



(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 I574166 B

(45)公告日：中華民國 106 (2017) 年 03 月 11 日

(21)申請案號：104139443 (22)申請日：中華民國 104 (2015) 年 11 月 26 日
 (51)Int. Cl. : G06F15/80 (2006.01) G06F9/30 (2006.01)
 (30)優先權：2014/12/27 美國 14/583,644
 (71)申請人：英特爾股份有限公司(美國) INTEL CORPORATION (US)
 美國
 (72)發明人：柯柏 吉瑟斯 CORBAL, JESUS (ES)；歐德亞麥德維爾 艾蒙斯特阿法 OULD-AHMED-VALL, ELMOUSTAPHA (MR)；瓦倫泰 羅柏 VALENTINE, ROBERT (US)；查尼 馬克 CHARNEY, MARK J. (US)；吉卡 密林德 GIRKAR, MILIND B. (US)；托爾 布萊特 TOLL, BRET L. (US)；艾斯帕薩 羅傑 ESPASA, ROGER (ES)；索羅 吉勒姆 SOLE, GUILLEM (ES)；巴拉特 傑洛 BALART, JAIRO (ES)；希克曼 布萊恩 HICKMANN, BRIAN J. (US)
 (74)代理人：林志剛
 (56)參考文獻：
 TW 201241774A US 6594754B1
 審查人員：彭智輝
 申請專利範圍項數：24 項 圖式數：15 共 87 頁

(54)名稱

利用索引和立即數執行向量排列的方法和設備

METHOD AND APPARATUS FOR PERFORMING A VECTOR PERMUTE WITH AN INDEX AND AN IMMEDIATE

(57)摘要

本發明揭示用於執行向量排列的方法和設備。例如，處理器的一個實施例包含：來源向量暫存器，用以儲存複數個來源資料元件；目的地向量暫存器，用以儲存複數個目的地資料元件；控制向量暫存器，用以儲存複數個控制資料元件，各控制資料元件對應於該些目的地資料元件中的一者且包括 N 位元值，該 N 位元值指示來源資料元件是否被複製到該對應的目的地資料元件；向量排列邏輯，用以將各控制資料元件的該 N 位元值與立即數的 N 位元部分比較以決定是否將來源資料元件複製到對應的目的地資料元件，其中如果該些 N 位元值匹配，則該向量排列邏輯是用以使用包括在該控制資料元件中的索引值來識別來源資料元件，以及用以響應地將該來源資料元件複製到該目的地向量暫存器中的該對應的目的地資料元件。

An apparatus and method for performing a vector permute. For example, one embodiment of a processor comprises: a source vector register to store a plurality of source data elements; a destination vector register to store a plurality of destination data elements; a control vector register to store a plurality of control data elements, each control data element corresponding to one of the destination data elements and including an N bit value indicating whether a source data element is to be copied to the corresponding destination data element; vector permute logic to compare the N bit value of each control data element to an N bit portion of an immediate to determine whether to copy a source data element to the corresponding destination data element, wherein if the N bit values match, then the vector permute logic is to identify a source data element

using an index value included in the control data element and to responsively copy the source data element to the corresponding destination data element in the destination vector register.

指定代表圖：

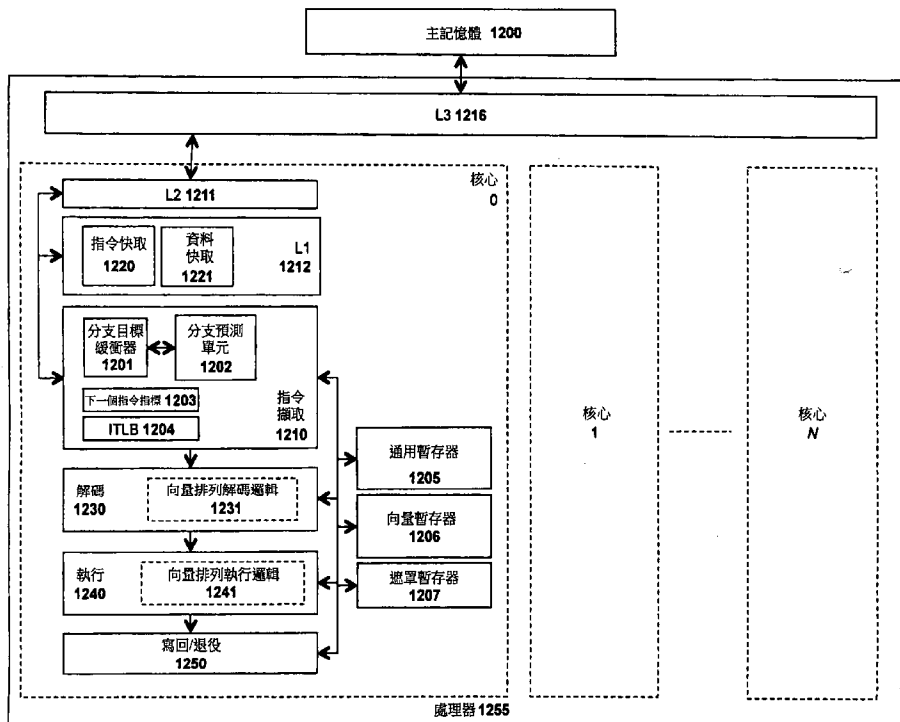


圖 12

符號簡單說明：

- 1200 . . . 主記憶體
- 1201 . . . 分支目標緩衝器
- 1202 . . . 分支預測單元
- 1203 . . . 下一個指令指標
- 1204 . . . 指令轉譯後備緩衝器
- 1205 . . . 通用暫存器
- 1206 . . . 向量暫存器
- 1207 . . . 遮罩暫存器
- 1210 . . . 指令擷取單元
- 1211 . . . 第 2 階 (L2)快取
- 1212 . . . 第 1 階 (L1)快取
- 1216 . . . 第 3 階 (L3)快取
- 1220 . . . 指令快取
- 1221 . . . 資料快取
- 1230 . . . 解碼單元
- 1231 . . . 向量排列解碼邏輯
- 1240 . . . 執行單元
- 1241 . . . 向量排列執行邏輯
- 1250 . . . 寫回/退役
- 1255 . . . 處理器

發明摘要

※申請案號：104139443

※申請日：104年11月26日

※IPC分類：G06F 15/80 (2006:01)
9/30 (2006:01)

【發明名稱】(中文/英文)

利用索引和立即數執行向量排列的方法和設備

Method and apparatus for performing a vector permute with an index and an immediate

【中文】

本發明揭示用於執行向量排列的方法和設備。例如，處理器的一個實施例包含：來源向量暫存器，用以儲存複數個來源資料元件；目的地向量暫存器，用以儲存複數個目的地資料元件；控制向量暫存器，用以儲存複數個控制資料元件，各控制資料元件對應於該些目的地資料元件中的一者且包括 N 位元值，該 N 位元值指示來源資料元件是否被複製到該對應的目的地資料元件；向量排列邏輯，用以將各控制資料元件的該 N 位元值與立即數的 N 位元部分比較以決定是否將來源資料元件複製到對應的目的地資料元件，其中如果該些 N 位元值匹配，則該向量排列邏輯是用以使用包括在該控制資料元件中的索引值來識別來源資料元件，以及用以響應地將該來源資料元件複製到該目的地向量暫存器中的該對應的目的地資料元件。

【 英文 】

An apparatus and method for performing a vector permute. For example, one embodiment of a processor comprises: a source vector register to store a plurality of source data elements; a destination vector register to store a plurality of destination data elements; a control vector register to store a plurality of control data elements, each control data element corresponding to one of the destination data elements and including an N bit value indicating whether a source data element is to be copied to the corresponding destination data element; vector permute logic to compare the N bit value of each control data element to an N bit portion of an immediate to determine whether to copy a source data element to the corresponding destination data element, wherein if the N bit values match, then the vector permute logic is to identify a source data element using an index value included in the control data element and to responsively copy the source data element to the corresponding destination data element in the destination vector register.

【代表圖】

【本案指定代表圖】：第(12)圖。

【本代表圖之符號簡單說明】：

- 1200：主記憶體
- 1201：分支目標緩衝器
- 1202：分支預測單元
- 1203：下一個指令指標
- 1204：指令轉譯後備緩衝器
- 1205：通用暫存器
- 1206：向量暫存器
- 1207：遮罩暫存器
- 1210：指令擷取單元
- 1211：第2階(L2)快取
- 1212：第1階(L1)快取
- 1216：第3階(L3)快取
- 1220：指令快取
- 1221：資料快取
- 1230：解碼單元
- 1231：向量排列解碼邏輯
- 1240：執行單元
- 1241：向量排列執行邏輯
- 1250：寫回單元
- 1255：處理器

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：
無

發明專利說明書

(本說明書格式、順序，請勿任意更動)

【發明名稱】(中文/英文)

利用索引和立即數執行向量排列的方法和設備

Method and apparatus for performing a vector permute with an index and an immediate

● 【技術領域】

本發明一般係關於電腦處理器的領域。尤其是，本發明係關於一種用於利用索引和立即數執行向量排列的方法和設備。

【先前技術】

指令集、或指令集架構 (ISA) 是與編程相關的電腦架構的部分，包括本地資料類型、指令、暫存器架構、定址模式、記憶體架構、中斷和異常處理以及外部輸入和輸出 (I/O)。應當指出的是，術語“指令”在本說明書中一般是指巨集指令 (macro-instructions) - 其為提供給處理器用於執行的指令 - 與微指令 (micro-instructions) 或微運算 (micro-ops) 相反 - 其是解碼巨集指令的處理器的解碼器的結果。微指令或微運算可以被配置以指示處理器上的執行單元以執行運算來實施與巨集指令相關的邏輯。

ISA 與微架構不同，其為一組用於實施指令集的處理器設計技術。具有不同微架構的處理器可以共享一通用的

指令集。例如，Intel® Pentium 4 處理器、Intel® Core™ 處理器、以及來自加州的桑尼維爾的 Advanced Micro Devices, Inc.的處理器，實施幾乎相同的版本的 x86 指令集（已經加入更新版本的一些擴展），但有不同的內部設計。例如，ISA 相同的暫存器架構可以使用習知技術以不同的微架構、不同的方式來實施，包括專用的實體暫存器、使用暫存器重新命名機制的一或多個動態分配實體暫存器（例如，使用暫存器別名表（Register Alias Table, RAT）、重新排序緩衝器（Reorder Buffer, ROB）和退休暫存器檔）。除非另有說明，在本說明書中所用的短語暫存器架構、暫存器檔和暫存器是用於指代是對軟體/程式設計師可見的那些，和指令指定的暫存器的方式。在需要區別之處，形容詞“邏輯（的）”、“架構（的）”或“軟體可見（的）”將被用來表示在暫存器架構中的暫存器/檔，而不同的形容詞將被用以指定在給定的微架構中的暫存器（例如，實體暫存器、重新排序緩衝器、退休暫存器、暫存器池）。

指令集包括一或多個指令格式。給定的指令格式定義各種欄位（位元數、位元位置）以，除了其他事項之外，指定將要被執行的運算和將要對其執行運算的運算元。透過指令模板（或子格式）的定義，有些指令格式進一步被細分。例如，給定的指令格式的指令模板可以被定義為具有不同子集的指令格式的欄位（所包括的欄位通常係相同的順序，但至少有一些具有不同的位元位置，因為其包括

有更少欄位) 及 / 或被定義為具有不同解釋的給定欄位。給定的指令是使用給定的指令格式 (以及如果被定義, 在該指令格式的指令模板中給定的一者之中) 來表達, 並指定運算和運算元。指令串流是指令的特定順序, 其中該序列中的每個指令是以指令格式的指令的發生 (以及如果被定義, 該指令格式的指令模板的給定的一者) 。

● 【圖式簡單說明】

更好地理解本發明的可以從結合以下附圖與以下詳細描述來獲得, 其中:

圖 1A 和 1B 是根據本發明的實施例, 示出通用向量友好指令格式和其指令模板的方框圖;

圖 2A-D 是根據本發明的實施例, 示出示例性特定向量友好指令格式的方框圖;

圖 3 是根據本發明的一個實施例的暫存器架構的方框圖; 以及

圖 4A 是根據本發明的實施例, 示出一示例性的有序之擷取、解碼、退休管線和一示例性的暫存器重新命名、無序發出/執行管線兩者的方框圖;

圖 4B 是根據本發明的實施例, 示出被包括在處理器中的一示例性實施例的有序之擷取、解碼、退休核心和一示例性的暫存器重新命名、無序發出/執行架構核心兩者的方框圖;

圖 5A 是單一處理器核心的方框圖, 除了它的連接至

晶粒上互連網路外；

圖 5B 是根據本發明實施例示出在圖 5A 中的處理器核心的一部分的擴展圖；

圖 6 是根據本發明實施例的具有整合的記憶體控制器和圖形的單核處理器和多核處理器的方框圖；

圖 7 根據本發明的一個實施例示出系統的方框圖；

圖 8 根據本發明的實施例示出第二系統的方框圖；

圖 9 根據本發明的實施例示出第三系統的方框圖；

圖 10 根據本發明的實施例示出片上系統（system on a chip, SoC）的方框圖；

圖 11 根據本發明的實施例示出對比軟體指令轉換器之使用以將在來源指令集中的二進制指令轉換成在目標指令集中的二進制指令的方框圖；

圖 12 示出本發明的實施例可以在其上實施的示例性處理器；

圖 13 根據使用立即數值的本發明的實施例示出向量排列邏輯；

圖 14 示出可能如何執行一排列，其使用來自不同的來源暫存器之資料元件；以及

圖 15 根據本發明的一個實施例示出方法。

【發明內容及實施方式】

在以下的描述中，出於解釋的目的，許多具體的細節被闡述以便提供徹底理解本發明下述的實施例。然而，這

將對本發明領域中的通常知識者是顯而易見的，本發明之實施例可在沒有這些具體細節的情況下實踐。在其他實施例中，習知的結構和裝置以方框圖的形式示出，以避免模糊本發明的實施例的基本原理。

示例性處理器架構和資料類型

指令集包括一或多個指令格式。給定的指令格式定義各種欄位（位元數、位元位置）以，除了其他事項之外，指定將要執行的運算（opcode, 運算碼）和將要對其執行運算的運算元。透過指令模板（或子格式）的定義，有些指令格式進一步被細分。例如，給定的指令格式的指令模板可以被定義為具有不同子集的指令格式的欄位（所包括的欄位通常係相同的順序，但至少有一些具有不同的位元位置，因為其包括有更少欄位）及／或被定義為具有不同解釋的給定欄位。因此，每一個 ISA 的指令是使用給定的指令格式（以及如果被定義，在該指令格式的指令模板中給定的一者之中）來表達，並且包括用於指定運算和運算元的欄位。例如，示例性 ADD 指令具有特定的運算碼和包括運算碼欄位的指令格式，以指定運算碼和運算元欄位來選擇運算元（來源 1/目的地和來源 2）；以及在指令串流中的此 ADD 指令的出現將在選擇特定的運算元的運算元欄位中具有特定內容。已經發布及／或公布一組 SIMD 擴充，其被稱為進階向量擴充（Advanced Vector Extensions, AVX）（AVX1 和 AVX2），並且使用向量擴

充 (Vector Extensions, VEX) 編碼機制 (例如 , 參見 2011 年 10 月之 Intel® 64 和 IA-32 架構軟體開發者手冊 , 以及參見 2011 年 6 月看到 Intel® 進階向量擴充編程參考) 。

示例性指令格式

本發明所描述的指令的實施例可以利用不同的格式體現。此外, 示例性的系統、架構、和管線詳述如下。指令的實施例可以在這樣的系統、架構、和管線執行, 但不局限於這些詳細說明。

A. 通用向量友好指令格式

向量友好指令格式是適用於向量指令的指令格式 (例如, 有針對向量運算的某些欄位)。所描述的實施例中, 向量和純量運算兩者都透過向量友好指令格式所支持, 替代的實施例中只使用向量運算的向量友好指令格式。

圖 1A 和 1B 是根據本發明的實施例, 示出通用向量友好指令格式和其指令模板的方框圖。圖 1A 是根據本發明的實施例, 示出通用向量友好指令格式和其類別 A 指令模板的方框圖; 而圖 1B 是根據本發明的實施例, 示出通用向量友好指令格式和其類別 B 指令模板的方框圖。具體地講, 被定義為類別 A 和類別 B 指令模板的通用向量友好指令格式 100, 這兩者都包括無記憶體存取 105 指令模板和記憶體存取 120 指令模板。在向量友好指令格式的上

下文中之術語通用是指不繫於任何特定指令集的指令格式。

而本發明所描述的發明之實施例，其中向量友好指令格式支持下列：64 位元組 (byte) 向量運算元長度 (或大小) 其具有 32 位元 (4 位元組) 或 64 位元 (8 位元組) 資料元件寬度 (或大小) (以及因此，64 位元組向量由 16 個雙字 (doubleword) 大小元件或替代地，8 個四倍字 (quadword) 大小元件組成)；64 位元組向量運算元長度 (或大小) 其具有 16 位元 (2 位元組) 或 8 位元 (1 位元組) 資料元件寬度 (或大小)；32 位元組向量運算元長度 (或大小) 其具有 32 位元 (4 位元組)、64 位元 (8 位元組)、16 位元 (2 位元組) 或 8 位元 (1 位元組) 資料元件寬度 (或大小)；以及 16 位元組向量運算元長度 (或大小) 其具有 32 位元 (4 位元組)、64 位元 (8 位元組)、16 位元 (2 位元組) 或 8 位元 (1 位元組) 資料元件寬度 (或大小)；替代實施例可以支持更多、更少及/或不同的向量運算元大小 (例如，256 位元組向量運算元) 其具有更多、更少或不同的資料元件寬度 (例如，128 位元 (16 位元組) 的資料元件寬度)。

圖 1A 中的類別 A 指令模板包括：1) 無記憶體存取 105 指令模板內，示出了無記憶體存取、完全捨入型控制類型運算 110 指令模板以及無記憶體存取資料變換類型運算 115 指令模板；以及 2) 記憶體存取 120 指令模板內，示出了記憶體存取暫時 125 指令模板以及記憶體存取非暫

時 130 指令模板。圖 1B 中的類別 B 指令模板包括：1) 無記憶體存取 105 指令模板內，示出了無記憶體存取、寫入遮罩控制、部分捨入型控制類型運算 112 指令模板以及無記憶體存取寫入遮罩控制向量大小 (vsize) 類型運算 117 指令模板；以及 2) 記憶體存取 120 指令模板內，示出了記憶體存取寫入遮罩控制 127 指令模板。

通用向量友好指令格式 100 包括在圖 1A-1B 所示的順序中於下方所列出的以下欄位。

格式欄位 140-在此欄位的一特定值 (指令格式識別符值) 唯一識別該向量友好指令格式，且因此在指令串流中的向量友好指令格式中的指令發生。因此，就該欄位不需要用於僅具有通用向量友好指令格式的指令集之意義來看，該欄位是可選的。

基底運算欄位 (Base operation field) 142-其內容區分不同的基底運算。

暫存器索引欄位 (Register index field) 144-它的內容，直接地或透過位址產生，指定來源的位置和目的地運算元，無論是在暫存器或記憶體中。這些包括足夠數目的位元用以從 $P \times Q$ 個 (例如 32×512 、 16×128 、 32×1024 、 64×1024) 暫存器檔選擇 N 個暫存器。而在一個實施例中， N 可以是多達三個來源和一個目的地暫存器，替代的實施例可支持更多或更少的來源和目的地暫存器 (例如，可以支持多達兩個來源，其中這些來源中的一者還可作為目的地；可以支持多達到三個來源，其中這些來源中的一

者還可作為目的地；可以支持多達兩個來源和一個目的地）。

修飾符欄位（Modifier field）146-其內容區分在通用向量指令格式中的指令的出現，其指定從那些不在無記憶體存取 105 指令模板和記憶體存取 120 指令模板之間的記憶體存取。記憶體存取運算讀取及／或寫入至記憶體階層（在某些情況下，使用在暫存器中的值來指定來源及／或目的地位址），而非記憶體存取運算則不讀取及／或寫入（例如，來源和目的地為暫存器）。而在一個實施例中，此欄位還在三種不同的方式之間選擇以執行記憶體位址計算，替代實施例可支持更多、更少或不同的方式以執行記憶體位址計算。

擴充運算欄位（Augmentation operation field）150-其內容區分各種不同的運算中的哪一個以在除基底運算之外被執行。此欄位是上下文特定。在本發明的一個實施例中，該欄位被劃分成類別欄位 168、 α 欄位 152 及 β 欄位 154。擴充運算欄位 150 允許將在單個指令中而不是 2 個、3 個或 4 個的指令中被執行之運算的通用群組。

標度欄位（Scale field）160-其內容允許索引欄位的内容之縮放用於記憶體位址產生（例如，使用 $2^{\text{標度}} * \text{索引} + \text{基底}$ 之位址產生）。

位移欄位（Displacement Field）162A-其內容被使用作為記憶體位址產生的一部分（例如，使用 $2^{\text{標度}} * \text{索引} + \text{基底} + \text{位移}$ 之位址產生）。

位移因子欄位 (Displacement Factor Field) 162B (注意，在位移因子欄位 162B 的正上方的位移欄位 162A 之並列指示一個或另一個被使用) -其內容作為位址產生的部分；其指定位移因子是藉由記憶體存取 (N) 的大小而被縮放 -其中 N 是記憶體存取中的位元組數 (例如，對於採用 $2^{\text{標度}} * \text{索引} + \text{基底} + \text{經定標之位移}$ 之位址產生) 。冗餘低階位元被忽略，且因此，位移因子欄位的內容與記憶體運算元總大小 (N) 相乘，以產生最終的位移以被使用於計算有效位址。N 的值是依據全運算碼欄位 (full opcode field) 174 (在本文後續描述) 以及資料操縱欄位 154C 在運行期間藉由處理器硬體所決定。就以下意義來看，位移欄位 162A 以及位移因子欄位 162B 為可選的：在它們沒有被用於無記憶體存取 105 指令模板，以及 / 或不同的實施例可能只實施兩者之一或兩者皆無之時。

資料元件寬度欄位 (Data element width field) 164-其內容區分數個資料元件寬度中的哪一者被使用 (在一些實施例中用於所有指令；在一些實施例中只用於指令中的一些) 。就以下意義來看，此欄位是可選的：如果只有一個資料元素寬度被支持及 / 或使用運算碼中的一些態樣而使資料元件寬度被支持，則不需要此欄位。

寫入遮罩欄位 (Write mask field) 170-其內容，在每資料元件位置基礎上，控制在目的地向量運算元中的資料元件位置是否反應出基底運算和擴充運算的結果。類別 A 指令模板支持合併寫入遮蓋 (merging-writemasking) ，

而類別 B 指令模板支持合併寫入遮蓋和歸零寫入遮蓋兩者。當合併時，向量遮罩允許目的地中的任何一組元件被保護而在任何運算的執行期間（由基底運算和擴充運算所指定）免於更新；在另一個實施例中，保留其對應的遮罩位元具有 0 之目的地的每一個元件的舊值。相反地，當歸零向量遮罩允許目的地中的任何一組元件在任何運算的執行期間（由基底運算和擴充運算所指定）被歸零時；在一個實施例中，當對應的遮罩位元具有值 0 時，目的地的元件被設定為 0。此功能性的一子集為用以控制被執行的運算的向量長度之能力（也就是，從第一個到最後一個正被修改的元件的跨距）；然而，被修改的元件不須要是連續的。寫入遮罩欄位（write mask field）170 允許部分的向量運算，其包括載入、儲存、運算、邏輯等。而本發明的實施例描述為，其中寫入遮罩欄位 170 的內容在數個包含將被使用的寫入遮罩的寫入遮罩暫存器中選擇一者（以及因此寫入遮罩欄位 170 的內容非直接地識別將被執行的遮蓋），替代的實施例替代或額外的允許遮罩寫入欄位 170 的內容用以直接地指定將被執行的遮罩。

立即數欄位（Immediate field）172-其內容允許立即數的規範。就以下意義來看，此欄位是可選的：其不出現在不支持立即數的通用向量友善格式的實施中且其不出現在不使用立即數的指令中。

類別欄位（Class field）168-其內容區分不同類別間的指令。參照圖 1A 至 1B，此欄位的內容在類別 A 和類

別 B 指令之間選擇。在圖 1A 至 1B 中，圓弧框被用以指示欄位中出現的特定值（例如，分別於圖 1A 至 1B 中用於類別欄位 168 的類別 A 168A 和類別 B 168B）。

類別 A 的指令模板

在類別 A 的無記憶體存取 105 指令模板的情況中， α 欄位 152 被解譯為 RS 欄位 152A，其內容區分不同的擴充運算類型中的哪一者將被執行（例如，捨入（round）152A.1 和資料變換（data transform）152A.2 分別被指定用於非記憶體存取、捨入類型運算 110 和無記憶體存取、資料變換類型運算 115 指令模板），而 β 欄位 154 區分所指定的類型的運算中的哪一者將被執行。在無記憶體存取 105 指令模板中，沒有出現標度欄位 160、位移欄位 162A 和位移標度欄位 162B。

無記憶體存取指令模板 - 完全捨入控制類型運算（Full Round Control Type Operation）

在無記憶體存取完全捨入控制類型運算 110 指令模板中， β 欄位 154 被解譯為捨入控制欄位 154A，其內容提供靜態捨入。而在本發明所描述的實施例中，捨入控制欄位 154A 包括抑制所有浮點數異常（suppress all floating point exceptions, SAE）欄位 156 以及捨入運算控制欄位 158，替代實施例可支持將這兩者概念編碼進相同欄位或具有這些概念/欄位中的一者或另一者（例如，可只具有

捨入運算控制欄位 158)。

SAE 欄位 156-其內容區分是否禁能異常事件回報；當 SAE 欄位 156 的內容指示抑制被致能，給定的指令並未回報任何種類的浮點數異常旗標且未引起任何浮點數異常處理器。

捨入運算控制欄位 (Round operation control field) 158 -其內容區分一組捨入運算中的哪一者要執行 (例如，向上捨入、向下捨去、向零捨入和就近捨入)。因此，捨入運算控制欄位 158 允許在每一指令基礎上改變捨入模式。在其中處理器包括用於指定捨入模式的控制暫存器的本發明的一個實施例中，捨入運算控制欄位 150 的內容覆寫該暫存器值。

無記憶體存取指令模板-資料變換類型運算

在無記憶體存取資料變換類型運算 115 指令模板中， β 欄位 154 被解譯為資料變換欄位 154B，其內容區分將要執行的數個資料變換中的哪一者 (例如，無資料變換、拌和 (swizzle)、廣播)。

在類別 A 的記憶體存取 120 指令模板的情況中， α 欄位 152 被解譯為驅逐提示欄位 (eviction hint field) 152B，其內容區別哪一個驅逐提示將被使用 (在圖 1A 中，時效性 (temporal) 152B.1 和非時效性 (non-temporal) 152B.2 分別被指定用於記憶體存取、時效性 125 指令模板和記憶體存取、非時效性 130 指令模板)，

而 β 欄位 154 被解譯為資料操縱欄位 154C，其內容區分數種資料操縱運算（亦習知為基元）中的哪一者將被執行（例如，無操縱；廣播；來源向上轉換；以及目的地向下轉換）。記憶體存取 120 指令模板包括標度欄位 160 以及可選地位移欄位 162A 或位移標度欄位 162B。

向量記憶體指令執行來自記憶體的向量加載和至記憶體的向量儲存，具有轉換支援。如同常規向量指令，向量記憶體指令以逐個資料元件方式傳送來自記憶體的資料或傳送資料到記憶體，其中由被選為寫入遮罩的向量遮罩的內容決定實際傳送的元件。

記憶體存取指令模板-時效性

時效性資料是可能足夠快被再次使用以受益於快取的資料。然而，這就是提示，以及不同的處理器可以不同的方式來實施它，包括完全地忽視該提示。

記憶體存取指令模板-非時效性

非時效性資料是不太可能足夠快地被再次使用以受益於第一層快取中的快取且應被給定優先驅逐之資料。然而，這就是提示，以及不同的處理器可以不同的方式來實施它，包括完全地忽視該提示。

類別 B 的指令模板

在類別 B 的指令模板的情況中，將 α 欄位 152 解譯為

寫入遮罩控制 (Z) 欄位 152C，其之內容區別由寫入遮罩欄位 170 所控制之寫入遮蓋是否應為合併或歸零。

在類別 B 的無記憶體存取 105 指令模板的情況中，將 β 欄位 154 的一部分解譯為 RL 欄位 157A，其之內容區分不同的擴充運算類型中的哪一者將被執行（例如，針對無記憶體存取寫入遮罩控制部分捨入控制類型運算 112 指令模板及無記憶體存取寫入遮罩控制 VSIZE 類型運算 117 指令模板分別指名捨入 157A.1 及向量長度 (VSIZE) 157A.2)，而 β 欄位 154 的其餘部分則區分所指明的類型的運算中的哪一者將被執行。在無記憶體存取 105 指令模板中，沒有出現標度欄位 160、位移欄位 162A 和位移標度欄位 162B。

在無記憶體存取寫入遮罩控制部分捨入控制類型運算 110 指令模板中， β 欄位 154 的其餘部分被解譯為捨入運算欄位 159A 且禁能異常事件回報（給定的指令不會回報任何種類的浮點異常旗標且不會引起任何浮點異常處理器）。

捨入運算控制欄位 (Round operation control field) 159A -如同捨入運算控制欄位 158，其內容區分一組捨入運算中的哪一者要執行（例如，向上捨入、向下捨去、向零捨入和就近捨入）。因此，捨入運算控制欄位 159A 允許在每一指令基礎上改變捨入模式。在處理器包括用於指明捨入模式的控制暫存器的本發明的一個實施例中，捨入運算控制欄位 150 的內容覆寫該暫存器值。

無記憶體存取寫入遮罩控制 VSIZE 類型運算 117 指令模板中， β 欄位 154 的其餘部分被解譯為向量長度欄位 159B，其之內容區分將要執行的數個資料向量長度中的哪一者（例如，128、256、或 512 位元組）。

在類別 B 的記憶體存取 120 指令模板的情況中，將 β 欄位 154 的一部分解譯為廣播欄位 157B，其之內容區分廣播類型資料操縱運算是否執行，而將 β 欄位 254 的其餘部分解譯為向量長度欄位 159B。記憶體存取 120 指令模板包括標度欄位 160 以及可選地位移欄位 162A 或位移標度欄位 162B。

關於通用向量友善指令格式 100，顯示全運算碼欄位 174，包括格式欄位 140、基底運算欄位 142 及資料元件寬度欄位 164。示出一個實施例，其中全運算碼欄位 174 包括所有這些欄位，在不支持他們全部的實施例中，全運算碼欄位 174 包括少於這些欄位的全部。全運算碼欄位 174 提供運算碼（opcode）。

擴充運算欄位 150、資料元件寬度欄位 164 和寫入遮罩欄位 170 允許在通用向量友善指令格式中每指令基礎上指明這些特徵。

寫入遮罩欄位和資料元件寬度欄位的組合創造鍵入指令，它們允許基於不同的資料元件寬度來施加遮罩。

在類別 A 及類別 B 內可見的各種指令模板在不同情況中有益。本發明的一些實施例中，不同的處理器或處理器內的不同核心可能只支持類別 A、只支持類別 B 或支持

兩個類別。例如，預期用於通用計算的高效能通用無序核心可以僅支持類別 B、主要預期用於圖形及/或科學（通量）計算的核心可僅支持類別 A 以及預期用於兩者的核心可支持兩者（當然，具有來自兩個類別的模板和指令的某些混合但不是所有的模板和指令都來自這兩個類別的核心是落入本發明的範圍內）。而且，單一處理器可包括多個核心，其中所有的核心支持相同的類別或其中不同的核心支持不同的類別。例如，在具有分開的圖形和通用核心的處理器中，預期主要用於圖形及/或科學計算的圖形核心中的一者可僅支持類別 A，而通用核心的一或多者可以是高效能通用核心，其具有無序執行和預期用於通用計算的暫存器重新命名，僅支持類別 B。不具有分開的圖形核心的另一處理器，可包括支持類別 A 和類別 B 兩者的一或多個通用有序或無序核心。當然，在本發明的不同實施例中，來自一個類別的特徵也可在其他類別中實施。用高階語言所寫的程式可以被放入（例如，即時編譯或靜態編譯）各種不同的執行形式，包括：1）僅具有用於執行之由目標處理器所支持的類別的指令之形式；或 2）具有使用所有類別的指令的不同組合所編寫的替代例程且具有基於當前執行碼的處理器所支持的指令來選擇例程以執行的控制流程碼之形式。

B. 示例性特定向量友好指令格式

圖 2 是根據本發明的實施例，示出示例性特定向量友

好指令格式的方框圖。圖 2 示出特定向量友好指令格式 200，在其指定欄位的位置、大小、解譯和次序以及那些欄位中的一些欄位的值之意義上，向量友好指令格式是特定的。特定向量友好指令格式 200 可被用於擴展 x86 指令集，且因此欄位中的一些與在現有的 x86 指令集和其擴展中（例如，AVX）所使用的那些相似或相同。這種格式與具有擴展之現有的 x86 指令集的前綴編碼欄位、實際運算碼位元組欄位、MOD R/M 欄位、SIB 欄位、位移欄位以及立即數欄位保持一致。將描述從圖 2 的欄位映射到圖 1 的欄位。

應該理解的是，即使出於說明性目的在通用向量友好指令格式 100 的上下文中參考特定向量友好指令格式 200 描述本發明的實施例，本發明除了所聲明之處外並不受限於特定向量友好指令格式 200。例如，通用向量友好指令格式 100 設想用於各種欄位的各種可能的大小，而特定向量友好指令格式 200 被示出為具有特定的大小的欄位。藉由特定例子，資料元件寬度欄位 164 被闡釋為特定向量友好指令格式 200 中的一個位元欄位，而本發明並不受限於此（也就是說，通用向量友好指令格式 100 設想其他大小的資料元件寬度欄位 164）。

通用向量友好指令格式 100 包括在圖 2A 所示的順序中於下方所列出的以下欄位。

EVEX 前綴（位元組 0-3）202-是以四位元組形式編碼。

格式欄位 140 (EVEX 位元組 0, 位元[7:0]) - 第一位元組 (EVEX 位元組 0) 是格式欄位 140 且其包含 0x62 (在本發明的一個實施例中使用於分辨向量友好指令格式的唯一值)。

第二到第四位元組 (EVEX 位元組 1-3) 包括提供特定能力的數個位元欄位。

REX 欄位 205 (EVEX 位元組 1, 位元[7-5]) - 由 EVEX.R 位元欄位 (EVEX 位元組 1, 位元[7]-R)、EVEX.X 位元欄 (EVEX 位元組 1, 位元[6]-X)、以及 157BEX 位元組 1, 位元[5]-B) 所組成。EVEX.R、EVEX.X 和 EVEX.B 位元欄位提供與對應的 VEX 位元欄位相同的功能性, 以及使用 1 的補數形式來編碼, 即 ZMM0 被編碼為 1111B, ZMM15 被編碼為 0000B。指令的其他欄位編碼本領域中已知的暫存器索引的較低三個位元 (rrr、xxx 和 bbb), 使得 Rrrr、Xxxx 和 Bbbb 可藉由添加 EVEX.R、EVEX.X 和 EVEX.B 來形成。

REX'欄位 110-這是 REX'欄位 110 的第一部分且是被用於編碼擴展的 32 暫存器集的上部 16 或下部 16 的 EVEX.R'位元欄位 (EVEX 位元組 1, 位元[4]-R')。在本發明的一個實施例中, 此位元, 以及下面所指示的其他位元, 是以位元反轉格式所儲存, 以與 BOUND 指令區別 (習知的 x86 32 位元模式), BOUND 指令的實際運算碼位元組是 62, 但在 MOD R/M 欄位 (以下描述) 中不接受 MOD 欄位中的值 11; 本發明的替代實施例不以反轉格式

儲存此位元和下面的其他所指示的位元。值 1 被用來編碼下部 16 暫存器。換句話說，R'Rrrr 是藉由組合 EVEX.R'、EVEX.R 和來自其他欄位的其他 RRR 來形成。

運算碼映射欄位 215 (EVEX 位元組 1, 位元 [3:0]-mmmm) - 其內容編碼暗示的前導運算碼位元組 (0F、0F 38、或 0F 3)。

資料元件寬度欄位 164 (EVEX 位元組 2, 位元 [7]-W) 是由符號 EVEX.W 所表示。EVEX.W 被用來定義資料類型 (32 位元資料元件或 64 位元資料元件) 的粒度 (大小)。

EVEX.vvvv 220 (EVEX 位元組 2, 位元 [6:3]-vvvv) - EVEX.vvvv 的角色可包括以下：1) EVEX.vvvv 編碼以反轉 (1 的補數) 形式指定的第一來源暫存器運算元，且對帶有 2 個或以上的來源運算元的指令有效；2) EVEX.vvvv 編碼目的地暫存器運算元，該目的地暫存器運算元是以 1 的補數形式指定以用於某些向量位移；或 3) EVEX.vvvv 不編碼任何運算元，該欄位被保留且應包含 1111b。因此，EVEX.vvvv 欄位 220 編碼以反轉 (1 的補數) 形式所儲存的第一來源暫存器的指定符的 4 個低階位元。依據指令，額外不同的 EVEX 位元欄位被用於擴展指定符的大小至 32 個暫存器。

EVEX.U 168 類別欄位 (EVEX 位元組 2, 位元 [2]-U) - 如果 EVEX.U = 0, 則它指示類別 A 或 EVEX.U0；如果 EVEX.U = 1, 則它指示類別 B 或 EVEX.U1。

前綴編碼欄位 225 (EVEX 位元組 2, 位元[1:0]-pp) - 為基底運算欄位提供額外位元。除了提供對以 EVEX 前綴格式的傳統 SSE 指令的支持外, 這也具有壓縮 SIMD 前綴的益處 (不像需要一位元組來表示 SIMD 前綴, EVEX 前綴僅需 2 個位元)。在一個實施例中, 欲在傳統格式和 EVEX 前綴格式兩者中支持使用 SIMD 前綴 (66H、F2H、F3H) 的傳統 SSE 指令, 將這些傳統 SIMD 前綴編碼到 SIMD 前綴編碼欄位中; 並在被提供給解碼器的 PLA 之前, 在運行時被擴展到傳統 SIMD 前綴中 (因此 PLA 可以無須修改就執行這些傳統指令的傳統格式和 EVE 格式兩者)。儘管較新的指令可以直接將 EVEX 前綴編碼欄位的內容用作運算碼擴展, 某些實施例以相似的方式以便獲得一致性, 但允許這些傳統 SIMD 前綴指明不同的含義。替代的實施例可以重新設計 PLA 以便支持 2 位元 SIMD 前綴編碼, 且因此不需擴展。

● α 欄位 152 (EVEX 位元組 3, 位元[7]-EH; 亦稱為 EVEX.EH、EVEX.rs、EVEX.RL、EVEX.write mask control (寫入遮罩控制) 和 EVEX.N; 也用 α 描述) - 如先前所描述, 此欄位是上下文特定的。

β 欄位 154 (EVEX 位元組 3, 位元[6:4]-SSS, 亦稱為 EVEX.s₂₋₀、EVEX.r₂₋₀、EVEX.rr1、EVEX.LL0、EVEX.LLB; 也用 $\beta\beta\beta$ 描述) - 如先前所描述, 此欄位是上下文特定的。

REX'欄位 110-這是 REX'欄位的剩餘部分且可以是被

用於編碼擴展的 32 暫存器集的上部 16 或下部 16 的 EVEX.V' 位元欄位 (EVEX 位元組 3, 位元 [3]-V')。此位元以位元反轉格式被儲存。值 1 被用來編碼下部 16 暫存器。換言之, 藉由結合 EVEX.V'、EVEX.vvvv 來形成 V'VVVV。

寫入遮罩欄位 170 (EVEX 位元組 3, 位元 [2:0]-kkk) -其內容如前述般指明寫入遮罩暫存器中之一個暫存器的索引。在本發明之一實施例中, 特定值 EVEX.kkk = 000 具有特殊行為, 意味著針對特定指令不使用寫入遮罩 (這可以各種方式實現, 包括使用實體接線至所有一或繞過遮蓋硬體的硬體之寫入遮罩)。

真實運算碼欄位 230 (位元組 4) 亦習知為運算碼位元組。運算碼的部分被指明在此欄位中。

MOD R/M 欄位 240 (位元組 5) 包括 MOD 欄位 242、Reg 欄位 244、及 R/M 欄位 246。如先前所描述的, MOD 欄位 242 的內容在記憶體存取和無記憶體存取運算之間做區分。Reg 欄位 244 的角色可被總結為兩種情況: 編碼目的地暫存器運算元或來源暫存器運算元, 或被看待成運算碼擴展, 且不被用以編碼任何指令運算元。R/M 欄位 246 的角色可包括以下: 編碼參照記憶體位址的指令運算元, 或編碼目的地暫存器運算元或來源暫存器運算元。

標度、索引、基底 (SIB) 位元組 (位元組 6) -如先前所描述的, 標度欄位 150 的內容被用於記憶體位址產生。SIB.xxx 254 和 SIB.bbb 256-這些欄位的內已經先前參

照相關於暫存器索引 Xxxx 和 Bbbb。

位移欄位 162A (位元組 7-10) -當 MOD 欄位 242 包含 10 時，位元組 7-10 是位移欄位 162A，且和傳統 32 位元位移 (disp32) 作用相同且作用如位元組粒度。

位移因子欄位 162B (位元組 7) -當 MOD 欄位 242 包含 01 時，位元組 7 是位移因子欄位 162B。此欄位的位置和傳統 x86 指令集 8 位元位移 (disp8) 相同，其作用如位元組粒度。由於 disp8 為正負號擴展的，其僅可定址於 -128 和 127 位元組偏移之間：就 64 位元組快取線來說，disp8 使用 8 位元，其僅可被設定成四個真正有用的值 -128、-64、0 和 64；由於時常需要更大範圍，使用 disp32；然而，disp32 需要 4 位元組。相較於 disp8 及 disp32，位移因子欄位 162B 係 disp8 的重新解譯；當使用位移因子欄位 162B 時，由位移因子欄位的內容乘上記憶體運算元存取的大小 (N) 來決定實際的位移。這種位移稱為 $\text{disp8} * N$ 。這減少平均指令長度 (針對位移使用的單位元組但具有大許多的範圍)。這種壓縮的位移係基於在有效位移為記憶體存取之粒度的倍數之假設上，且因此，不需編碼位址偏移的冗餘低階位元。換言之，位移因子欄位 162B 取代傳統 x86 指令集 8 位元位移。因此，以與 x86 指令集 8 位元位移的相同方式編碼位移因子欄位 162B (所以在 ModRM/SIB 編碼規則中無改變)，而僅有的例外是 disp8 被過載至 $\text{disp8} * N$ 。換句話說，編碼規則或編碼長度沒有改變，但在硬體對位移值的解譯中有變

(其需要以記憶體運算元的大小定標位移來獲得逐位元組之位址偏移)。

立即數欄位 272 如前述般運算。

全運算碼欄位

圖 2B 為描述根據本發明之一個實施例的構成全運算碼欄位 174 之特定向量友善指令格式 200 的欄位之方框圖。具體地，全運算碼欄位 174 包括格式欄位 140、基底運算欄位 142 和資料元件寬度 (W) 欄位 164。基底運算欄位 142 包括前綴編碼欄位 225、運算碼映射欄位 215 及真實運算碼欄位 230。

暫存器索引欄位

圖 2C 為描述根據本發明之一個實施例的構成暫存器索引欄位 144 之特定向量友善指令格式 200 的欄位之方框圖。具體地，暫存器索引欄位 144 包括 REX 欄位 205、REX' 欄位 210、MODR/M.reg 欄位 244、MODR/M.r/m 欄位 246、VVVV 欄位 220、xxx 欄位 254 以及 bbb 欄位 256。

擴充運算欄位

圖 2D 為描述根據本發明之一個實施例的構成擴充運算欄位 150 之特定向量友善指令格式 200 的欄位之方框圖。當類別 (U) 欄位 168 含有 0 時，其代表 EVEX.U0

(類別 A 168A)；當其含有 1 時，其代表 EVEX.U1 (類別 B 168B)。當 $U=0$ 且 MOD 欄位 242 包含 11 (代表無記憶體存取運算) 時， α 欄位 152 (EVEX 位元組 3，位元 [7]-EH) 被解譯為 rs 欄位 152A。當 rs 欄位 152A 包含 1 (捨入 152A.1) 時， β 欄位 154 (EVEX 位元組 3，位元 [6:4]-SSS) 被解譯為捨入控制欄位 154A。捨入控制欄位 154 包括一位元 SAE 欄位 156 和兩位元捨入運算欄位 158。當 rs 欄位 152A 包含 0 (資料變換 152A.2) 時， β 欄位 154 (EVEX 位元組 3，位元 [6:4]-SSS) 被解譯為三位元資料變換欄位 154B。當 $U=0$ 且 MOD 欄位 242 包含 00、01、或 10 (代表記憶體存取運算) 時， α 欄位 152 (EVEX 位元組 3，位元 [7]-EH) 被解譯為驅逐提示 (EH) 欄位 152B，以及 β 欄位 154 (EVEX 位元組 3，位元 [6:4]-SSS) 被解譯為三位元資料操縱欄位 154C。

當 $U=1$ ， α 欄位 152 (EVEX 位元組 3，位元 [7]-EH) 被解譯為寫入遮罩控制 (Z) 欄位 152C。當 $U=1$ 且 MOD 欄位 242 包含 11 (代表無記憶體存取運算) 時， β 欄位 154 的部分 (EVEX 位元組 3，位元 [4]-S₀) 被解譯為 RL 欄位 157A；當其包含 1 (捨入 157A.1) 時， β 欄位 154 的其餘部分 (EVEX 位元組 3，位元 [6-5]-S₂₋₁) 被解譯為捨入運算欄位 159A，而當 RL 欄位 157A 包含 0 (VSIZE 157.A2) 時， β 欄位 154 (EVEX 位元組 3，位元 [6-5]-S₂₋₁) 的其餘部分被解譯為向量長度欄位 159B (EVEX 位元組 3，位元 [6-5]-L₁₋₀)。當 $U=1$ 且 MOD 欄

位 242 包含 00、01、或 10（代表記憶體存取運算）時， β 欄位 154（EVEX 位元組 3，位元[6:4]-SSS）被解譯為向量長度欄位 159B（EVEX 位元組 3，位元[6-5]-L_{1,0}）及廣播欄位 157B（EVEX 位元組 3，位元[4]-B）。

C. 示例性暫存器架構

圖 3 是根據本發明的一個實施例的暫存器架構 300 的方框圖。在所描述的實施例中，存在有 512 位元寬的 32 個向量暫存器 310；這些暫存器稱為 zmm0 至 zmm31。下部 16 zmm 暫存器的低階 256 位元覆加在暫存器 ymm0-16 上。下部 16 zmm 暫存器的低階 128 位元（ymm 暫存器之低階 128 位元）覆加在暫存器 xmm0-15 上。特定向量友善指令格式 200 在如下表中所示之這些被覆加的暫存器檔上運算。

可調向量長度	類別	運算	暫存器
不包括向量長度欄位 159B 的指令模板	A(圖 1A; U=0)	110、115、125、130	zmm 暫存器(向量長度為 64 位元組)
	B(圖 1B; U=1)	112	zmm 暫存器(向量長度為 64 位元組)
包括向量長度欄位 159B 的指令模板	B(圖 1B; U=1)	117、127	取決於向量長度欄位 159B, zmm、ymm 或 xmm 暫存器(向量長度為 64 位元組、32 位元組、或 16 位元組)

換言之，向量長度欄位 159B 在最大長度及一或多個其他較短長度之間做選擇，其中每個這種較短長度為前一長度之一半的長度；且無向量長度欄位 159B 的指令模板

在最大向量長度上運算。此外，在一個實施例中，特定向量友善指令格式 200 的類別 B 指令模板在分包或純量單/雙精準浮點資料及分包或純量整數資料上運算。純量運算為在 zmm/ymm/xmm 暫存器中的最低階資料元件位置上執行的運算；取決於實施例，較高階資料元件位置保持與在指令之前一樣或是被歸零。

寫入遮罩暫存器 315-在所示實施例中，有 8 個寫入遮罩暫存器 (k0 至 k7)，各具有 64 位元的大小。在替代的實施例中，寫入遮罩暫存器 315 是 16 位元的大小。如先前所描述，在本發明的一個實施例中，向量遮罩暫存器 k0 不可用為寫入遮罩；當正常會指示 k0 的編碼被用於寫入遮罩時，其選擇 0xFFFF 之固線寫入遮罩，實際上針對該指令而禁能寫入遮蓋。

通用暫存器 325-在所示實施例中，有十六個 64 位元的通用暫存器，其連同現有 x86 定址模式一起用於定址記憶體運算元。以名稱 RAX、RBX、RCX、RDX、RBP、RSI、RDI、RSP、及 R8 至 R15 參照這些暫存器。

純量浮點堆疊暫存器檔 (x87 堆疊) 345，其上別名有 MMX 分包整數暫存器檔 350-在所示實施例中，該 x87 堆疊為 8 元件堆疊，其被用於在 32/64/80 位元浮點資料上使用 x87 指令集擴展來執行純量浮點運算；而 MMX 暫存器用來在 64 位元分包整數資料上執行運算，以及針對在 MMX 與 XMM 暫存器之間執行的一些運算保持運算元。

本發明的替代實施例可使用更寬或更窄的暫存器。另外，本發明之替代實施例可使用更多、更少或不同的暫存器檔及暫存器。

D. 示例性核心架構、處理器以及電腦架構

處理器核心可以不同的方式實施、用於不同的目的、以及在不同的處理器中。例如，這種核心的實施可包括：

- 1) 意圖用於通用計算的通用有序核心；
- 2) 意圖用於通用計算的高效能通用無序核心；
- 3) 主要意圖用於圖形及／或科學（通量）計算的專用核心。

不同的處理器的實施可包括：

- 1) 包括意圖用於通用計算的一或多個通用有序核心及／或意圖用於通用計算的一或多個通用無序核心的 CPU；
- 以及 2) 包括主要意圖用於圖形及／或科學（通量）的一或多個專用核心的共處理器。

這種不同的處理器導致不同的電腦系統架構，其包括：

- 1) 在與該 CPU 不同的晶片上的共處理器；
- 2) 在與 CPU 相同封裝中的不同晶片上的共處理器；
- 3) 在與 CPU 相同晶粒上的共處理器（在此情況中，這種共處理器是有時候被稱為專用邏輯，像是整合圖形及／或科學（通量）邏輯、或被稱為專用核心）；
- 以及 4) 晶片上系統，其可包括在相同晶粒上所描述的 CPU（有時候被稱為應用核心或應用處理器）、上述共處理器以及額外的功能。

示例性核心架構接著被描述，接著示例性處理器和電腦架構的描述。

圖 4A 是根據本發明的實施例，示出一示例性的有序

管線和一示例性的暫存器重新命名、無序發出/執行管線兩者的方框圖。圖 4B 是根據本發明的實施例，示出被包括在處理器中的一示例性實施例的有序架構核心和一示例性的暫存器重新命名、無序發出/執行架構核心兩者的方框圖。圖 4A 至 B 中的實線框描述有序管線和有序核心，而可選的額外虛線框描述暫存器重新命名、無序發出/執行管線和核心。鑑於在有序態樣是無序態樣的一個子集，將對無序態樣進行說明。

在圖 4A 中，處理器管線 400 包括擷取階段 402、長度解碼階段 404、解碼階段 406、分配階段 408、重新命名階段 410、排程（也稱為調度或發出）階段 412、暫存器讀取/記憶體讀取階段 414、執行階段 416、寫回/記憶體寫入階段 418、異常處理階段 422、以及提交階段 424。

圖 4B 示出了包括耦合到執行引擎單元 450 的前端單元 430 之處理器核心 490，並且兩個單元都連接到記憶體單元 470。核心 490 可以是精簡指令集計算（RISC）核心、複雜指令集計算（CISC）核心、超長指令字組（VLIW）核心、或混合的或替代的核心類型。作為另一種選擇，核心 490 可以是專用的核心，像是，例如，網路或通信核心、壓縮引擎、共處理器核心、通用計算圖形處理單元（GPGPU）核心、圖形核心，或類似物。

前端單元 430 包括耦合到指令快取單元 434 的分支預測單元 432，指令快取單元 434 耦合到指令轉譯後備緩衝區（translation lookaside buffer, TLB）436，轉譯後備緩

衝區 436 耦合到指令擷取單元 438，指令擷取單元 438 耦合到解碼單元 440。解碼單元 440（或解碼器）可以解碼指令，並產生作為輸出的一或多個微操作、微碼入口點、微指令、其他指令、或其它控制信號，其是從原始的指令解碼、或者以其他方式反映原始的指令、或從原始的指令衍生出來的。解碼單元 440 可使用各種不同的機制來實現。合適的機制的例子包括，但不限於，查找表、硬體實施、可編程邏輯陣列（PLA），微碼唯讀記憶體（ROM）等。在一個實施例中，核心 490 包括其儲存用於某些巨指令的微碼之微碼 ROM 或其他媒體（例如，在解碼單元 440 中或者是在前端單元 430 內）。解碼單元 440 耦合到執行引擎單元 450 中的重新命名/分配器單元 452。

執行引擎單元 450 包括耦合到退休單元 454 和一組一或多個排程器單元 456 的重新命名/分配器單元 452。排程器單元 456 表示任何數目的不同排程器，包括保留站、中央指令視窗等。排程器單元 456 耦合實體暫存器檔單元 458。每個實體暫存器檔單元 458 代表一或多個實體暫存器檔，其不同的各者儲存一或多個不同的資料類型，如純量整數、純量浮點、分包整數、分包浮點、向量整數、向量浮點、狀態（例如，是下一個指令將被執行的位址的指令指標）等。在一個實施例中，實體暫存器檔單元 458 包括向量暫存器單元、寫入遮罩暫存器單元、和純量暫存器單元。這些暫存器單元可以提供架構向量暫存器、向量遮罩暫存器和通用暫存器。實體暫存器檔單元 458 是藉由退

休單元 454 重疊，來說明其中的暫存器重新命名和無序執行的各種方式可以被實施（例如，使用重新排序緩衝器和退休暫存器檔；使用未來檔、歷史緩衝器和退休暫存器檔；使用暫存器地圖和暫存器的池等等）。退休單元 454 和實體暫存器檔單元 458 耦合到執行叢集 460。執行叢集 460 包括一組一或多個執行單元 462 和一組一或多個記憶體存取單元 464。執行單元 462 可以執行各種操作（例如，移位、加、減、乘）和各種類型的資料（例如，純量浮點、分包整數、分包浮點、向量整數、向量浮點）。雖然一些實施例可以包括多個專用於特定功能或一組功能的執行單元，其他實施例可僅包括一個執行單元或多個執行單元，其所有執行所有功能。排程器單元 456、實體暫存器檔單元 458 和執行叢集 460 示出為可能複數，因為某些實施例中創建用於特定類型的資料/運算的獨立管線（例如，純量整數管線、純量浮點/分包整數/分包浮點/向量整數/向量浮點管線，及/或記憶體存取管線，其每個都具有自己的排程器單元、實體暫存器檔單元及/或執行叢集—和在一個單獨的記憶體存取管線的情況下，某些實施例中被實現，其中只有該管線的執行叢集具有記憶體存取單元 464）。還應該理解的是，其中，使用單獨的管線時，這些管線中的一或多個可以是無序發出/執行，其餘的有序。

該組記憶體存取單元 464 耦合到記憶體單元 470，它包括耦合到資料快取單元 474 的資料 TLB 單元 472，該資

料快取單元 474 耦合到第 2 階 (L2) 快取單元 476。在一個示範性實施例中，記憶體存取單元 464 可以包括載入單元、儲存位址單元、儲存資料單元，其中的每一個耦接到在記憶體單元 470 中資料 TLB 單元 472。指令快取單元 434 被進一步耦合到記憶體單元 470 的第 2 階 (L2) 快取單元 476 中。L2 快取單元 476 被耦合到快取的一或多個其他階以及最終耦合到主要記憶體。

以舉例的方式，示例性暫存器重新命名、無序發出/執行核心架構可實施管線 400 如下：1) 指令擷取 438 執行擷取與長度解碼階段 402 和 404；2) 解碼單元 440 執行解碼階段 406；3) 重新命名/分配器單元 452 執行分配階段 408 和重新命名階段 410；4) 排程器單元 456 執行排程階段 412；5) 實體暫存器檔單元 458 和記憶體單元 470 執行暫存器讀取/記憶體讀取階段 414；執行叢集 460 執行執行階段 416；6) 記憶體單元 470 以及實體暫存器檔單元 458 執行寫回/記憶體寫入階段 418；7) 各種單元可涉及異常處理階段 422；以及 8) 退休單元 454 以及實體暫存器檔單元 458 執行提交階段 424。

核心 490 可以支持一或多個指令集 (例如，x86 指令集 (與已經添加較新的版本的一些擴展)；加州桑尼維爾，MIPS 科技的 MIPS 指令集；加州桑尼維爾，ARM 控股的 ARM 指令集 (具有可選的額外擴展，像是 NEON))，包括本發明所描述的指令。在一個實施例中，核心 490 包括邏輯，其用以支持分包的資料指令集擴展

（例如，AVX1、AVX2），藉此使用分包資料允許由許多多媒體應用所使用的運算被執行。

但是應當理解的是，核心可以支持多執行緒（執行兩個或多個平行組運算或執行緒），並且可以多種方式這樣做，包括時間分片的多執行緒、同步多執行緒（其中單個實體核心提供邏輯核心用於實體核心是同步地多執行緒的每一個執行緒），或它們的組合（例如，時間切片擷取和解碼以及此後的同時多執行緒，像是 Intel® Hyperthreading（超線程）技術）。

而暫存器重新命名是描述在無序執行的上下文中，但是應該理解的是，暫存器重新命名可以在有序結構中被使用。儘管所描述的處理器的實施例還包括分開的指令和資料快取單元 434/474 和共享的 L2 快取單元 476，替代的實施例可具有用於指令和資料兩者的單一內部快取，像是，例如，第 1 階（Level 1, L1）的內部快取，或多階的內部快取。在一些實施例中，系統可包括內部快取和對核心及／或處理器是外部的快取的組合。可替換地，所有的快取可以是對核心及／或處理器是外部的。

圖 5A-B 示出了更具體的示例性的有序核心架構的方框圖，該核心將是晶片中的幾個邏輯區塊（包括相同類型及／或不同類型的其他核心）中的一者。邏輯區塊透過一高帶寬互連網路（例如，一個環形網路）與一些固定功能邏輯、記憶體 I/O 介面、及其它必要的 I/O 邏輯通訊，這取決於應用。

圖 5A 是根據本發明的實施例的一單一處理器核心的方框圖，隨著其連接到片上（on-die）互連網路 502，以及隨著其第 2 階（Level 2, L2）快取 504 的本地子集。在一個實施例中，指令解碼器 500 支持 x86 指令集與分包資料指令集擴展。L1 快取 506 允許對快取記憶體至純量和向量單元的低延遲存取。雖然在一個實施例中（為了簡化設計），純量單元 508 和向量單元 510 使用單獨的暫存器組（分別是純量暫存器 512 和向量暫存器 514）和在它們之間傳輸的資料被寫入記憶體，然後接著從第 1 階（L1）快取 506 讀回，本發明的替代實施例可使用不同的方法（例如，使用單一暫存器組或包括允許資料在兩個暫存器檔之間傳輸而不會被寫和讀回的通信路徑）。

L2 快取 504 的本地子集是被分成單獨的本地子集、每一處理器核心一個的 L2 快取的部分。每一個處理器核心對 L2 快取 504 的其自己本地子集具有直接存取路徑。由處理器核心讀出的資料儲存在其 L2 快取子集 504 中，並且可以迅速地進行存取，與存取他們自己的本地 L2 快取子集的其它處理器核心平行。由處理器核心所寫入的資料儲存在其自己的 L2 快取子集 504 中，並如果需要的話從其它子集刷新。環狀網路確保共享資料的一致性。環狀網路是雙向的，以允許如處理器核心、L2 快取和其他邏輯區塊以在晶片內彼此進行通信。每個環狀資料路徑是每方向 1012 位元寬。

圖 5B 是根據本發明實施例在圖 5A 中的處理器核心

的一部分的擴展圖；圖 5B 包括 L1 資料快取 506A、L1 快取 504 的部分以及有關的向量單元 510 和向量暫存器 514 的更多的細節。具體地，向量單元 510 是 16 寬向量處理單元（VPU）（見 16 寬 ALU 528），其執行一或多個整數、單精度浮點和雙精度浮點指令。VPU 支持在記憶體輸入上以拌和單元 520 拌和暫存器輸入、以數字轉換單元 522A-B 作數字轉換、以及以複製單元 524 複製。寫入遮罩暫存器 526 允許預測結果向量寫入。

圖 6 是根據本發明的實施例，其可具有一個以上的核心，可能具有整合記憶體控制器，以及可具有整合圖形的處理器 600 的方框圖。圖 6 中的實線框示出具有單核心 602A 的處理器 600、系統代理 610、一組一或多個匯流排控制器單元 616，而可選的額外虛線框示出了替代處理器 600，其具有多個核心 602A-N、系統代理單元 610 中的一組一或多個整合記憶體控制器單元 614、以及專用邏輯 608。

因此，處理器 600 的不同實施可以包括：1) 具有專用邏輯 608 的 CPU 係整合圖形及／或科學（通量）邏輯（其可包括一或多個核心），並且核心 602A-N 是一或多個通用核心（例如，通用有序核心、通用無序核心、兩者的組合）；2) 具有核心 602A-N 的共處理器是大量專用核心，其意圖主要用於圖形及／或科學（通量）；以及 3) 具有核心 602A-N 的共處理器，其是大量的通用有序核心。因此，處理器 600 可以是通用處理器、共處理器或

專用處理器，諸如，例如，網路或通訊處理器、壓縮引擎、圖形處理器、GPGPU（通用圖形處理單元）、高通量可透過許多整合核心（MIC）共處理器（包括 30 個或更多的核心）、嵌入式處理器，或類似物。處理器可以在一或多個晶片上實施。處理器 600 可以是一部分的及／或可以使用任意數量的製程技術而在一或多個基板上實施，像是，例如，BiCMOS、CMOS 或 NMOS。

記憶體階層包括核心內的快取中的一或多個階、一組一或多個共享快取單元 606、及耦合到該組整合記憶體控制器單元 614 的外部記憶體（未示出）。該組共享快取單元 606 可包括一或多個中級快取（例如，第 2 階（L2）、第 3 階（L3）、第 4 階（L4）、或其他階的快取、最末階快取（LLC）、及／或它們的組合。而在一個實施例中，環狀基互連單元 612 使整合圖形邏輯 608、該組共享快取單元 606 以及系統代理單元 610/整合記憶體控制器單元 614 互連，替代實施例可使用任何數量的習知技術用於互連這些單元。在一個實施例中，在一或多個快取單元 606 和核心 602-A 到 N 之間維持一致性。

在一些實施例中，一個核心 602A-N 的一個或多個能夠多線程。系統代理 610 包括這些元件協調和運行核心 602A-N。系統代理單元 610 可包括例如功率控制單元（PCU）和顯示單元。在 PCU 可以是或包括需要用於調節核心 602A-N 和整合圖形邏輯 608 的功率狀態的邏輯和元件。顯示單元是用於驅動一或多個外部連接的顯示器。

核心 602A-N 可以是就結構指令集而言為均相或非均相；也就是，兩個或多個核心 602A-N 可以是能夠執行相同的指令集，而其他也是能夠只執行該指令集或不同的指令集的一個子集。

圖 7-10 是示例性電腦結構的方框圖。其他系統的設計和組態在本領域中習知為用於筆記型電腦、桌上型電腦、手持個人電腦、個人數位助理、工程工作站、伺服器、網路裝置、網路集線器、交換器、嵌入式處理器、數位信號處理器（DSP）、圖形裝置、視訊遊戲裝置、機上盒、微控制器、手機、攜帶式媒體播放器、手持裝置、和各種其他電子裝置，也是合適的。在一般情況下，能夠併入如本發明所揭示的處理器及／或其他執行邏輯大量的各種系統或電子裝置通常是合適的。

現在參照圖 7 示出根據本發明的一個實施例的系統 700 的方框圖。系統 700 可以包括一或多個處理器 710、715，其耦合到控制器集線器 720。在一個實施例中，控制器集線器 720 包括圖形記憶體控制器集線器（GMCH）790 和輸入/輸出集線器（IOH）750（其可以是不同的晶片）；GMCH 790 包括其耦合於記憶體 740 和共處理器 745 的記憶體和圖形控制器；IOH 750 耦合輸入/輸出（I/O）裝置 760 至 GMCH 790。可替代地，記憶體和圖形控制器中的一者或兩者都被整合在處理器內（如本文所述），記憶體 740 和共處理器 745 利用 IOH 750 被直接地耦合到處理器 710 以及在單一晶片中的控制器集線器

720。

額外的處理器 715 的可選性以圖 7 中的虛線來表示。每個處理器 710、715 可包括一或多個在此描述的處理核心且其可以是一些處理器 600 的一些版本。

記憶體 740 可以是，例如，動態隨機存取記憶體（DRAM）、相變記憶體（PCM）、或者是兩者的組合。對於至少一個實施例中，控制器集線器 720 經由像是前側匯流排（FSB）的多點匯流排與處理器 710、715、像是 QuickPath 互連（QPI）的點對點介面、或類似的連接 795 通信。

在一個實施例中，共處理器 745 是一個專用處理器，諸如，例如，高通量的 MIC 處理器、網路或通信處理器、壓縮引擎、圖形處理器、GPGPU、嵌入式處理器或類似物。在一個實施例中，控制器集線器 720 可以包括整合的圖形加速器。

就包括架構、微架構、熱、功率消耗特性等一系列的優點衡量標準而論，在實體資源 710、715 之間可以有各種差異。

在一個實施例中，處理器 710 執行其控制一般類型的資料處理運算的指令。指令內的嵌入可能是共處理器指令。處理器 710 識別這些共處理器指令為應該由附加的共處理器 745 所執行的類型。因此，處理器 710 在共處理器匯流排或其它互連上發出這些共處理器指令（或代表共處理器指令的控制信號）至共處理器 745。共處理器 745 接

受並執行接收到的共處理器指令。

現在參照圖 8 示出根據本發明的一個實施例的第一更詳細的示例性系統 800 的方框圖。如圖 8 所示，多處理器系統 800 是點對點互連系統，且包括第一處理器 870 和第二處理器 880 其經由點對點互連 850 耦合。處理器 870 和 880 各者可能是處理器 600 的某版本。在本發明的一個實施例中，處理器 870 和 880 分別為處理器 710 和 715，而共處理器 838 是共處理器 745。在另一個實施例中，處理器 870 和 880 分別為處理器 710、共處理器 745。

處理器 870 和 880 示出為分別包括整合記憶體控制器 (IMC) 單元 872 和 882。處理器 870 還包括作為其匯流排控制器單元點對點 (P-P) 介面 876 和 878 的一部分；同樣地，第二處理器 880 包括 P-P 介面 886 和 888。處理器 870 和 880 可以經由點對點 (P-P) 介面 850 使用 P-P 介面電路 888 和 878 來交換資訊。如圖 8，IMC 872 和 882 將處理器耦合到各自的記憶體，即記憶體 832 和記憶體 834，其可以是本地連接到相應處理器的主要記憶體的部分。

處理器 870 和 880 可各經由各別 P-P 介面 852、854 使用點對點介面電路 876、894、886 和 898 與晶片組 890 交換資訊。晶片組 890 可選地經由一高效能介面 839 與共處理器 838 交換資訊。在一個實施例中，共處理器 838 是一個專用處理器，諸如，例如，高通量的 MIC 處理器、網路或通信處理器、壓縮引擎、圖形處理器、GPGPU、嵌

入式處理器或類似物。

共享快取（未示出）可以被包括在兩個處理器之外或任一處理器中，但經由 P-P 互連與處理器連接，以使得如果處理器被置於低功率模式時，任一個或兩個處理器的本地快取資訊可被儲存在共享快取中。

晶片組 890 可以經由介面 896 耦合到第一匯流排 816。在一個實施例中，第一匯流排 816 可以是周邊組件互連（PCI）匯流排、或像是 PCI Express 匯流排或另一種第三代 I/O 互連匯流排的匯流排，但是本發明的範圍並不局限於此。

如圖 8 所示，各種 I/O 裝置 814 可以被耦合到第一匯流排 816，連同匯流排橋 818，其耦合第一匯流排 816 到第二匯流排 820。在一個實施例中，一或多個額外處理器 815，如共處理器、高通量的 MIC 處理器、GPGPU 的處理器、加速器（如，例如，圖形加速器或數位信號處理（DSP）單元）、現場可編程閘陣列，或任何其它的處理器，皆耦合到第一匯流排 816。在一個實施例中，第二匯流排 820 可以是低針腳計數（low pin count, LPC）匯流排。在一個實施例中，各種裝置可以被耦合到第二匯流排 820，其包括例如，鍵盤及／或滑鼠 822、通信裝置 827 和像是硬碟或其它大容量存儲裝置的儲存單元 828，其可以包括指令/碼和資料 830。此外，音頻 I/O 824 可以被耦合到第二匯流排 820。注意，其它架構也是可能的。例如，除了圖 8 的點對點架構，系統可以實現多點匯流排或

其它這種架構。

現在參照圖 9 示出根據本發明的一個實施例的第二更詳細的示例性系統 900 的方框圖。像在圖 8 和 9 中具有相似的附圖標記的元件，以及為了避免模糊圖 9 的其他方面而使圖 8 的某些方面已經從圖 9 中刪去。

圖 9 示出了處理器 870、880 可分別包括整合記憶體及 I/O 控制邏輯 (“CL”) 872 和 882。因此，CL 872、882 包括整合記憶體控制器單元，以及包括 I/O 控制邏輯。圖 9 示出，不僅是記憶體 832、834 耦合到 CL 872、882，而且該 I/O 裝置 914 也耦合到控制邏輯 872、882。傳統 I/O 裝置 915 耦合到晶片組 890。

現在參照圖 10 示出根據本發明的一個實施例的 SoC 1000 的方框圖。在圖 6 中相似的元件具有類似的參考圖標。另外，虛線方框是更進階的 SoC 上的可選功能。在圖 10 中，互連單元 1002 被耦接到：應用處理器 1010，其包括一組一或多個核心 202A-N 及共享快取單元 606；系統代理單元 610；匯流排控制器單元 616；整合記憶體控制器單元 614；一組或一或多個共處理器 1020，其可包括整合圖形邏輯、影像處理器、音訊處理器和視訊處理器；靜態隨機存取記憶體 (SRAM) 單元 1030；直接記憶體存取 (DMA) 單元 1032；及用於耦合到一或多個外部顯示器的顯示單元 1040。在一個實施例中，共處理器 1020 包括一個專用處理器，諸如，例如，網路或通信處理器、壓縮引擎、GPGPU、高通 MIC 處理器、嵌入式處理器或類似

物。

本文公開的機制的實施例可以以硬體、軟體、韌體或這些實施方式的組合來實現。本發明的實施例可以被實施為在可編程系統上執行的電腦程式或程式碼，系統包含至少一個處理器、儲存系統（包括易失性和非易失性記憶體及／或儲存元件）、至少一輸入裝置，以及至少一輸出裝置。

例如在圖 8 所示的碼 830 的程式碼，可應用於輸入指令以執行本文描述的功能和產生輸出資訊。輸出資訊可以被應用於一或多個輸出裝置，以習知的方式。對於本申請案的目的，處理系統包括具有處理器的任何系統，像是，例如：數位信號處理器（DSP）、微控制器、特定應用積體電路（ASIC）、或微處理器。

程式碼可以以高階程序或物件導向編程語言以與處理系統進行通信來實現。程式碼還可以以組合或機器語言來實現，如果需要的話。事實上，本文描述的機制並不限於任何特定的程式語言的範圍。在任何情況下，語言可以是編譯或解譯語言。

至少一個實施例的一或多個方面可以由儲存在機器可讀媒體上代表的指令來實施，其表示處理器內的各種邏輯，其當由機器讀取時，使機器製造邏輯以執行本文描述的技術。這樣的表示，習知為“IP 核心”，可以被儲存在有形的、機器可讀媒體上和被供給到各種客戶或生產設施以載入到製造機器，其實際上製造邏輯或處理器。

這樣的機器可讀儲存媒體可以包括，但不受限於，藉由機器或裝置所形成或製造的非暫態、實體配置的物體，其包括儲存媒體，例如：硬碟或任何包括軟碟、光碟、唯讀光碟記憶體（CD-ROM）、可重寫光碟（CD-RW）和磁光碟的其他類型的硬碟；像是唯讀記憶體（ROM）、像是動態隨機存取記憶體（DRAM）和靜態隨機存取記憶體（SRAM）的隨機存取記憶體（RAM）、可擦除可編程唯讀記憶體（EPROM）、快閃記憶體、電可擦除可編程唯讀記憶體（EEPROM）、相變記憶體（PCM）的半導體裝置；磁或光卡，或適用於儲存電子指令的任何其它類型的媒體

因此，本發明的實施例還包括非暫態的、包含指令或包含設計資料的有形機器可讀媒體，像是硬體描述語言（HDL），其限定本文所描述的結構、電路、設備、處理器及／或系統功能。這樣的實施例也可以被稱為程式產品。

在某些情況下，指令轉換器可用於將指令從源指令集轉換為目標指令集。例如，指令轉換器可轉譯（例如，使用靜態二進制轉譯、包括動態編譯的動態二進制轉譯）、變形、模擬、或以其他方式轉換指令成一或多個其它指令以被核心所處理。指令轉換器可以以軟體、硬體、韌體，或其組合來實現。指令轉換器可能是開啟處理器、關閉處理器、或部開啟和部分關閉處理器。

圖 11 是根據本發明的實施例對比使用軟體指令轉換

器用以將在來源指令集中的二進制指令轉換成在目標指令集中的二進制指令之方框圖。在所示實施例中，指令轉換器是一個軟體指令轉換器，雖然可替代地，指令轉換器可以以軟體、韌體、硬體、或它們的各種組合來實現。圖 11 示出高階語言 1102 的程式可以使用 x86 編譯器 1104 編譯，以產生 x86 的二進制碼 1106，其可被具有至少一個 x86 指令集核心 1116 的處理器本身執行。具有至少一個 x86 指令集核心 1116 的處理器表示可以執行與具有至少一個 x86 指令集核心的 Intel 處理器的實質上相同功能的任何處理器，其藉由相容地執行或反之處理 (1) Intel x86 指令集核心的指令集的實質部分或 (2) 目標在具有至少一個 x86 指令集核心的 Intel 處理器上運行的應用程式的物件碼版本或其他軟體，以實現實質上與具有至少一個 x86 指令集核心的 Intel 處理器相同的結果。在 x86 編譯器 1104 表示編譯器，其可操作以產生 x86 的二進制碼 1106 (例如，物件碼)，其可利用或不利用額外連結處理而在具有至少一個 x86 指令集核心 1116 的處理器上執行。類似地，圖 11 示出在高階語言 1102 中的程式可以使用替代的指令集編譯器 1108 來編譯，以產生替代的指令集二進制碼 1110，其可以由不含至少一個 x86 指令集核心 1114 的處理器本身執行 (例如，具有執行加州桑尼維爾的 MIPS 科技的 MIPS 指令集及 / 或執行加州桑尼維爾的 ARM 控股的 ARM 指令集之核心的處理器)。指令轉換器 1112 被用於轉換 x86 二進制碼 1106 成可由不具有

x86 指令集核心 1114 的處理器本身執行的碼。此轉換的碼可能不同於替代的指令集二進制碼 1110，因為能夠這樣的指令轉換器很難製造；然而，此轉換後的碼將完成一般操作，並由來自替換指令集的指令所構成。因此，指令轉換器 1112 代表軟體、韌體、硬體、或其組合，透過仿真、模擬或任何其他製程，允許不具有 x86 指令集處理器或核心的處理器或其他電子裝置來執行 x86 二進制碼 1106。

利用索引和立即數執行向量排列的方法和設備

本發明的一個實施例包括指令（在此有時候被稱為 VPERMBI/WI），其利用立即數控制來執行向量位元組/字組排列，其中只對在控制暫存器中的最高有效位元（MSB）與立即數值的最低有效位元（LSB）匹配之元件執行排列。如以下所討論的，這些指令可被用於有效率地實施表格的查找，其常見地被使用在許多高效能計算（high performance computing, HPC）工作負載中。

如圖 12 所描述，在本發明的實施例可被實施於其上的示例性處理器 1255 包括具有向量排列解碼邏輯 1231 的解碼器 1230 以及具有向量排列執行邏輯 1241 的執行邏輯 1240，向量排列解碼邏輯 1231 用於解碼在此所述的向量排列指令以產生複數個微運算(uops) 以及向量排列執行邏輯 1241 用於執行向量排列微運算。示例性處理器的額外細節將不會被描述。應該注意的是，然而，本發明的基

本原則不受限於微碼處理器或處理器架構的任何特定類型。

所描述的處理器架構包括一組通用暫存器 (GPR) 1205、一組向量暫存器 1206 及一組遮罩暫存器 1207。在一個實施例中，多個向量資料元件被分包成每個向量暫存器 1206，其可以具有 512 位元寬度，用於儲存兩個 256 位元的值、四個 128 位元值、八個 64 位元值、十六個 32 位元值等。然而，本發明的根本原理不受限於任何特定的向量資料大小/類型。在一個實施例中，遮罩暫存器 1207 包括被用於在儲存在向量暫存器 1206 中所儲存的值上而執行位元遮蓋運算 (例如，實施如上所述遮罩暫存器 k0-k7) 八個 64 位元運算元遮罩暫存器。然而，本發明的根本原理不受限於任何特定的遮罩暫存器大小/類型。

單一處理器核心 (“核心 0”) 的細節為簡單起見示於圖 12。然而，應當理解，在圖 12 中所示的每個核心可具有相同組的邏輯的作為核心 0。例如，每個核心可包括一個專用的第 1 階 (L1) 快取 1212 和第 2 階 (L2) 快取 1211，用於根據一個指定的快取管理策略來快取指令和資料。L1 快取 1212 包括用於儲存指令的單獨的指令快取 1220 和用於儲存資料的單獨的資料快取 1221。儲存在各種處理器快取內的指令和資料以可為固定大小的快取線的粒度進行管理 (例如，64、128、512 位元組的長度)。此示例性實施例的每個核心具有指令擷取單元 1210，用於從主記憶體 1200 及/或共享的第 3 階 (L3) 快取 1216

擷取指令；用於解碼指令的解碼單元 1220（例如，將程式指令解碼成微運算或“uops”）；用於執行指令的執行單元 1240；以及用於退役指令和寫回結果的寫回單元 1250。

指令擷取單元 1210 包括各種習知的元件，包括用於儲存將從記憶體 1200（或快取中的一個）擷取的下一個指令的位址的下一個指令指標 1203；用於儲存最近使用的虛擬到實體指令位址的地圖的指令轉譯後備緩衝器（instruction translation look-aside buffer, ITLB）1204，以提高位址轉譯的速度；分支預測單元 1202 用於推測性預測指令分支位址；以及用於儲存分支位址和目標位址的分支目標緩衝器（branch target buffer, BTB）1201。一旦擷取，指令隨後被串流到指令管線的其餘階段，指令管線包括解碼單元 1230、執行單元 1240、以及寫回單元 1250。每個這些單元的結構和功能是由那些本發明領域中的通常知識者理解並在此將不再詳細描述，以避免模糊本發明的不同實施例的相關方面。

如所提及的，在一個實施例中，向量排列指令交換連續位元或儲存在來源暫存器中的位元的群組的位置以及在立即數的控制之下將結果儲存在目的地暫存器中，該立即數指明將被反轉/交換的位元群組的大小。在一個實施例中，只允許兩個位元組大小的功率，使得所有位元組具有一對可與其交換。

圖 13 示出描述向量排列邏輯 1300 的一個實施例，其

回應於控制暫存器 1310 中的一組排列控制元件及立即數值 1311 而將來自來源暫存器 1305 的資料元件排列至目的地暫存器 1315。在一個實施例中，在控制暫存器 1310 中的每一個元件對應於目的地暫存器 1315 中的資料元件，並包括索引值，其識別來源暫存器 1305 內的元件以被複製到目的地。例如，在圖 13 中，在控制暫存器 1310（例如，最低有效位元組）中的每個位元組對應於目的地暫存器 1315 中在相對應的位置的位元組並且包括索引值，其識別來自來源暫存器 1305 的資料元件以被複製到該位置。在一個實施例中，索引值包括每一個位元組的 6 個位元，其識別來自來源暫存器 1305 的 64 個來源位元組 B0 ~ B63 中的一者。

如所提到的，在一個實施例中，只對在控制暫存器 1310 中對應的控制位元組中的最高有效位元（MSB）與立即數值 1311 的最低有效位元（LSB）匹配的那些目的地資料元件執行排列。在圖 13 中，例如，立即數值 1311（其被醒目顯示）的兩個最低有效位元是 01。因此，只有具有最高顯示位元位置中為 01 的那些控制元件被用於識別來自來源的資料元件以被複製到目的地。因此，第一控制位元組 01000011 被用於將來自來源的位元組 B3 排列至目的地 1315（即，因為它從 01 開始，並且具有識別位元組 B3 的 000011 的索引值）中的第一位元組。但是，由於下一個控制位元組在最高有效位元位置不具有 01（即，它具有值 10110010），它不能被用於識別來自來

源 1305 的一值，以被複製到目的地中的下一個位元組位置（如由圖 13 中的下一個位元組位置所指示）。在一個實施例中，在目的地中的這個位元組位置沒有被修改，使得先前儲存在其中的任何值被保持。第三個控制位元組是 01000100。因為它從 01（對應於立即數的最低有效位元）開始，它的索引值 000100 被用來識別位元組 B4 以從來源複製到目的地。接著該排列，對應於以 01 開始的控制位元組的所有目的地位元組被控制位元組的索引值識別的來源位元組所填滿，而對應於以 00、10 或 11 開始的控制位元組的所有目的地位元組則並未被修改。

在所描述的實施例中，來源暫存器 1305、控制暫存器 1310 以及目的地暫存器 1315 各包括能夠儲存 64 位元組值的 512 位元向量暫存器。相同的基本原理可以使用不同大小的資料元件被應用。例如，在一個實施例中，16 位元資料元件（字組）被儲存在來源 1305 中且使用其最高有效 N 位元對應於立即數的最低有效 N 位元的控制元件而被排列至目的地 1315。也可以使用 32 位元、64 位元，及／或 128 位元的資料元件大小。另外，如圖 13 所示，可以提供遮罩值 1312 對儲存在目的地中的結果執行寫入遮蓋。

在一個實施例中，本文所描述的技術可以被用來將來自多個來源暫存器的資料元件排列到單一目的地暫存器中。這可以說對於像是在散佈跨越於多個來源暫存器的表格中的表格查找之應用特別有用。

透過舉例的方式，圖 14 示出了一個實施例，其中來源資料元件散佈跨越四個暫存器：zmm1、zmm2、zmm3 和 zmm4。向量排列邏輯 1300 可以執行 VPERMBI 指令的以下四種情況，每一者具有不同的立即數值（0x0、0x1、0x2 和 0x3），以有效地致能儲存在 zmm31 內的不同控制元件（即，具有對應於立即數的 LSB 之 MSB 的那些控制元件）：

```
vpermbi zmm0, zmm31, zmm1, 0x0
```

```
vpermbi zmm0, zmm31, zmm2, 0x1
```

```
vpermbi zmm0, zmm31, zmm3, 0x2
```

```
vpermbi zmm0, zmm31, zmm4, 0x3
```

換言之，對於具有立即數為 0x0 的第一指令，只有那些來自 zmm31 的控制元件被用於從 zmm1 至 zmm0 排列資料元件，控制元件具有對應於立即數值（例如，00）的 LSB 之 MSB。類似地，對於具有立即數為 0x1 的第二指令，只有那些來自 zmm31 的控制元件被用於從 zmm2 至 zmm0 排列資料元件，控制元件具有對應於立即數值（例如，01）的 LSB 之 MSB。對於具有立即數為 0x2 的第三指令，只有那些來自 zmm31 的控制元件被用於從 zmm3 至 zmm0 排列資料元件，控制元件具有對應於立即數值（例如，10）的 LSB 之 MSB；對於具有立即數為 0x3 的第四指令，只有那些來自 zmm31 的控制元件被用於從 zmm4 至 zmm0 排列資料元件，控制元件具有對應於立即數值（例如，11）的 LSB 之 MSB。在最後，目的地暫存

器 $zmm0$ 包含依據立即數和來自 $zmm31$ 的控制元件的每一者而排列自四個暫存器 $zmm1-zmm4$ 每一者的資料元件。

在一個實施例中，上述技術可以被用於執行有效率的表格查找運算，其中 256B 表格被儲存在 $zmm1$ 、 $zmm2$ 、 $zmm3$ 和 $zmm4$ 中。在所示描述的指令序列後，目的地 $zmm0$ 暫存器包含 64 個已查找的位元組。

● 根據本發明的一個實施例的方法示於圖 15。方法可以在本文所述之架構的上下文內被執行，但不局限於任何特定的處理器架構。

在 1501，向量排列指令是從記憶體擷取或從指令快取讀取。在 1502，將被排列的資料元件被儲存在來源向量暫存器中以及控制元件被儲存在控制向量暫存器中。在 1503，對於在控制向量暫存器中的每一個控制元件，最高有效 N 位元是與立即數的最低有效 N 位元比較。在 1504
● 決定如果有匹配，則在 1505，從控制元件讀取索引值以識別來源資料元件且該來源資料元件被複製到對應於控制元件的目的地資料元件。如果控制元件的 MSB 不等於立即數的 LSB，則在 1506，對應的目的地資料元件保持不變。注意，以運算 1503 開始的流程圖的部分在被執行用於控制向量暫存器每一個控制元件。

在前述的說明書中，本發明的實施例已經參照其具體的示例性實施例被描述。然而，很明顯的是可以對其進行各種修改和改變而不脫離本發明的更寬的精神和範圍，如

所附申請專利範圍中所闡述。因此，說明書和附圖是應被認為是說明性的意義而不是限制性意義。

本發明的實施例可包括已在上面描述的各種步驟。這些步驟可以體現在其可用於使通用或專用處理器執行步驟的機器可執行指令。或者，這些步驟可以由包含用於執行步驟之硬線邏輯的特定硬體組件所執行，或由編程的電腦組件和客製化硬體組件的任何組合所執行。

如本文所述，指令可以指的是像是特定應用積體電路（ASIC）的硬體的特定配置，其配置為執行某些操作或具有預定的功能或儲存在體現為非暫態電腦可讀媒體的記憶體中的軟體指令。因此，在圖中所示的技術可以使用在一或多個電子裝置儲存並執行的碼和資料來實現（例如，終端站、網路元件等等）。這種電子裝置使用如非暫態電腦機器可讀儲存媒體的電腦機器可讀媒體（例如，磁碟；光碟；隨機存取記憶體；唯讀記憶體；快閃記憶體裝置；相變記憶體）和暫態電腦機器可讀通信媒體（例如，電、光、聲或其它形式的傳播信號—像是載波、紅外信號、數位信號等等）（於內部及／或透過網路與其它電子設備）儲存和通信碼和資料。此外，這種電子裝置通常包括一組耦合到像是一或多個儲存裝置（非暫態機器可讀儲存媒體）、使用者輸入/輸出裝置（例如，鍵盤、觸控螢幕及／或顯示器）以及網路連接的一或多個其它組件的一或多個處理器。該組處理器與其他組件的耦合通常是通過一或多個匯流排和橋接器（也稱為匯流排控制器）。儲存裝置

和攜帶網路流量的信號分別表示一或多個機器可讀儲存媒體和機器可讀通信媒體。因此，給定電子裝置的儲存裝置通常儲存用於在該電子裝置的該組一或多個處理器上執行的碼及／或資料。當然，本發明的一個實施例的一或多個部分可使用軟體、韌體及／或硬體的不同組合來實現。在此詳細描述全篇中，出於解釋的目的，許多具體的細節被闡述以便提供徹底理解本發明。然而，這將對本發明領域中的通常知識者是顯而易見的，本發明可在沒有這些具體細節的情況下實踐。在某些情況下，習知的結構和功能沒有被精細描述，以避免模糊本發明的標的。因此，本發明的範圍和精神應在隨後申請專利範圍中判斷。

【符號說明】

105：無記憶體存取

110：無記憶體存取、完全捨入型控制類型運算

115：無記憶體存取、資料變換類型運算

112：無記憶體存取、寫入遮罩控制、部分捨入控制類型運算

117：無記憶體存取寫入遮罩控制 v 大小類型運算

120：記憶體存取

125：記憶體存取、時效性

127：記憶體存取寫入遮罩控制

130：記憶體存取、非時效性

140：格式欄位

- 142 : 基底運算欄位
- 144 : 暫存器索引欄位
- 146 : 修飾符欄位
- 146A : 無記憶體存取
- 146B : 記憶體存取
- 150 : 擴充運算欄位
- 152 : α 欄位
- 152A : RS 欄位
- 152A.1 : 捨入
- 152A.2 : 資料變換
- 152B : 驅逐提示欄位
- 152B.1 : 時效性
- 152B.2 : 非時效性
- 154 : β 欄位
- 154A : 捨入控制欄位
- 154B : 三位元資料變換欄位
- 154C : 三位元資料操縱欄位
- 156 : 抑制所有浮點數異常欄位
- 158 : 捨入運算控制欄位
- 160 : 標度欄位
- 162A : 位移欄位
- 162B : 位移標度欄位
- 164 : 資料元件寬度欄位
- 170 : 寫入遮罩欄位

- 172 : 立即數欄位
- 200 : 特定向量友善指令格式
- 202 : EVEX 前綴
- 205 : REX 欄位
- 210 : REX' 欄位
- 215 : 運算碼映射欄位
- 220 : EVEX.vvvv
- 225 : 前綴編碼欄位
- 230 : 真實運算碼欄位
- 240 : MOD R/M 欄位
- 242 : MOD 欄位
- 244 : Reg 欄位
- 246 : R/M 欄位
- 254 : xxx 欄位
- 256 : bbb 欄位
- 300 : 暫存器架構
- 310 : 向量暫存器
- 315 : 寫入遮罩暫存器
- 325 : 通用暫存器
- 345 : 純量浮點堆疊暫存器檔
- 350 : MMX 分包整數平暫存器檔
- 400 : 處理器管線
- 402 : 擷取階段
- 404 : 長度解碼階段

- 406：解碼階段
- 408：分配階段
- 410：重新命名階段
- 412：排程階段
- 414：暫存器讀取/記憶體讀取階段
- 416：執行階段
- 418：寫回/記憶體存寫入階段
- 422：異常處理階段
- 424：提交階段
- 430：前端單元
- 432：分支預測單元
- 434：指令快取單元
- 436：轉譯後備緩衝區
- 438：指令擷取單元
- 440：解碼單元
- 450：執行引擎單元
- 452：重新命名/分配器單元
- 454：退休單元
- 456：排程器單元
- 458：實體暫存器檔單元
- 460：執行叢集
- 462：執行單元
- 464：記憶體存取單元
- 470：記憶體單元

- 472 : 資料 TLB 單元
- 474 : 資料快取單元
- 476 : 第 2 階 (L2) 快取單元
- 500 : 指令解碼器
- 502 : 片上互連網路
- 504 : 第 2 階 (L2) 快取
- 506 : L1 快取
- 506A : L1 資料快取
- 508 : 純量單元
- 510 : 向量單元
- 512 : 純量暫存器
- 514 : 向量暫存器
- 520 : 拌和單元
- 522A : 數字轉換單元
- 522B : 數字轉換單元
- 524 : 複製單元
- 526 : 寫入遮罩暫存器
- 528 : 16 位元寬 ALU
- 600 : 處理器
- 602A ~ 602N : 核心
- 604A ~ 604N : 快取單元
- 606 : 共享快取單元
- 608 : 整合圖形邏輯
- 610 : 系統代理單元

- 612 : 環狀基互連單元
- 614 : 整合記憶體控制器單元
- 616 : 匯流排控制器單元
- 700 : 系統
- 710 : 處理器
- 715 : 處理器
- 720 : 控制器集線器
- 740 : 記憶體
- 745 : 共處理器
- 750 : 輸入/輸出集線器 (IOH)
- 760 : 輸入/輸出 (I/O) 裝置
- 790 : 圖形記憶體控制器集線器
- 795 : 連接
- 800 : 系統
- 814 : I/O 裝置
- 815 : 額外處理器
- 816 : 第一匯流排
- 818 : 匯流排橋
- 820 : 第二匯流排
- 822 : 鍵盤及/或鼠標
- 824 : 音頻 I/O
- 827 : 通信裝置
- 828 : 儲存單元
- 830 : 指令/碼和資料

- 832 : 記憶體
- 834 : 記憶體
- 838 : 共處理器
- 839 : 高效能介面
- 850 : 點對點互連
- 852 : P-P 介面
- 854 : P-P 介面
- 870 : 第一處理器
- 872 : 整合記憶體及 I/O 控制邏輯
- 876 : 點對點介面電路
- 878 : 點對點介面電路
- 880 : 第二處理器
- 882 : I/O 控制邏輯
- 886 : P-P 介面
- 888 : P-P 介面
- 890 : 晶片組
- 892 : 介面
- 894 : 點對點介面電路
- 896 : 介面
- 898 : 點對點介面電路
- 900 : 系統
- 914 : I/O 裝置
- 915 : 傳統 I/O 裝置
- 1000 : SoC

- 1002 : 互連單元
- 1010 : 應用處理器
- 1020 : 共處理器
- 1030 : 靜態隨機存取記憶體 (SRAM) 單元
- 1032 : 直接記憶體存取 (DMA) 單元
- 1040 : 顯示單元
- 1102 : 高階語言
- 1104 : x86 編譯器
- 1106 : x86 的二進制碼
- 1108 : 替代的指令集編譯器
- 1110 : 替代的指令集二進制碼
- 1112 : 指令轉換器
- 1114 : x86 指令集核心
- 1116 : x86 指令集核心
- 1200 : 主記憶體
- 1201 : 分支目標緩衝器
- 1202 : 分支預測單元
- 1203 : 下一個指令指標
- 1204 : 指令轉譯後備緩衝器
- 1205 : 通用暫存器
- 1206 : 向量暫存器
- 1207 : 遮罩暫存器
- 1210 : 指令擷取單元
- 1211 : 第 2 階 (L2) 快取

1212 : 第 1 階 (L1) 快取

1216 : 第 3 階 (L3) 快取

1220 : 指令快取

1221 : 資料快取

1230 : 解碼單元

1231 : 向量排列解碼邏輯

1240 : 執行單元

● 1241 : 向量排列執行邏輯

1250 : 寫回單元

1255 : 處理器

1300 : 向量排列邏輯

1305 : 來源暫存器

1310 : 控制暫存器

1311 : 立即數值

1312 : 遮罩值

● 1315 : 目的地暫存器

申請專利範圍

1. 一種處理器，包含：

來源向量暫存器，用以儲存複數個來源資料元件；

目的地向量暫存器，用以儲存複數個目的地資料元件；

控制向量暫存器，用以儲存複數個控制資料元件，各控制資料元件對應於該些目的地資料元件中的一者且包括 N 位元值，該 N 位元值指示來源資料元件是否被複製到該對應的目的地資料元件；

向量排列邏輯，用以將各控制資料元件的該 N 位元值與立即數的 N 位元部分比較以決定是否將來源資料元件複製到對應的目的地資料元件，其中如果該些 N 位元值匹配，則該向量排列邏輯是用以使用包括在該控制資料元件中的索引值來識別來源資料元件，以及用以響應地將該來源資料元件複製到該目的地向量暫存器中的該對應的目的地資料元件。

2. 如申請專利範圍第 1 項所述之處理器，其中該立即數的該 N 位元部分包含該立即數的最低有效 N 位元部分，且其中該控制資料元件的該 N 位元部分包含該控制資料元件的最高有效 N 位元部分。

3. 如申請專利範圍第 2 項所述之處理器，其中 $N = 2$ 。

4. 如申請專利範圍第 1 項所述之處理器，其中該些來源資料元件、該些目的地資料元件以及該些控制資料元

件每一者包含位元組。

5. 如申請專利範圍第 4 項所述之處理器，其中每一個控制位元組的 6 個位元被用於該索引值以索引在該來源向量暫存器中的 64 個來源位元組的一者。

6. 如申請專利範圍第 5 項所述之處理器，其中每一個控制位元組的 2 個位元將被與該立即數的 2 位元部分比較，以決定是否將來源資料元件複製到該對應的目的地資料元件。

7. 如申請專利範圍第 1 項所述之處理器，其中該些來源資料元件、該些目的地資料元件以及該些控制資料元件每一者包含字組。

8. 如申請專利範圍第 1 項所述之處理器，其中該向量排列邏輯是用以執行一或多個向量排列指令以執行其運算。

9. 如申請專利範圍第 8 項所述之處理器，其中該向量排列邏輯是用以執行複數個向量排列指令，各具有不同的立即數值及不同的來源向量暫存器以將來自該些不同的來源向量暫存器之值排列至該目的地向量暫存器中。

10. 如申請專利範圍第 1 項所述之處理器，進一步包含：

遮罩暫存器，用以儲存遮罩值，其中該向量排列邏輯是用以使用該遮罩值在被複製到該目的地向量暫存器的資料元件上執行寫入遮蓋。

11. 一種方法，包含：

將複數個來源資料元件儲存在來源向量暫存器中；

將複數個目的地資料元件儲存在目的地向量暫存器中；

將複數個控制資料元件儲存在控制向量暫存器中，各控制資料元件對應於該些目的地資料元件中的一者且包括 N 位元值，該 N 位元值指示來源資料元件是否被複製到該對應的目的地資料元件；

● 將各控制資料元件的該 N 位元值與立即數的 N 位元部分比較以決定是否將來源資料元件複製到對應的目的地資料元件，其中如果該些 N 位元值匹配，則使用包括在該控制資料元件中的索引值來識別來源資料元件，響應地將該來源資料元件複製到該目的地向量暫存器中的該對應的目的地資料元件。

● 12. 如申請專利範圍第 11 項所述之方法，其中該立即數的該 N 位元部分包含該立即數的最低有效 N 位元部分，且其中該控制資料元件的該 N 位元部分包含該控制資料元件的最高有效 N 位元部分。

13. 如申請專利範圍第 12 項所述之方法，其中 $N = 2$ 。

14. 如申請專利範圍第 11 項所述之方法，其中該些來源資料元件、該些目的地資料元件以及該些控制資料元件每一者包含位元組。

15. 如申請專利範圍第 14 項所述之方法，其中每一個控制位元組的 6 個位元被用於該索引值以索引在該來源

向量暫存器中的 64 個來源位元組的一者。

16. 如申請專利範圍第 15 項所述之方法，其中每一個控制位元組的 2 個位元將被與該立即數的 2 位元部分比較，以決定是否將來源資料元件複製到該對應的目的地資料元件。

17. 如申請專利範圍第 11 項所述之方法，其中該些來源資料元件、該些目的地資料元件以及該些控制資料元件每一者包含字組。

18. 如申請專利範圍第 11 項所述之方法，其中執行一或多個向量排列指令以執行所述操作。

19. 如申請專利範圍第 18 項所述之方法，其中執行複數個向量排列指令，各具有不同的立即數值及不同的來源向量暫存器以將來自該些不同的來源向量暫存器之值排列至該目的地向量暫存器中。

20. 如申請專利範圍第 11 項所述之方法，進一步包含：

在遮罩暫存器儲存遮罩值，以及使用該遮罩值在被複製到該目的地向量暫存器的資料元件上執行寫入遮蓋。

21. 一種系統，包含：

記憶體，用以儲存資料和指令，包括向量排列指令和資料；

複數個核心，用以執行該些指令和處理該資料；

圖形處理器，用以回應於特定指令來執行圖形操作；

網路介面，用於在網路上接收和傳送資料；

使用者輸入介面，用於從滑鼠或鼠標控制裝置接收使用者輸入；以及

在該些核心的一或多個中的來源向量暫存器，用以儲存複數個來源資料元件；

在該些核心的一或多個中的目的地向量暫存器，用以儲存複數個目的地資料元件；

在該些核心的一或多個中的控制向量暫存器，用以儲存複數個控制資料元件，各控制資料元件對應於該些目的地資料元件中的一者且包括 N 位元值，該 N 位元值指示來源資料元件是否被複製到該對應的目的地資料元件；

在該些核心的一或多個中的向量排列邏輯，用以將各控制資料元件的該 N 位元值與立即數的 N 位元部分比較以決定是否將來源資料元件複製到對應的目的地資料元件，其中如果該些 N 位元值匹配，則該向量排列邏輯是以使用包括在該控制資料元件中的索引值來識別來源資料元件，以及用以響應地將該來源資料元件複製到該目的地向量暫存器中的該對應的目的地資料元件。

22. 如申請專利範圍第 21 項所述之系統，其中該立即數的該 N 位元部分包含該立即數的最低有效 N 位元部分，且其中該控制資料元件的該 N 位元部分包含該控制資料元件的最高有效 N 位元部分。

23. 如申請專利範圍第 22 項所述之系統，其中 $N = 2$ 。

24. 如申請專利範圍第 21 項所述之系統，其中該些

來源資料元件、該些目的地資料元件以及該些控制資料元件每一者包含位元組。

圖 1A

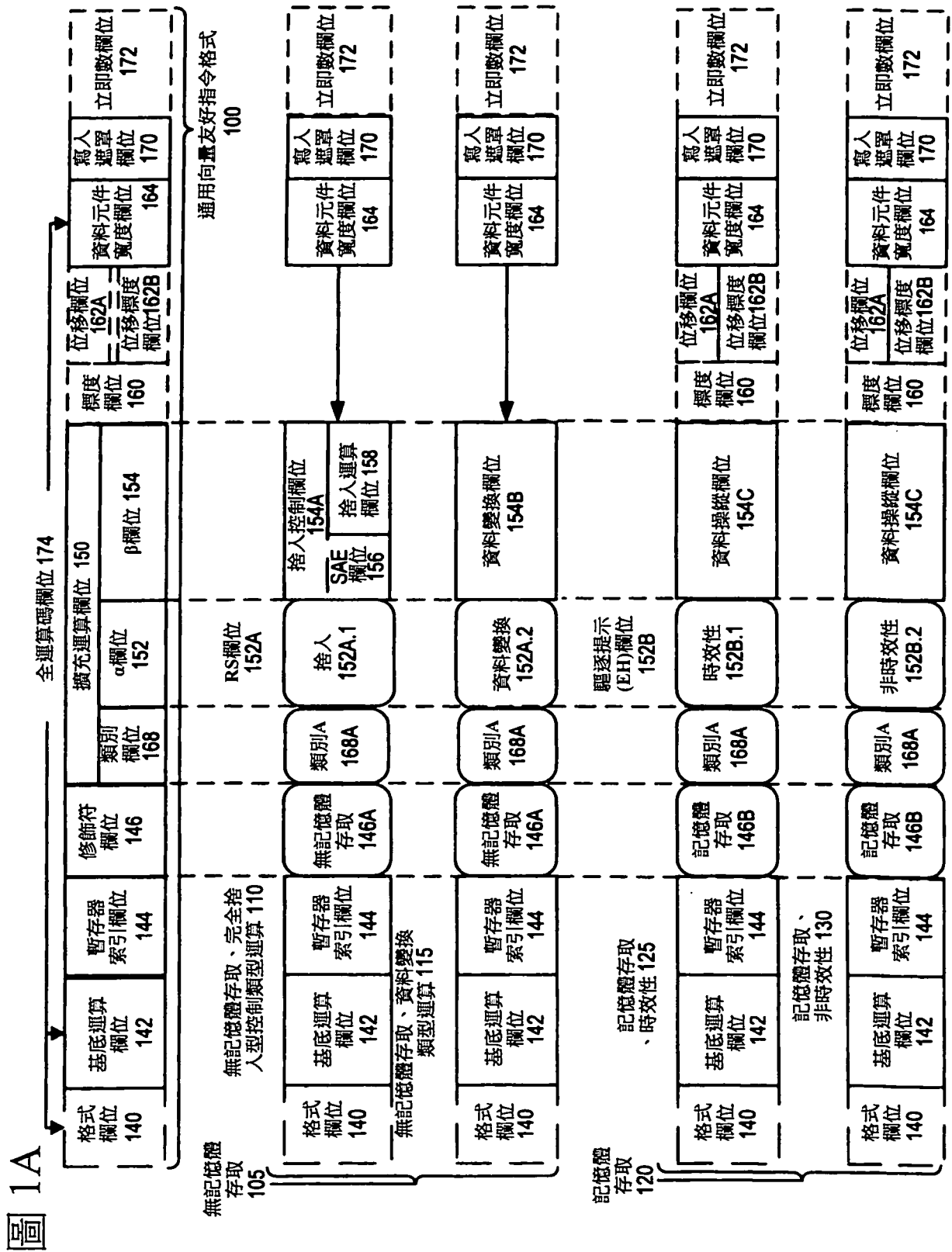


圖 1A

圖 1B

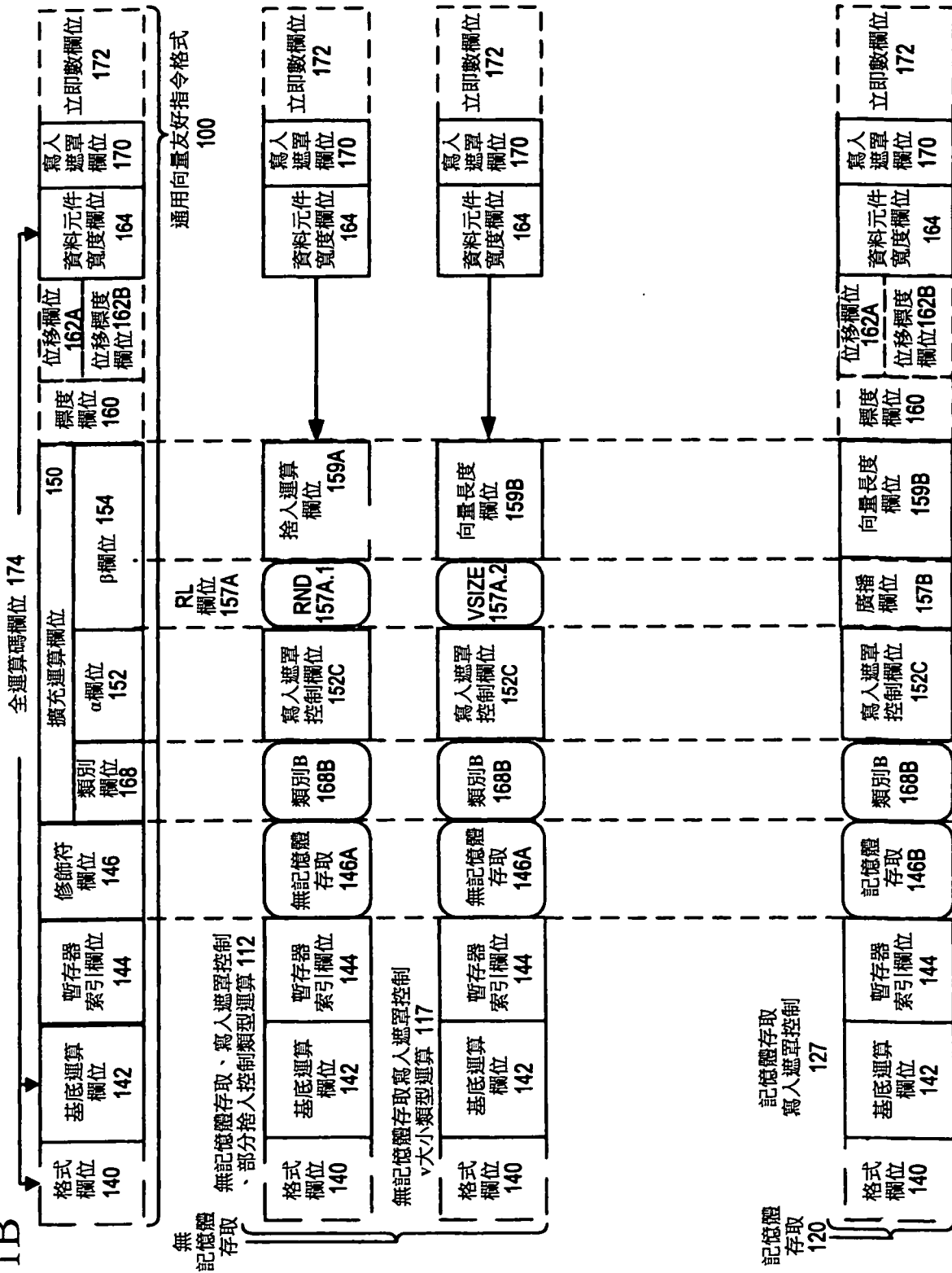


圖 2A

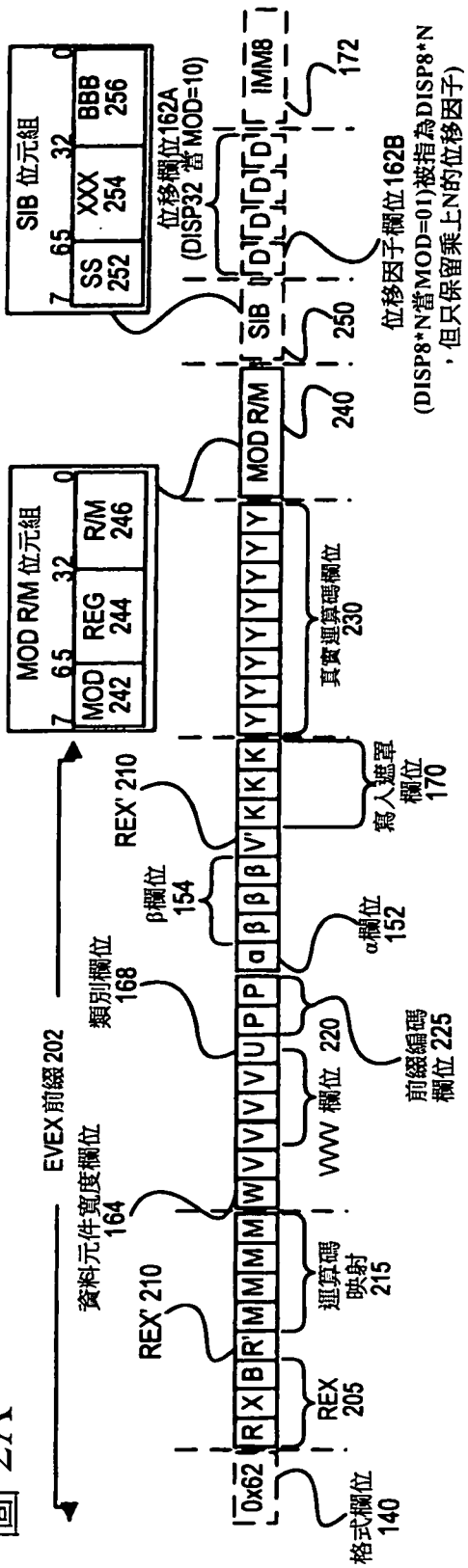


圖 2B

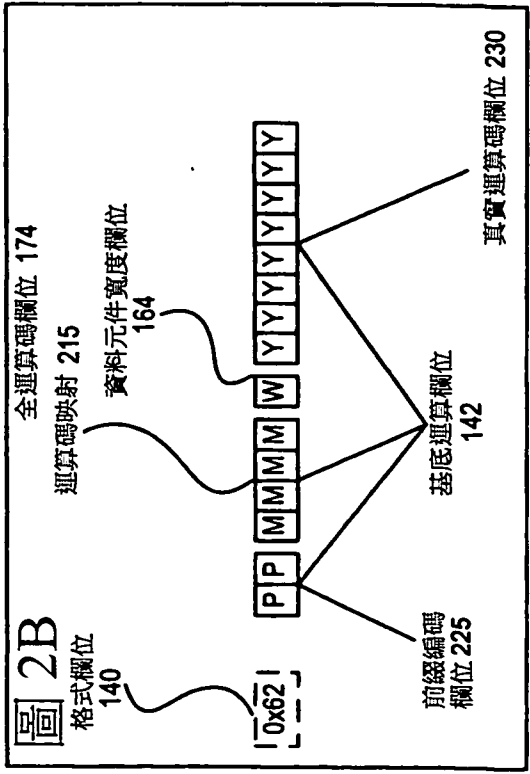


圖 2C

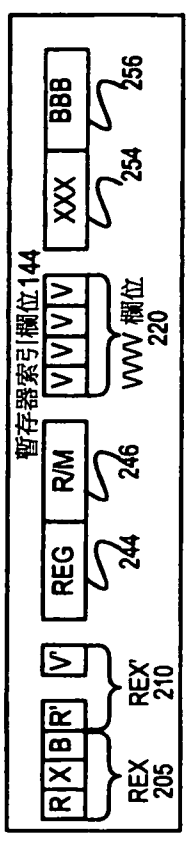
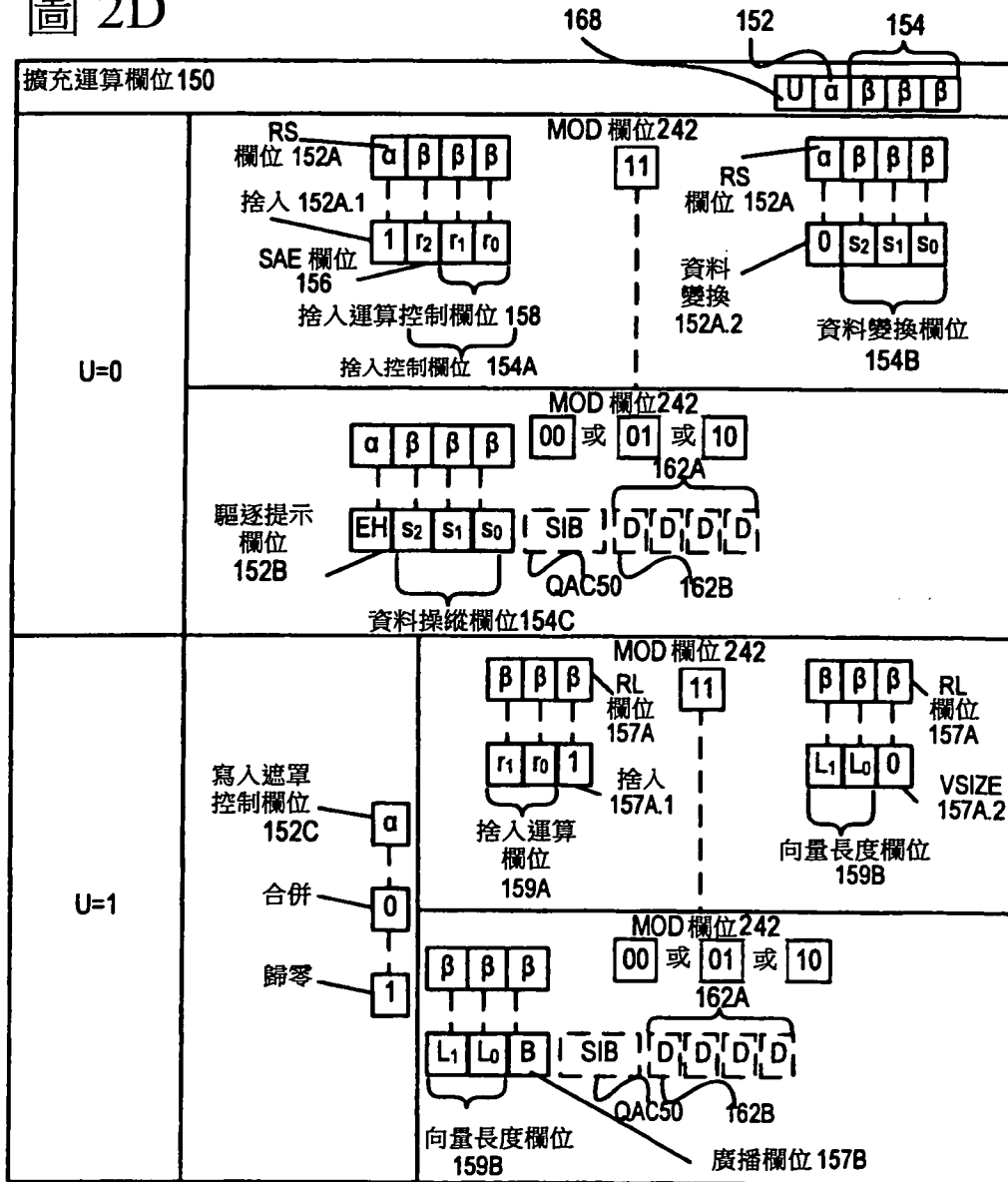


圖 2D



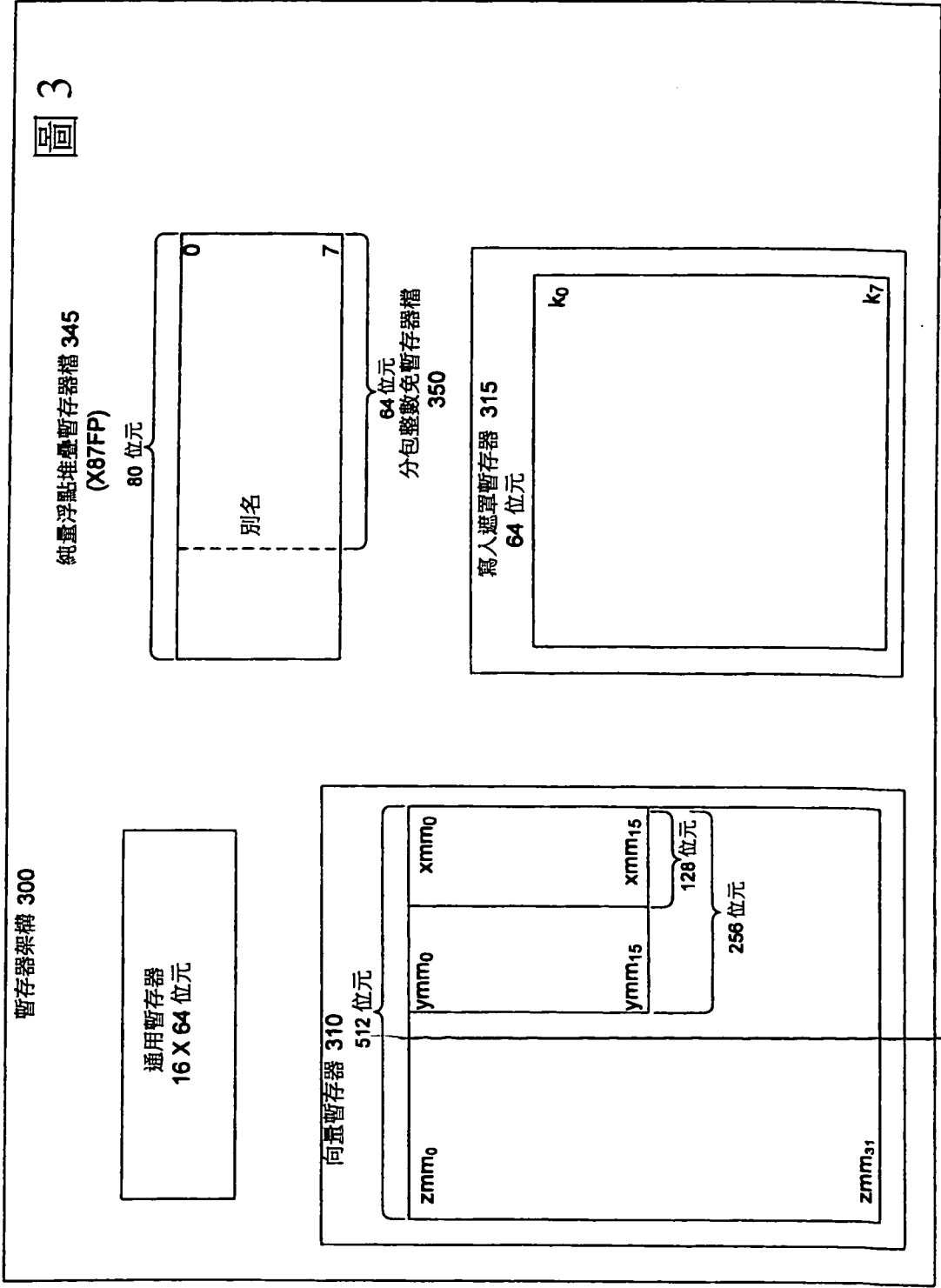


圖 3

圖 4A

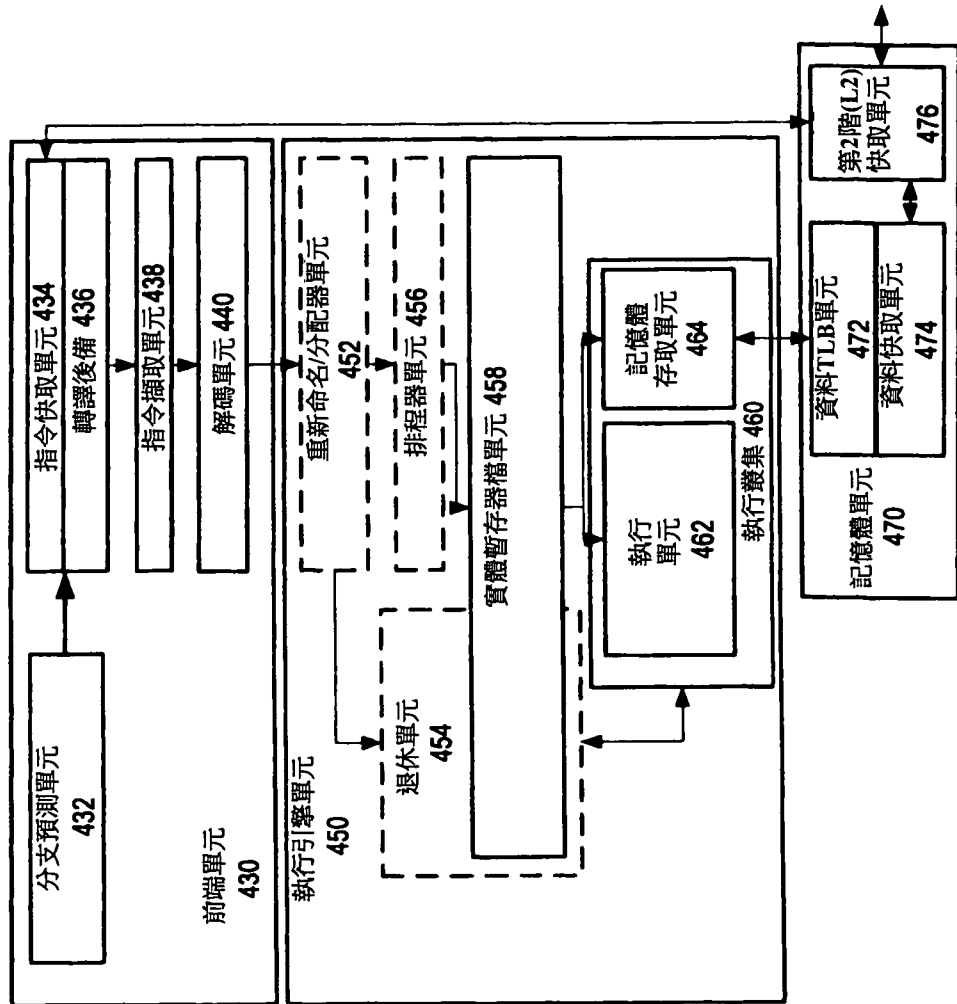
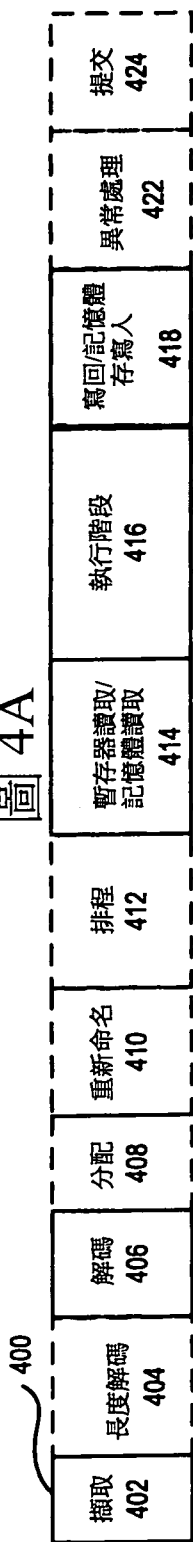


圖 4B

圖 5A

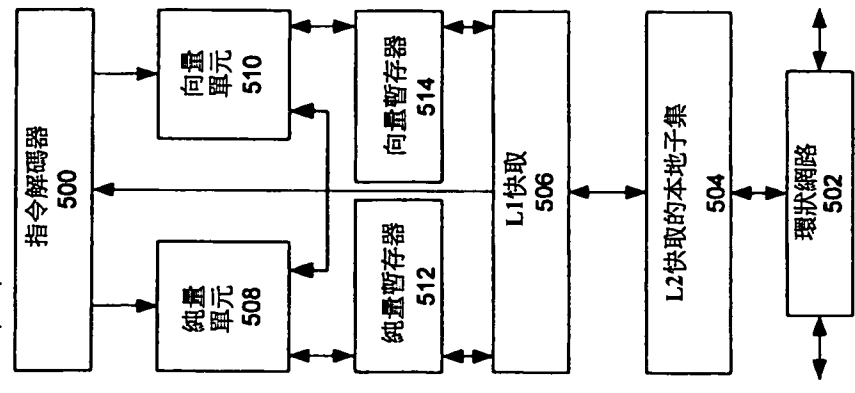
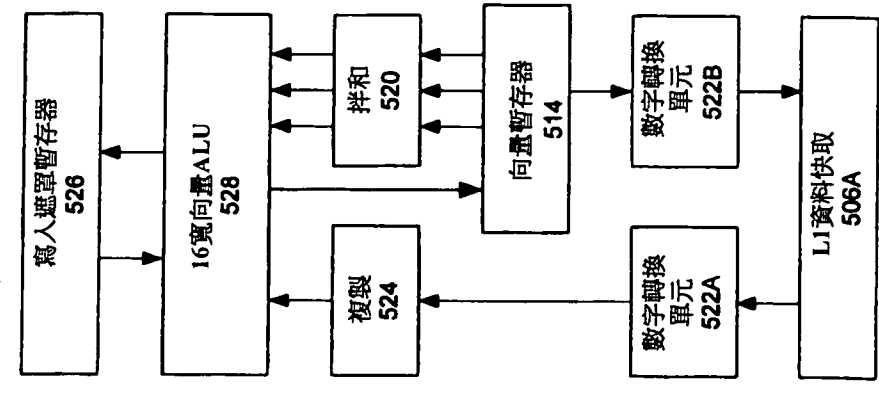


圖 5B



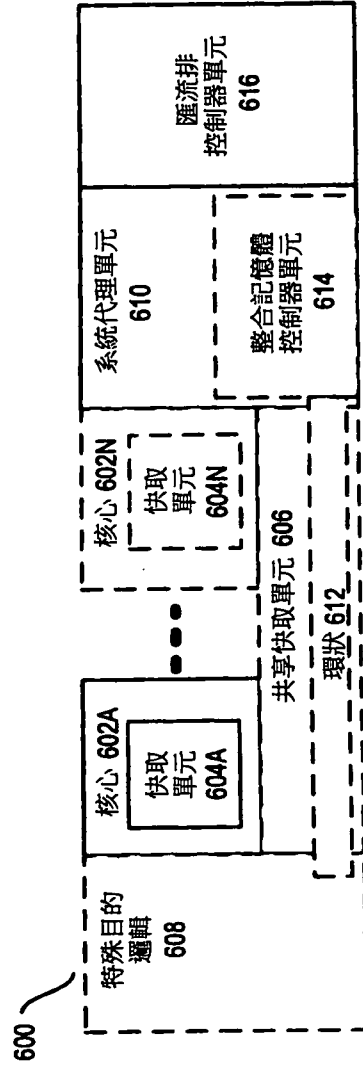


圖 6

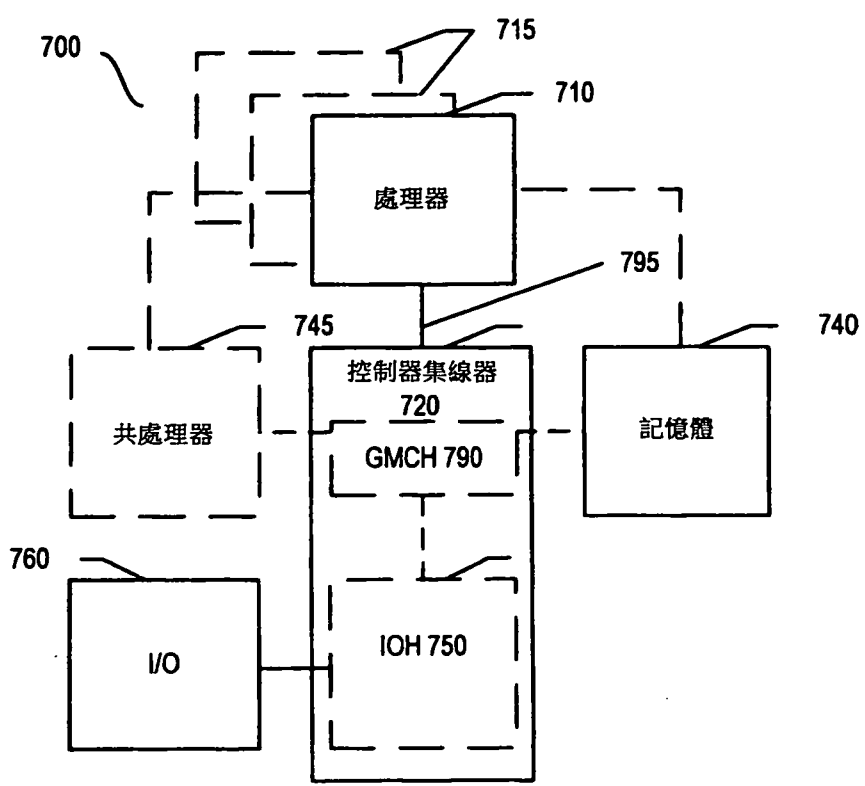


圖 7

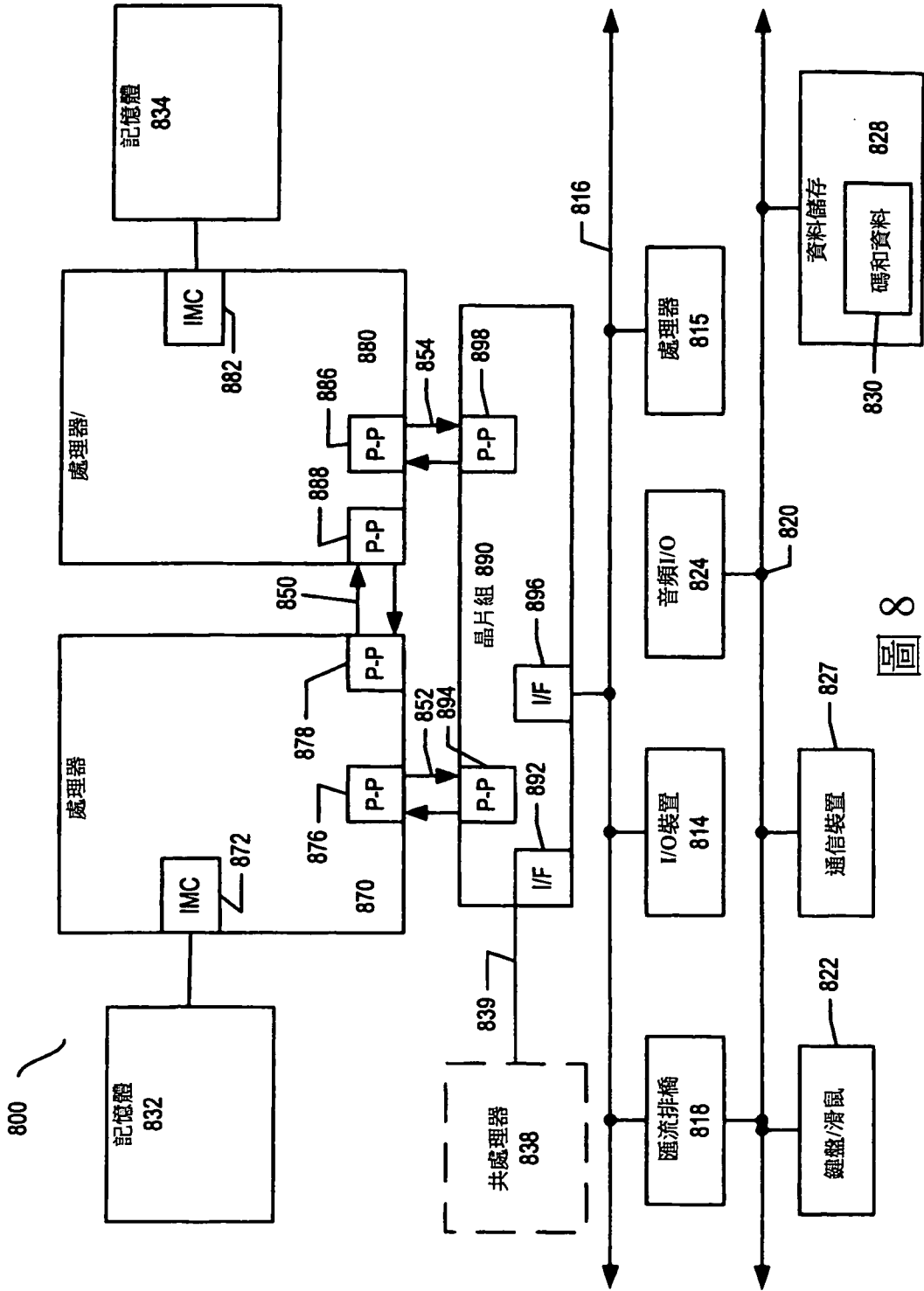


圖 8

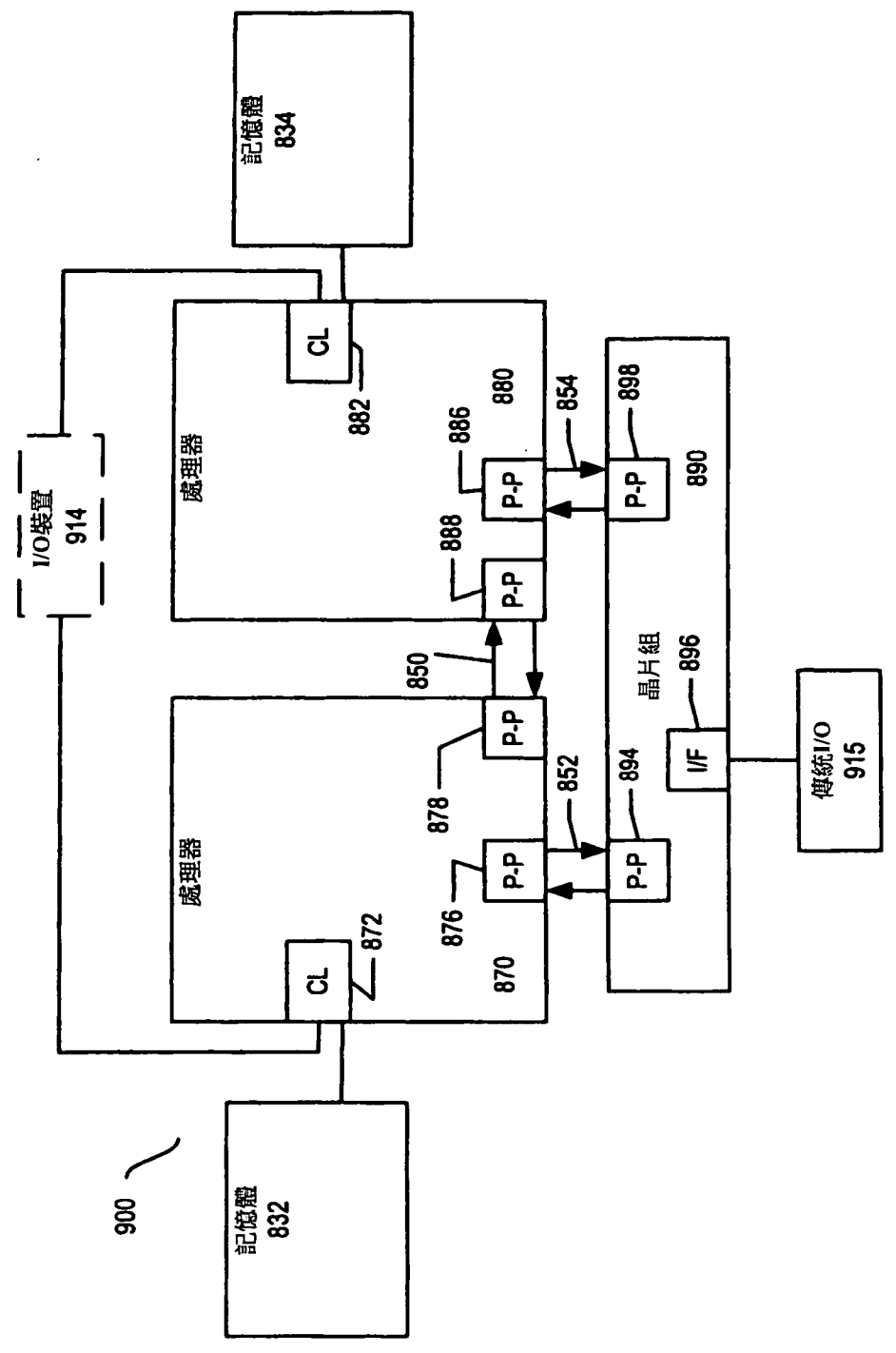


圖 9

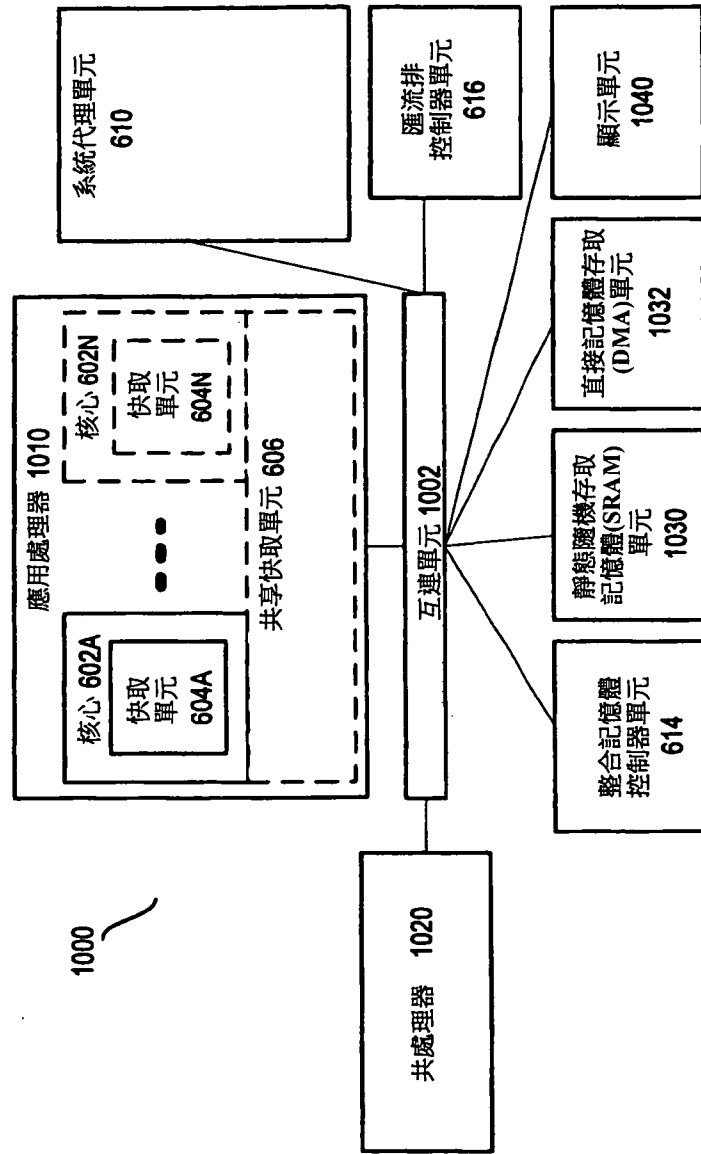


圖 10

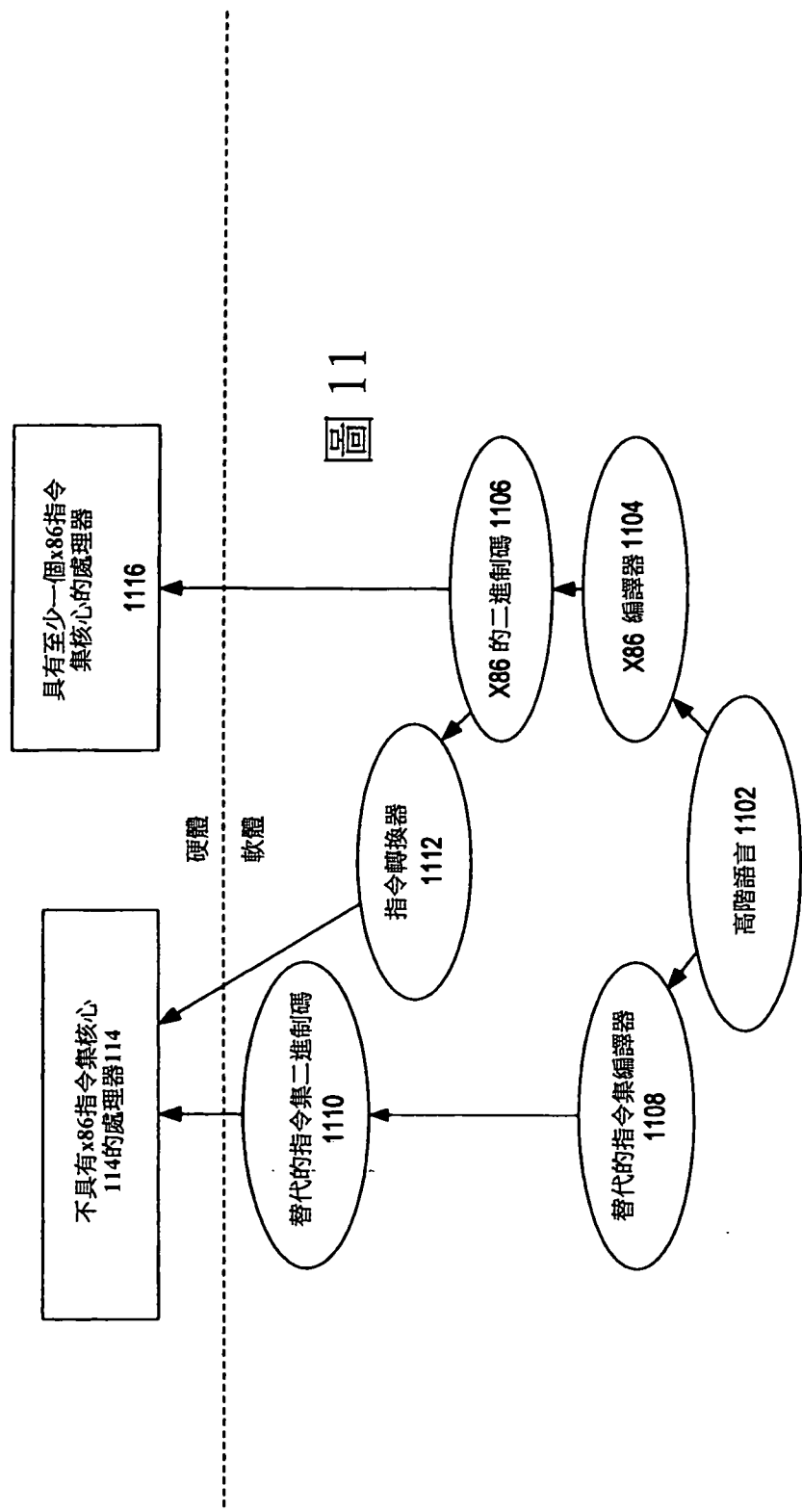


圖 11

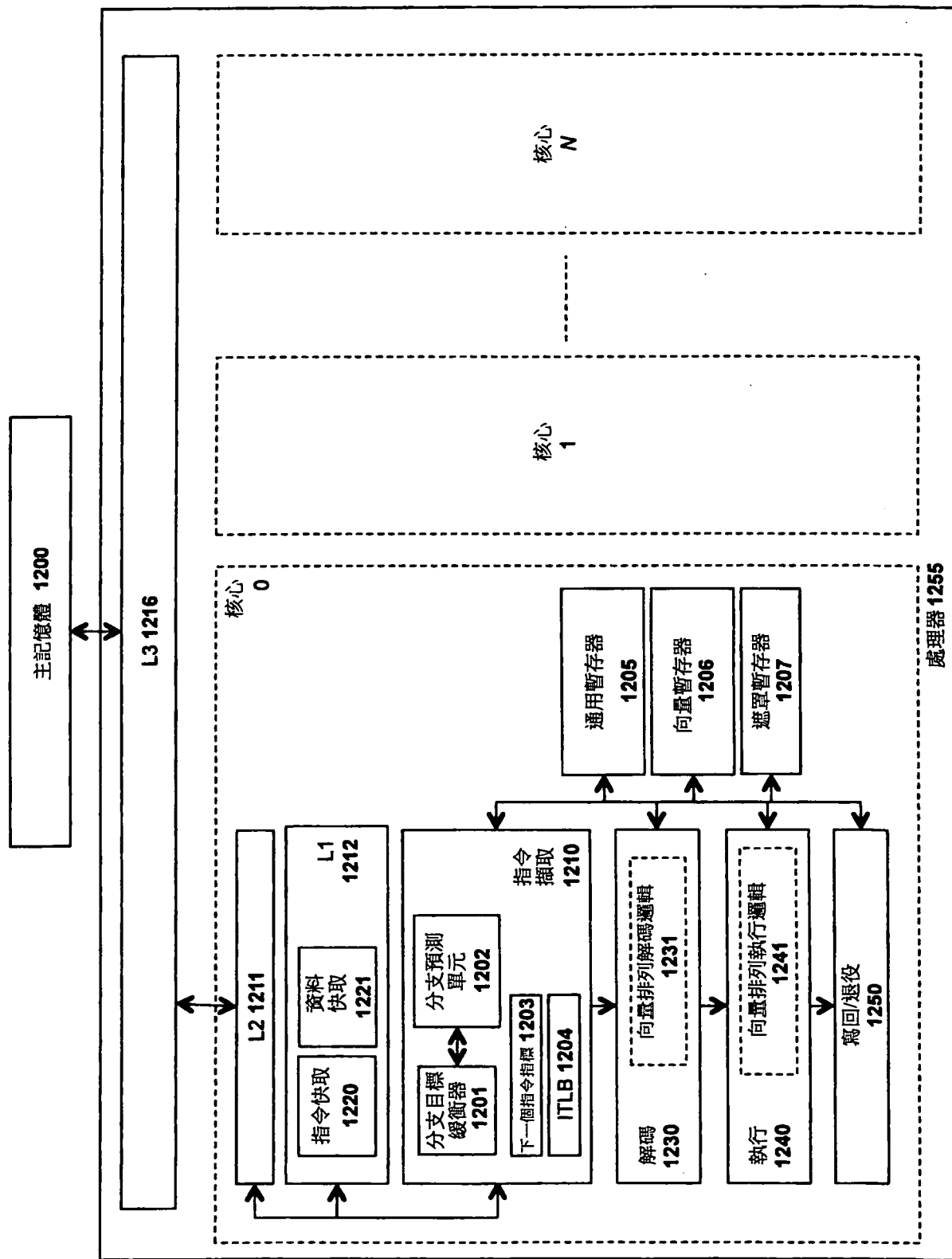


圖 12

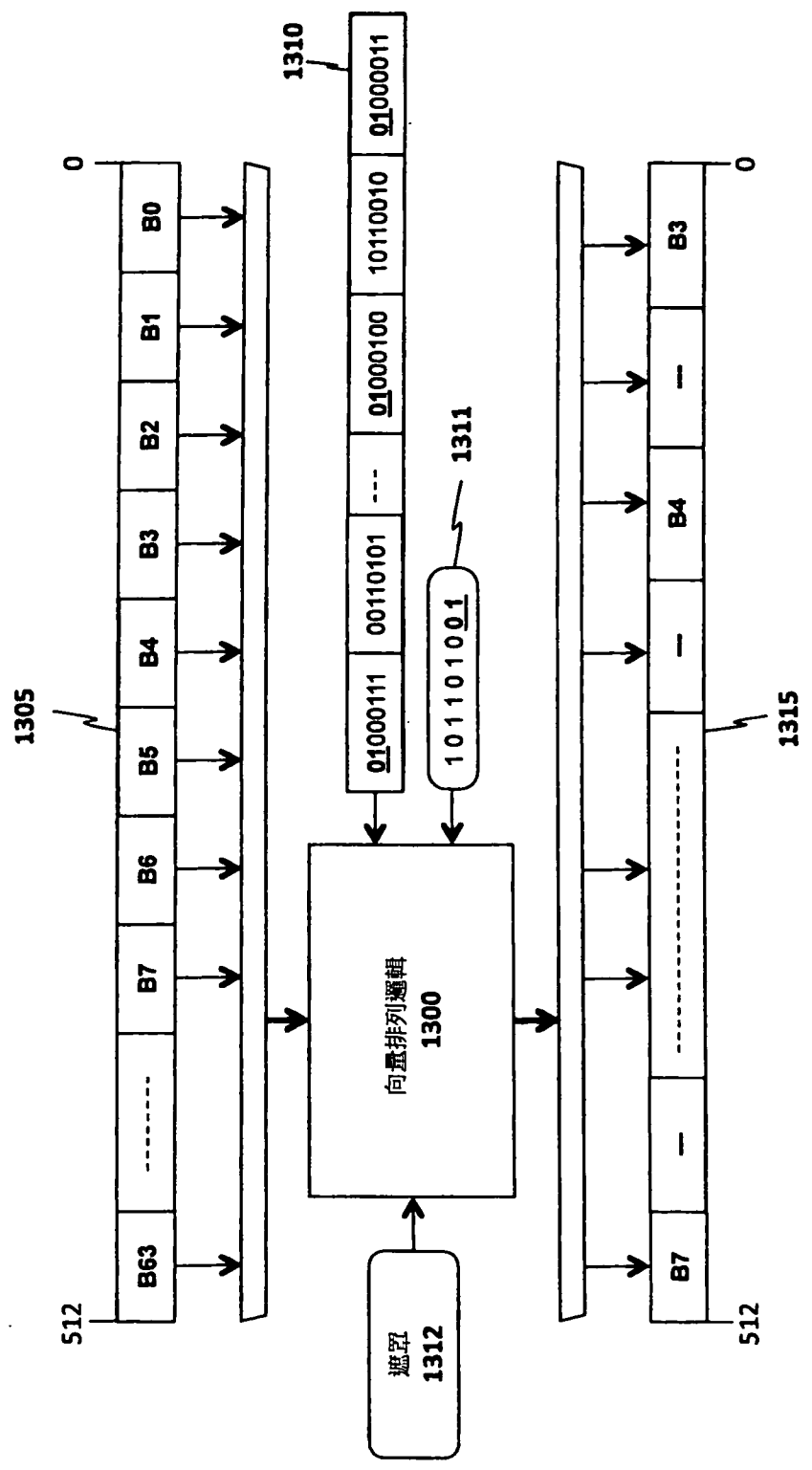


圖 13

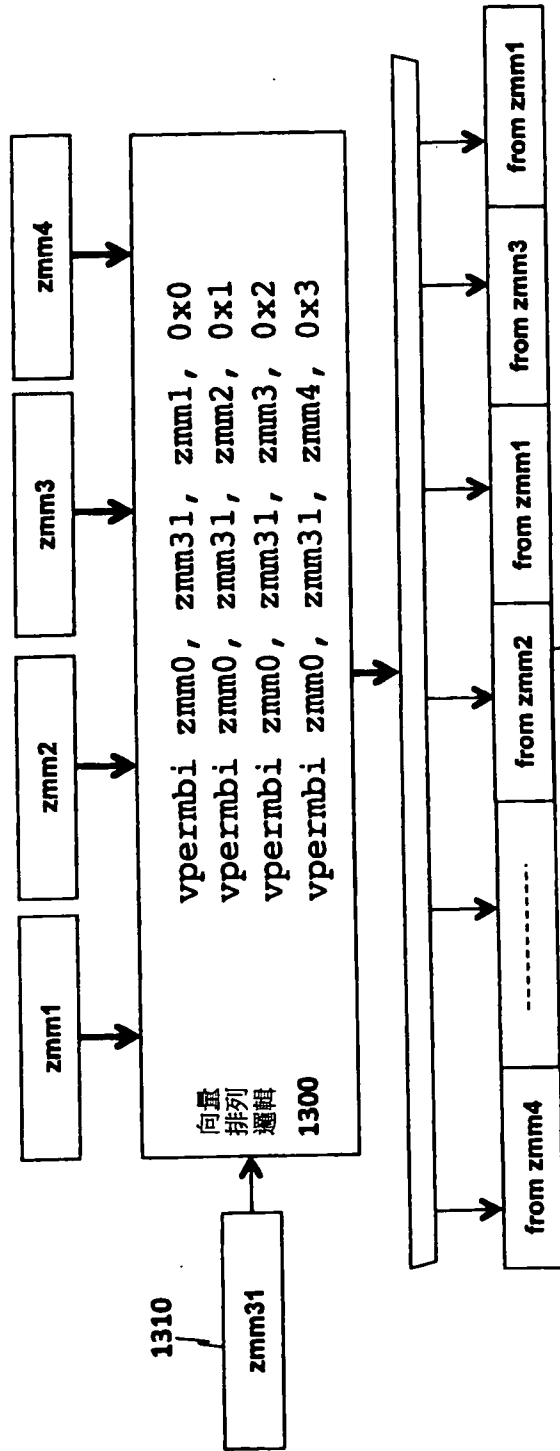


圖 14

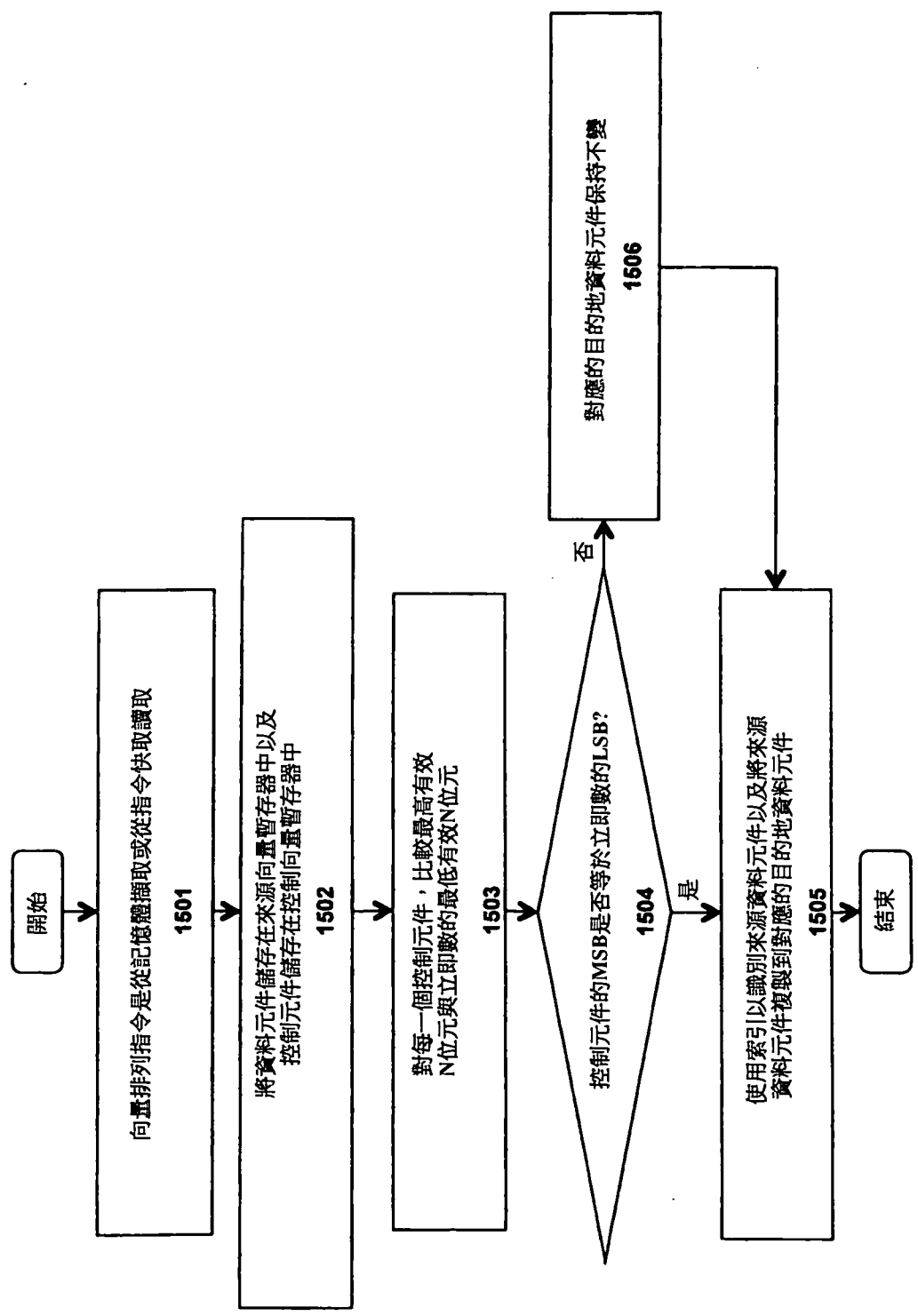


圖 15