每一個指令來調用(invoked)。JIT 編譯器或轉譯器代表截然不同的解譯途徑。就 JIT 轉換而言，其通常具有比解譯器高的額外負擔，但其產生更為最佳化的轉譯碼(translated code)及具有較高執行效能的轉譯碼。在多數的仿擬實施方案中，第一次需要轉譯時，將轉譯作為解譯來進行以減少額外負擔，在看過(執行)程式碼許多次之後，調用 JIT 轉譯以建立比較最佳化的轉譯。

然而，程式碼變換程序還是引起許多問題。JIT 編譯程序本身就對處理器造成為數不小的額外負擔。這可在啟動應用程式時造成較大延遲。另外，管理變換碼在系統記憶體中的儲存造成多趟往返系統記憶體的行程(trip)，且包括記憶體映射(memory mapping)及分配管理額外負擔，因而招致可觀的潛時損失(latency penalty)。此外，變更應用程式的執行區涉及在系統記憶體及程式碼快取中重新安置變換碼，然後從頭開始起始程序。解譯程序涉及的額外負擔比 JIT 轉譯小，但其額外負擔是每個指令都要重複，因此還是相當可觀。即使產生程式碼，所產生的程式碼也無法最佳化。

【發明內容】

本發明之具體實施例實施一種實現客戶指令至本機指令轉

譯程序之硬體為基加速(hardware based acceleration)的演算法(algorithm)及裝置。

在一具體實施例中，本發明實施為硬體為基轉譯加速器(hardware based translation accelerator)。該硬體為基轉譯加速器包括：一客戶提取邏輯組件(guest fetch logic component)，用於存取複數個客戶指令；一客戶提取緩衝器(guest fetch buffer)，其耦接至該客戶提取邏輯組件及一分支預測組件(branch prediction component)，用於將該複數個客戶指令組合成(assemble)一客戶指令區塊(guest instruction block)；及複數個轉換表(conversion table)，其耦接至該客戶提取緩衝器，用於將該客戶指令區塊轉譯為對應的本機轉換區塊(native conversion block)。

該硬體為基轉譯加速器另外包括：一本機快取(native cache)，其耦接至該等轉換表，用於儲存該對應的本機轉換區塊；及一轉換後備緩衝器(conversion look aside buffer, CLB)，其耦接至該本機快取，用於儲存該客戶指令區塊至對應的本機轉換區塊的一映射，其中在收到針對客戶指令的後續請求時，索引該轉換後備緩衝器以決定是否發生一命中(hit)，其中該映射指示該客戶指令在該本機快取中具有一對應的轉換本機指令(converted native instruction)。回應於該命中，該轉換後備緩衝器傳送(forward)轉譯的本機指令以便執行。

上述為發明內容，按必要性含有簡單、一般性的說明，因而省略部分細節；所以熟習本技術者應明白，發明內容僅為說

明性的，而無意以任何方式加以限制。參考以下提出的非限制性詳細說明，即可明瞭本發明僅如申請專利範圍所定義的其他方面、發明特徵、及優點。

【實施方式】

雖然已經結合一具體實施例說明本發明，但無意使本發明受限於本文所提的特定形式。相反地，其意在於涵蓋如可在隨附申請專利範圍定義之本發明範疇中合理包括的替代方案、修改、及等效物。

在以下「實施方式」中，提出許多特定細節，諸如特定方法順序、結構、元件、及連接。但應明白，要實踐本發明之具體實施例，未必要利用這些及其他特定細節。在其他情形中，已省略或未以特定細節說明熟知的結構、元件、或連接，以免不必要地混淆本說明。

說明書中提及「一具體實施例」是指結合該具體實施例說明的特定特徵、結構、或特性包括在本發明的至少一具體實施例中。說明書中各處出現的用語「在一具體實施例中」未必全部有關相同的具體實施例，也不是與其他具體實施例互斥的獨立或替代具體實施例。此外，說明可由某些具體實施例展現而未由其他具體實施例展現的各種特徵。同樣地，說明可為某些具體實施例但非其他具體實施例之必要條件的必要條件。

以下「實施方式」的某些部分就電腦記憶體內資料位元操作的過程、步驟、邏輯區塊、處理、及其他符號表示法提出討

論。這些說明及表示法係熟習資料處理技術者所使用的方法，以最有效的方式將其運作本質傳達給其他熟習本技術者。過程、電腦執行步驟、邏輯區塊、程序等在此且一般被設想為導致所要結果之自相一致的步驟或指令序列。此等步驟是需要物理量之物理操控的步驟。通常，但非一定，這些物理量採取電腦可讀儲存媒體之電信號或磁信號的形式，且能夠在電腦系統中儲存、傳輸、結合、比較及以其他方式操控。已證實有時很方便的是(主要為了一般的使用)將這些信號稱為位元、值、元件、符號、字元(character)、條件項(term)、數字等。

然而，應明白，所有這些術語及相似術語係為了和相應物理量相關聯，且只是應用於這些物理量的便利標記。除非另外明確說明為明顯不同於以下論述，應明白，在本發明所有方面，利用諸如「處理(processing)」或「存取(accessing)」或「寫入」或「儲存」或「複製」等術語的論述，是指電腦系統或類似電子計算裝置的動作與程序，電腦系統或類似電子計算裝置操控電腦系統暫存器與記憶體及其他電腦可讀媒體中表示為物理(電子)量的資料，及將其變換為電腦系統記憶體或暫存器或其他此類資訊儲存器、傳送或顯示裝置中同樣表示為物理量的其他資料。

本發明之具體實施例藉由大幅加速將客戶指令架構之客戶指令轉譯為本機指令架構之本機指令的程序以在本機處理器上執行而發揮功能。本發明之具體實施例利用硬體單元實施轉換程序的硬體加速。客戶指令可以來自若干不同的指令架構。實例架構包括 Java 或 JavaScript、x86、MIPS、SPARC 等。這些

客戶指令被快速轉換為本機指令，並以管線送到本機處理器硬體以便快速執行。此舉比傳統的軟體控制轉換程序提供高出許多的效能等級。

在一具體實施例中，本發明實施靈活轉換程序，其可使用若干不同指令架構作為輸入。在此具體實施例中，將處理器的前端實施成其可為軟體所控制，同時利用硬體加速轉換處理，以達到更高的效能等級。此實施方案在多方面達到好處。可處理及轉換不同的客戶架構，且各個客戶架構得到硬體加速的好處，以享有更高的效能等級。軟體控制前端可提供應用程式在處理器上執行的較大靈活度。硬體加速可達成以接近本機硬體速度執行客戶應用程式的客戶指令。在以下說明中，圖 1 至圖 4 顯示本發明之具體實施例處置客戶指令序列及處置在這些客戶指令序列內的較近分支(near branch)及較遠分支(far branch)的方式。圖 5 顯示根據本發明之一具體實施例之例示性硬體加速轉換處理系統的概觀。

圖 1 顯示本發明之一具體實施例所操作的例示性指令序列。如圖 1 所描繪，指令序列 100 包含 16 個指令，從圖 1 的頂部開始至底部。如圖 1 中可見，序列 100 包括四個分支指令 101-104。

本發明具體實施例之一目的在於將整個群組的指令處理為單一基元單元(single atomic unit)。此基元單元稱為「區塊」。指令區塊可充分延伸超過圖 1 顯示的 16 個指令。在一具體實施例中，區塊將包括足夠的指令以填滿固定大小(如，64 位元組、128

位元組、256 位元組等)，或直到遇到出口條件(exit condition)。在一具體實施例中，結束指令區塊的出口條件是遇到較遠分支指令。如本文具體實施例的說明中所使用，較遠分支是指其目標位址駐存在目前指令區塊之外的分支指令。換言之，在給定的客戶指令區塊內，較遠分支具有駐存在某個其他區塊中或在給定指令區塊外之某個其他指令序列中的目標。同樣地，較近分支是指其目標位址駐存在目前指令區塊內的分支指令。另外，應注意，本機指令區塊可含有多個客戶較遠分支。在以下討論中將進一步說明這些術語。

圖 2 顯示根據本發明之一具體實施例描繪基於區塊的轉換程序的示意圖，其中將客戶指令區塊轉換為本機轉換區塊。如圖 2 中圖解，顯示將複數個客戶指令區塊 201 轉換為對應的複數個本機轉換區塊 202。

本發明之具體實施例藉由將客戶指令區塊的指令轉換為本機轉換區塊之對應的指令而發揮功能。各個區塊 201 由客戶指令組成。如上述，這些客戶指令可來自若干不同的客戶指令架構(如，Java 或 JavaScript、x86、MIPS、SPARC 等)。多個客戶指令區塊可轉換為一或多個對應的本機轉換區塊。此轉換基於每個指令而發生。

圖 2 亦圖解基於分支預測將客戶指令區塊組合成序列的方式。此屬性(attribute)使本發明之具體實施例能夠基於較遠分支的預測結果組合客戶指令序列。基於較遠分支預測，將客戶指令序列從多個客戶指令區塊組合成及轉換為對應的本機轉換區

塊。此做法將在以下圖 3 及圖 4 中進一步說明。

圖 3 顯示根據本發明之一具體實施例圖解將客戶指令區塊的各個指令轉換為本機轉換區塊之對應的本機指令之方式的示意圖。如圖 3 中圖解，客戶指令區塊駐存在客戶指令緩衝器 301 內。同樣地，本機轉換區塊駐存在本機指令緩衝器 302 內。

圖 3 顯示本發明之具體實施例的屬性，其中將客戶分支指令的目標位址轉換為本機分支指令的目標位址。例如，客戶指令分支各包括識別特定分支之目標位址的偏移量(offset)。這在圖 3 中顯示為客戶偏移量或 G_offset。隨著客戶指令被轉換，此偏移量通常因為本機指令產生對應客戶指令之功能性所需的不同長度或序列而不同。例如，客戶指令的長度可與其對應之本機指令的長度不同。因此，轉換程序藉由計算對應的本機偏移量補償此差異。這在圖 3 中顯示為本機偏移量或 N_offset。

應注意，由於未預測在客戶指令區塊內具有目標的分支(稱為「較近分支」)，因而未改變指令序列流。

圖 4 顯示根據本發明之一具體實施例圖解以處置本機轉換區塊來處理較遠分支之方式的示意圖。如圖 4 中圖解，將客戶指令描繪為記憶體中的客戶指令序列 401。同樣地，將本機指令描繪為記憶體中的本機指令序列 402。

在一具體實施例中，每一個指令區塊(客戶指令區塊及本機指令區塊二者)以較遠分支結束(如，即使本機區塊可能含有多個

客戶較遠分支)。如上述，區塊將包括足夠的指令以填滿固定大小(如，64 位元組、128 位元組、256 位元組等)，或直到遇到出口條件，諸如最後一個客戶較遠分支指令。如果已經處理若干客戶指令以組合客戶指令區塊且尚未遇到較遠分支，則插入客戶較遠分支以結束區塊。此較遠分支只要一躍(jump)即可跳到下一個後續區塊(subsequent block)。這確保指令區塊以引向記憶體中另一本機指令區塊或另一客戶指令序列的分支結束。另外，如圖 4 中顯示，區塊可包括在其指令序列內不駐存在區塊末端處的客戶較遠分支。這以客戶指令較遠分支 411 及對應的本機指令客戶較遠分支 412 來顯示。

在圖 4 具體實施例中，預測取用較遠分支 411。因此，指令序列跳至較遠分支 411 的目標，即客戶指令 F。同樣地，在對應的本機指令中，接續在較遠分支 412 之後的是本機指令 F。未預測較近分支。因此，這些較近分支未以較遠分支的相同方式改變指令序列。

以此方式，本發明之具體實施例產生轉換區塊的軌跡(trace)，其中各個區塊包含若干(如，3-4)個較遠分支。此軌跡係基於客戶較遠分支預測。

在一具體實施例中，在本機轉換區塊內的較遠分支包括用於相對分支路徑之相反位址的客戶位址。如上述，基於較遠分支的預測產生指令序列。直到執行對應的本機轉換區塊，才會知道真正的預測結果。因此，一旦偵測到錯誤的預測，則檢查錯誤的較遠分支，以獲得相對分支路徑的相反客戶位址。轉換

程序接著從相反客戶位址繼續，其現在是真正的分支路徑。以此方式，本發明之具體實施例使用所包括之用於相對分支路徑的相反客戶位址，以從較遠分支之預測結果是錯誤的狀況中恢復。因此，如果較遠分支預測結果是錯誤的，則程序知道去何處找到正確的客戶指令。同樣地，如果較遠分支預測結果是正確的，則忽略相反客戶位址。應注意，如果正確預測在本機指令區塊內的較遠分支，則在 CLB 中不需要其目標區塊的登錄點(entry point)。然而，一旦發生失誤的預測，則需要在 CLB 中插入目標區塊的新登錄。此功能係以保存 CLB 容量(capacity)為目標而履行。

圖 5 顯示例示性硬體加速轉換系統 500 的示意圖，其根據本發明之一具體實施例圖解將客戶指令區塊及其對應的本機轉換區塊儲存在快取中的方式。如圖 5 中圖解，使用轉換後備緩衝器 506 快取客戶及本機區塊之間的位址映射；致使透過低潛時可用性(low latency availability)將最常遇到的本機轉換區塊存取至處理器 508 中。

圖 5 示意圖圖解在高速低潛時快取(high-speed low latency cache memory)(轉換後備緩衝器 506)內維持時常遇到的本機轉換區塊的方式。圖 5 中描繪的組件實施硬體加速轉換處理，以達到更高的效能等級。

客戶提取邏輯單元 502 運作為從系統記憶體 501 提取客戶指令的硬體為基客戶指令提取單元(hardware-based guest instruction fetch unit)。給定應用程式的客戶指令駐存在系統記憶

體 501 內。在程式初始後，硬體為基客戶提取邏輯單元 502 開始預提取客戶指令至客戶提取緩衝器 503 中。客戶提取緩衝器 503 累積客戶指令，然後將這些指令組合成客戶指令區塊。使用轉換表 504，將這些客戶指令區塊轉換為對應的本機轉換區塊。轉換的本機指令在本機轉換緩衝器 505 內累積，直到本機轉換區塊完成。接著將本機轉換區塊傳輸至本機快取 507，及將映射儲存在轉換後備緩衝器 506 中。接著使用本機快取 507 以供給本機指令至處理器 508 以便執行。在一具體實施例中，藉由客戶提取邏輯狀態機(guest fetch logic state machine)產生由客戶提取邏輯單元 502 實施的功能性。

隨著此程序繼續，轉換後備緩衝器 506 填滿客戶區塊至本機區塊的位址映射。轉換後備緩衝器 506 使用一或多個演算法(如，最近最少使用等)以確保將較常遇到的區塊映射保持在緩衝器內，且將很少遇到的區塊映射逐出緩衝器。以此方式，將熱門的本機轉換區塊映射儲存在轉換後備緩衝器 506 內。另外，應注意，在本機區塊內精確預測的較遠客戶分支不需要在 CLB 中插入新的映射，因為其目標區塊固定在單一映射本機區塊內，因此保存 CLB 結構的較小容量效率。此外，在一具體實施例中，構造 CLB 以僅儲存最後的客戶至本機位址映射。此做法亦保存 CLB 的較小容量效率。

客戶提取邏輯 502 查看轉換後備緩衝器 506 以決定客戶指令區塊的位址是否已被轉換為本機轉換區塊。如上述，本發明之具體實施例提供用於轉換處理的硬體加速。因此，在從系統記憶體 501 提取客戶位址以進行新的轉換前，客戶提取邏輯 502

將查看轉換後備緩衝器 506 有無預先存在的本機轉換區塊映射。

在一具體實施例中，對於轉換後備緩衝器以客戶位址範圍或以個別客戶位址進行索引。客戶位址範圍是已轉換為本機轉換區塊之客戶指令區塊之位址的範圍。轉換後備緩衝器所儲存的本機轉換區塊映射經由其對應之客戶指令區塊之對應的客戶位址範圍而被進行索引。因此，客戶提取邏輯可比較客戶位址與轉換區塊的客戶位址範圍或個別客戶位址(其映射保持在轉換後備緩衝器 506 中)，以決定預先存在的本機轉換區塊是否駐存在本機快取 507 中儲存的位置內或在圖 6 的程式碼快取中。如果預先存在的本機轉換區塊位在本機快取或程式碼快取中，則將對應的本機轉換指令從這些快取直接傳送到處理器。

以此方式，熱門的客戶指令區塊(如，經常執行的客戶指令區塊)具有其維持在高速低潛時轉換後備緩衝器 506 內的對應熱門本機轉換區塊映射。隨著區塊被接觸，適當的替換策略(replacement policy)確保熱門的區塊映射繼續留在轉換後備緩衝器內。因此，客戶提取邏輯 502 可馬上識別所請求的客戶位址先前是否已被轉換，並可將先前轉換的本機指令直接傳送至本機快取 507，以由處理器 508 執行。這些做法省下許多週期，因為至系統記憶體的路程可花費 40 至 50 個週期或更多。這些屬性(如，CLB、客戶分支序列預測、客戶及本機分支緩衝器、對先前的本機快取)允許本發明之具體實施例的硬體加速功能性達成客戶應用程式的應用程式效能至同等本機應用程式的應用程式效能的 80%至 100%內。

在一具體實施例中，客戶提取邏輯 502 一再地預提取客戶指令，用於與來自處理器 508 的客戶指令請求(guest instruction request)無關的轉換。本機轉換區塊可在系統記憶體 501 中那些用於較少使用區塊的轉換緩衝器「程式碼快取」(conversion buffer code cache)內累積。轉換後備緩衝器 506 亦保持最常使用的映射。因此，如果所請求的客戶位址未映射至轉換後備緩衝器中的客戶位址，則客戶提取邏輯可檢查系統記憶體 501 以決定客戶位址是否對應於其中儲存的本機轉換區塊。

在一具體實施例中，轉換後備緩衝器 506 係實施為快取並利用快取一致性協定(cache coherency protocol)，以與儲存在較高階快取中的較大轉換緩衝器及系統記憶體 501 維持一致性。儲存在轉換後備緩衝器 506 內的本機指令映射亦寫回(write back)較高階的快取及系統記憶體 501。至系統記憶體的寫回維持一致性。因此，可使用快取管理協定以確保熱門的本機轉換區塊映射儲存在轉換後備緩衝器 506 內，及冷門的本機轉換映射區塊儲存在系統記憶體 501 中。因此，較大形式的轉換緩衝器 506 駐存在系統記憶體 501 中。

應注意，在一具體實施例中，可使用例示性硬體加速轉換系統 500 實施若干不同的虛擬儲存方案。例如，可使用客戶指令區塊及其對應的本機轉換區塊儲存在快取內的方式支援虛擬儲存方案。同樣地，可使用用來快取客戶及本機區塊之間的位址映射的轉換後備緩衝器 506 支援虛擬儲存方案(如，虛擬至實體記憶體映射的管理)。

在一具體實施例中，圖 5 的架構實施虛擬指令集處理器/電腦，其使用可接收若干不同指令架構作為輸入的靈活轉換程序。在此虛擬指令集處理器中，將處理器的前端實施成其可為軟體所控制，同時利用硬體加速轉換處理，以達到更高的效能等級。使用此實施方案，可處理及轉換不同的客戶架構(guest architecture)，且各個客戶架構得到硬體加速的好處，以享有更高的效能等級。實例客戶架構包括 Java 或 JavaScript、x86、MIPS、SPARC 等。在一具體實施例中，「客戶架構」可以是本機指令(如，來自本機應用程式/巨集操作(macro-operation))，及轉換程序產生最佳化本機指令(如，最佳化本機指令/微操作(micro-operation))。軟體控制前端可提供應用程式在處理器上執行的較大靈活度。如上述，硬體加速可以接近本機硬體速度執行客戶應用程式的客戶指令。

圖 6 顯示根據本發明之一具體實施例之硬體加速轉換系統 600 的詳細實例。系統 600 以上述系統 500 的實質上相同方式履行。然而，系統 600 顯示說明例示性硬體加速程序之功能性的額外細節。

系統記憶體 601 包括資料結構(data structure)，其包含：客戶碼 602、轉換後備緩衝器 603、最佳化器碼(optimizer cod)604、轉換器碼(converter code)605、及本機碼快取 606。系統 600 亦顯示共用硬體快取(shared hardware cache)607，在其中交錯及共用客戶指令及本機指令兩者。客戶硬體快取 610 從共用硬體快取 607 抓取最常接觸的這些客戶指令。

客戶提取邏輯 620 預提取來自客戶碼 602 的客戶指令。客戶提取邏輯 620 與 TLB 609 介接，TLB 609 運作為將虛擬客戶位址轉譯為對應的實體客戶位址的轉換後備緩衝器。TLB 609 可將命中直接傳送至客戶硬體快取 610。由客戶提取邏輯 620 提取的客戶指令儲存在客戶提取緩衝器 611 中。

轉換表 612 及 613 包括替代欄位(substitute field)及控制欄位(control field)，並運作為將接收自客戶提取緩衝器 611 的客戶指令轉譯為本機指令的多層轉換表(multilevel conversion table)。

多工器(multiplexer)614 及 615 將轉換的本機指令傳輸至本機轉換緩衝器 616。本機轉換緩衝器 616 累積轉換的本機指令以組合本機轉換區塊。接著將這些本機轉換區塊傳輸至本機硬體快取 608，及將映射保持在轉換後備緩衝器 630 中。

轉換後備緩衝器 630 包括用於以下的資料結構：客戶分支位址(guest branch address)631、本機位址 632、轉換的區塊範圍(converted block range)633、程式碼快取及轉換後備緩衝器管理位元(code cache and conversion look aside buffer management bit)634、及動態分支偏差值位元(dynamic branch bias bit)635。客戶分支位址 631 及本機位址 632 包含指示哪些對應的本機轉換區塊駐存在轉換的區塊範圍 633 內的客戶位址範圍。快取管理協定及替換策略確保熱門的本機轉換區塊映射駐存在轉換後備緩衝器 630 內，同時冷門的本機轉換區塊映射駐存在系統記憶體 601 的轉換後備緩衝器資料結構 603 內。

如同系統 500 的情況，系統 600 試圖確保熱門的區塊映射駐存在高速低潛時轉換後備緩衝器 630 內。因此，在一具體實施例中，當提取邏輯 640 或客戶提取邏輯 620 查看以提取客戶位址時，提取邏輯 640 首先可檢查客戶位址，以決定對應的本機轉換區塊是否駐存在本機碼快取 606 內。此舉允許決定請求的客戶位址是否在本機碼快取 606 中具有對應的本機轉換區塊。如果請求的客戶位址不駐存在緩衝器 603 或 608、或緩衝器 630 內，則從客戶碼 602 提取客戶位址及若干後續客戶指令，及經由轉換表 612 及 613 實施轉換程序。

圖 7 顯示根據本發明之一具體實施例具有輔助軟體為基加速轉換管線(secondary software-based accelerated conversion pipeline)之硬體加速轉換系統 700 的實例。

組件 711-716 包含軟體實施的載入儲存路徑(load store path)，其例如在特殊高速記憶體 760 內。如圖 7 所描繪，客戶提取緩衝器 711、轉換表 712-713 及本機轉換緩衝器 716 包含特殊高速記憶體 760 的分配部分(allocated portion)。在許多觀點上，特殊高速記憶體 760 用作極低階(low-level)快速快取(如，L0 快取)。

箭頭 761 圖解藉以經由與指令提取路徑(instruction fetch path)(如，來自提取的解碼邏輯(fetched decode logic))相對的載入儲存路徑來加速轉換的屬性。

在圖 7 具體實施例，高速記憶體 760 包括用於進行比較的

特殊邏輯。由於這一點，可以軟體實施轉換加速。例如，在另一具體實施例中，由使用處理器執行管線(processor execution pipeline)的軟體操控儲存組件 711-716 的標準記憶體 760，該軟體在處理器執行管線處將該等組件 711-716 的值載入一或多個 SIMD 暫存器並實施履行 SIMD 暫存器中欄位間之比較的比較指令，及視需要履行遮罩操作(mask operation)及結果掃描操作(result scan operation)。可使用通用微處理器硬體(general purpose microprocessor hardware)(諸如，例如，使用比較一者與多者之比較指令)來實施載入儲存路徑。

應注意，由具有特殊屬性或位址範圍的指令存取記憶體 760。例如，在一具體實施例中，客戶提取緩衝器具有用於各個客戶指令項目(guest instruction entry)的 ID。每個客戶指令均建立 ID。此 ID 允許容易從客戶緩衝器映射至本機轉換緩衝器。ID 允許容易進行客戶偏移量至本機偏移量的計算，不管客戶指令與對應的本機指令相比的不同長度。此做法在上文圖 3 中已做圖示。

在一具體實施例中，由使用計算所提取客戶指令之長度之長度解碼器(length decoder)的硬體來計算 ID。然而，應注意，可以硬體或軟體履行此功能性。

一旦指派 ID 後，可經由 ID 存取本機指令緩衝器。ID 允許客戶偏移量至本機偏移量的偏移量轉換。

圖 8 顯示根據本發明之一具體實施例圖解 CLB 結合程式碼

快取及儲存於記憶體中的客戶指令至本機指令映射而發揮功能之方式的例示性流程圖。

如上述，使用 CLB 來儲存具有對應的轉換本機位址(儲存在程式碼快取記憶體內)之客戶位址的映射(如，客戶至本機位址映射)。在一具體實施例中，以客戶位址的一部分索引 CLB。客戶位址可分割成索引、標籤(tag)、及偏移量(如，記憶塊大小(chunk size))。此客戶位址包含用以識別 CLB 項目中對應於索引之匹配的標籤。如果在標籤上發生命中，則對應的項目將儲存指標(pointer)，其指示可在程式碼快取記憶體 806 中何處找到對應的轉換本機指令記憶塊(如，轉換本機指令的對應區塊)。

應注意，本文使用術語「記憶塊」指轉換本機指令區塊的對應記憶體大小。例如，記憶塊的大小可取決於轉換本機指令區塊的不同大小而不同。

在一具體實施例中，關於程式碼快取記憶體 806，以一組固定大小記憶塊(如，各個記憶塊類型具有不同大小)分配程式碼快取。在系統記憶體及所有較低階硬體快取(如，本機硬體快取 608、共用硬體快取 607)中，可將程式碼快取邏輯分割成若干組及方式。CLB 可使用客戶位址以進行索引及標籤上比較用於程式碼快取記憶塊(code cache chunk)的方式標籤(way tag)。

圖 8 描繪 CLB 硬體快取 804 以描繪為方式 x 及方式 y 的兩種方式儲存客戶位址標籤。應注意，在一具體實施例中，可透過以結構化方式將指標儲存於本機碼記憶塊，完成使用 CLB 結

構之客戶位址至本機位址的映射(如，從客戶至本機位址映射)。各個方式與標籤相關聯。CLB 係以客戶位址 802(包含標籤)進行索引。在 CLB 中發生命中時，傳回對應於標籤的指標。使用此指標以索引程式碼快取記憶體。這在圖 8 中以一行文字「程式碼記憶塊的本機位址=Seg#+F(pt)」顯示，其代表以下事實：程式碼記憶塊的本機位址隨指標及區段號碼(segment number)而變。在本具體實施例中，區段是指記憶體中用於虛擬映射指標範疇(pointer scope)之點的基礎(如，允許將指標陣列映射於實體記憶體中的任何區域)。

或者，在一具體實施例中，可經由如圖 8 中以一行文字「程式碼記憶塊的本機位址=seg#+ 索引 * (64 個記憶塊大小) + way# * (記憶塊大小)」顯示的第二方法，索引程式碼快取記憶體。在此具體實施例中，可將程式碼快取組織成其方式結構(way-structure)匹配 CLB 方式結構化(way structuring)，使得在 CLB 的方式及程式碼快取記憶塊的方式之間存在 1:1 映射。當在特定 CLB 方式中發生命中時，即程式碼快取的對應方式中的對應程式碼記憶塊具有本機碼。

仍然參考圖 8，如果 CLB 的索引未中(miss)，則可檢查記憶體的較高階層是否有命中(如，L1 快取、L2 快取等)。如果在這些較高快取等級中沒有發生命中，則檢查系統記憶體 801 中的位址。在一具體實施例中，客戶索引指向包含例如 64 個記憶塊的項目。讀出 64 個記憶塊中各者的標籤，並將其與客戶標籤進行比較，以決定是否發生命中。此程序在圖 8 中以虛線方框 805 顯示。如果在與系統記憶體中的標籤比較之後沒有任何命中，

則在記憶體的任何階層等級不存在轉換，及必須轉換客戶指令。

應注意，本發明之具體實施例管理以類似快取方式儲存客戶至本機指令映射之記憶體的各個階層等級。這原本來自快取為基記憶體(cache-based memory)(如，CLB 硬體快取、本機快取、L1 及 L2 快取等)。然而，CLB 亦包括「程式碼快取+CLB 管理位元」，其係用來實施系統記憶體 801 內之客戶至本機指令映射的最近最少使用(least recently used, LRU)替換管理策略。在一具體實施例中，CLB 管理位元(如，LRU 位元)為軟體所管理。以此方式，使用記憶體的所有階層等級儲存最近使用之最常遇到的客戶至本機指令映射。對應地，這導致記憶體的所有階層等級同樣地儲存最常遇到的轉換本機指令。

圖 8 亦顯示 CLB 中儲存的動態分支偏差值位元及/或分支歷史記錄位元(branch history bit)。使用這些動態分支位元追蹤用於組合客戶指令序列之分支預測的行為。使用這些位元追蹤哪些分支預測是最常正確預測的，及哪些分支預測是最常不正確預測的。CLB 亦儲存轉換區塊範圍的資料。此資料使程序能夠讓程式碼快取記憶體中對應之客戶指令已修改(如，在自我修改程式碼中)的轉換區塊範圍變成無效。

圖 9 顯示根據本發明之一具體實施例圖解實體儲存堆疊快取實施方案(physical storage stack cache implementation)及客戶位址至本機位址映射的例示性流程圖。如圖 9 所描繪，可將快取實施為實體儲存堆疊 901。

圖 9 具體實施例圖解可將程式碼快取實施為可變結構快取(variable structure cache)的方式。取決於不同具體實施例的需求，可變結構快取可完全為硬體實施及控制、完全為軟體實施及控制、或軟體強制及控制與基本硬體啟用的部分混合。

圖 9 具體實施例有關在管理分配及替換客戶至本機位址映射及其在實際實體儲存中的對應轉譯的工作之間求取最佳平衡。在本具體實施例中，此可透過使用結合指標與可變大小記憶塊的結構來完成。

使用多方式標籤陣列(multi-way tag array)來儲存用於不同大小群組之實體儲存的指標。每次需要分配特定的儲存大小(如，其中儲存大小對應於位址)時，則據此分配各對應於此大小之儲存區塊的群組。此舉允許本發明之具體實施例精確地分配儲存，以儲存可變大小的指令軌跡。圖 9 顯示群組如何可以屬於不同大小。顯示兩個例示性群組大小：「群組大小 4 的替換候選者」及「群組大小 2 的替換候選者」。除了對應於位址的標籤之外，指標係也儲存在將位址映射成實體儲存位址的標籤陣列中。標籤可包含兩個或兩個以上的子標籤。例如，標籤結構 902 中最前面的 3 個標籤包含分別如所顯示的子標籤 A1 B1、A2 B2 C2 D2、及 A3 B3。因此，標籤 A2 B2 C2 D2 包含群組大小 4，而標籤 A1 B1 包含群組大小 2。群組大小遮罩(group size mask)亦指示群組的大小。

實體儲存於是可像堆疊般管理，致使每次分配新的群組時，可將新的群組放置在實體儲存堆疊的頂部。藉由覆寫項目

的標籤而使項目失效，藉此恢復所分配的空間。

圖 9 亦顯示延伸方式標籤結構(extended way tag structure)903。在一些情形中，標籤結構 902 中的項目在延伸方式標籤結構 903 中將具有對應的項目。這取決於項目及標籤結構是否具有設定之延伸方式位元(如，設定為一)。例如，設定為一的延伸方式位元指示在延伸方式標籤結構中有對應的項目。延伸方式標籤結構允許處理器以與標準標籤結構的不同方式延伸參考區域性(locality of reference)。因此，儘管以一個方式索引標籤結構 902(如，索引(j))，但以不同方式(如，索引(k))索引延伸方式標籤結構。

在典型的實施方案中，索引(j)可以是在索引(k)中的更多項目。這是因為在大多數的限制中，主要的標籤結構 902 比延伸方式標籤結構 903 大上許多，其中例如(j)可涵蓋 1024 個項目(如，10 位元)，而(k)可涵蓋 256(如，8 位元)。

這使得本發明之具體實施例能夠合併用於匹配已變得很熱門(如，極常遇到)之軌跡的額外方式。例如，如果無法在標籤結構 902 中找到在熱門集(hot set)內的匹配，則藉由設定延伸方式位元，可使用延伸方式標籤結構儲存用於熱門軌跡的額外方式。應注意，此可變快取結構在僅需要時使用儲存於堆疊上之快取程式碼/資料的儲存，例如，如果在特定程式階段(phase)期間從未存取任何快取集(cache set)(由索引位元指示的項目)，則堆疊上將沒有該快取集的儲存分配。與典型快取(其中快取集具有針對每一個快取集的固定實體資料儲存)相比，這提供高效率

的有效儲存容量增加。

亦可以有指示一個集合或一個群組的集合為冷門的(cold)位元(如，意思是這些集合已經很久未被存取)。在此例中，這些集合的堆疊儲存看起來像是在分配堆疊儲存內的氣泡(bubble)。此時，可為其他熱門集要求使用這些冷門集(cold set)的分配指標。此程序是儲存回收再利用程序(storage reclamation process)，其中在記憶塊已在堆疊內分配好之後，該記憶塊屬於的整個集合其後變成冷門。可促進此回收再利用的所需機制及結構(圖 9 中未顯示以免使所顯示的方面變得混亂或混淆)是：每一個集合有一個冷門集指示符(cold set indicator)(項目索引)，及回收再利用程序(其中用於這些冷門集之方式的指標被重複使用用於其他熱門集的方式)。此舉允許回收再利用這些堆疊儲存氣泡(記憶塊)。當不處於回收再利用模式時，將新的記憶塊分配於堆疊的頂部，當堆疊具有冷門集(如，集合方式/記憶塊長時間未被存取)時，回收再利用動作允許需要被分配在另一集合中的新記憶塊去重複使用被回收再利用的指標及其在堆疊內的相關聯記憶塊儲存(其屬於冷門集)。

應注意，圖 9 具體實施例充分適於使用標準記憶體，其實施為與特殊快取記憶體相對。此屬性是由於以下事實所造成：實體儲存堆疊係藉由讀取指標、讀取索引、及分配位址範圍加以管理。在此實施方案中不需要特殊的快取為基電路結構。

應注意，在一具體實施例中，可使用圖 9 架構以實施不涉及轉換或程式碼變換的資料快取及快取方案。因此，可使用圖 9

架構實施更為標準化的快取(如，L2 資料快取等)。這麼做將比習用的固定結構快取等提供更大的有效容量。

圖 10 顯示根據本發明之一具體實施例描繪硬體加速轉換系統 1000 之額外例示性細節的示意圖。直線 1001 圖解將傳入的客戶指令與複數個群組遮罩及標籤相比較的方式。其目的是快速識別客戶指令的類型，然後將其指派給對應的群組。群組遮罩及標籤藉由匹配客戶指令的子欄位(subfield)以識別客戶指令所屬於的特定群組而發揮功能。遮罩遮掩客戶指令樣式(guest instruction pattern)的不相關位元，以特別查看相關位元。諸如表格 1002 的表格按優先順序處理的方式儲存遮罩-標籤對(mask-tag pair)。

藉由按優先權方向(在此例中描繪為由上至下的方向)讀取至表格中，來匹配樣式。以此方式，藉由按遮罩-標籤儲存的優先權方向讀取，來匹配樣式。以其優先權順序及樣式匹配功能性所檢查的不同遮罩係對應地按其優先權順序加以應用。在找到命中時，即從儲存映射的對應表格(如，表格 1003)讀取樣式的對應映射。第二級表格 1004 圖解可以串列循序方式(cascading sequential manner)存取多個轉換表直到完全轉換客戶指令的階層方式(hierarchical manner)。如上述，轉換表包括替代欄位及控制欄位，並運作為將接收自客戶提取緩衝器的客戶指令轉譯為本機指令的多層轉換表。

以此方式，緩衝器中的各個位元組串流(byte stream)發送到轉換表，其中各個等級的轉換表連續偵測位元欄位。在偵測到

相關位元欄位時，表格替代本機的等效欄位。

表格亦產生幫助用於此等級以及下一個等級表格(如，第二級表格 1004)之替代程序(substitution process)的控制欄位。下一個表格使用先前表格的控制欄位以識別下一個相關位元欄位，其以本機等效欄位替代。第二級表格於是可產生控制欄位以幫助第一級表格，以此類推。一旦所有客戶位元欄位均以本機位元欄位替代，指令即完全轉譯且傳送至本機轉換緩衝器。本機轉換緩衝器接著寫入程式碼快取，及其客戶至本機位址映射記錄於 CLB 中，如上所述。

圖 11A 顯示本發明之具體實施例所實施之例示性樣式匹配程序的示意圖。如圖 11A 所描繪，由標籤、樣式、及遮罩決定目的地。樣式解碼的功能性包含履行位元比較(如，逐位元 XOR)、履行位元 AND(如，逐位元 AND)、及後續檢查所有零位元(如，所有位元的 NOR)。

圖 11B 顯示根據本發明之一具體實施例之基於 SIMD 暫存器之樣式匹配程序的示意圖 1100。如示意圖 1100 中所描繪，顯示四個 SIMD 暫存器 1102-1105。這些暫存器實施如所示樣式解碼程序的功能性。使用傳入樣式 1101 在各個標籤上履行平行位元比較(如，逐位元 XOR)，及其結果履行與遮罩的位元 AND(如，逐位元 AND)。匹配指示符結果如所示各儲存於其相應的 SIMD 位置。接著如所示履行掃描，及掃描在 SIMD 元件中所遇到的第一個「真(true)」是其中所有 i 位元的方程式 $(P_i \text{ XOR } T_i) \text{ AND } M_i = 0$ 為真的元件，其中 P_i 是相應的樣式， T_i 是相應

的標籤及 M_i 是相應的遮罩。

圖 12 顯示根據本發明之一具體實施例之統一暫存器檔案 (unified register file) 1201 的圖式。如圖 12 所描繪，統一暫存器檔案 1201 包括個 2 部分 1202-1203 及項目選擇器 (entry selector) 1205。統一暫存器檔案 1201 支援硬體狀態更新的架構推測 (architecture speculation)。

統一暫存器檔案 1201 可實施最佳化影子暫存器 (optimized shadow register) 及認可的 (committed) 暫存器狀態管理程序。此程序支援硬體狀態更新的架構推測。在此程序下，本發明之具體實施例可支援影子暫存器功能性及認可的暫存器功能性，且不需要在暫存器記憶體之間進行任何交叉複製。例如，在一具體實施例中，統一暫存器檔案 1201 的功能性大部分由項目選擇器 1205 提供。在圖 12 具體實施例中，各個暫存器檔案項目由分別來自部分 1 及部分 2 的兩對暫存器 R 及 R' 構成。在任何給定時間，從各個項目讀取的暫存器不是來自部分 1 的 R ，就是來自部分 2 的 R' 。基於項目選擇器 1205 針對各個項目儲存的 x 及 y 位元的值，針對暫存器檔案的各個項目，有 4 個不同的組合。

x 及 y 位元的值如下。

00: R 無效; R' 認可	(在讀取了讀取請求 R' 後)
01: R 推測; R' 認可	(在讀取了讀取請求 R 後)
10: R 認可; R' 推測	(在讀取了讀取請求 R' 後)
11: R 認可; R' 無效	(在讀取了讀取請求 R 後)

以下是各個指令/事件的影響(impact)。在指令「寫回」後，00 變成 01，及 11 變成 10。在指令「認可」後，01 變成 11，及 10 變成 00。在發生回復事件(rollback event)後，01 變成 00，及 10 變成 11。

這些變更主要是儲存在暫存器檔案項目選擇器 1205 中的狀態變更，並基於發生這些變更時的事件而發生。應注意，認可指令及回復事件需要達到認可階段，以在項目選擇器 1205 中造成位元轉變。

以此方式，才能夠在影子暫存器狀態內繼續執行，而不會破壞認可的暫存器狀態。當影子暫存器狀態準備認可時，更新暫存器檔案項目選擇器致使以上述方式從該部分讀取有效結果。以此方式，僅視需要藉由更新暫存器檔案項目選擇器，則推測執行結果(speculative execution result)可在發生例外時，回復到最近認可點(commit point)。同樣地，認可點可向前移動，藉此僅藉由更新暫存器檔案項目選擇器，認可推測執行結果。此功能性在不需要在暫存器記憶體之間進行任何交叉複製下提供。

以此方式，統一暫存器檔案可經由暫存器檔案項目選擇器 1205，實施複數個推測暫用影子暫存器(speculative scratch shadow register, SSSR)及複數個認可暫存器(committed register, CR)。例如，在認可時，SSSR 暫存器變成 CR 暫存器。在回復時，SSSR 狀態回復為 CR 暫存器。

圖 13 顯示根據本發明之一具體實施例支援推測架構狀態 (speculative architectural state) 及暫態架構狀態 (transient architectural state) 之統一影子暫存器檔案及管線架構 (unified shadow register file and pipeline architecture) 1300 的示意圖。

圖 13 具體實施例描繪包含支援包含架構 1300 的組件，其支援包含架構推測狀態之指令及結果以及支援包含暫態之指令及結果。如本文中使用的，認可架構狀態包含可由處理器上執行的程式存取(如，讀取及寫入)的可見暫存器 (visible register) 及可見記憶體。相反地，推測架構狀態包含未被認可及因此並非全域可見的暫存器及/或記憶體。

在一具體實施例中，有四個由架構 1300 啟用的使用模型 (usage model)。第一使用模型包括硬體狀態更新的架構推測，如上文在圖 12 的討論中描述。

第二使用模型包括雙範疇使用 (dual scope usage)。此使用模型可用於將 2 個執行緒 (thread) 提取至處理器中，其中一個執行緒在推測狀態中執行，及另一個執行緒在非推測狀態中執行。在此使用模型中，兩個範疇被提取至機器 (machine) 中，及同時存在於機器中。

第三使用模型包括指令從一個形式至另一個形式的 JIT (及時) 轉譯或編譯。在此使用模型中，經由軟體 (例如，JIT) 重排架構狀態。第三使用模型可適用於例如客戶至本機指令轉譯、虛

擬機器至本機指令轉譯、或將本機微指令重新映射/轉譯為更最佳化的本機微指令。

第四使用模型包括暫態環境切換(transient context switching)，而不用在從暫態環境返回後存檔及復原先前環境。此使用模型適用於可因若干理由發生的環境切換。一個此種理由例如可以是經由例外處置環境(exception handling context)對例外進行精確處置。第二、第三、及第四使用模型將在以下圖 14-17 的討論中進一步說明。

再次參考圖 13，架構 1300 包括實施上述 4 個使用模型的若干組件。統一影子暫存器檔案 1301 包括：第一部分，認可暫存器檔案 1302；第二部分，影子暫存器檔案 1303；及第三部分，最新指示符陣列(latest indicator array)1304。還包括推測撤回記憶體緩衝器(speculative retirement memory buffer，SMB)1342 及最新指示符陣列 1341。架構 1300 包含亂序架構(out of order architecture)，因此，架構 1300 另外包括重排緩衝器及撤回視窗(reorder buffer and retirement window)1332。重排及撤回視窗 1332 另外包括機器撤回指標(machine retirement pointer)1331、就緒位元陣列(ready bit array)1334，及每指令最新指示符(per instruction latest indicator)，諸如指示符 1333。

根據本發明之一具體實施例進一步詳細說明第一使用模型，即，硬體狀態更新的架構推測。如上述，架構 1300 包含亂序架構。架構 1300 的硬體能夠認可亂序指令結果(如，亂序載入及亂序儲存及亂序暫存器更新)。架構 1300 以上文在圖 12 的

討論中說明的方式利用統一影子暫存器檔案，以支援在認可暫存器及影子暫存器之間的推測執行。另外，架構 1300 利用推測載入儲存緩衝器(speculative load store buffer)及推測撤回記憶體緩衝器 1342 支援推測執行。

架構 1300 將使用這些組件與重排緩衝器及撤回視窗 1332，以允許其狀態正確撤回到認可暫存器檔案 1302 及可見記憶體 1350，即使機器以亂序方式將這些組件在內部撤回到統一影子暫存器檔案及撤回記憶體緩衝器。例如，該架構將使用統一影子暫存器檔案 1301 及推測記憶體 1342，以基於例外發生與否實施回復及認可事件。此功能性使暫存器狀態能夠將亂序撤回到統一影子暫存器檔案 1301，及使推測撤回記憶體緩衝器 1342 將亂序撤回到可見記憶體 1350。隨著推測執行繼續進行及亂序指令執行繼續進行，如果未漏掉預測任何分支且沒有發生例外，則機器撤回指標 1331 向前移動直到觸發認可事件。認可事件造成統一影子暫存器檔案藉由使其認可點向前移動而認可其環境，及造成推測撤回記憶體緩衝器根據機器撤回指標 1331 對記憶體 1350 認可其環境。

例如，考慮到在重排緩衝器及撤回視窗 1332 內顯示的指令 1-7，就緒位元陣列 1334 顯示準備要執行的"X"附加指令(beside instruction)，及未準備要執行的"/"附加指令。據此，指令 1、2、4、及 6 被允許以亂序繼續進行。其後，如果發生例外，諸如漏掉預測指令 6 分支，則可回復在指令 6 後發生的指令。或者，如果沒有任何例外發生，則可藉由相應地移動機器撤回指標 1331，認可所有指令 1-7。

使用最新指示符陣列 1341、最新指示符陣列 1304 及最新指示符 1333 以允許亂序執行。例如，即使指令 2 在指令 5 之前載入暫存器 R4，則一旦指令 5 準備發生，將忽略指令 2 的載入。最新載入將根據最新指示符取代之前的載入。

在重排緩衝器及撤回視窗 1332 內發生分支預測或例外時，將觸發回復事件。如上述，在發生回復時，統一影子暫存器檔案 1301 將回復至其最後一個認可點，及推測撤回記憶體緩衝器 1342 將被排清(flushed)。

圖 14 顯示根據本發明之一具體實施例包括雙範疇使用之第二使用模型的示意圖 1400。如上述，此使用模型可用於將 2 個執行緒提取至處理器中，其中一個執行緒在推測狀態中執行，及另一個執行緒在非推測狀態中執行。在此使用模型中，兩個範疇被提取至機器中，及同時存在於機器中。

如在示意圖 1400 中顯示，2 個範疇/軌跡 1401 及 1402 已被提取至機器中。在此實例中，範疇/軌跡 1401 是目前的非推測範疇/軌跡。範疇/軌跡 1402 是新的推測範疇/軌跡。架構 1300 啟用推測及暫用狀態(speculative and scratch state)，其允許 2 個執行緒使用這些執行狀態。一個執行緒(如，1401)在非推測範疇中執行，及另一個執行緒(如，1402)使用推測範疇。兩個範疇均可同時提取至機器中並同時存在，其中各個範疇不同地設定其相應的模式。第一個範疇是非推測性的，及另一個範疇是推測性的。因此第一個範疇在 CR/CM 模式中執行，及另一個在 SR/SM

模式中執行。在 CR/CM 模式中，讀取及寫入認可暫存器，及記憶體寫入進入記憶體。在 SR/SM 模式中，暫存器寫入進入 SSSR，及暫存器自最新寫入讀取，而記憶體寫入撤回記憶體緩衝器(SMB)。

一個實例將是排序的目前範疇(如，1401)及推測性的下一個範疇(如，1402)。因為在目前範疇之後提取下一個範疇，所以相依性將會受到重視，因此兩個範疇均可在機器中執行。例如，在範疇 1401 中，在「認可 SSSR 為 CR」處，直至此時的暫存器及記憶體處於 CR 模式，而程式碼在 CR/CM 模式中執行。在範疇 1402 中，程式碼在 SR 及 SM 模式中執行，並在發生例外時可回復。以此方式，兩個範疇在機器中同時執行，但各在不同模式中執行並相應地讀取及寫入暫存器。

圖 15 顯示根據本發明之一具體實施例之第三使用模型的示意圖 1500，其包括暫態環境切換，在從暫態環境返回後，不必存檔及復原先前環境。如上述，此使用模型適用於可因若干理由發生的環境切換。一個此種理由例如可以是經由例外處置環境對例外進行精確處置。

第三使用模型發生於機器正執行轉譯碼及其遇到環境切換時(如，轉譯碼中的例外或需要後續程式碼轉譯時)。在目前的範疇(如，在例外之前)中，SSSR 及 SMB 尚未認可其對客戶架構狀態的推測狀態。目前狀態在 SR/SM 模式中運行。在例外發生時，機器切換至例外處置器(exception handler)(如，轉換器)以精確地處理例外。插入回復，這造成暫存器狀態回復為 CR 及 SMB

被排清。轉換器碼將在 SR/CM 模式中運行。在執行轉換器碼期間，SMB 將其環境撤回到記憶體而不用等待認可事件。暫存器被寫入 SSSR 且不用更新 CR。其後，當轉換器結束時且在切換回執行的轉換碼之前，轉換器回復為 SSSR(如，回復 SSSR 為 CR)。在此程序期間，最後認可的暫存器狀態是在 CR 中。

這顯示在示意圖 1500 中，其中先前的範疇/軌跡 1501 已從 SSSR 認可為 CR。目前的範疇/軌跡 1502 是推測性的。暫存器及記憶體及此範疇是推測性的，及執行在 SR/SM 模式下發生。在此實例中，例外發生在範疇 1502 中，及程式碼在轉譯之前需要以原始順序重新執行。此時，SSSR 被回復及 SMB 被排清。接著，JIT 程式碼 1503 執行。JIT 程式碼將 SSSR 回復至範疇 1501 的末端並排清 SMB。在 SR/CM 模式下執行 JIT。在 JIT 結束時，SSSR 被回復為 CR，及目前的範疇/軌跡 1504 接著在 CR/CM 模式中以原始的轉譯順序重新執行。以此方式，以確切的目前順序精確地處置例外。

圖 16 顯示根據本發明之一具體實施例描繪指令序列中的例外是因為後續程式碼需要轉譯之案例的示意圖 1600。如在示意圖 1600 中顯示，先前的範疇/軌跡 1601 以遠跳(far jump)至未轉譯的目的地而結束。在跳至遠跳目的地之前，認可 SSSR 為 CR。JIT 程式碼 1602 接著執行以轉譯遠跳目的地的客戶指令(如，以建立本機指令的新軌跡)。在 SR/CM 模式下執行 JIT。在 JIT 執行結束時，暫存器狀態從 SSSR 回復為 CR，及由 JIT 轉譯的新範疇/軌跡 1603 開始執行。新的範疇/軌跡繼續在 SR/SM 模式中從先前範疇/軌跡 1601 的最後認可點執行。

圖 17 顯示根據本發明之一具體實施例之第四使用模型的示意圖 1700，其包括暫態環境切換，在從暫態環境返回後，不必存檔及復原先前環境。如上述，此使用模型適用於可因若干理由發生的環境切換。一個此種理由例如可以是經由例外處置環境處理輸入或輸出。

示意圖 1700 顯示在 CR/CM 模式下執行的先前範疇/軌跡 1701 以叫用(call)函數 F1 而結束的案例。直至此時的暫存器狀態從 SSSR 認可為 CR。函數 F1 範疇/軌跡 1702 接著開始在 SR/CM 模式下以推測的方式執行。函數 F1 接著以返回主範疇/軌跡 1703 而結束。此時，暫存器狀態從 SSSR 回復為 CR。主範疇/軌跡 1703 重返在 CR/CM 模式中執行。

圖 18 顯示根據本發明之一具體實施例之例示性微處理器管線(microprocessor pipeline)1800 的圖式。微處理器管線 1800 包括實施上述硬體加速轉換程序之功能性的硬體轉換加速器。在圖 18 具體實施例中，硬體轉換加速器耦接至提取模組 1801，其後是解碼模組 1802、分配模組 1803、分派模組(dispatch module)1804、執行模組 1805 及撤回模組 1806。應注意，微處理器管線 1800 只是實施本發明之具體實施例上述功能性之管線的一個實例。熟習本技術者應瞭解，可實施包括上述解碼模組功能性的其他微處理器管線。

為了解說的目的，已參考特定具體實施例做出以上說明。然而，以上闡釋之討論的目的不在詳盡窮舉或限制本發明於揭

示的精確形式。可按照以上教示進行許多修改及變化。具體實施例的選擇與說明係為了對本發明的原理及實際應用提出最好的解說，藉此讓熟習本技術者以適於所想特定用途的各種修改，充分利用本發明及各種具體實施例。

【圖式簡單說明】

本發明藉由舉例而非限制，以附圖的各個圖式進行解說，圖中相似參考數字代表相似元件。

圖 1 顯示本發明之一具體實施例所操作的例示性指令序列。

圖 2 顯示根據本發明之一具體實施例描繪基於區塊的轉譯程序的示意圖，其中將客戶指令區塊轉換為本機轉換區塊。

圖 3 顯示根據本發明之一具體實施例圖解將客戶指令區塊的各個指令轉換為本機轉換區塊之對應的本機指令之方式的示意圖。

圖 4 顯示根據本發明之一具體實施例圖解以處置本機轉換區塊處理較遠分支之方式的示意圖。

圖 5 顯示根據本發明之一具體實施例之例示性硬體加速轉換系統的示意圖，其圖解將客戶指令區塊及其對應的本機轉換區塊儲存在快取中的方式。

圖 6 顯示根據本發明之一具體實施例之硬體加速轉換系統的詳細實例。

圖 7 顯示根據本發明之一具體實施例具有輔助軟體加速轉換管線之硬體加速轉換系統的實例。

圖 8 顯示根據本發明之一具體實施例圖解 CLB 結合程式碼快取及儲存於記憶體中的客戶指令至本機指令映射而發揮功能之方式的例示性流程圖。

圖 9 顯示根據本發明之一具體實施例圖解實體儲存堆疊程式碼快取實施方案及客戶指令至本機指令映射的例示性流程圖。

圖 10 顯示根據本發明之一具體實施例描繪硬體加速轉換系統之額外例示性細節的示意圖。

圖 11A 顯示本發明之具體實施例所實施之例示性樣式匹配程序的示意圖。

圖 11B 顯示根據本發明之一具體實施例之基於 SIMD 暫存器之樣式匹配程序的示意圖。

圖 12 顯示根據本發明之一具體實施例之統一暫存器檔案的圖式。

圖 13 顯示根據本發明之一具體實施例支援推測架構狀態及暫態架構狀態之統一影子暫存器檔案及管線架構 1300 的示意圖。

圖 14 顯示根據本發明之一具體實施例包括雙範疇使用之第二使用模型的示意圖。

圖 15 顯示根據本發明之一具體實施例之第三使用模型的示意圖，其包括暫態環境切換，在從暫態環境返回後，不必存檔及復原先前環境。

圖 16 顯示根據本發明之一具體實施例描繪指令序列中的例外是因為後續程式碼需要轉譯之案例的示意圖。

圖 17 顯示根據本發明之一具體實施例之第四使用模型的示意圖，其包括暫態環境切換，在從暫態環境返回後，不必存檔及復原先前環境。

圖 18 顯示根據本發明之一具體實施例之例示性微處理器管線的圖式。

【主要元件符號說明】

100	指令序列
101~104	分支指令
201	客戶指令區塊
202	本機轉換區塊
301	客戶指令緩衝器
302	本機指令緩衝器
401	記憶體中的客戶指令序列
402	記憶體中的本機指令序列
411	客戶指令較遠分支
412	對應的本機指令客戶較遠分支
500	硬體加速轉換系統
501	系統記憶體
502	客戶提取邏輯單元
503	客戶提取緩衝器
504	轉換表
505	本機轉換緩衝器
506	轉換後備緩衝器
507	本機快取
508	處理器
600	硬體加速轉換系統
601	系統記憶體
602	客戶碼
603	轉換後備緩衝器
604	最佳化器碼

605	轉換器碼
606	本機碼快取
607	共用硬體快取
608	緩衝器
609	TLB
610	客戶硬體快取
611	客戶提取緩衝器
612	轉換表
613	轉換表
614	多工器
615	多工器
616	本機轉換緩衝器
620	客戶提取邏輯
630	轉換後備緩衝器
631	轉換的區塊登錄點位址
632	本機位址
633	轉換的位址範圍
634	程式碼快取及轉換後備緩衝器管理位元
635	動態分支偏差值位元
640	提取邏輯
650	處理器
700	硬體加速轉換系統
711	客戶提取緩衝器
712	轉換表
713	轉換表
716	本機轉換緩衝器

760	特殊高速記憶體
801	系統記憶體
802	客戶位址
804	CLB 硬體快取
805	虛線方框
806	程式碼快取記憶體
901	實體儲存堆疊
902	標籤結構
903	延伸方式標籤結構
1000	硬體加速轉換系統
1001	直線
1002	表格
1003	表格
1004	第二級表格
1100	示意圖
1101	傳入樣式
1102~1105	SIMD 暫存器
1201	統一暫存器檔案
1202~1203	部分
1205	項目選擇器
1300	統一影子暫存器檔案及管線架構
1301	統一影子暫存器檔案
1302	認可暫存器檔案
1303	影子暫存器檔案
1304	最新指示符陣列
1331	機器撤回指標

1332	重排緩衝器及撤回視窗
1333	最新指示符
1334	就緒位元陣列
1341	最新指示符陣列
1342	推測撤回記憶體緩衝器
1350	可見記憶體
1400	示意圖
1401	目前的非推測範疇/軌跡
1402	新的推測範疇/軌跡
1500	示意圖
1501	先前的範疇/軌跡
1502	目前的範疇/軌跡
1503	JIT 程式碼
1504	目前的範疇/軌跡
1600	示意圖
1601	先前的範疇/軌跡
1602	JIT 程式碼
1603	新的範疇/軌跡
1700	示意圖
1701	先前的範疇/軌跡
1702	函數 F1 範疇/軌跡
1703	主範疇/軌跡
1800	微處理器管線
1801	提取模組
1802	解碼模組
1803	分配模組

1804	分派模組
1805	執行模組
1806	撤回模組
1807	統一影子暫存器檔案
1808	推測記憶體緩衝器
1809	全域可見記憶體/快取

發明專利說明書

(本說明書格式、順序，請勿任意更動，※記號部分請勿填寫)

※申請案號：101102835

※申請日：101年01月30日

※IPC分類：G06F15/82(2006.01)

G06F9/45(2006.01)

一、發明名稱：(中文/英文)

用於將客戶指令轉譯為本機指令之硬體加速組件

HARDWARE ACCELERATION COMPONENTS FOR
TRANSLATING GUEST INSTRUCTIONS TO NATIVE
INSTRUCTIONS

二、中文發明摘要：

本發明揭示一種硬體為基轉譯加速器。該硬體包括：
一客戶提取邏輯組件，用於存取客戶指令；一客戶提取緩衝器，其耦接至該客戶提取邏輯組件及一分支預測組件，用於將客戶指令組合成一客戶指令區塊；及轉換表，其耦接至該客戶提取緩衝器，用於將該客戶指令區塊轉譯為一對應的本機轉換區塊。該硬體另外包括：一本機快取，其耦接至該等轉換表，用於儲存該對應的本機轉換區塊；及一轉換後備緩衝器，其耦接至該本機快取，用於儲存該客戶指令區塊至對應的本機轉換區塊的一映射，其中在收到針對客戶指令的後續請求時，索引該轉換後備緩衝器以決定是否發生命中，其中該映射指示該客戶指令在該本機快取中具有一對應的轉換本機指令。

三、英文發明摘要：

A hardware based translation accelerator. The hardware

includes a guest fetch logic component for accessing guest instructions; a guest fetch buffer coupled to the guest fetch logic component and a branch prediction component for assembling guest instructions into a guest instruction block; and conversion tables coupled to the guest fetch buffer for translating the guest instruction block into a corresponding native conversion block. The hardware further includes a native cache coupled to the conversion tables for storing the corresponding native conversion block, and a conversion look aside buffer coupled to the native cache for storing a mapping of the guest instruction block to corresponding native conversion block, wherein upon a subsequent request for a guest instruction, the conversion look aside buffer is indexed to determine whether a hit occurred, wherein the mapping indicates the guest instruction has a corresponding converted native instruction in the native cache.

七、申請專利範圍：

1. 一種硬體為基轉譯加速器，包含：

- 一客戶提取邏輯組件，用於存取複數個客戶指令；
- 一客戶提取緩衝器，其耦接至該客戶提取邏輯組件及一分支預測組件，用於將該複數個客戶指令組合成一客戶指令區塊；
- 複數個轉換表，其耦接至該客戶提取緩衝器，用於將該客戶指令區塊轉譯為一對應的本機轉換區塊；
- 一本機快取，其耦接至該等轉換表，用於儲存該對應的本機轉換區塊；
- 一轉換後備緩衝器，其耦接至該本機快取，用於儲存該客戶指令區塊至對應的本機轉換區塊的一映射；
- 其中在收到針對一客戶指令的一後續請求時，索引該轉換後備緩衝器以決定是否發生一命中，其中該映射指示該客戶指令在該本機快取中具有一對應的轉換本機指令；及
- 回應於該命中，該轉換後備緩衝器傳送轉譯的本機指令以便執行。

2. 如申請專利範圍第 1 項所述之硬體為基轉譯加速器，其中該客戶提取邏輯組件存取與一處理器無關之該複數個客戶指令。

3. 如申請專利範圍第 1 項所述之硬體為基轉譯加速器，其中該轉換後備緩衝器包含使用替換策略以維持儲存於其中的最常遇到的本機轉換區塊的一快取。

4. 如申請專利範圍第 1 項所述之硬體為基轉譯加速器，其中在一系統記憶體內維持一轉換緩衝器，及在該轉換後備緩衝器及該

轉換緩衝器之間維持快取一致性。

5. 如申請專利範圍第 4 項所述之硬體為基轉譯加速器，其中該轉換緩衝器大於該轉換後備緩衝器，及使用寫回策略以在該轉換緩衝器及該轉換後備緩衝器之間維持一致性。

6. 如申請專利範圍第 1 項所述之硬體為基轉譯加速器，其中將該轉換後備緩衝器實施為耦接至一處理器之一管線的一高速低潛時快取記憶體。

7. 一種用於一處理器以將客戶指令加速轉譯為本機指令之系統，包含：

一客戶提取邏輯組件，用於存取複數個客戶指令；

一客戶提取緩衝器，其耦接至該客戶提取邏輯組件及一分支預測組件，用於將該複數個客戶指令組合成一客戶指令區塊；

複數個轉換表，其耦接至該客戶提取緩衝器，用於將該客戶指令區塊轉譯為一對應的本機轉換區塊；

一本機快取，其耦接至該等轉換表，用於儲存該對應的本機轉換區塊；

一轉換後備緩衝器，其耦接至該本機快取，用於儲存該客戶指令區塊至對應的本機轉換區塊的一映射；

其中在收到針對一客戶指令的一後續請求時，索引該轉換後備緩衝器以決定是否發生一命中，其中該映射指示該客戶指令在該本機快取中具有一對應的轉換本機指令；及

回應於該命中，該轉換後備緩衝器傳送轉譯的本機指令以便執行。

8. 如申請專利範圍第 7 項所述之系統，其中該客戶提取邏輯組件存取與該處理器無關之該複數個客戶指令。

9. 如申請專利範圍第 7 項所述之系統，其中該轉換後備緩衝器包含使用替換策略以維持儲存於其中的最常遇到的本機轉換區塊的一快取。

10. 如申請專利範圍第 7 項所述之系統，其中在一系統記憶體內維持一轉換緩衝器，及在該轉換後備緩衝器及該轉換緩衝器之間維持快取一致性。

11. 如申請專利範圍第 10 項所述之系統，其中該轉換緩衝器大於該轉換後備緩衝器，及使用寫回策略以在該轉換緩衝器及該轉換後備緩衝器之間維持一致性。

12. 如申請專利範圍第 7 項所述之系統，其中將該轉換後備緩衝器實施為耦接至該處理器之一管線的一高速低潛時快取記憶體。

13. 一種實施轉譯指令之一方法的微處理器，該微處理器包含：
 一微處理器管線；
 一硬體加速器模組，其耦接至該微處理器管線，其中該硬體加速器模組另外包含：
 一客戶提取邏輯組件，用於存取複數個客戶指令；
 一客戶提取緩衝器，其耦接至該客戶提取邏輯組件及一分支預測組件，用於將該複數個客戶指令組合成一客戶指令區塊；

複數個轉換表，其耦接至該客戶提取緩衝器，用於將該客戶指令區塊轉譯為一對應的本機轉換區塊；

一本機快取，其耦接至該等轉換表，用於儲存該對應的本機轉換區塊；

一轉換後備緩衝器，其耦接至該本機快取，用於儲存該客戶指令區塊至對應的本機轉換區塊的一映射；

其中在收到針對一客戶指令的一後續請求時，索引該轉換後備緩衝器以決定是否發生一命中，其中該映射指示該客戶指令在該本機快取中具有一對應的轉換本機指令；及

回應於該命中，該轉換後備緩衝器傳送轉譯的本機指令以便執行。

14. 如申請專利範圍第 13 項所述之微處理器，其中該客戶提取邏輯組件存取與該微處理器無關之該複數個客戶指令。

15. 如申請專利範圍第 13 項所述之微處理器，其中該轉換後備緩衝器包含使用替換策略以維持儲存於其中的最常遇到的本機轉換區塊的一快取。

16. 如申請專利範圍第 13 項所述之微處理器，其中在一系統記憶體內維持一轉換緩衝器，及在該轉換後備緩衝器及該轉換緩衝器之間維持快取一致性。

17. 如申請專利範圍第 16 項所述之微處理器，其中該轉換緩衝器大於該轉換後備緩衝器，及使用寫回策略以在該轉換緩衝器及該轉換後備緩衝器之間維持一致性。

18. 如申請專利範圍第 13 項所述之微處理器，其中將該轉換後備緩衝器實施為耦接至該微處理器管線的一高速低潛時快取記憶體。

19. 如申請專利範圍第 13 項所述之微處理器，其中該硬體加速器模組的功能包含一平行客戶指令提取管線(parallel guest instruction fetch pipeline)，其平行於一本機微處理器提取管線(native microprocessor fetch pipeline)而發揮功能。

20. 一種實施轉譯指令之一方法的微處理器，該微處理器包含：
 一微處理器管線；
 一加速器模組，其包含耦接至該微處理器管線的高速記憶體，其中該加速器模組另外包含：
 一客戶提取邏輯器，用於存取複數個客戶指令；
 一客戶提取記憶體，其耦接至該客戶提取邏輯器，用於將該複數取記戶指令組合成一客戶指令區塊；
 複數個轉換表，用於將該客戶指令區塊轉譯為一對應的本機轉換區塊；

一本機轉換緩衝器，用於儲存該對應的本機轉換區塊；
 其中在收到針對一客戶指令的一後續請求時，索引儲存該客戶指令區塊至對應的本機轉換區塊之一映射的一轉換後備緩衝器以決定是否發生一命中，其中該映射指示該客戶指令在該本機快取中具有一對應的轉換本機指令；及

回應於該命中，該轉換後備緩衝器傳送轉譯的本機指令以便執行。

21. 如申請專利範圍第 20 項所述之微處理器，其中該高速記憶體包含該微處理器之一 L0 快取。

22. 如申請專利範圍第 20 項所述之微處理器，其中該加速器模組另外包含該微處理器之一載入儲存指令提取路徑。

八、圖式：

100

1	Inst A	
2	Inst B	
3	Inst C	
4	Branch.c1 (11, 6)	101
5	Inst D	
6	Inst E	
7	Branch.c2 (10, 2)	102
8	Inst F	
9	Branch.c3 (15, 5)	103
10	Inst G	
11	Inst H	
12	Inst I	
13	Inst J	
14	Branch.c4 (16, 1)	104
15	Inst K	
16	Far Branch	

圖 1

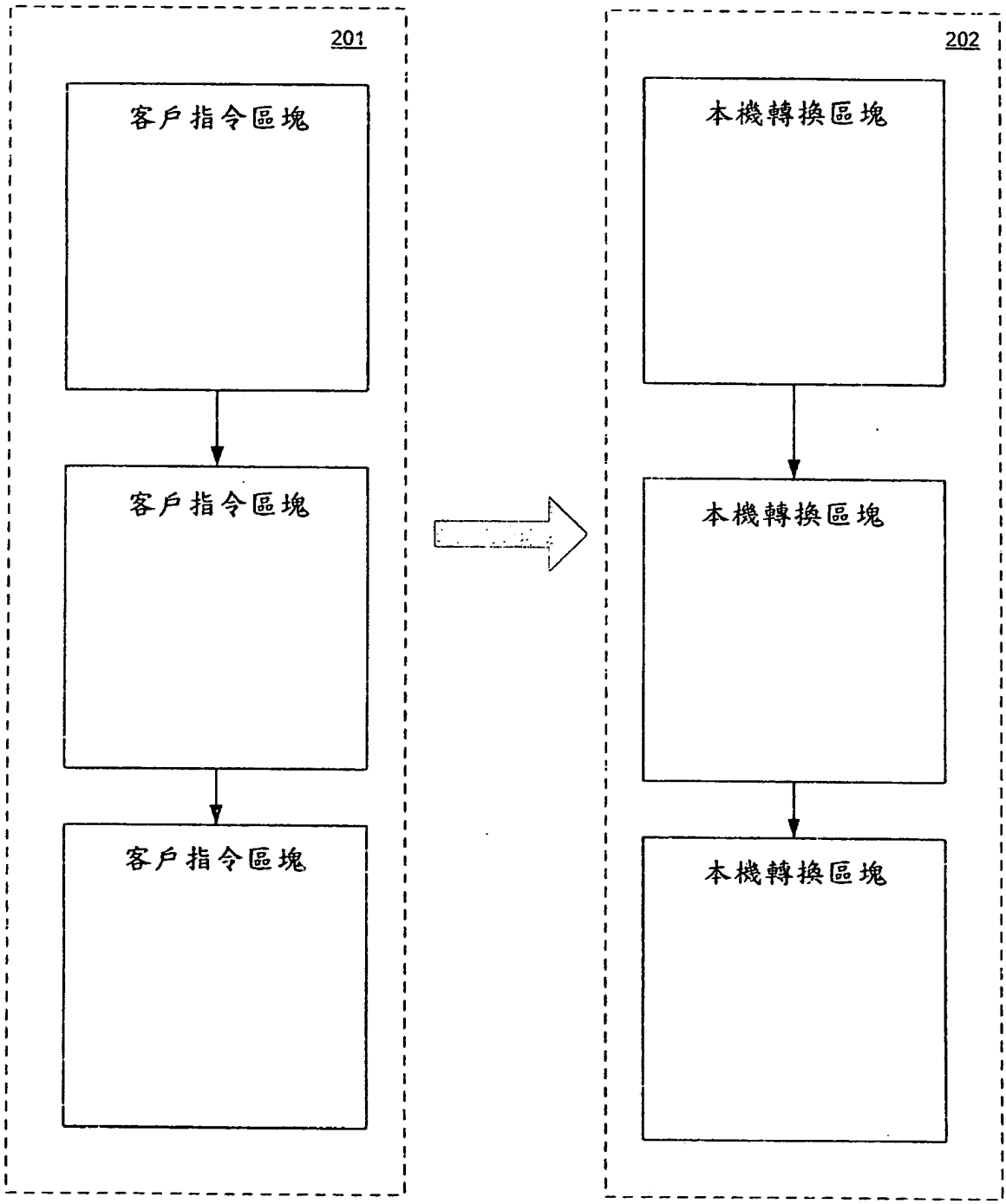


圖2

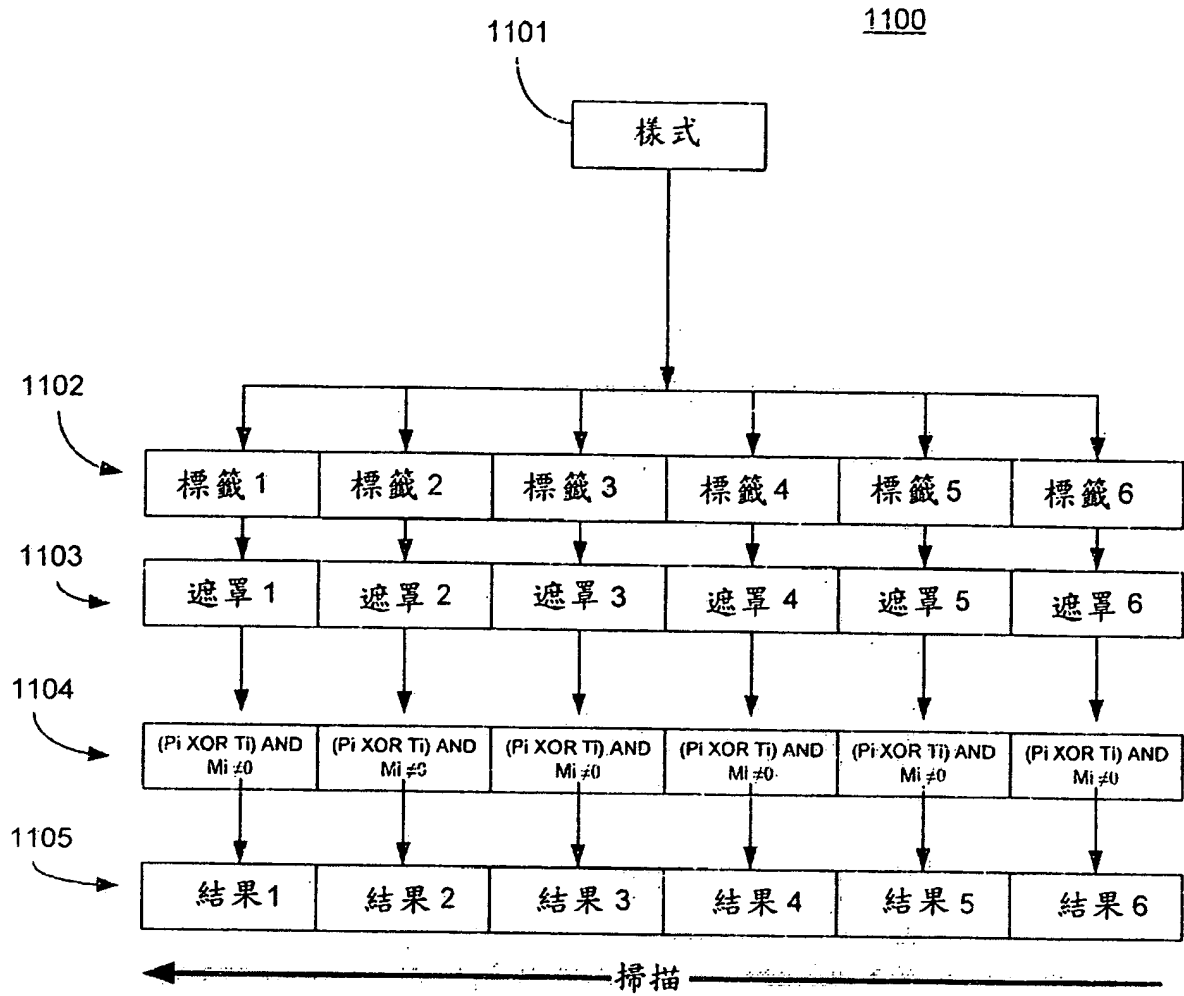


圖 11B

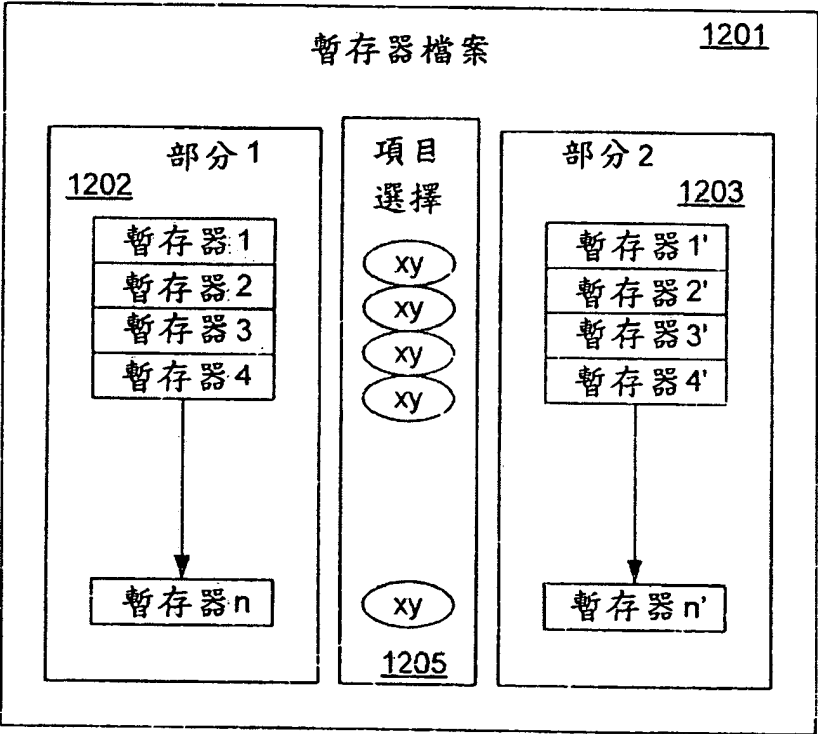


圖 12

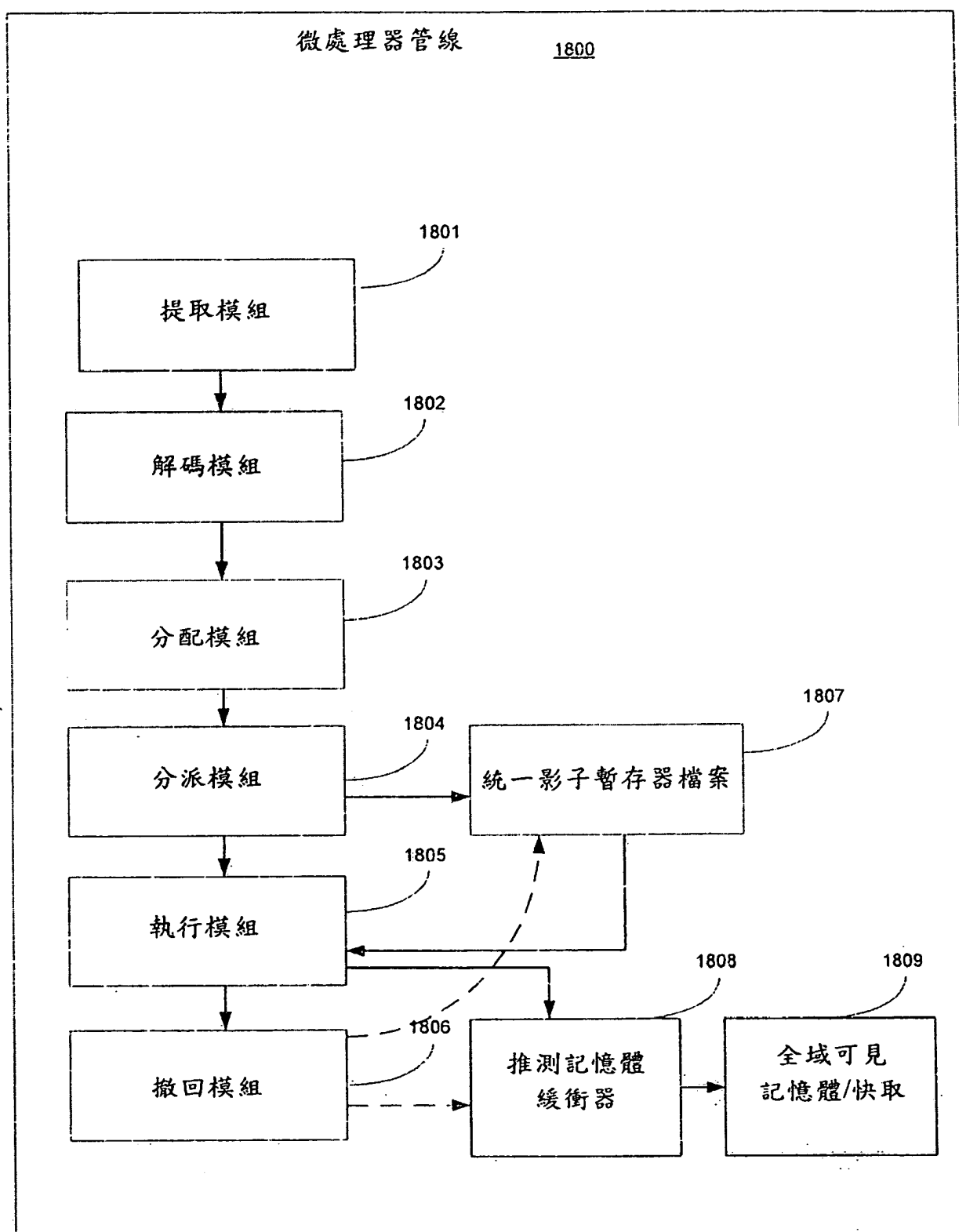


圖 18

四、指定代表圖：

(一)本案指定代表圖為：圖 5。

(二)本代表圖之元件符號簡單說明：

500	硬體加速轉換系統
501	系統記憶體
502	客戶提取邏輯
503	客戶提取緩衝器
504	轉換表
505	本機轉換緩衝器
506	轉換後備緩衝器
507	本機指令緩衝器
508	處理器

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：無。