



(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 I610228 B

(45) 公告日：中華民國 107 (2018) 年 01 月 01 日

(21) 申請案號：104138336 (22) 申請日：中華民國 104 (2015) 年 11 月 19 日
 (51) Int. Cl. : G06F9/30 (2006.01) G06F9/38 (2006.01)
 (30) 優先權：2014/12/23 美國 14/581,738
 (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)
 (74) 代理人：林志剛
 (56) 參考文獻：
 US 2009/0187746A1 US 2012/0278591A1
 審查人員：林信宏
 申請專利範圍項數：25 項 圖式數：27 共 103 頁

(54) 名稱

用於執行向量位元反轉和交叉的方法和設備

METHOD AND APPARATUS FOR PERFORMING A VECTOR BIT REVERSAL AND CROSSING

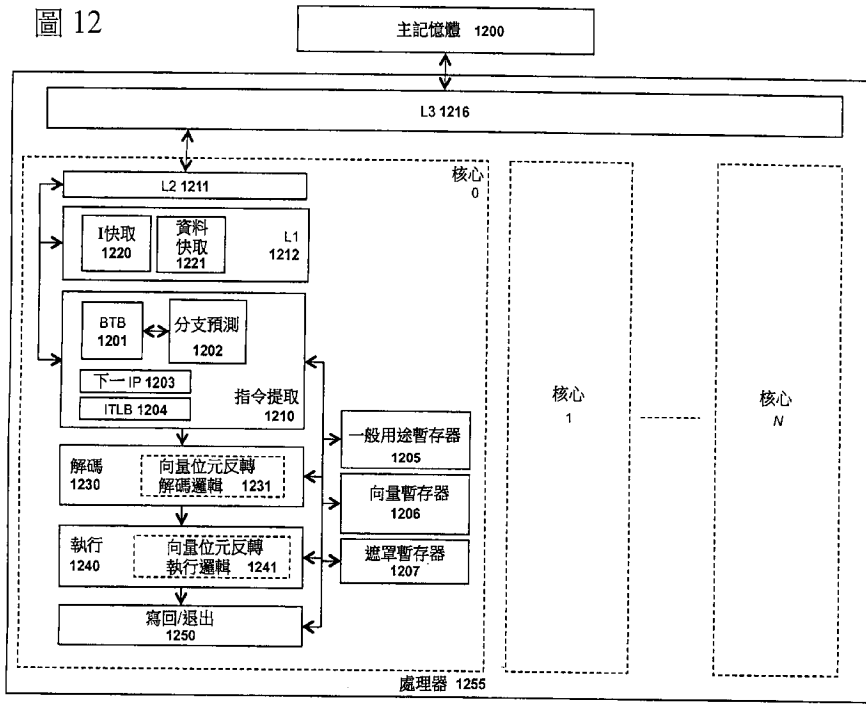
(57) 摘要

揭示用於執行向量位元反轉指令及交叉的設備及方法。舉例而言，處理器的一實施例包括：第一源向量暫存器，用以儲存第一複數個源位元群組，其中，用於位元群組的大小將於指令的立即值中被指定；第二源向量，用以儲存第二複數個源位元群組；向量位元反轉及交叉邏輯，用以從立即值決定位元群組大小以及用以回應地反轉第一源向量暫存器內鄰接的位元群組的位置，以產生一組反轉的位元群組，其中，向量位元反轉及交叉邏輯又將該組反轉的位元群組與第二複數個位元群組交錯；以及，目的地向量暫存器，用以儲存與第一複數個位元群組交錯之反轉的位元群組。

An apparatus and method for performing a vector bit reversal and crossing. For example, one embodiment of a processor comprises: a first source vector register to store a first plurality of source bit groups, wherein a size for the bit groups is to be specified in an immediate of an instruction; a second source vector to store a second plurality of source bit groups; vector bit reversal and crossing logic to determine a bit group size from the immediate and to responsively reverse positions of contiguous bit groups within the first source vector register to generate a set of reversed bit groups, wherein the vector bit reversal and crossing logic is to additionally interleave the set of reversed bit groups with the second plurality of bit groups; and a destination vector register to store the reversed bit groups interleaved with the first plurality of bit groups.

指定代表圖：

圖 12



符號簡單說明：

- 1200 . . . 主記憶體
- 1220 . . . I 快取
- 1221 . . . 資料快取
- 1201 . . . BTB
- 1202 . . . 分支預測
- 1203 . . . 下一 IP
- 1204 . . . ITLB
- 1210 . . . 指令提取
- 1230 . . . 解碼
- 1231 . . . 向量位元反轉解碼邏輯
- 1240 . . . 執行
- 1241 . . . 向量位元反轉執行邏輯
- 1250 . . . 寫回/退出
- 1205 . . . 一般用途暫存器
- 1206 . . . 向量暫存器
- 1207 . . . 遮罩暫存器
- 1255 . . . 處理器
- 1211 . . . L1 快取記憶體
- 1212 . . . L2 快取記憶體
- 1216 . . . L3 快取記憶體

發明專利說明書

(本申請書格式、順序，請勿任意更動)

【發明名稱】(中文/英文)

用於執行向量位元反轉和交叉的方法和設備

Method and apparatus for performing a vector bit reversal and crossing

【技術領域】

本發明通常係關於電腦處理器領域。更特別地，本發明關於用於執行向量位元反轉和交叉的方法和設備。

【先前技術】

指令集、或是指令集架構 (ISA) 是電腦架構的一部份，所述電腦架構的一部份是與程式化有關、包含本體資料型式、指令、暫存器架構、定址模式、記憶體架構、中斷及例外處理、以及外部輸入及輸出 (I/O)。應注意，指令一詞於此大致上意指巨集指令，其是提供給處理器用於執行的指令，巨集指令係與微指令或微運算 (micro-op) 相反的，微指令或微運算是將巨集指令解碼之處理器的解碼器的結果。微指令或微運算可以配置成指令處理器上的執行單元執行運算以實施與巨集指令相關的邏輯。

ISA 與用以實施指令集的處理器的處理器設計技術集之微架構有所區別。設有不同微架構的處理器共用共同的指令集。舉例而言，Intel[®] Pentium 4 處理器、Intel[®] Core[™] 處理器、及來自加州太陽谷的超微公司的處理器實

5

施幾乎相同版本的 x86 指令集（某些程度上增加更新的版本），但具有不同的內部設計。舉例而言，在使用習知的技術之不同微架構中，以不同方式實施相同的 ISA 暫存器架構，其包含專用的實體暫存器、使用暫存器重命名機制（例如使用暫存器別名表（RAT）、重排序緩衝器（ROB）及退出暫存器檔案。除非另外指明，否則，此處使用暫存器架構、暫存器檔案、及暫存器之文句以表示軟體／程式人員可看到者、及指令指明暫存器的方式。在需要明確性的情形中，將使用形容詞「邏輯的」、「架構的」、或「軟體可看見的」來表示暫存器架構中的暫存器／檔案，而不同的形容詞會被用以指明給定的微架構中的目的地暫存器（例如，實體暫存器、重排序緩衝器、退出暫存器、暫存器池）。

指令集包含一或更多指令格式。給定的指令格式界定不同的欄位（位元數目、位元位置）以特別指明要被執行的運算（運算碼）以及運算要於其上執行的運算元。某些指令格式可以經由指令樣板（或副子令格式）的界定而進一步分解。舉例而言，給定的指令格式的指令樣板可以被界定為具有不同子集合的指令格式欄位（所包含的多個欄位典型上是相同次序，但是，因為有較少的欄位被包含，所以，至少某些欄位具有不同的位元位置）及/或被界定為具有被不同解譯的給定欄位。使用給定的指令格式（以及，假使被界定時，在該指令格式的多個指令樣板中的給定之一中）以表示給定的指令，以及，指明運算及運算

元。指令串是特定的指令序列，其中，在序列中的各指令是依指令格式的指令之存在（以及，假使被界定时，是该指令格式的多個指令樣板中的給定之一）。

【圖式簡單說明】

從配合下述附圖的詳細說明，將可以更佳地瞭解本發明。

圖 1A 及 1B 是方塊圖，顯示根據本發明的實施例之泛型向量友善指令格式及其指令樣板；

圖 2A-D 是方塊圖，顯示根據本發明的實施例之舉例說明的特定向量友善指令格式；

圖 3 是根據本發明的一實施例之暫存器架構的方塊圖；

圖 4A 是方塊圖，顯示根據本發明的實施例之舉例說明的有序提取、解碼、退出管線及舉例說明的暫存器重命名、亂序議題/執行管線；

圖 4B 是方塊圖，顯示根據本發明的實施例之包含在處理器中的有序提取、解碼、退出核心及舉例說明的暫存器重命名、亂序議題/執行架構核心；

圖 5A 是方塊圖，顯示單一核心、與其與晶粒上互連網路的連接；

圖 5B 顯示根據本發明的實施例之圖 5A 的部份處理器核心的放大視圖；

圖 6 是方塊圖，顯示根據本發明的實施例之單核心處 5

理器及具有集成的圖形和記憶體控制器之多核心處理器；

圖 7 顯示根據本發明的一實施例之系統的方塊圖；

圖 8 顯示根據本發明的實施例之第二系統的方塊圖；

圖 9 顯示根據本發明的實施例之第三系統的方塊圖；

圖 10 顯示根據本發明的實施例之系統晶片的方塊圖；

圖 11 是方塊圖，對比根據本發明的實施例之將源指令集中的二進位指令轉換成目標指令集中的二進位指令之軟體指令轉換器的使用；

圖 12 顯示本發明的實施例實施於其上的舉例說明的處理器；

圖 13-18 顯示使用不同的立即值之根據本發明的一實施例之向量位元反轉邏輯；

圖 19 顯示根據本發明的一實施例之方法；

圖 20 顯示位元反轉運算集，被執行以反轉暫存器中的位元次序；

圖 21 顯示本發明的一實施例實施於其上的舉例說明的處理器；

圖 22-24 顯示本發明的一實施例中使用的向量位元反轉及交叉技術；

圖 25 顯示舉例說明的使用向量位元反轉及交叉技術而執行的轉置運算；

圖 26A-B 顯示舉例說明的使用向量位元反轉及交叉技術而執行的傾斜原始運算；及

圖 27 顯示根據本發明的一實施例之方法。

【發明內容及實施方式】

在下述說明中，為了說明之目的，揭示保留特定細節以助於更完整瞭解下述的本發明的實施例。但是，習於此技藝者顯然清楚知道，沒有這些特定細節中的某些細節，仍可實施本發明的實施例。在其它情形中，未詳細顯示習知的結構及裝置，以免模糊本發明的實施例之基本原理。

舉例說明的處理器架構及資料型式

指令集包含一或更多指令格式。給定的指令格式界定不同的欄位（位元數目、位元位置）以特別指明要被執行的運算（運算碼）以及運算要對其執行的運算元。某些指令格式可以經由指令樣板（或副子令格式）的界定而進一步分解。舉例而言，給定的指令格式的指令樣板可以被界定為具有不同子集合的指令格式欄位（所包含的多個欄位典型上是相同次序，但是，因為有較少的欄位被包含，所以，至少某些欄位具有不同的位元位置）及/或被界定為具有被不同解譯的給定欄位。因此，使用給定的指令格式（以及，假使被界定時，在該指令格式的多個指令樣板中的給定之一中）以表示 ISA 的各指令，以及，包含用於指明運算及運算元。舉例而言，舉例說明的 ADD 指令具有特定運算碼及指令格式，指令格式包含運算碼欄位以指定該運算碼及運算元欄位以選取運算元的運算元（源 1/目 5

的地及源 2)；以及，在指令串中此 ADD 指令的存在將在選取特定運算元的運算元欄位中具有特定內容。稱為進階向量擴充 (AVX)(AVX1 及 AVX2)以及使用向量擴充 (VEX)碼化設計的 SIMD 擴充集已經發行及/或公開(舉例而言，請參照 2011 年 10 月之 Intel®64 和 IA-32 架構軟體開發者手冊；以及，2011 年 6 月之 Intel®進階向量擴充程式參考)。

舉例說明的指令格式

此處所述的指令實施例可以以不同格式具體實施。此外，於下詳述舉例說明的系統、架構、及管線。在些系統、架構、及管線上執行指令的實施例，但是不限於詳述的實施例。

A. 泛型向量友善指令格式

向量友善指令格式是適用於向量指令的指令格式(例如，有某些欄位特定用於向量運算)。雖然說明經由向量友善指令格式而支援向量及純量運算之實施例，但是，替代的實施例可以僅使用向量友善指令格式向量運算。

圖 1A-1B 是方塊圖，顯示根據本發明的實施例之泛型向量友善指令格式及其指令樣板。圖 1A 是方塊圖，顯示根據本發明的實施例之泛型向量友善指令格式及其等級 A 指令樣板；而圖 1B 是方塊圖，顯示根據本發明的實施例之泛型向量友善指令格式及其等級 B 指令樣板。具體而

言，等級 A 及等級 B 指令樣板被界定用於泛型向量友善指令格式 100，等級 A 及等級 B 指令樣板都未包含記憶體存取 105 指令樣板及記憶體存取 120 指令樣板。在向量友善指令格式的內容中泛型一詞係指指令格式未被束縛於任何特定的指令集。

將說明本發明的實施例，其中，向量友善指令格式支援下述：具有 32 位元（4 位元組）或是 64 位元（8 位元組）資料元件寬度（或大小）之 64 位元組向量運算元長度（或大小）（因此，64 位元組向量由 16 個雙倍字大小元或是替代地由 8 個四倍字大小元組成）；具有 16 位元（2 位元組）或是 8 位元（1 位元組）資料元件寬度（或大小）之 64 位元組向量運算元長度（或大小）；具有 32 位元（4 位元組）、64 位元（8 位元組）、16 位元（2 位元組）或是 8 位元（1 位元組）資料元件寬度（或大小）之 32 位元組向量運算元長度（或大小）；以及，具有 32 位元（4 位元組）、64 位元（8 位元組）、16 位元（2 位元組）或是 8 位元（1 位元組）資料元件寬度（或大小）之 16 位元組向量運算元長度（或大小）；替代實施例以更多、更少或是不同的資料元件寬度（例如，128 位元（16 位元組）資料元件寬度）來支援更多、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）。

在圖 1A 中的等級 A 指令樣板包含：1) 在無記憶體存取 105 指令樣板之內，顯示無記憶體存取、完全捨入控制型運算 110 指令樣板及無記憶體存取、資料轉換型式運算

115 指令樣板；以及，2)在記憶體存取 120 指令樣板之內，有顯示記憶體存取、暫時 125 指令樣板及記憶體存取、非暫時 130 指令樣板。圖 1B 中的等級 B 指令樣板包含：1)在無記憶體存取 105 指令樣板之內，顯示無記憶體存取、寫入遮罩控制、部份捨入控制型運算 112 指令樣板及無記憶體存取、寫入遮罩控制、向量長度型式運算 117 指令樣板；以及，2)在記憶體存取 120 指令樣板之內，顯示有記憶體存取、寫入遮罩控制 127 指令樣板。

泛型向量友善指令格式 100 包含以圖 1A-1B 中所示的次序列出的下述欄位。

格式欄位 140 – 在此欄位中的特定值（指令格式識別符值）獨特地識別向量友善指令格式，並因而識別指令串中向量友善指令格式的指令發生。確切而言，以此欄位對於僅有泛型向量友善指令格式之指令集並非所需的觀點而言，此欄位是選加的。

基礎運算欄位 142 – 其內容區別不同的基礎運算。

暫存器索引欄位 144 – 其內容直接或是經由位址產生而指明源及目的地運算元在暫存器或在記憶體中的位置。這些包含足夠數目的位元以從 $P \times Q$ (例如 32×512 、 16×128 、 32×1024 、 64×1024) 暫存器檔案中選取 N 個暫存器。雖然在一實施例中， N 可以高達三個源及一個目的地暫存器，但是，替代實施例可以支援更多或是更少的源及目的地暫存器（例如，可以支援高達二個源，其中，這些源中之一也作為目的地，可以支援高達三個源，其中，這

些源中之一也作為目的地，可以支援高達二個源及一個目的地）。

修飾符欄位 146 – 其內容區別指定及未指定記憶體存取的泛型向量指令格式中指令的發生；亦即，在無記憶體存取 105 指令樣板與記憶體存取 120 指令樣板之間作區分。記憶體存取運算對記憶體層級結構讀取及/或寫入（在某些情形中，使用暫存器中的值指定源及/或目的地址），而非記憶體存取運算未如此作（例如，源及目的地是暫存器）。雖然在一實施例中，此欄位也在三不同方式之間選取以執行記憶體位址計算，但是，替代的實施例可以支援更多、更少、或是不同的方式以執行記憶體位址計算。

擴增運算欄位 150 – 其內容區別多種不同運算中那一運算要加上基礎運算被執行。此欄位是內容特定的。在本發明的一實施例中，此欄位分成等級欄位 168、阿爾發欄位 152、貝它欄位 154。擴增運算欄位 150 允許共同的運算組以單一指令而非 2、3 或 4 指令執行。

比例欄位 160 – 其內容允許索引欄位的內容的比例用於記憶體位址產生（例如，用於使用 $2^{\text{scale}} \cdot \text{索引} + \text{基礎}$ 之位址產生）。

位移欄位 162A – 其內容作為記憶體位址產生的一部份（例如，用於使用 $2^{\text{scale}} \cdot \text{索引} + \text{基礎} + \text{位移}$ 之位址產生）。

位移因數欄位 162B（注意，位移欄位 162A 直接在位 5

移因數欄位 162B 上方並列係表示一或另一者被使用) – 其內容作為位址產生的一部份；其指明要依記憶體存取 (N) 的大小而比例化之位移因數，其中，N 是記憶體存取中的位元組數目 (例如，用於使用 $2^{\text{scale}} * \text{索引} + \text{基礎} + \text{比例化位移}$ 的位址產生)。冗餘的低階位元被忽略，因此，位移因數欄的內容乘以記憶體運算元全部大小 (N) 以產生計算有效位址時使用的最後位移。根據全運算碼欄位 174(稍後說明)及資料操縱欄位 154C，在運行時間時由處理器硬體決定 N 的值。在位移欄位 162A 及位移因數欄位 162B 不用於無記憶體存取 105 指令樣板及 / 或不同實施例僅實施二者中之一或不實施二者的情形中，它們是選加的。

資料元件寬度欄位 164 – 其內容區別一些資料元件寬度中的那一寬度是要被使用(在某些實施例中對所有指令；在其它實施例中僅對某些指令)。在假使使用運算碼的某些態樣而僅有一資料元件寬度被支援及 / 或資料元件寬度被支援時不需要此欄位的情形中，此欄位是選加的。

寫入遮罩欄位 170 – 其內容以每一資料元件位置為基礎而控制目的地向量運算元中的資料元件位置是否反應基礎運算及擴增運算的結果。等級 A 指令樣板支援合併寫入遮罩，而等級 B 指令樣板支援合併及零化寫入遮罩。當合併時，向量遮罩允許目的地中任何組的元在任何運算的執行期間受保護 (由基礎運算及擴增運算指明) 而免於更新；在其它的一實施例中，保留對應的遮罩位元具有 0 之

目的地的各元的舊值。相反地，當零化向量遮罩時允許目的地中的任何組的元在任何運算（由基礎運算及擴增運算指定）執行期間被零化；在一實施例中，當對應的遮罩位元具有 0 值時，目的地的元被設定為 0。此功能的子集合是控制執行的運算的向量長度之能力（亦即，被修改的元的展幅，從第一至最後一個）；但是，被修改的元不必是連續的。如此，寫入遮罩欄位 170 允許部份向量運算，包含載入、儲存、算術、邏輯、等等。雖然說明本發明的實施例，其中，寫入遮罩欄位 170 的內容選取一些含有要被使用的寫入遮罩之寫入遮罩暫存器中之一（因此，寫入遮罩欄位 170 的內容間接地識別要被執行的遮罩），但是，替代實施例取代地或增加地允許遮罩寫入欄位 170 的內容直接指定要被執行的遮罩。

立即欄位 172 — 其內容允許立即性的規格。在此欄位未出現在未支援立即性的泛型向量友善格式實施中以及其未出現在未使用立即性的指令中之情形中，此欄位是選加的。

等級欄位 168 — 其內容區別不同等級的指令。參考圖 1A-B，此欄位的內容在等級 A 與等級 B 指令之間選取。在圖 1A-B 中，四角圓化的方形被用以標示特定值存在於欄位中（例如，分別在圖 1A-B 中用於等級欄位 168 的等級 A 168A 及等級 B 168B）。

等級 A 的指令樣板

5

在等級 A 的非記憶體存取 105 指令樣板的情形中，阿爾發欄位 152 被解譯為 RS 欄位 152A，其內容區別不同的擴增運算型式中那一型式是要被執行（例如，捨入 152A.1 及資料轉換 152A.2 分別被指定用於無記憶體存取、捨入型式運算 110 及無記憶體存取、資料轉換型式運算 115 指令樣板），而貝他欄位 154 區別被指定的型式的運算中那一運算是被執行。在無記憶體存取 105 指令樣板中，比例欄位 160、位移比例欄位 162A、及位移比例欄位 162B 未出現。

無記憶體存取指令樣板－完全捨入控制型運算

在無記憶體存取完全捨入控制型運算 110 指令樣板中，貝他欄位 154 被解譯為捨入控制欄位 154A，其內容提供靜態捨入。雖然在上述的發明實施例中，捨入控制欄位 154A 包含抑制所有浮點例外（SAE）欄位 156 及捨入運算控制欄位 158，但是，替代實施例可以支援將這二概念編碼成相同欄位或是僅具有這些概念／欄位中之一或另一者（例如，可以僅具有捨入運算控制欄位 158）。

SAE 欄位 156－其內容區別是否將例外事件報告禁能；當 SAE 欄位的 156 內容標示抑制被賦能時，給定的指令不報告任何種類的浮點例外旗標及不喚起任何浮點例外處理器。

捨入運算控制欄位 158－其內容區別捨入運算組中何者要執行（例如，捨進、捨去、捨入至零、及捨入至最接

近的)。因此，捨入運算控制欄位 158 允許以每一指令為基礎之捨入模式的改變。在處理器包含用於指明捨入模式的控制暫存器之發明的一實施例中，捨入運算控制欄位 150 的內容置換該暫存器值。

無記憶體存取指令樣板－資料轉換型式運算

在無記憶體存取資料轉換型式運算 115 指令樣板中，貝他欄位 154 被解譯為資料轉換欄位 154B，其內容區別多個資料轉換中何者要被執行（例如，無資料轉換、調和、廣播）。

在等級 A 的記憶體存取 120 指令樣板的情形中，阿爾發欄位 152 被解譯為逐出暗示欄位 152B，其內容區別逐出暗示中那一者是要被使用（在圖 1A 中，暫時 152B.1 及非暫時 152B.2 分別被指定用於記憶體存取、暫時 125 指令樣板以及記憶體存取、非暫時 130 指令樣板），而貝他欄位 154 被解譯為資料操縱欄位 154C，其內容區別多個資料操縱運算（也稱為基元）中何者是要被執行（例如，無操縱；廣播；源的上轉；及目的地的下轉）。記憶體存取 120 指令樣板包含比例欄位 160，且選加地包含位移欄位 162A 或位移比例欄位 162B。

藉由轉換支援，向量記憶體指令對記憶體執行向量載入及向量儲存。如正常的向量指令般，向量記憶體指令以像資料元件方式對記憶體傳輸資料，而真正被傳送的元是由被選為寫入遮罩的向量遮罩的內容所指定。

記憶體存取指令樣板－暫時

暫時資料是很可能足夠快再被使用而從快取獲利之資料。但是，這是暗示，以及，不同的處理器可以以不同方式實施它，包含完全忽略暗示。

記憶體存取指令樣板－非暫時

非暫時資料是不太可能足夠快再被使用以從第一層快取記憶體中的快取獲利之資料，且應被授予逐出優先權。但是，這是暗示，以及，不同的處理器可以以不同方式實施它，包含完全忽略暗示。

等級 B 的指令樣板

在等級 B 的指令樣板的情形中，阿爾發欄位 152 被解譯為寫入遮罩控制 (Z) 欄位 152C，其內容區別由寫入遮罩欄位 170 控制的寫入遮罩是否應為合併或零化。

在等級 B 的非記憶體存取 105 指令樣板的情形中，貝他欄位 154 的部份是被解譯為 RL 欄位 157A，其內容區別不同的擴增運算型式中那一運算型式是要被執行（例如，捨入 157A.1 以及向量長度(VSIZE)157A.2 分別被指定用於無記憶體存取、寫入遮罩控制、部份捨入控制型運算 112 指令樣板以及無記憶體存取、寫入遮罩控制、VSIZE 型式運算 117 指令樣板），而貝他欄位 154 的其餘部份區別具有指定型式的運算中何者是要被執行。在無記憶體存

取 105 指令樣板中，比例欄位 160、位移欄位 162A、及位移比例欄位 162B 不存在。

在無記憶體存取中，寫入遮罩控制、部份捨入控制型運算 110 指令樣板、貝他欄位 154 的其它部份被解譯為捨入運算欄位 159A 且例外事件報告被禁能（被給定的指令不報告任何種類的浮點例外旗標及不喚起任何浮點例外處理器）。

捨入運算控制欄位 159A – 正如同捨入運算控制欄位 158 般，其內容區別捨入運算組中何者要執行（例如，捨進、捨去、捨入至零、及捨入至最接近的）。因此，捨入運算控制欄位 159A 允許以每一指令為基礎之捨入模式的改變。在處理器包含用於指明捨入模式的控制暫存器之發明的一實施例中，捨入運算控制欄 150 的內容置換該暫存器值。

在無記憶體存取、寫入遮罩控制，VSIZE 型式運算 117 指令樣板中，貝他欄位 154 的其餘部份被解譯為向量長度欄位 159B，其內容區別要對多個資料向量長度中的那一資料向量長度執行（例如，128、256、或 512 位元組）。

在等級 B 的記憶體存取 120 指令樣板的情形中，貝它欄位 154 的一部份被解譯為廣播欄位 157B，其內容區別廣播型資料操縱運算是否要被執行，而貝它欄位 154 中的其它部份被解譯為向量長度欄位 159B。記憶體存取 120 指令樣板包含比例欄位 160，以及，選加地包含位移欄位

5

162A 或是位移比例欄位 162B。

關於泛型向量友善指令格式 100，完全運算碼欄位 174 顯示為包含格式欄位 140、基礎運算欄位 142、及資料元件寬度欄位 164。雖然顯示完全運算碼欄位 174 包含所有這些欄位的一實施例，但是，在未支援它們全部的實施例中，完全運算碼欄位 174 包含小於所有這些欄位的欄位。完全運算碼欄位 174 提供運算碼(opcode)。

擴增運算欄位 150、資料元件寬度欄位 164、及寫入遮罩欄位 170 允許以泛型向量友善指令格式的指令為基礎來指明這些特點。

寫入遮罩欄位及資料元件寬度欄位的結合會產生型式化的指令，其中，它們允許根據不同的資料元件寬度來施加遮罩。

等級 A 及等級 B 之內發現的各種指令樣板在不同的情形中是有利的。在本發明的某些實施例中，不同的處理器或是處理器之內不同的核心可以僅支援等級 A、僅支援等級 B、或支援這二等級。舉例而言，要用於一般目的之計算的高性能目的之亂序核心僅支援等級 B，主要用於圖形及/或科學（通量）計算的核心僅支援等級 A，以及用於支援二等級的核心可以支援二等級（當然，具有來自二等級的樣板及指令的某些混合，但非來自二等級的所有樣板及指令是在本發明的範圍之內）。而且，單一處理器包含多核心，所有這些核心支援相同等級，或者，其中，不同的核心支援不同的等級。舉例而言，在設有分別的圖形

及一般用途核心的處理器中，主要用於圖形及/或科學計算的複數個圖形核心之一僅支援等級 A，而一或更多一般用途核心可為具有用於僅支援等級 B 的一般用途計算之亂序執行及暫存器重命名之高性能一般用途核心。不具有分別的圖形核心之另一處理器可以包含支援等級 A 和等級 B 的一或更多一般用途有序或亂序核心。當然，在本發明的不同實施例中，來自一等級的特點也可在其它等級中實施。以高階語言撰寫的程式將被置於（舉例而言，僅即時被編譯或是靜態地被編譯成）各式各樣之不同的可執行形式中，包含：1)僅具有由用於執行的標的處理器所支援的等級的指令之形式；或是，2)具有使用所有等級的指令之不同組合撰寫的替代常式以及具有控制流程碼之形式，控制流程碼係根據由目前正執行碼的處理器所支援的指令而選取要執行的常式。

舉例說明的特定向量友善指令格式

圖 2 是方塊圖，顯示根據本發明的實施例之舉例說明的特定向量友善指令格式。圖 2 顯示特定向量友善指令格式 200，其在指定欄位的位置、大小、解譯、及次序、以及用於那些欄位中的某些欄位之值的情形中是特定的。特定向量友善指令格式 200 可以被用以擴充 x86 指令集，因而某些欄位類似於或同於現有的 x86 指令集中使用的欄位以及其擴充（例如，AVX）。此格式維持與具有延伸的現有 x86 指令集的前置編碼欄位、實數運算碼位元組欄位、

5

MOD R/M 欄位、SIB 欄位、位移欄位、及立即欄位一致。顯示來自圖 2 的欄位映射至來自圖 1 的欄位。

應瞭解，雖然為了說明而在泛型向量友善指令格式 100 的脈絡中參考特定向量友善指令格式 200，以說明本發明的實施例，但是，除非特別申明，否則本發明不侷限於特定向量友善指令格式 200。舉例而言，泛型向量友善指令格式 100 慮及用於各種欄位的各種可能大小，而特定向量友善指令格式 200 顯示為具有特定大小的欄位。具體舉例而言，雖然資料元件寬度欄位 164 顯示為特定向量友善指令格式 200 中的一位元欄位，但是，本發明不限於此（亦即，泛型向量友善指令格式 100 慮及資料元件寬度欄位 164 的其它大小）。

泛型向量友善指令格式 100 包含依圖 2A 中所示的次序而於下列出的下述欄位。

EVEX 前置(位元組 0-3)202-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 以及其含有 0x62(用於區別發明的一實施例中向量友善指令格式的獨特值)。

第二—第四位元組 (EVEX 位元組 1-3)包含提供特定能力的一些位元欄位。

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 位元欄位(EVEX 位元組 1，位元[6]-X)、及 157BEX 位元組 1，位

元[5]-B)組成。EVEX.R、EVEX.X及EVEX.B位元欄位提供與對應的VEX位元欄位相同的功能性，且使用1s互補形式來編碼，亦即，ZMM0被編碼為1111B，ZMM15被編碼為0000B。如同此技藝中所知般，指令的其它欄位將暫存器索引的較低的三個位元編碼(rrr, xxx,及bbb)，以致於藉由加上EVEX.R、EVEX.X、及EVEX.B，可以形成Rrrr、Xxxx、及Bbbb。

REX'欄位110—這是REX'欄位110的第一部份且是用以將擴充的32暫存器組的上16或下16編碼之EVEX.R'位元欄位(EVEX位元組1，位元[4]-R')。在本發明的一實施例中，此位元與如下標示的其它位元以位元反轉格式儲存，以與BOUND指令區別(在已知的x86 32位元模式中)，BOUND指令的實數運算碼位元組是62，但是，在MOD R/M欄位(下述)中未接受MOD欄位中11的值；本發明的替代實施例未以反轉格式儲存此位元及下述其它標示的位元。1的值用以將下16暫存器編碼。換言之，藉由結合來自其它欄位的EVEX.R'、EVEX.R、及其它RRR以形成R'Rrrr。

運算碼映射欄位215(EVEX位元組1，位元[3:0]-mmmm)—它的內容將暗指的領先運算碼位元組(0F, 0F 38, 或0F 3)編碼。

資料元件寬度欄位164(EVEX位元組2，位元[7]-W)—由記號EVEX.W表示。EVEX.W用以界定資料型式的粒度(大小)(32位元資料元件或64位元資料元

件)。

EVEX.vvvv 220(EVEX 位元組 2, 位元[6:3]-vvvv) – EVEX.vvvv 的角色可以包含下述：1)EVEX.vvvv 將以反轉 (1s 互補) 形式指定的第一源暫存器運算元編碼，且對於具有 2 或更多源運算元的指令是有效的；2)EVEX.vvvv 將對某些向量偏移以 1s 互補形式指定的目的地暫存器運算元編碼；或者 3)EVEX.vvvv 未將任何運算元編碼，欄位被保留且應含有 1111b。因此，EVEX.vvvv 欄位 220 將依反轉 (1s 互補) 形式儲存的第一源暫存器指定符的 4 低階位元編碼。取決於指令，額外的不同 EVEX 位元欄位被用以擴充指定符大小至 32 暫存器。

EVEX.U 168 等級欄位 (EVEX 位元組 2, 位元[2]-U) – 假使 EVEX.U=0, 則其標示等級 A 或 EVEX.U0；假使 EVEX.U=1, 則其標示等級 B 或 EVEX.U1。

前置編碼欄位 225(EVEX 位元組 2, 位元[1:0]-pp) – 提供額外的位元用於基礎運算欄位。除了提供支援給 EVEX 前置格式的舊有 SSE 指令，這也具有使 SIMD 前置小巧化 (而非要求位元組來表示 SIMD 前置, EVEX 前置僅要求 2 位元) 的優點。在一實施例中，為了支援以舊有格式及 EVEX 前置格式等二各式使用 SIMD 前置 (66H、F2H、F3H) 之舊有 SSE 指令，這些舊有 SIMD 前置被編碼成 SIMD 前置編碼欄位；以及，在被提供給解碼器的 PLA (以致於 PLA 可以執行這些舊有指令的舊有及 EVEX 等二格式而不用修改) 之前，在運行時間時被擴展成舊有

SIMD 前置。雖然較新的指令可以直接使用 EVEX 前置編碼欄位的內容作為運算碼擴充，但是，某些實施例為了一致性而以類似方式擴展但允許不同的意義由這些舊有 SIMD 前置規定。替代實施例可以重新設計 PLA 以支援 2 位元 SIMD 前置編碼，因而不要求擴展。

阿爾發欄位 152(EVEX 位元組 3，位元[7]—EH；也稱為 EVEX.EH、EXEX.rs、EVEX.RL、EVEX.寫入遮罩控制、及 EVEX.N；也以 α 顯示)—如以往所述般，此欄位是內容特定的。

貝他欄位 154(EVEX 位元組 3，位元[6:4]—SSS；也稱為 EVEX.s₂₋₀、EVEX.r₂₋₀、EVEX.rr1、EVEX.LL0、EVEX.LLB；也以 $\beta\beta\beta$ 顯示)—如以往所述般，此欄位是內容特定的。

REX'欄位 110—這是 REX'欄位的餘部且是可被用以將擴充的 32 暫存器組的上 16 或下 16 編碼之 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儲存。1 的值被用以將下 16 暫存器編碼。換言之，藉由結合 EVEX.V'、EVEX.vvvv 以形成 V'VVVV。

寫入遮罩欄位 170(EVEX 位元組 3，位元[2:0]-kkk)—如同先前所述般，其內容指定寫入遮罩暫存器中的暫存器索引。在發明的一實施例中，特定值 EVEX.kkk=000 具有特別的表現，暗指無寫入遮罩用於特定的指令（這可以以各種方式實施，包含使用實體接線至所有或硬體之寫入遮罩，所述硬體是繞過遮罩硬體）。

實數運算碼欄位 230(位元組 4)也稱為運算碼位元組。在此欄位中指定運算碼的一部份。

MOD R/M 欄位 240(位元組 5)包含 MOD 欄位 242、Reg 欄位 244、及 R/M 欄位 246。如同先前所述般，MOD 欄位 242 的內容區別記憶體存取與非記憶體存取運算。Reg 欄位 244 的角色可以總合為二情形：將目的地暫存器運算元或源暫存器運算元編碼，或是被當作運算碼擴充來處理且不被用以將任何指令運算元編碼。R/M 欄位 246 的角色可以包含下述：將參考記憶體位址的指令運算元編碼，或者，將目的地暫存器運算元或源暫存器運算元編碼。

比例、索引、基礎(SIB)位元組(位元組 6)－如同先前所述般，比例欄位 150 的內容是用於記憶體位址產生。SIB.xxx 254 及 SIB.bbb 256－這些欄位的內容先前已被述及與暫存器索引 Xxxx 及 Bbbb 有關。

位移欄位 162A(位元組 7-10)－當 MOD 欄位 242 含有 10 時，位元組 7-10 是位移欄位 162A，且其與舊有 32 位元位移($disp_{32}$)相同工作並以位元組粒度工作。

位移因數欄位 162B(位元組 7)－當 MOD 欄位 242 含有 01 時，位元組 7 是位移因數欄位 162B。此欄位的位置與以位元組粒度工作的舊有 x86 指令集 8 位元位移($disp_8$)的位置相同。由於 $disp_8$ 是正負號擴充，所以，其僅可以在 -128 與 127 位元組差距之間定址；以 64 位元組快取線的觀點而言， $disp_8$ 使用可以被設定為僅四個真

正有用的值 -128、-64、0、及 64 之 8 位元；由於通常需要更大的範圍，所以，使用 `disp32`；但是，`disp32` 要求 4 位元組。與 `disp8` 和 `disp32` 相反，位移因數欄位 162B 是 `disp8` 的再解譯；當使用位移因數欄位 162B 時，真實的位移由位移因數欄位的內容乘以記憶體運算元存取 (N) 所決定。此型式的位移被稱為 `disp8*N`。這降低平均的指令長度（用於位移但是具有更大範圍的單一位元組）。此被壓縮的位移是根據有效的位移是記憶體存取的粒度之倍數的假設，因此，位址差距的冗餘低階位元不需被編碼。換言之，位移因數欄位 162B 替代舊有 x86 指令集 8 位元位移。因此，位移因數欄位 162B 以同於 x86 指令集 8 位元位移的方式編碼（以致於 ModRM/SIM 編碼規則不變），僅有的例外是 `disp8` 被過載至 `disp8*`。換言之，編碼規則或編碼長度沒有變化，但是僅有硬體對位移值的解譯有變化（這需要將位移依記憶體運算元的大小來比例化以取得位元組方式的位址差距）。

立即欄位 172 如先前所述地操作。

完全運算碼欄位

圖 2B 是方塊圖，顯示根據本發明的一實施例之構成完全運算碼欄位 174 之特定的向量友善指令格式 200 的欄位。具體而言，完全運算碼欄位 174 包含格式欄位 140、基礎運算欄位 142、及資料元件寬度 (W) 欄位 164。基礎運算欄位 142 包含前置編碼欄位 225、運算碼映射欄位 s

215、及實數運算碼欄位 230。

暫存器索引欄位

圖 2C 是方塊圖，顯示根據本發明的一實施例之構成暫存器索引欄位 144 之特定的向量友善指令格式 200 的欄位。具體而言，暫存器索引欄位 144 包含 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、及 bbb 欄位 256。

擴增運算欄位

圖 2D 是方塊圖，顯示根據本發明的一實施例之構成擴增運算欄位 150 之特定向量友善指令格式 200 的欄位。當等級(U)欄位 168 含有 0 時，其意謂 EVEX.U0(等級 A 168A)；當其含有 1 時，其表示 EVEX.U1(等級 B 168B)。當 U=0 及 MOD 欄位 242 含有 11 時（表示無記憶體存取運算），阿爾發欄位 152（EVEX 位元組 3，位元[7]-EH）被解譯為 rs 欄位 152A。當 rs 欄位 152A 含有 1(捨入 152A.1)時，貝它欄位 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 154A 包含一位元 SAE 欄位 156 及二位元捨入運算欄位 158。當 rs 欄位 152A 含有 0(資料轉換 152A.2)時，貝它欄位 154（EVEX 位元組 3，位元[6:4]-SSS）被解譯為三位元資料轉換欄位 154B。當 U=0 及 MOD 欄位 242 含有 00、

01、或 10(表示記憶體存取運算)時，阿爾發欄位 152 (EVEX 位元組 3，位元 [7]-EH) 被解譯為逐出暗示 (EH) 欄位 152B 及貝它欄位 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 時，阿爾發欄位 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 及 MOD 欄位 242 含有 11(表示無記憶體存取運算)時，貝它欄位 154 的一部份 (EVEX 位元組 3，位元 [4]-S₀) 被解譯為 RL 欄位 157A；當其含有 1 (捨入 157A.1) 時，貝它欄位 154 的其餘部份 (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 含有 0 (VSIZE 157.A2) 時，貝它欄位 154 的其餘部份 (EVEX 位元組 3，位元 [6-5]-S₂₋₁) 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 及 MOD 欄位 242 含有 00、01 或 10(表示記憶體存取運算)時，貝它欄位 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]-L₁₋₀) 及廣播欄位 157B(EVEX 位元組 3，位元 [4]-B)。

C. 舉例說明的暫存器架構

圖 3 是根據本發明的一實施例之暫存器架構 300 的方塊圖。在所示的實施例中，有 512 位元寬的 32 個向量暫存器 310；這些暫存器稱為 zmm0 至 zmm31。較低的 16 zmm 暫存器的低階 256 位元覆蓋於暫存器 ymm0-16 之 s

上。較低的 16-zmm 暫存器的低階 128 位元(ymm 暫存器的低階 128 位元)覆蓋於暫存器 xmm0-15 之上。如下述表格中所示般，特定向量友善指令格式 200 對這些被覆蓋的暫存器檔案操作。

可調整的向量長度	等級	操作	暫存器
不包含向量長度欄位 159B 的指令樣板	A(圖 1A ; U=0)	110、115、125、130	zmm 暫存器(向量長度是 64 位元組)
	B(圖 1B ; U=1)	112	zmm 暫存器(向量長度是 64 位元組)
包含向量長度欄位 159B 的指令樣板	B(圖 1B ; U=1)	117、127	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)取決於向量長度欄位 159B

換言之，向量長度欄位 159B 在最大長度與一或更多其它較短的長度之間選取，其中，每一此較短長度是先前長度的長度之一半；以及，不具向量長度欄位 159B 的指令樣板對最大向量長度運算。此外，在一實施例中，特定向量友善指令格式 200 的等級 B 指令樣板對緊縮的或純量的單一／雙倍精度浮點資料及緊縮的或純量的整數資料運算。純量運算是對 zmm/ymm/xmm 暫存器中最低階資料元件位置執行的運算；更高階的資料元件位置視實施例而保留成與它們在指令之前相同或者零化。

寫入遮罩暫存器 315— 在所示的實施例中，有 8 個大小均為 64 位元之寫入遮罩暫存器 (k0 至 k7)。在替代實

施例中，寫入遮罩暫存器 315 大小為 16 位元。如同先前所述般，在本發明的一實施例中，向量罩暫存器 k0 無法作為寫入遮罩；當正常標示 k0 的編碼用於寫入遮罩時，其選取 0xFFFF 的實體接線的寫入遮罩，有效地使用於該指令的寫入遮罩禁能。

一般用途暫存器 325 – 在所示的實施例中，有十六個 64 位元一般用途暫存器，它們與現有的 x86 定址模式一起用以將記憶體運算元定址。這些暫存器以名稱 RAX、RBX、RCX、RDX、RBP、RSI、RDI、RSP、及 R8 至 R15 表示。

純量浮點堆疊暫存器檔案 (x87 堆疊) 345，MMX 緊縮整數平坦暫存器檔案 350 疊於其上 – 在所示的實施例中，x87 堆疊是八元堆疊，用以使用 x87 指令集延伸，對 32/64/80 位元浮點資料執行純量浮點運算；而 MMX 暫存器被用以對 64 位元緊縮暫存器資料執行運算，以及為了某些在 MMX 及 XMM 暫存器之間執行的運算而固持運算元。

本發明的替代實施例可以使用更寬或更窄的暫存器。此外，本發明的替代實施例可以使用更多、更少、或不同的暫存器檔案及暫存器。

D. 舉例說明的核心架構、處理器、及電腦架構

為了不同目的而以不同方式、及在不同處理器中，實施處理器核心。舉例而言，這些核心的實施包含：1)要用 5

於一般用途計算的一般用途有序核心；2) 要用於一般用途計算的高性能一般用途亂序核心；3) 主要用於圖形及/或科學（輸貫量）計算的特定用途的核心。不同處理器的實施包含：1) CPU，包含要用於一般用途計算的一或更多一般用途有序核心及/或要用於一般用途計算的一或更多一般用途亂序核心；及 2) 副處理器，包含主要用於圖形及/或科學（輸貫量）計算的一或更多特定用途核心。這些不同的處理器導致不同的電腦系統架構，包含：1) 在與 CPU 分別的晶片上的副處理器；2) 在與 CPU 相同的封裝中在分別的晶粒上的副處理器；3) 在與 CPU 相同的晶粒上的副處理器（在此情形中，此副處理器有時被稱為特定用途邏輯，例如集成的圖形及/或科學(輸貫量)邏輯，或是作為特定用途核心）；及 4) 系統晶片，在相同晶粒上包含所述 CPU(有時稱為應用核心或應用處理器)、上述副處理器、及其它功能。於下說明舉例說明的核心架構，接著是舉例說明的處理器及電腦架構的說明。

圖 4A 是方塊圖，顯示根據本發明的實施例之舉例說明的有序管線及舉例說明的暫存器重命名、亂序議題/執行管線。圖 4B 是方塊圖，顯示根據本發明的實施例之要包含於處理器中之舉例說明的暫存器重命名、亂序議題/執行架構核心以及有序架構核心之舉例說明的實施例。圖 4A-B 中的實線方塊顯示有序管線及有序核心，而選加的虛線方塊顯示暫存器重命名、亂序議題/執行管線及核心。在有序態樣是亂序態樣的子集合之假設下，說明亂序

態樣。

在圖 4A 中，處理器管線 400 包含提取級 402、長度解碼級 404、解碼級 406、分配級 408、重命名級 410、排程（也稱為派遣或核發）級 412、暫存器讀取/記憶體讀取級 414、執行級 416、寫回/記憶體寫入級 418、例外處理級 422、及提交級 424。

圖 4B 顯示處理器核心 490，處理器核心 490 包含耦合至執行引擎單元 450 之前端單元 430，引擎單元 450 及前端單元 430 都耦合至記憶體單元 470。核心 490 可為精簡指令集計算（RISC）核心、複雜指令集計算（CISC）核心、很長指令字（VLIW）核心、或是混合或替代核心型式。關於又另一選項，核心 490 可為特別用途的核心，舉例而言，例如網路或通訊核心、壓縮引擎、副處理器核心、一般用途計算圖形處理單元(GPGPU)核心、圖形核心、等等。

前端單元 430 包含分支預測單元 432，分支預測單元 432 耦合至指令快取單元 434，指令快取單元 434 耦合至指令轉譯旁看緩衝器（TLB）436，指令轉譯旁看緩衝器（TLB）436 耦合至指令提取單元 438，指令提取單元 438 耦合至解碼單元 440。解碼單元 440(或解碼器)將指令解碼，以及產生微碼進入點、微指令、其它指令、或是從原始指令解碼、或是從原始指令以其它方式反應、或是從原始指令導出的其它控制訊號，以作為輸出的一或更多微運算。使用各種不同的機構，以實施解碼單元 440。適當的

5

機構實例包含但不限於查詢表、硬體實施、可編程邏輯陣列(PLA)、微碼唯讀記憶體(ROM)、等等。在一實施例中，核心 490 包含儲存用於某些巨集指令的微碼（例如，在解碼單元 440 中或是在前端單元 430 之內）之微碼 ROM 或是其它媒體。解碼單元 440 耦合至執行引擎單元 450 中的重命名/分配器單元 452。

執行引擎單元 450 包含重命名/分配器單元 452，命名/分配器單元 452 耦合至退出單元 454 及一或更多排程器單元 456 的集合。排程器單元 456 代表任何數目的不同排程器，包含保留站、中央指令窗、等等。排程器單元 456 耦合至實體暫存器檔案單元 458。各實體暫存器檔案單元 458 代表一或更多實體暫存器檔案，不同的實體暫存器檔案儲存例如純量整數、純量浮點、緊縮整數、緊縮浮點、向量整數、向量浮點、等等一或更多不同的資料型式、狀態（例如，指令指標，指令指標是要被執行的下一指令的位址）、等等。在一實施例中，實體暫存器檔案單元 458 包括向量暫存器單元、寫入遮罩暫存器單元、及純量暫存器單元。這些暫存器單元提供架構向量暫存器、向量單暫存器、及一般用途暫存器。實體暫存器檔案單元 458 由退出單元 454 重疊，以顯示實施暫存器重命名及失序執行的各種方式（例如，使用重排序緩衝器及退出暫存器檔案；使用未來檔案、歷史緩衝器、及退出暫存器檔案；使用暫存器映射及暫存器池；等等）。退出單元 454 及實體暫存器檔案單元 458 耦合至執行叢集 460。執行叢集 460 包含

一或更多執行單元 462 的集合以及一或更多記憶體存取單元 464 的集合。執行單元 462 執行不同的運算（例如，偏移、加法、減法、乘法）以及對不同型式的資料（例如，純量浮點、緊縮整數、緊縮浮點、向量整數、向量浮點）執行運算。雖然某些實施例包含專用於特定功能或功能集的一些執行單元，但是，其它實施例可以僅包含一執行單元或是多個都執行所有功能的執行單元。由於某些實施例產生用於某些型式的資料/運算之分別的管線，所以，排程器單元 456、實體暫存器檔案 458、及執行叢集 460 顯示為可能為複數的（例如，均具有它們自己的排程器單元、實體暫存器檔案單元、及/或執行叢集的純量整數管線、純量浮點/緊縮整數/緊縮浮點/向量整數/向量浮點管線及/或記憶體存取管理，以及，在分別的記憶體存取管線的情形中，實施某些實施例，其中，僅有此管線的執行叢集具有記憶體存取單元 464）。也應瞭解，在使用分別的管線時，這些管線中之一或更多可以是亂序核發/執行，而其它是有序的。

記憶體存取單元 464 的集合耦合至記憶體單元 470，記憶體單元 470 包含耦合至資料快取單元 474 的資料 TLB 單元 472，資料快取單元 474 耦合至階層 2 (L2) 快取單元 476。在一舉例說明的實施例中，記憶體存取單元 464 包含載入單元、儲存位址單元、及儲存資料單元，各單元耦合至記憶體單元 470 中的資料 TLB 單元 472。指令快取單元 434 又耦合至記憶體單元 470 中的階層 2(L2)快取單

元 476。L2 快取單元 476 耦合至一或更多其它階層的快取記憶體以及最後耦合至主記憶體。

舉例而言，舉例說明的暫存器重命名、亂序核發/執行核心架構如下所述地實施管線 400：1)指令提取 438 執行提取及長度解碼級 402 和 404；2)解碼單元 440 執行解碼級 406；3)重命名/分配器單元 452 執行分配級 408 及重命名級 410；4)排程器單元 456 執行排程級 412；5)實體暫存器檔案單元 458 及記憶體單元 470 執行暫存器讀取/記憶體讀取級 414；執行叢集 460 執行執行級 416；6)記憶體單元 470 及實體暫存器檔案單元 458 執行寫回/記憶體寫入級 418；7)各種單元涉及例外處理級 422；以及，8)退出單元 454 及實體暫存器檔案單元 458 執行提交級 424。

核心 490 支援一或更多指令集（例如，x86 指令集（增加較新版本的某些擴充）；加州太陽谷的 MIPS Technologies 的 MIPS 指令集；加州太陽谷的 ARM Holdings 的 ARM 指令集（加上選加的例如 NEON 等增加擴充）），包含此處所述的指令。在一實施例中，核心 490 包含邏輯以支援緊縮資料指令集擴充（例如 AVX1、AVX2），藉以允許由很多多媒體應用所使用的運算使用緊縮資料而執行。

應瞭解，核心支援多緒（執行二或更多平行的運算或緒組），以及，以各種方式如此執行，這些方式包含時間切割多緒、同時多緒（其中，單一實體核心提供用於實體

核心同時正多緒化的多個緒中的各緒之邏輯核心)、或是其組合(例如,時間切割提取及解碼以及其後的同時多緒,例如 Intel® Hyperthreading 技術中所示)。

雖然在亂序執行的環境中說明暫存器重命名,但是,應瞭解,暫存器重命名可用於有序架構中。雖然所示的處理器的實施例也包含分別的指令及資料快取單元 434/474 以及共用的 L2 快取單元 476,但是,替代實施例具有用於指令及資料之單一內部快取,例如階層 1(L1)內部快取、或是多階層的內部快取。在某些實施例中,系統包含內部快取及外部快取的組合,外部快取是核心及/或處理器的外部。替代地,所有的快取可以是核心及/或處理器的外部。

圖 5A-B 顯示更特定之舉例說明的有序核心架構的方塊圖,其中,核心是晶片中數個邏輯區塊(包含相同型式及/或不同型式的其它核心)中之一。這些邏輯區塊視應用而經由設有某些固定功能邏輯之高頻寬互連網路(例如,環式網路)、記憶體 I/O 介面、及其它所需 I/O 邏輯而通訊。

圖 5A 是根據本發明的實施例之單一處理器核心、以及其對晶粒上互連網路 502 的連接及其本地子集合的階級 2(L2)快取記憶體 504 之方塊圖。在一實施例中,指令解碼器 500 支援設有緊縮資料指令集擴充的 x86 指令集。L1 快取記憶體 506 允許對快取記憶體低潛時存取至純量及向量單元。雖然在一實施例中(為了簡化設計),純量單元 5

508 和向量單元 510 使用分開的暫存器集（分別為純量暫存器 512 和向量暫存器 514）及在它們之間傳送的資料被寫至記憶體，然後從階層 1(L1) 快取記憶體 506 讀回，但是，本發明的替代實施例可以使用不同的方式（例如，使用單一暫存器集或是包含通訊路徑，允許資料在二暫存器檔案之間傳送而不用寫入及讀回）。

L2 快取記憶體 504 的本地子集合是被分割成多個分別的本地子集合之通用 L2 快取的部份，每一處理器核心有一分別的本地子集合。各處理器核心對於它自己的 L2 快取記憶體 504 的本地子集合具有直接存取路徑。與存取它們自己的本地 L2 快取記憶體子集合之其它處理器核心相平行地，由處理器核心讀取的資料被儲存在 L2 快取子集合 504 中且可以被快速地存取。由處理器核心寫入的資料儲存在它自己的 L2 快取子集合 504 中，且假使需要時從其它子集合湧入。環式網路確保共用資料的同調性。環式網路是雙向的以允許例如處理器核心、L2 快取記憶體及其它邏輯區塊以在晶片之內彼此通訊。各環式資料路徑是每一方向 1012 位元寬。

圖 5B 是根據本發明的實施例的圖 5A 中的處理器核心的部份之放大視圖。圖 5B 包含 L1 快取記憶體 504 的 L1 資料快取記憶體 506A 部份、以及關於向量單元 510 和向量暫存器 514 的更多細節。具體而言，向量單元 510 是 16-寬的向量處理單元 (VPU)（請參見 16-寬的 ALU 528），其執行整數、單精準浮點、及雙倍精準浮點指令

中之一或更多。VPU 以調和單元 520 支援調和暫存器輸入、以數字轉換單元 522A-B 支援數字轉換、以及以記憶體輸入上的複製單元 524 支援複製。寫入遮罩暫存器 526 允許預測造成的向量寫入。

圖 6 是根據本發明的實施例之具有一個以上的核心、具有集成的記憶體控制器、及具有集成的圖形的處理器 600 的方塊圖。圖 6 中的粗線框顯示具有單核心 602A、系統代理器 610、一或更多匯流排控制器單元 616 的組之處理器 600，而選加的虛線框顯示設有多核心 602A-N、系統代理器單元 610 中一或更多整合的記憶體控制器元 614 的組、以及特定用途邏輯 608 的替代處理器 600。

因此，處理器 600 的不同實施包含：1)CPU，設有特定用途邏輯 608 及核心 602A-N，特定用途邏輯 608 是集成的圖形及/或科學(通量)邏輯(包含一或更多核心)，核心 602A-N 是一或更多一般用途核心(例如，一般用途有序核心、一般用途亂序核心、二者之組合)；2)副處理器，設有核心 602A-N，核心 602A-N 是大量主要用於圖形及/或科學(通量)的特定用途核心；以及，3)副處理器，設有核心 602A-N，核心 1202A-N 是大量一般用途的有序核心。因此，處理器 600 可為一般用途處理器、副處理器或特定用途處理，舉例而言，網路或通訊處理器、壓縮引擎、圖形處理器 GPGPU(一般用途圖形處理單元)，高通量多集成核心(MIC)副處理器(包含 30 或更多核心)、嵌入式處理器、等等。處理器可以實施在一或更多晶片 5

上。使用例如 BiCMOS、CMOS、或 NMOS 等多種製程技術，處理器 600 可以實施在一或更多基底上及/或其一部份。

記憶體階層架構包含在核心內的一或更多階層的快取記憶體、一組或一或更多共用快取單元 606、及耦合至集成的記憶體控制器單元 614 組之外部記憶體(未顯示)。共用快取單元 606 組可以包含例如階層 2(L2)、階層 3(L3)、階層 4(L4)、或其它階層快取記憶體等一或更多中級快取、最後階層快取記憶體(LLC)、及/或其組合。雖然在一實施例中，環式基礎互連單元 612 將整合圖形邏輯 608、共用快取單元 606 的組、及系統代理器單元 610/集成的記憶體控制器單元 614 互連，但是，替代實施例可以使用任何數目的已知技術來互連這些單元。在一實施例中，在一或更多快取單元 606 與核心 602-A-N 之間維持同調性。

在某些實施例中，核心 602A-N 中之一或更多核心能夠多緒化。系統代理器 610 包含協調及操作核心 602A-N 之那些組件。系統代理器單元 610 可以包含例如電力控制單元(PCU)及顯示單元。PCU 可以是或包含調節核心 602A-N 及集成的圖形邏輯 608 的電力狀態所需的邏輯及組件。顯示單元是用於驅動一或更多外部連接的顯示器。

以架構指令集的觀點而言，核心 602A-N 可以是同質的或異質的；亦即，核心 602A-N 中之二或更多能夠執行相同的指令集，而其它核心能夠僅執行該指令集的子集合

或不同的指令集。

圖 7-10 是舉例說明的電腦架構的方塊圖。用於膝上型電腦、桌上型電腦、手持 PC、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、數位訊號處理器(DSP)、圖形裝置、電動遊戲裝置、機上盒、微控制器、行動電話、可攜式媒體播放器、手持裝置、及各式各樣的其它電子裝置之此領域中熟知的其它系統設計及配置也是適合的。一般而言，能夠併有此處所揭示的處理器及／或其它執行邏輯的複數個種類的系統或電子裝置通常是適合的。

現在參考圖 7，其顯示根據本發明的一實施例之系統 700 的方塊圖。系統 700 可以包含耦合至控制器集線器 720 之一或更多處理器 710、715。在一實施例中，控制器集線器 720 包含圖形記憶體控制器集線器(GMCH)790 及輸入/輸出集線器(IOH)750(可以在分別的晶片上)；GMCH 790 包含記憶體及圖形控制器，記憶體 740 及副處理器 745 耦合至記憶體及圖形控制器；IOH 750 將輸入/輸出(I/O)裝置 760 耦合至 GMCH 790。替代地，記憶體及圖形控制器中之一或二者集成於處理器之內(如上所述)，記憶體 740 及副處理器 745 直接耦合至處理器 710、及設有 IOH 750 之單晶片中的控制器集線器 720。

在圖 7 中以虛線標示額外的處理器 715 的選加本質。每一處理器 710、715 包含此處所述的一或更多處理核心以及可以是某些版本的處理器 600。

舉例而言，記憶體 740 可以是動態隨機存取記憶體 (DRAM)、相位改變記憶體 (PCM)、或是二者的組合。對於至少一實施例，控制器集線器 720 經由多接點匯流排而與處理器 710、715 通訊，多接匯流排為例如前側匯流排 (FSB)、例如快速路徑互連 (QPI) 等點對點介面、或是類似連接 795。

在一實施例中，副處理器 745 是特定用途處理器，例如高通量 MIC 處理器、網路或通訊處理器、壓縮引擎、圖形處理器、GPGPU、嵌入式處理器、等等。在一實施例中，控制器集線器 720 包含集成的圖形加速器。

以包含架構、微架構、熱、耗電特徵、等等準則計量光譜的觀點而言，實體資源 710、715 之間有各種差異。

在一實施例中，處理器 710 執行控制一般型式的資料處理運算之指令。處理器指令可以嵌入在指令之內。處理器 710 將這些副處理器指令視為應由附加的副處理器 745 執行的型式。因此，處理器 710 在副處理器匯流排上或其它互連上核發這些副處理器指令（或是代表副處理器指令的控制訊號）給副處理器 745。副處理器 745 接受及執行收到的處理器指令。

現在參考圖 8，其顯示根據本發明的實施例之第一更特定舉例說明的系統 800 的方塊圖。如圖 8 中所示般，多處理器系統 800 是點對點互連系統，以及包含經由點對點互連 850 耦合的第一處理器 870 和第二處理器 880。各處理器 870 及 880 可以是某版本的處理器 600。在本發明的

一實施例中，處理器 870 和 880 分別是處理器 710 和 715，而副處理器 838 是副處理器 745。在另一實施例中，處理器 870 及 880 分別是處理器 710 和副處理器 745。

處理器 870 及 880 顯示為分別包含集成的記憶體控制器(IMC)單元 872 和 882。處理器 870 也包含點對點(P-P)介面 876 和 878 作為它的匯流排控制器單元的一部份；類似地，第二處理器 880 包含 P-P 介面 886 和 888。處理器 870、880 可以使用 P-P 介面電路 878、888 而經由點對點(P-P)介面 850 來交換資訊。如同圖 8 中所示般，IMC 872 和 882 將處理器耦合至各別記憶體，亦即記憶體 832 和記憶體 834，它們可以是本地附著至各別處理器的主記憶體的部份。

處理器 870、880 使用點對點介面電路 876、894、886、898，經由個別的 P-P 介面 852、854 而均可以與晶片組 890 交換資訊。晶片組 890 經由高性能介面 839，而與副處理器 838 選加地交換資訊。在一實施例中，副處理器 838 是特別用途處理器，例如高通量 MIC 處理器、網路或通訊處理器、壓縮引擎、圖形處理器、GPGPU、嵌入式處理器、等等。

共用的快取記憶體（未顯示）可以包含在任一處理器中或是二處理器的外部，未經由 P-P 互連與處理器連接，以致於假使處理器置於低電力模式中時，任一或二處理器的本地快取資訊可以儲存在共用的快取記憶體中。

晶片組 890 可以經由介面 896 而耦合至第一匯流排 816。在一實施例中，第一匯流排 816 可以是週邊元件互連(PCI)匯流排，或是例如 PCI 快速匯流排或其它第三代 I/O 互匯流排等匯流排，但是，本發明的範圍不侷限於此。

如圖 8 中所示般，各式各樣的 I/O 裝置 814 與匯流排橋接器 818 耦合至第一匯流排 816，匯流排橋接器 818 耦合第一匯流排 816 至第二匯流排 820。在一實施例中，例如副處理器、高通量 MIC 處理器、GPGPU、加速器（例如，圖形加速器或是數位訊號處理（DSP）單元）、現場可編程閘陣列、或是任何其它處理器等一或更多額外的處理器 815 耦合至第一匯流排 816。在一實施例中，第二匯流排 820 可以是低接腳數（LPC）匯流排。在一實施例中，各種裝置可以耦合至第二匯流排 820，舉例而言，包含鍵盤及／或滑鼠 822、通訊裝置 827 及例如包含指令/碼及資料 830 的碟片驅動器或其它大量儲存裝置的儲存單元 828。此外，音頻 I/O 824 可以耦合至第二匯流排 820。注意，其它架構是可能的。舉例而言，取代圖 8 的點對點架構，系統可以實施多接點匯流排或其它此類架構。

現在參考圖 9，其顯示根據本發明的實施例之第二更特定的舉例說明的系統 900 之方塊圖。圖 8 及 9 中類似的元件帶有類似的代號，以及，圖 8 的某些態樣在圖 9 中被省略，以免模糊圖 9 的其它態樣。

圖 9 顯示處理器 870、880 分別包含集成的記憶體及

I/O 控制邏輯 (CL) 872 和 882。因此，CL 872、882 包含集成的記憶體控制器單元及包含 I/O 控制邏輯。圖 9 顯示不僅記憶體 832、834 耦合至 CL 872、882，I/O 裝置 914 也耦合至控制邏輯 872、882。舊有 I/O 裝置 915 耦合至晶片組 890。

現在參考圖 10，其顯示根據本發明的實施例之系統晶片 (SoC) 1000 的方塊圖。圖 6 中類似的元件帶有類似的代號。而且，虛線框關於更多先進的 SoC 的選加特點。在圖 10 中，互連單元 1002 耦合至：應用處理器 1010，包含一或更多核心 202A-N 的集合及共用快取單元 606；系統代理器單元 610；匯流排控制器單元 616；整合記憶體控制器單元 614；一組或一或更多副處理器 1020，包含集成的圖形邏輯、影像處理器、音頻處理器、以及視頻處理器；靜態隨機存取記憶體 (SRAM) 單元 1030；直接記憶體存取 (DMA) 單元 1032；以及用於耦合至一或更多外部顯示器的顯示單元 1040。在一實施例中，副處理器 1020 包含特定用途處理器，舉例而言，例如網路或通訊處理器、壓縮引擎、GPGPU、高通量 MIC 處理器、嵌入式處理器、等等。

此處揭示的機制實施例可以以硬體、軟體、韌體或這些實施方式的結合來實施。本發明的實施例可以實施成在可編程系統上執行的電腦程式或程式碼，可編程系統包括至少一處理器、儲存系統（包含依電性及非依電性記憶體及／或儲存元件）、至少一輸入裝置、及至少一輸出裝

置。

例如圖 8 中所示的程式碼 830 可以應用至輸入指令以執行此處所述的功能和產生輸出資訊。輸出資訊可以以已知方式應用至一或更多輸出裝置。為了此應用，處理系統包含具有例如數位訊號處理器(DSP)、微控制器、特定應用積體電路(ASIC)、或微處理器等處理器之任何系統。

程式碼可以以高階程序或物件導向的程式語言實施，以與處理系統通訊。假使需要時，程式碼也可以以組合或機器語言來實施。事實上，此處所述的機制在範圍上不侷限於任何特定的程式語言。在任何情形中，語言可以是經過編譯或解譯的語言。

至少一實施例的一或更多態樣可以由儲存在機器可讀取的媒體上之代表指令實施，代表指令代表處理器內各種邏輯，代表指令由機器讀取時促使機器製造邏輯以執行此處所述的技術。稱為「IP 核心」的這些代表可以儲存在實體的、機器可讀取的媒體並供應至各式各樣的客戶或製造設施，而載入至真正地產生邏輯的製造機器或處理器。

此機器可讀取的媒體包含但不限於由機器或裝置製造或形成的物件之非暫時、實體配置，包括例如硬碟等儲存媒體、包含軟碟、光碟、光碟唯讀記憶體(CD-ROM)、光碟可重寫記憶體(CD-RW)、及磁光碟等任何其它型式的碟片、例如唯讀記憶體(ROM)、例如動態隨機存取記憶體、靜態隨機存取記憶體等隨機存取記憶體(RAM)、可抹拭可編程唯讀記憶體(EPPROM)、快閃記憶體、電可抹拭可

編程唯讀記憶體 (EEPROM) 等半導體裝置、相位改變記憶體 (PCM)、磁性或光學卡、或適用於儲存電子指令的任何其它型式的媒體。

因此，本發明的實施例也包含含有指令或含有設計資料之非暫時、實體的機器可讀取的媒體，所述設計資料是例如硬體說明語言 (HDL)，其界定此處所述的結構、電路、裝置、處理器及／或系統特點。這些實施例也意指程式產品。

在某些情形中，指令轉換器可以用以將指令從源指令集轉換至標的指令集。舉例而言，指令轉換器可以將指令轉譯（例如，使用靜態二進位轉譯、包含動態編譯的動態二進位轉譯）、變體、模仿、或以其它方式轉換至由核心處理的一或更多其它指令。指令轉換器可以以軟體、硬體、韌體、或其結合來實施。指令轉換器可以是在處理器上、不在處理器上、或部份在處理器上及部份離開處理器。

圖 11 是方塊圖，將根據本發明的實施例之使用軟體指令轉換器以將源指令集中的二進位指令轉換成標的指令集中的二進位指令作對比。在顯示的實施例中，指令轉換器是軟體指令轉換器，但是，指令轉換器可以替代地以軟體、韌體、硬體、或其各種結合來實施。圖 11 顯示高階語言 1102 的程式，可以由 x86 編譯器 1104 編譯以產生 x86 二進位碼 1106，二進位碼 1106 可以由具有至少一 x86 指令集核心 1116 的處理器原地執行。具有至少一 x86 5

指令集核心 1116 的處理器代表藉由共容地執行或其它方式地處理下述，而能夠執行與具有至少一 x86 指令集核心的英特爾處理器實質上相同功能的任何處理器：(1)英特爾 x86 指令集核心的指令集的實質部份，或是(2)以在具有至少一 x86 指令集核心的英特爾處理器上執行為目標之應用程式或其它軟體的物件碼版本，以便取得與具有至少一 x86 指令集核心的英特爾處理器實質上相同的結果。

x86 編譯器 1104 代表可以操作以產生 x86 二進位碼 1106 (例如，物件碼)之編譯器，x86 二進位碼 1106 藉由或不藉由額外的鏈結處理，都可以在具有至少一 x86 指令集核心 1116 的處理器上執行。類似地，圖 11 顯示高階語言 1102 的程式，使用替代指令集編譯器 1108，可以將其編譯以產生替代指令集二進位碼 1110，替代指令集二進位碼 1110 可以由不具有至少一 x86 指令集核心 1114 的處理器 (例如，具有執行例如加州 Sunnyvale 的 MIPS Technologies 公司的 MIPS 指令集及/或加州 Sunnyvale 的 ARM Holdings 公司的 ARM 指令集之核心的處理器)原地執行。指令轉換器 1112 用以將 x86 二進位碼 1106 轉換成可以由不具有 x86 指令集核心 1114 的處理器原地執行的碼。由於能夠如此執行的指令轉換器難以製作，所以，此轉換的碼不易與替代指令集二進位碼 1110 相同；但是，所述轉換的碼將完成一般運算及由來自替代指令集的指令組成。因此，指令轉換器 1112 代表軟體、韌體、硬體或其組合，其經由模仿、模擬或任何其它處理而允許

不具有 x86 指令集處理器或核心的處理器或其它電子裝置執行 x86 二進位碼 1106。

用於執行向量位元反轉之方法和設備

需要向量位元反轉運算以有效率地實施多個位元操縱常式。本發明的一實施例包含向量位元反轉指令，所述向量位元反轉指令會交換儲存在源暫存器中的鄰接位或位元群組之位置。特別地，向量位元反轉指令的一實施例是由立即值控制，立即值係指明要被反轉/交換的位元群組的大小。舉例而言，在一實施例中，1 的立即值表示個別位元要被反轉，2 的立即值表示位元對要被反轉；4 的立即值表示 4 位元組（4 位元）要被反轉；8 的立即值表示位元組要被反轉；16 的立即值表示字要被反轉；32 的立即值表示雙倍字要被反轉。位元反轉指令的結果儲存在目的地暫存器中。此外，模式可以以立即值的 3 位元編碼，舉例而言，000 用於 1 位元；001 用於 2 位元；010 用於 4 位元；011 用於 8 位元；100 用於 16 位元；及 101 用於 32 位元。此外，除了使用立即值之外，資訊可以直接編碼在指令本身中，以及具有用於每一位元大小的數個指令版本（例如，`vrevcross1`、`vrevcross2`、`vrevcross4`、等等）。

如圖 12 所示，顯示舉例說明的處理器 1255，本發明的實施例可以於其上實施，處理器 1255 包含用於將向量位元反轉指令解碼之設有向量位元反轉解碼邏輯 1231 之

5

解碼器 1230、以及用於執行已解碼的位元反轉指令之設有位元反轉執行邏輯 1241 的執行邏輯 1240。

現在將說明舉例說明的處理器之其它細節。但是，應注意，本發明的基本原理不侷限於任何特定型式的處理器架構。

顯示的處理器架構包含一般用途暫存器(GRP) 1205 的組、向量暫存器 1206 的組、遮罩暫存器 1207 的組。在一實施例中，多個向量資料元件緊縮於各向量暫存器 1206 中，其具有 512 位元寬，用於儲存二個 256 位元值、四個 128 位元值、八個 64 位元值、十六個 32 位元值、等等。但是，本發明的基本原理不侷限於任何特定大小/型式的向量資料。在一實施例中，遮罩暫存器 1207 包含八個 64 位元運算元遮罩暫存器，用於對儲存在向量暫存器 1206(例如，實施成上述遮罩暫存器 k0-k7)中的值執行位元遮罩運算。但是，本發明的基本原理不侷限於任何特定遮罩大小/型式。

為了簡明起見，於圖 12 中顯示單一處理器核心（「核心 0」）的細節。但是，將瞭解，圖 12 中所示的各核心可以具有與核心 0 相同的邏輯集。舉例而言，各核心包含專用等級 1(L1)快取記憶體 1212 及等級 2(L2)快取記憶體 1211，用於根據指定的快取管理政策而快取指令及資料。L1 快取記憶體 1212 包含用於儲存指令之分別的指令快取記憶體 1220、以及用於儲存資料之分別的資料快取記憶體 1221。以可為固定大小（例如長度 64、128、

512 位元組) 之快取線的粒度，管理儲存於各式各樣的處理器快取記憶體內的指令及資料。本舉例說明的實施例之各核心具有：指令提取單元 1210，用於從主記憶體 1200 及/或共用等級 3(L3)快取記憶體 1216 提取指令；解碼單元 1220，用於將指令解碼（例如，將程式指令解碼成微運算或是「 μop 」）；執行單元 1240，用於執行指令；以及，寫回單元 1250，用於使指令退出及將結果寫回。

指令提取單元 1210 包含各式各樣習知的組件，包含：下一指令指標器 1203，用於儲存要從記憶體 1200(或是快取記憶體中之一)提取的下一指令之位址；指令轉譯旁看緩衝器 (ITLB) 1204，用於儲存最近使用的虛擬對實體指令位址對映，以增進位址轉譯速度；分支預測單元 1202，用於推測地預測指令分支位址；以及，分支標的緩衝器(BTB)1201，用於儲存分支位址及標的位址。一旦被提取時，指令接著被串流至包含解碼單元 1230、執行單元 1240、及寫回單元 1250 之指令管線的其餘級。這些單元中的各單元之結構及功能是具有此技藝中的一般技術者所熟知的，且於此將不會詳述，以免模糊本發明的不同實施例之有關態樣。

如上所述，在一實施例中，向量位元反轉指令會在立即值的控制之下，交換儲存在源暫存器中的位元群組或是鄰接位元的位置並將結果儲存在目的地暫存器中，立即值係指明要被反轉/交換的的位元群組的大小。在一實施例中，僅允許 2 位元群組大小的幕次以致於所有位元群組具

有要交換的對。

圖 13 顯示舉例說明的實施例，其包含用於儲存要交換的位元群組或是位元之第一源暫存器 SRC1 1301、以及用於儲存向量位元反轉指令的結果之目的地暫存器 DST 1302。在一實施例中，SRC1 及 DST 包括緊縮於 512 位元中向量暫存器（例如 ZMM0、XMM1、等等）之 64 位元的資料。但是，如先前所述，本發明的基本原理不侷限於任何特定大小/型式的運算元或暫存器。注意，為了簡明起見，圖 13 僅顯示儲存於 SRC1 1301 及 DST 1302 中的資料之一部份。

在操作時，根據立即值 1303 中提供的控制值，向量位元反轉邏輯 1300(如上所述，包含處理器管線中的解碼及執行級中的邏輯)將位元或位元群組從源暫存器 SRC1 1301 交換至目的地暫存器 DST 1302。在圖 13 中所示的特定實例中，立即值 1303 設定於 1，標示個別位元要被交換。結果，位元 1 及 0 的位置交換，位元 2 及 3、4 及 5、6 及 7、8 及 9、等等也如此交換，直到位元 62 及 63 被交換。在一實施例中，向量位元反轉邏輯 1300 包括一或更多多工器的集合，以在立即值的控制之下，將位元群組從源暫存器 SRC1 1301 反轉至目的地暫存器 DST 1302。

圖 14 顯示立即值 1303 具有標示成對位元要被交換之 2 的值之實施例。因此，如同所示，向量位元反轉邏輯 1300 將位元對 0-1 與位元對 2-3 交換；位元對 4-5 與位元

對 6-7 交換；位元對 8-9 與位元對 10-11 交換、等等，直到位元對 60-61 與位元對 62-63 交換。

圖 15 顯示立即值 1303 具有標示 4 位元組(4 位元的群組)要被交換之 4 的値之實施例。因此，如同所示，向量位元反轉邏輯 1300 將位元 0-3 與位元 4-7 交換；位元 8-11 與位元 12-15 交換；位元 16-19 與位元 20-23 交換、等等，直到位元 56-59 與位元 60-63 交換。

圖 16 顯示立即值 1303 具有標示位元組要被交換之 8 的値之實施例。因此，如同所示，向量位元反轉邏輯 1300 將位元組 0-7 與位元組 8-15 交換；位元組 16-23 與位元組 24-31 交換；等等，直到位元組 48-55 與位元組 56-63 交換。

圖 17 顯示立即值 1303 具有標示字要被交換之 16 的値之實施例。因此，如同所示，向量位元反轉邏輯 1300 將字 0-15 與字 16-31 交換；字 32-47 與字 48-63 交換。

最後，圖 18 顯示立即值 1303 具有標示雙倍字要被交換之 32 的値之實施例。因此，如同所示，向量位元反轉邏輯 1300 將雙倍字 0-31 與雙倍字 32-63 交換。

圖 19 顯示根據本發明的一實施例之方法。可以在上述架構的環境內執行所述方法，但不侷限於任何特定葯統架構。

在 1901，從系統記憶體提取或是從快取記憶體（例如 L1、L2、或 L3 快取記憶體）讀取向量位元反轉指令。

在 1902，為回應向量位元反轉指令的解碼/執行，要反轉/ 5

交換的輸入向量資料會儲存於源暫存器中。如同所述，在一實施例中，源暫存器是 512 位元向量暫存器以及要反轉/交換的向量資料包括緊縮於向量暫存器（例如 64 位元）內的一或更多資料元件。在 1903，從指令的立即值提供執行向量位元反轉所需的控制資料。在 1904，根據立即值而反轉位元群組或是個別位元。如上所述，在一實施例中，1 的立即值標示個別位元要被反轉；2 的立即值表示位元對要被反轉；4 的立即值表示 4 位元組（4 位元）要被反轉；8 的立即值表示位元組要被反轉；16 的立即值表示字要被反轉；32 的立即值表示雙倍字要被反轉。在 1905，包括反轉的位元或位元群組的資料儲存於目的地暫存器中，如上所述，目的地暫存器可為另一 512 位元的向量暫存器。反轉的位元或位元群組接著作為用於一或更多額外的指令之源。

舉例而言但非限定，可以執行位元反轉指令的序列以完全地反轉向量資料元件的所有位元。舉例而言，假使使用 32、16、8、4、2、及 1 的立即值、使用目的地結果作為用於各後續指令的源，對 64 位元值執行向量位元反轉指令的序列，然後，在最後的目的地暫存器中反轉 64 位元值的所有位元。圖 20 顯示此運算的一部份，其顯示在源暫存器 2001 依 7-0 排序的初始位元組。如同所示，以 4 的立即值，執行第一向量位元反轉指令以將目的地暫存器 2002 中的位元 3-0 與 7-4 反轉，然後作為用於第二向量位元反轉指令的源暫存器。第二位元反轉指令具有 2 的立即

值，將位元 3-2 與 1-0 反轉以及位元 7-6 與 5-4 反轉，以及將結果儲存在目的地暫存器 2003 中。最後，以 1 的立即值，使用來自暫存器 2003 的資料，執行第三位元反轉指令，將位元 1 與 0、位元 3 與 2、位元 5 與 4、及位元 7 與 6 反轉，以達成與初始集 (76543210) 完全反轉的位元集 (01234567)。

在一實施例中，對於 EVEV 編碼實施，第一源運算元、第二源運算元、及目的地運算元都是 ZMM 暫存器。在一實施例中，向量位元反轉指令採取下述形式，其中，DEST 是目的地以及 SRC1 包括含有要被反轉/交換的源：

```
VPREVSTEPQ DEST, SRC1, IMM8
```

此外，下述偽碼提供根據本發明的一實施例執行的運算之表示：

```
VPREVSTEPQ DEST, SRC1, IMMB (EVEX 編碼版)
(KL, VL) = (2,128),(4,256),(8,512)
FOR j:= 0 TO KL - 1
  IF k1[j] OR *no writemask* THEN
    IF EVEX.b AND SRC1 *Is memory* THEN
      x := SRC1.qword[0]
    ELSE
      x := SRC1.qword[j]
    h :=IMM8[5:0]
    IF h & 1 THEN
      b:= ((x & 0x5555555555555555UL) << 1)|
          ((x & 0xAAAAAAAAAAAAAAAAAUL) >> 1)
    ELSE IF h & 2 THEN
      b := ((x & 0x3333333333333333UL) << 2)|
          ((x & 0xCCCCCCCCCCCCCCCCUL) >> 2)
```

```

ELSE IF h & 4 THEN
    b := ((x & 0x0F0F0F0F0F0F0FUL) << 4)|
         ((X & 0xF0F0F0F0F0F0F0UL) >> 4)
ELSE IF h & 8 THEN
    b := ((x & 0x00FF00FF00FF00FFUL) << 8)|
         ((x & 0xFF00FF00FF00FF0UL) >> 8)
ELSE IF h & 16 THEN
    b := ((x & 0x0000FFFF0000FFFFUL) << 16)|
         ((x & 0xFFFF0000FFFF0000UL) >> 16)
ELSE IF h & 32 THEN
    b := ((x & 0x00000000FFFFFFFFFUL) << 32)|
         ((x & 0xFFFFFFFF00000000UL) >> 32)
ELSE // h Is 0
    b :=x
DEST.qword[j] := b
ELSE IF *zero masking* THEN
    DEST.qword[j] := 0
// else dest Is unchanged for merge-based masking
DEST[MAX_VL-1:VL] :=0;

```

因此，假定 $KL=8$ 及 $VL=512$ ，則使用 FOR 迴路（根據 j ）以選取位元群組要被辨識之不同的 64 位元四倍字中的各四倍字。具有「EVEX.b AND SRC3 *is memory*」之 IF 陳述表示假使「b」位元設定於 EVEX 位元欄中（典型上用於源廣播、捨入控制（與 L'L 相結合），或是抑制例外）以及假使正從系統記憶體讀取源資料時，則使用單一的 64 位元源值（亦即， $j=0$ ）。否則，根據 j 的目前值，選取要使用的資料元件。涉及 h 之 IF 及 ELSE 陳述指明要根據立即值（ $h := IMM8[5:0]$ ）反轉的資料元件的大小以

及設定識別要被反轉的特定位元欄之 b 的值。

用於執行向量位元反轉及交叉的方法及設備

本發明的一實施例執行如上所述的向量位元反轉，但是也提供結合反轉的位元或位元群組與來自第二源的資料之能力。特別地，在一實施例中，使用向量位元反轉及交叉指令，執行來自第一源的元件內的分組位元之反轉以及藉由選取替代樣式的立元群組而將它們與第二源結合。指令的運算由立即值控制：位元 5-0 控制位元群組的大小，而位元 6 賦能或禁能二源的結合。此外，結合的次序由位元 7 控制。

在操作時，位元群組可如上所述地配對，形成大小在 1 與 32 位元之間的位元區塊。僅有 2 位元群組大小的冪次會被允許，以致於所有的群組具有對。各群組會與其鄰居成對並與其互換（交換）。在替換之後且假使立即位元 6 被設定時，藉由從一源或以替代方式從另一源選取位元的群，將運算的結果與其它源運算元相結合。在一實施例中，要被選取的第一運算元由位元 7 控制（例如 0=首先從 SRC2 選取以及 1=首先從 SRC3 選取）。

如圖 21 所示，顯示舉例說明的處理器 1255，本發明的實施例可以於其上實施，處理器 1255 包含用於將向量位元反轉及交叉指令解碼之設有向量位元反轉及交叉解碼邏輯 2131 之解碼器 1230、以及用於執行已解碼的位元反轉及交叉指令之設有位元反轉及交叉執行邏輯 2141 的執

5

行邏輯 1240。在參考圖 12 的上述中提供說明舉例說明的處理器 1255 之其它細節。但是，應注意，本發明的基本原理不侷限於任何特定型式的處理器架構。

圖 22 顯示舉例說明的實施例，其包含第一源暫存器 SRC3 2201 及第二源暫存器 SRC2 2202 以用於儲存由向量位元反轉及交叉邏輯 2200 處理過的源位元或位元群組。也顯示目的地暫存器 DST 2204，其用於儲存向量位元反轉及交叉邏輯 2200 執行的向量位元反轉及交叉運算的結果。在一實施例中，SRC2 2202、SRC3 2201、及 DST 2204 包括緊縮於 512 位元向量暫存器（例如 ZMM0、ZMM1、等等）中的資料之 64 位元。但是，如先前所述般，本發明的基本原理不侷限於任何特定大小/型式的運算元或暫存器。

在操作時，在立即值 2203 的控制之下，向量位元反轉及交叉邏輯 2200（如上述，包含處理器管線的解碼及執行級中的邏輯）執行例如上述所述的反轉運算。在圖 22 中所示的特定實例中，立即值的位元 0-5 標示反轉運算是要對資料的 16 位元群組（亦即字）執行。因此，在所示的實例中在 SRC3 2201 中的元 A0-A3 中的各元及在 SRC2 2200 中的元 B0-B3 中的各元之長度為 16 位元。如同所示，為回應立即值，向量位元反轉及交叉邏輯 2200 對第一源暫存器 SRC3 2201 中的資料執行反轉（將 A3 與 A2 交換以及將 A1 與 A0 交換）。在一實施例中，向量位元反轉及交叉邏輯 2200 包括一或更多多工器的集合，而在

立即值的控制之下，將來自源暫存器 SRC1 1301 的位元群組反轉以及將資料元件交錯，而在目的地暫存器 DST 1302 中產生結果。

此外，在一實施例中，立即值的位元 6 標示向量位元反轉及交叉邏輯 2200 是否應將來自第一源暫存器 SRC3 2201 的經過反轉的位元群組與第二源暫存器 B0-B3 中的資料元件交錯。舉例而言，在位元 6 中 1 的值表示交錯應發生。如此，除了執行來自第一源暫存器 SRC3 2201 的位元群組之反轉之外，向量位元反轉及交叉邏輯 2200 選取來自不同源暫存器的其它資料元件。在所示的實例中，選取資料元件 A2（已被施加反轉運算）並將其置於目的地的最左方資料元件位置中，來自 SRC2 的 B2 置於下一資料元件位置，來自 SRC3 的 A0 接續於後，接著是來自 SRC2 的 B0。在一實施例中，立即值的位元 7 標示資料元件交錯的次序。舉例而言，在圖 22 中，位元 7 設定於 0，意指反轉的資料元件 A2 被置於最左方資料元件位置。

但是，如圖 23 所示，假使位元 7 被設定於 1，則將來自 SRC2 的資料元件 B3 置於 DST 的最左方位置，接著是來自 SRC3 的 A3，接著是來自 SRC2 的 B1、接著是來自 SRC3 的 A1。換言之，位元 7 標示來自 SRC3 的資料是否應佔據 DST 2204 中的偶數或奇數資料元件位置（以來自 SRC2 的資料佔據未由來自 SRC3 的資料佔據的其它位置）。總結，位元 6 標示是否應執行與第二源暫存器 SRC2 的交錯，以及，位元 7 標示執行交錯的次序。應注

意，向量位元反轉及交叉邏輯 2200 可以執行其運算以及將所有資料元件同時填充目的地暫存器 DST 2204(例如在單一循環中)。

圖 24 顯示立即值的位元 6 設定於 0 的實例。如此，關閉與來自第二源暫存器 SRC2 2202 的資料元件交錯。結果，在本實施例中，向量位元反轉及交叉邏輯 2200 對第一源暫存器 SRC3 2201 內的位元群組執行反轉運算以及將結果儲存在目的地暫存器 DST 2204 中(如參考圖 14-18 之上述所述般)。

此處所述的向量位元反轉及交叉技術可以用以有效率地執行各式各樣不同的功能。舉例而言，圖 25 顯示當執行一般轉置演繹法時如何應用這些技術。源暫存器 2501 包含資料元件 a0-a3，源暫存器 2502 包含資料元件 b0-b3，源暫存器 2503 包含資料元件 c0-c3，以及源暫存器 2504 包含資料元件 d0-d3。在一實施例中，來自暫存器 2501 的元被反轉及與來自暫存器 2502 的元交錯以填充暫存器 2512。因此，暫存器 2512 包含(從左至右)元 a2、b2、a0、b0。此外，來自暫存器 2502 的元被反轉及與來自暫存器 2501 的元交錯以填充暫存器 2511，造成元 a3、b3、a1、b1。在一實施例中，對源暫存器 2503 及 2504 執行相同運算以填充暫存器 2513 及 2514(亦即，將 2503 的元反轉及與 2504 交錯以填充 2514 及將 2504 的元反轉及與 2503 交錯以填充 2513)。

在下一級中，在暫存器 2511 中的元 a3-b3 與元 a1-b1

反轉（亦即，立即值增加以涵蓋第一級中使用的位元群組的二倍的位元群組）且將結果與暫存器 2513 中的資料元件交錯（再度地，使用較大的位元群組大小）。因此，如同所示，暫存器 2523 含有元序列：a1、b1、c1、d1（a1-b1 從暫存器 2511 反轉以及 c1-d1 從暫存器 2513 插入）。類似地，在暫存器 2513 中的元 c3-d3 與元 c1-d1 相反轉且結果與暫存器 2511 中的（未反轉）資料元件交錯。因此，暫存器 2521 含有元序列：a3、b3、c3、d3。根據相同原理但以不同的源暫存器 2512 及 2514，填充其餘的目的地暫存器 2522 及 2524。

如同所示，上述運算序列的結束結果是含於源暫存器 2501-2504 中的矩陣被轉置於目的地暫存器 2521-2524 中。

圖 26A 及 26B 顯示本發明的實施例如何用以執行傾斜原始運算之另一實例。特別地，在圖 26A 中，源暫存器 2601 含有資料元件 3、2、1、及 0，以及，源暫存器 2602 含有資料元件 3'、2'、1'、及 0'。在級 1 中，資料元件 3' 與 2' 交換及資料元件 1' 與 0' 交換，造成暫存器 2603 中所示的序列（亦即，2'、3'、0'、1'）。如同所示，在級 2 中，暫存器 2603 中的結果與暫存器 2601 中的資料交錯，造成序列 3,3',1,1'。

圖 26B 顯示類似的處理（與圖 26A 中的處理同時地執行）但是源暫存器反轉。特別地，源暫存器 2611 含有 3'、2'、1'、及 0' 及源暫存器 2612 含有 3、2、1、及 0 資 5

料元件。在級 1 中，資料元件 3 與 2 交換及資料元件 1 與 0 交換，造成暫存器 2603 中所示的序列（亦即，2、3、0、1）。如同所示，在級 2 中，暫存器 2603 中的結果與暫存器 2601 中的資料交錯，造成序列 2,2',0,0'。

雖然為了說明目的而提供上述實例，但是，本發明的基本原理不侷限於任何特定功能。可以在各式各樣的不同功能之環境內，實施上述向量位元反轉及交錯技術。

圖 27 顯示根據本發明的一實施例之方法。所述方法可以實施於上述架構的環境內，但不侷限於任何特定架構。

在 2701，從系統記憶體提取或是從快取記憶體（例如 L1、L2、或 L3 快取記憶體）讀取向量位元反轉及交叉指令。在 2702，為回應向量位元反轉指令的解碼/執行，將第一輸入向量資料儲存於第一源暫存器中以及將第二輸入向量資料儲存於第二源暫存器中。如同所述，在一實施例中，源暫存器是 512 位元向量暫存器以及要反轉/交換的向量資料包括緊縮於向量暫存器（例如 64 位元）內的一或更多資料元件。在 2703，從指令的立即值（或者，更具體而言，從例如上述的位元 0-5 等立即值的一部份）提供執行向量位元反轉所需的控制資料。在 2704，識別要反轉之來自第一源暫存器的位元群組或個別位元以及將位元群組反轉。

假使立即交錯位元被設定時，在 2705 判定時，則在 2706，來自第一源暫存器的反轉的位元群組與來自第二源

暫存器的位元群組交錯。如上所述，位元群組交錯的次序也取決於立即值（舉例而言，例如上述的立即值的位元 7 的值）。假使立即交錯位元未被設定時，則在 2707，將來自第一源的反轉的位元群組儲存在目的地暫存器中（例如忽略第二源暫存器中的資料）。

在一實施例中，對於 EVEX 編碼實施，第一源運算元是向量暫存器，及第二源運算元是向量暫存器或記憶體位置。目的地運算元是向量暫存器。如同所述，控制來自立即位元組：

```

VPREVCROSSQ DEST, SRC2, SRC3, IMM8 (EVEX 編碼版)
(KL, VL) = (2, 128), (4,256), (8,512)
For j ← 0 TO KL - 1
  i ← j * 64;
  IF kl[j] OR *no writemask*
  THEN
    IF EVEX.b AND SRC3 *is memory*
    THEN
      x ← SRC3[63:0];
    ELSE
      x ← SRC3[i+63:i];
    IF;
    k ← IMM8[5:0];
    k' ← 1 << (6 - lzcnt(k));
    IF k & 1
    THEN
      x ← ((x & 0x5555555555555555UL)<<1)|
          ((x & 0xAAAAAAAAAAAAAAAAUL)>>1);
    ELSIF k & 2
    THEN

```

```

        x ← ((x & 0x3333333333333333UL)<<2)|
            ((x & 0xCCCCCCCCCCCCCCCCUL)>> 2);
    ELSIF k & 4
    THEN
        x ← ((x & 0x0F0F0F0F0F0F0F0FUL)<<4)|
            ((x & 0xF0F0F0F0F0F0F0FUL)>>4);
    ELSIF k & 8
    THEN
        x ← ((x & 0x00FF00FF00FF00FFUL)<<8)|
            ((x & 0xFF00FF00FF00FF0UL)>>8);
    ELSEIF k & 16
    THEN
        x ← ((x & 0x0000FFFF0000FFFFUL)<<16)|
            ((x & 0xFFFF0000FFFF0000UL)>>16);
    ELSIF k & 32
    THEN
        x ← ((x & 0x00000000FFFFFFFFUL)<<32)|
            ((x & 0xFFFFFFFF00000000UL)>>32);

    FI;
    a ← (IMM8[7]) ? x : SRC2[i+63:i];
    b ← (IMM8[7]) ? SRC2[i+63:i] : x;

    IF IMM8[6]
    THEN
        IF k' & 1
        THEN
            DEST[i+63:i] ← (a & 0x5555555555555555UL)|
                (b & 0xAAAAAAAAAAAAAAAAAUL);
        ELSIF k' & 2
        THEN
            DEST[i+63:i] ← (a & 0x3333333333333333UL)|
                (b & 0xCCCCCCCCCCCCCCCCUL);

```

```

        ELSIF k' & 4
        THEN
            DEST[i+63:i] ← (a & 0x0F0F0F0F0F0F0F0FUL)|
                            (b & 0xF0F0F0F0F0F0F0FOUL)
        ELSIF k' & 8
        THEN
            DEST[i+63:i] ← (a & 0x00FF00FF00FF00FFUL)|
                            (b & 0xFF00FF00FF00FF00UL);
        ELSIF k' & 16
        THEN
            DEST[i+63:i] ← (a & 0x0000FFFF0000FFFFUL)|
                            (b & 0xFFFF0000FFFF0000UL);
        ELSIF k' & 32
        THEN
            DEST[i+63:i] ← (a & 0x00000000FFFFFFFFUL)|
                            (b & 0xFFFFFFFF00000000UL);
        FI;
    ELSE
        DEST[i+63:i] ← a;
    FI;
ELSE
    IF *merging-masking*
    THEN
        DEST[i+63:i] remains unchanged;
    ELSE
        DEST[i+63:i] ← 0;
    FI;
FI;
ENDFOR;
DEST[MAX_VL-1:VL] ← 0

```

如此，假定 $KL=8$ 及 $VL=512$ ，使用 FOR 迴路（根據

5

j)以選取位元群組要被辨識之不同的 64 位元四倍字中的各四倍字。涉及 k 的第一組 IF/ELSE 陳述從第一源暫存器旋轉及選取位元群組，以及，根據立即值（例如，假使 IMM8[6]設定於 1 時），涉及 k' 的第二組 IF/ELSE 陳述從第二源暫存器選取（亦即交錯）位元群組。

在前述說明書中，參考本發明的特定舉例說明的實施例，說明本發明的實施例。但是，顯然可知，在不悖離後附的申請專利範圍中揭示的本發明的廣義精神及範圍之下，可以作各種修改及變化。因此，說明書及圖式被視為是說明性的而非限制性的。

本發明的實施例包含上述說明之各式各樣的步驟。步驟可以以機器可執行的指令具體實施，這些指令可用以促使一般用途或特定用途的處理器執行這些步驟。替代地，這些步驟可以由含有用於執行步驟之實體接線的邏輯之特定硬體組件、或是程式化的電腦組件及客製化硬體組件的任何組合所執行。

如同此處所述般，指令意指例如特定應用積體電路（ASIC）等硬體的特定配置，ASIC 係配置成執行具有預定功能的某些運算或儲存在記憶體中的軟體指令，記憶體係以非暫時的電腦可讀取媒體具體實施。如此，使用儲存於及執行於一或更多電子裝置（例如終端站台、網路元件、等等）上的碼及資料，可以實施圖式所示的技術。這些電子裝置使用電腦可讀取的媒體以儲存及通訊（內部地及/或經由網路而與其它電子裝置通訊）碼及資料，電

腦可讀取的媒體可為例如非暫時的電腦機器－可讀取的儲存媒體（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）以及暫時的電腦可讀取通訊媒體（例如電的、光學的、聲的或其它形式的傳播訊號－例如載波、紅外線訊號、數位訊號、等等）。此外，這些電子裝置典型地包含一或更多耦合至一或更多其它組件的處理器，一或更多其它組件可為例如一或更多儲存裝置（非暫時的機器可讀取的儲存媒體）、使用者輸入/輸出裝置（例如鍵盤、觸控螢幕、及/或顯示器）、及網路連接。處理器組及其它組件的耦合典型上經由一或更多匯流排及橋接器（也稱為匯流排控制器）。載送網路交通之儲存裝置及訊號分別代表一或更多機器可讀取的儲存媒體以及機器可讀取的通訊媒體。因此，給定的電子裝置之儲存裝置典型地儲存用於在該電子裝置的一或更多處理器的組上執行之碼及/或資料。當然，可以使用軟體、韌體、及/或硬體的不同組合，實施本發明的實施例之一或更多部份。在此詳細說明中，為了說明之目的，揭示複數個特定細節以助於完整瞭解本發明。但是，習於此技藝者將清楚，不用這些特定細節中的某些細節，仍可實施本發明。在某些情形中，未特別詳細地說明熟知的結構及功能，以免模糊本發明的標的。因此，本發明的範圍及精神應以後附的申請專利範圍之觀點來判斷。

【符號說明】

- 300：暫存器架構
- 310：向量暫存器
- 315：寫入遮罩暫存器
- 325：一般用途暫存器
- 345：純量浮點堆疊暫存器檔案
- 350：MMX 緊縮整數平坦暫存器檔案
- 400：管線
- 490：核心
- 600：處理器
- 700：系統
- 800：系統
- 900：系統
- 1000：系統晶片
- 1200：主記憶體
- 1231：向量位元反轉解碼邏輯
- 1241：向量位元反轉執行邏輯
- 1255：處理器
- 1300：向量位元反轉邏輯
- 2131：向量位元反轉及交叉解碼邏輯
- 2141：向量位元反轉及交叉執行邏輯

發明摘要

※申請案號：104138336

※申請日：104年11月19日

※IPC分類：G06F 9/30 (2006.01)

G06F 9/38 (2006.01)

【發明名稱】(中文/英文)

用於執行向量位元反轉和交叉的方法和設備

Method and apparatus for performing a vector bit reversal and crossing

【中文】

揭示用於執行向量位元反轉指令及交叉的設備及方法。舉例而言，處理器的一實施例包括：第一源向量暫存器，用以儲存第一複數個源位元群組，其中，用於位元群組的大小將於指令的立即值中被指定；第二源向量，用以儲存第二複數個源位元群組；向量位元反轉及交叉邏輯，用以從立即值決定位元群組大小以及用以回應地反轉第一源向量暫存器內鄰接的位元群組的位置，以產生一組反轉的位元群組，其中，向量位元反轉及交叉邏輯又將該組反轉的位元群組與第二複數個位元群組交錯；以及，目的地向量暫存器，用以儲存與第一複數個位元群組交錯之反轉的位元群組。

【 英文 】

An apparatus and method for performing a vector bit reversal and crossing. For example, one embodiment of a processor comprises: a first source vector register to store a first plurality of source bit groups, wherein a size for the bit groups is to be specified in an immediate of an instruction; a second source vector to store a second plurality of source bit groups; vector bit reversal and crossing logic to determine a bit group size from the immediate and to responsively reverse positions of contiguous bit groups within the first source vector register to generate a set of reversed bit groups, wherein the vector bit reversal and crossing logic is to additionally interleave the set of reversed bit groups with the second plurality of bit groups; and a destination vector register to store the reversed bit groups interleaved with the first plurality of bit groups.

圖 1A

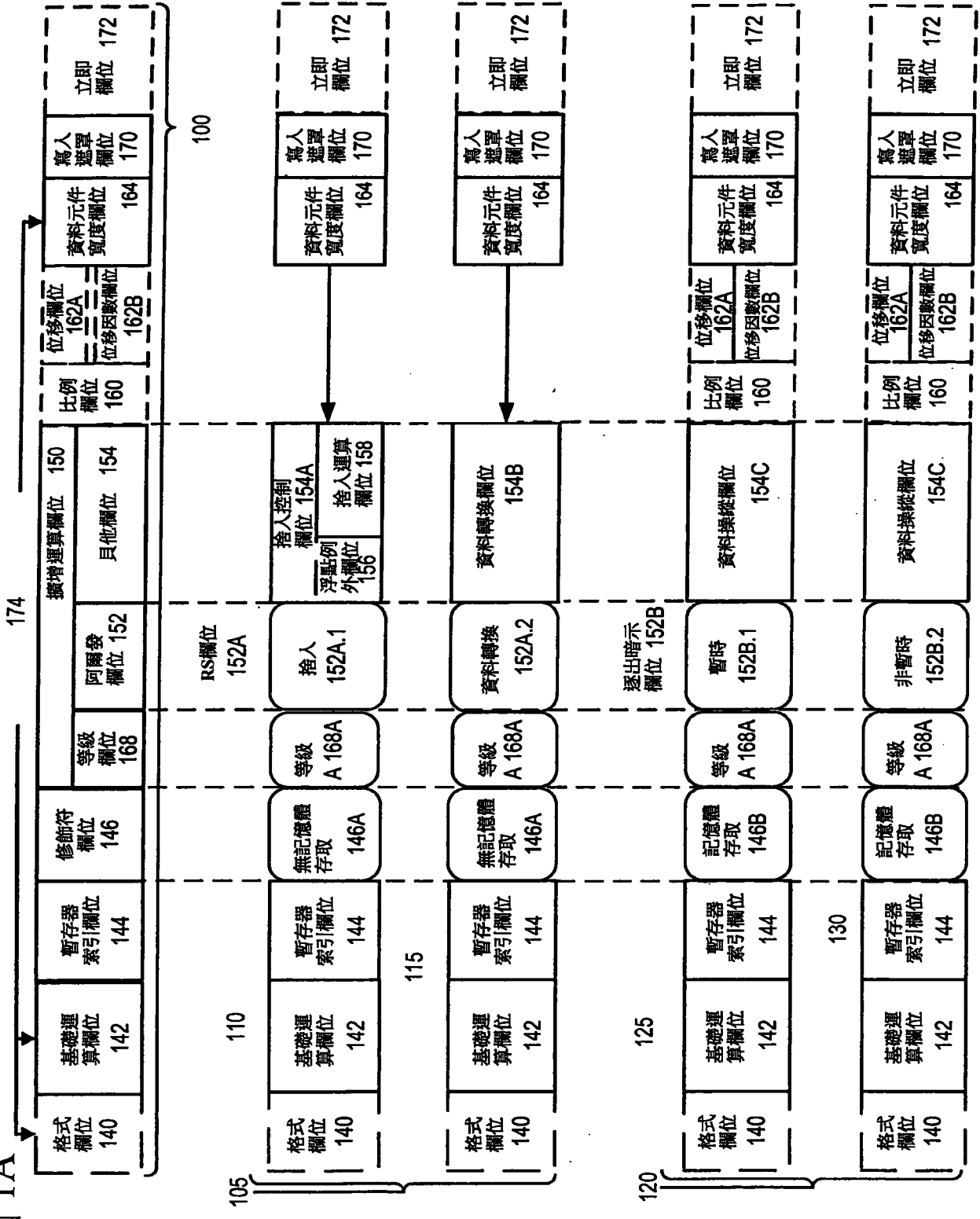


圖 1A

圖 1B

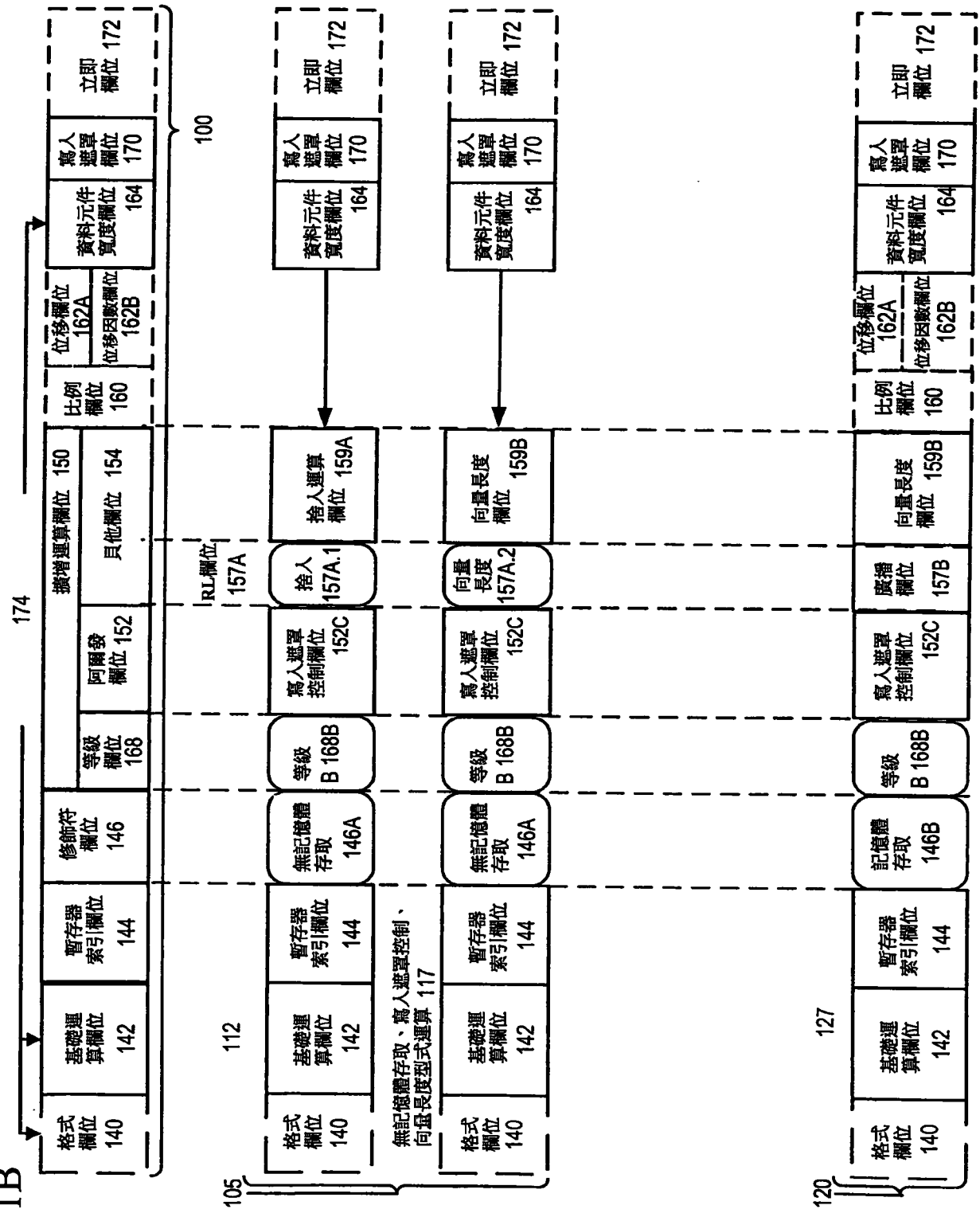


圖 2A

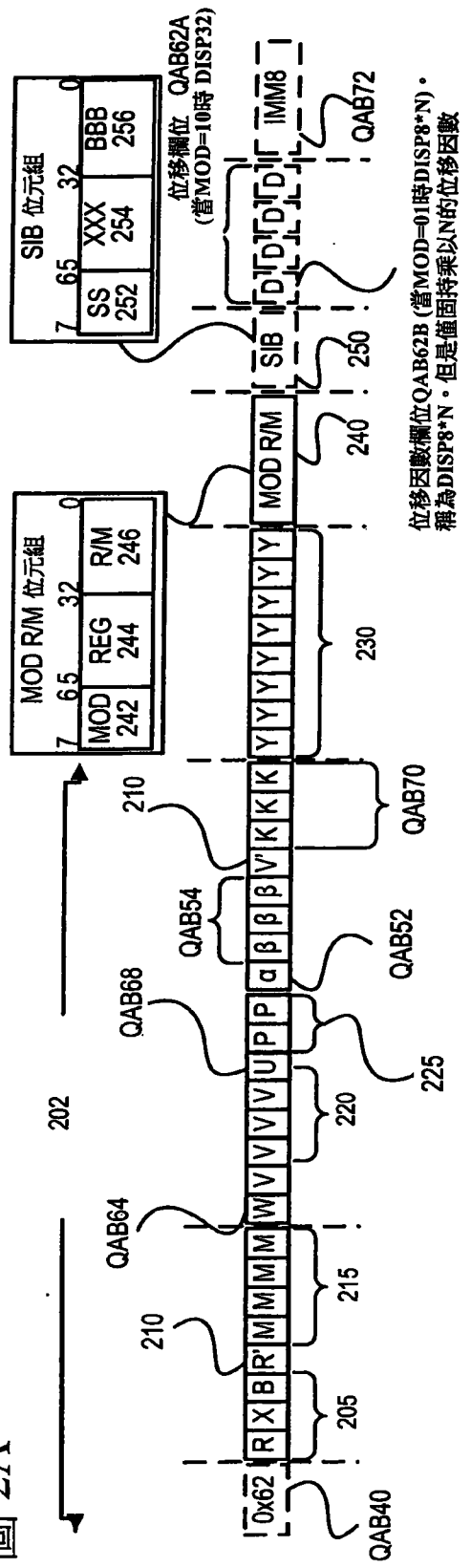


圖 2B

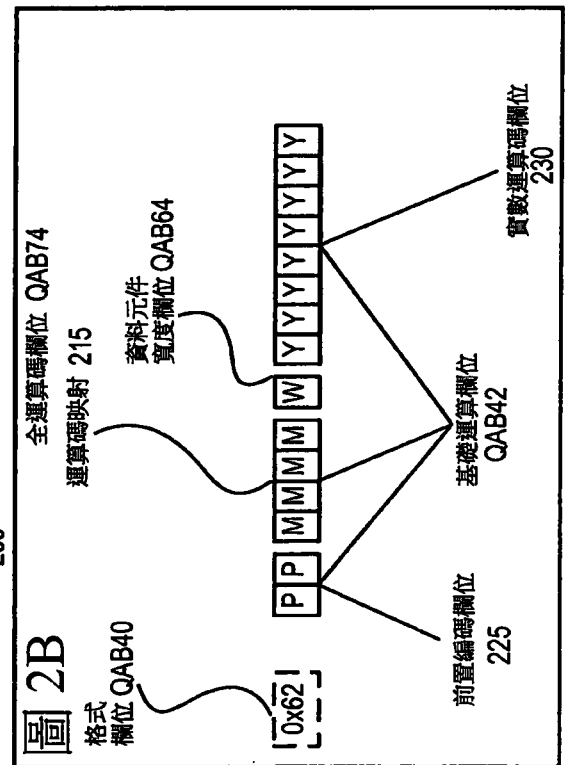


圖 2C

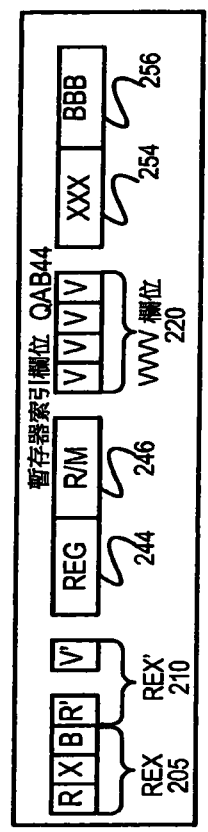
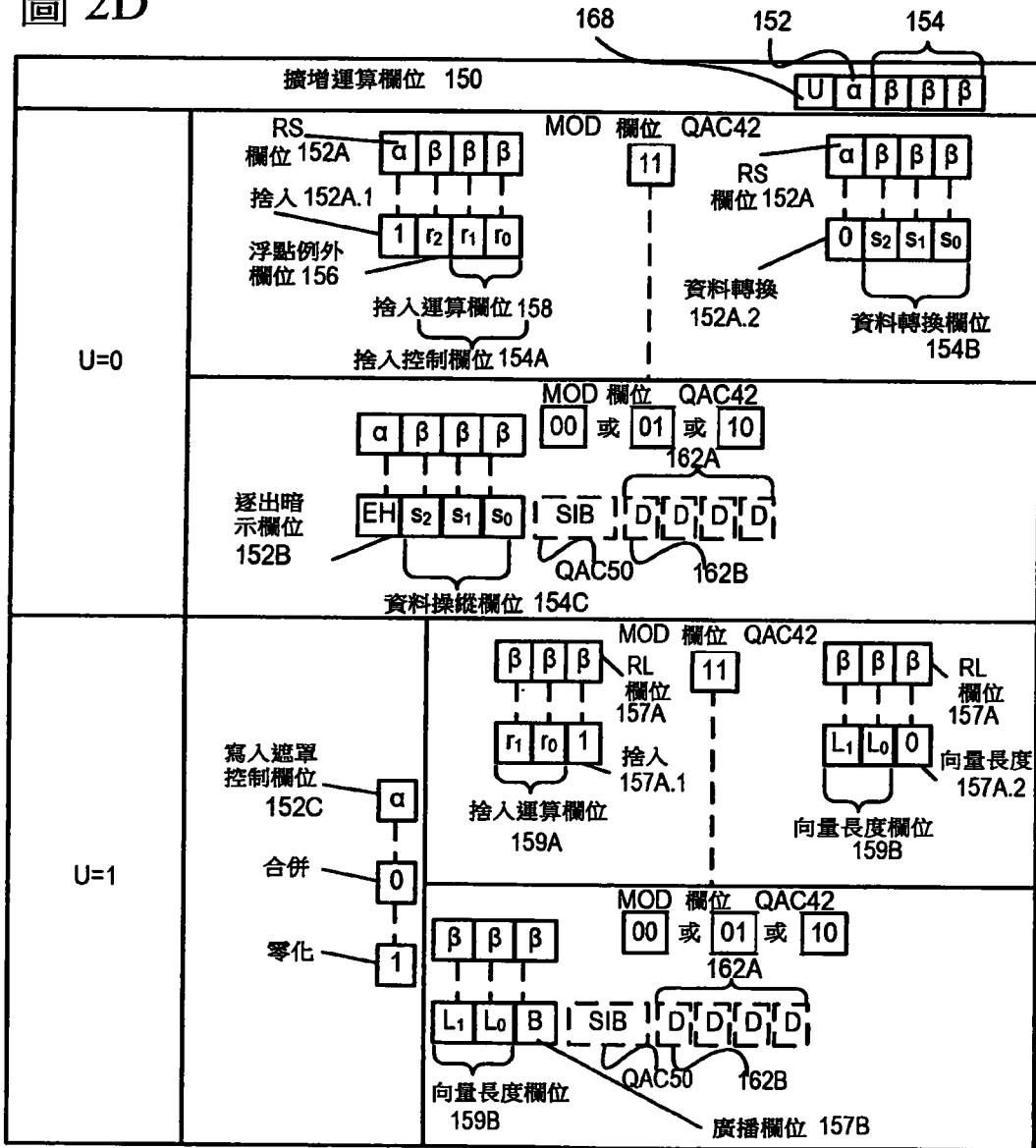


圖 2D



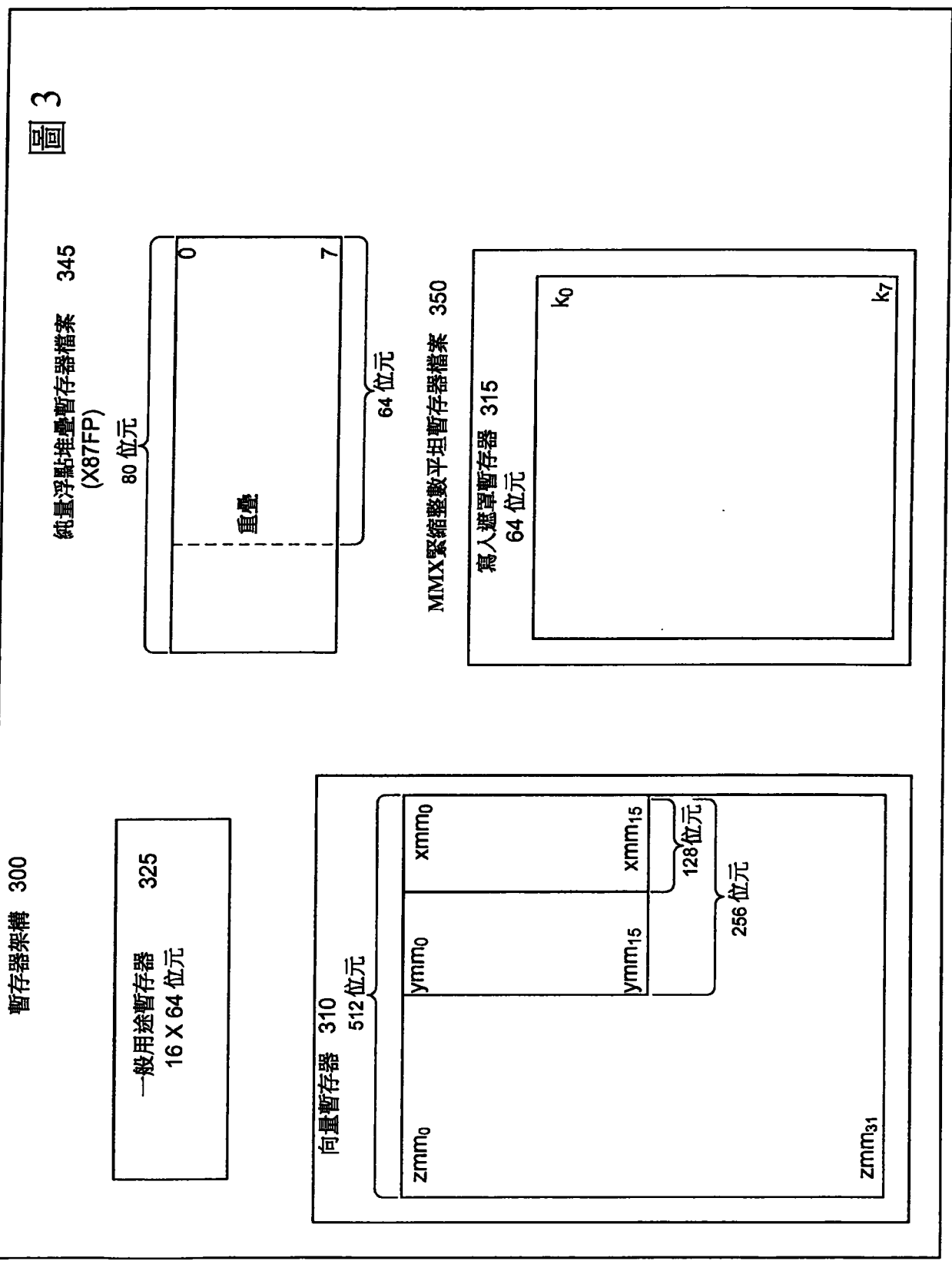
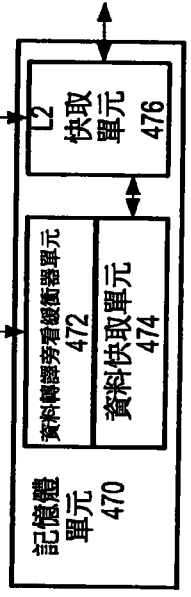
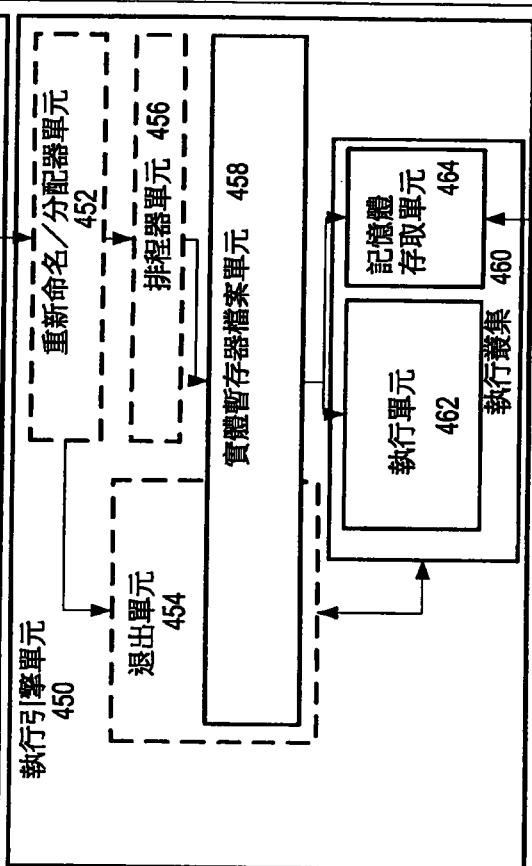
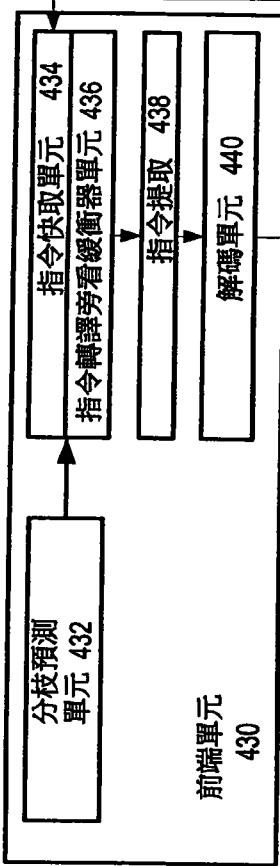
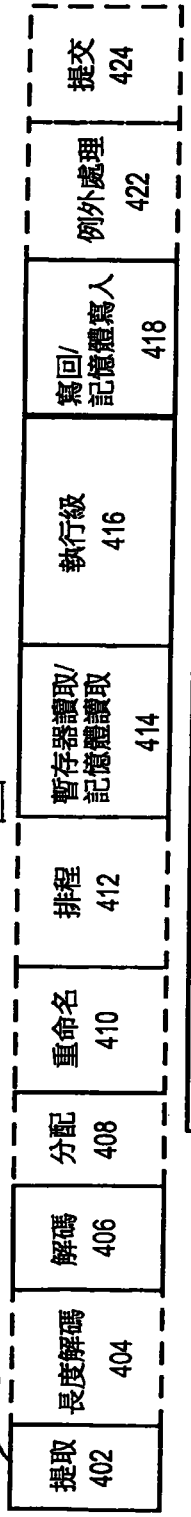


圖 3

400

圖 4A



490

圖 4B

圖 5A

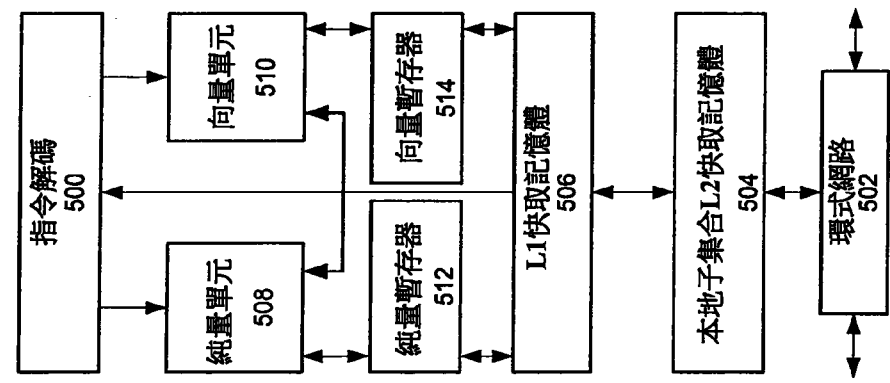
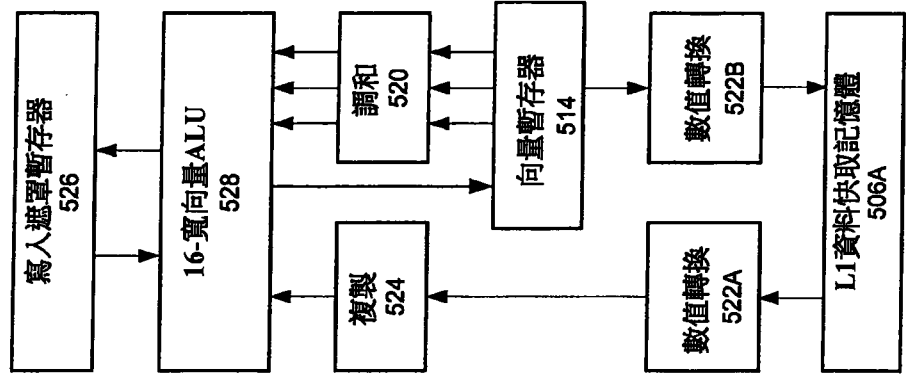


圖 5B



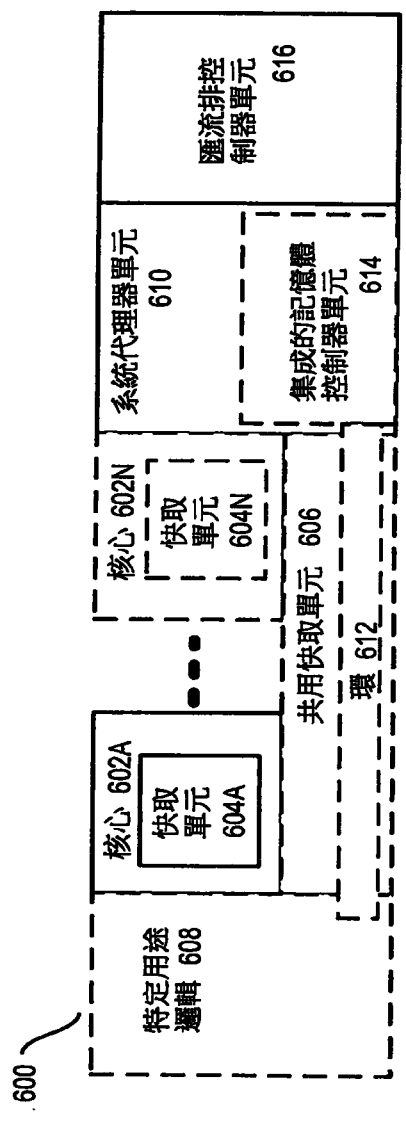


圖 6

圖 7

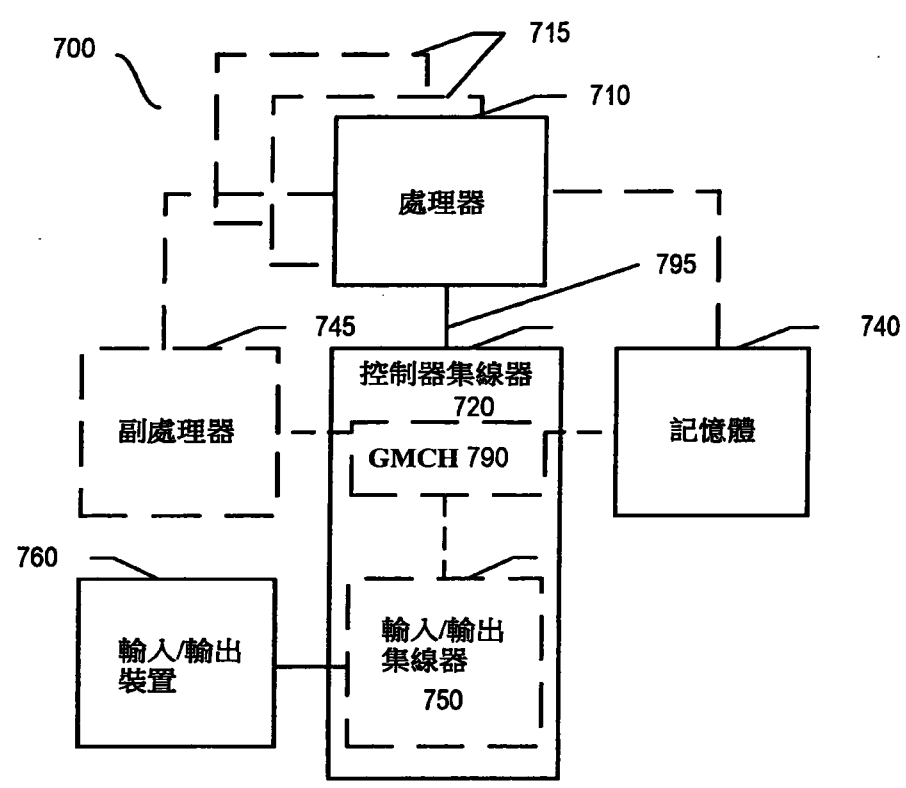


圖 8

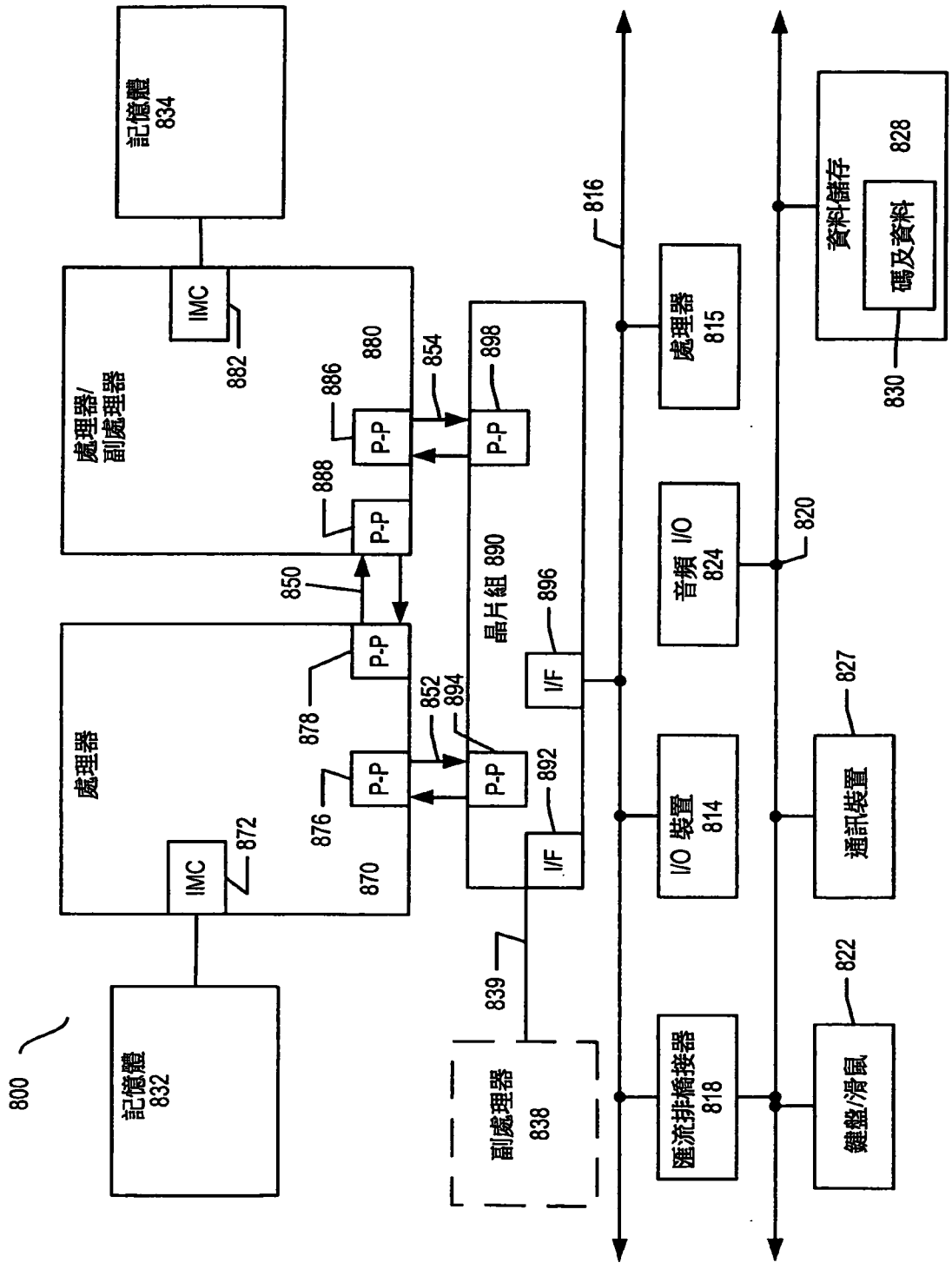
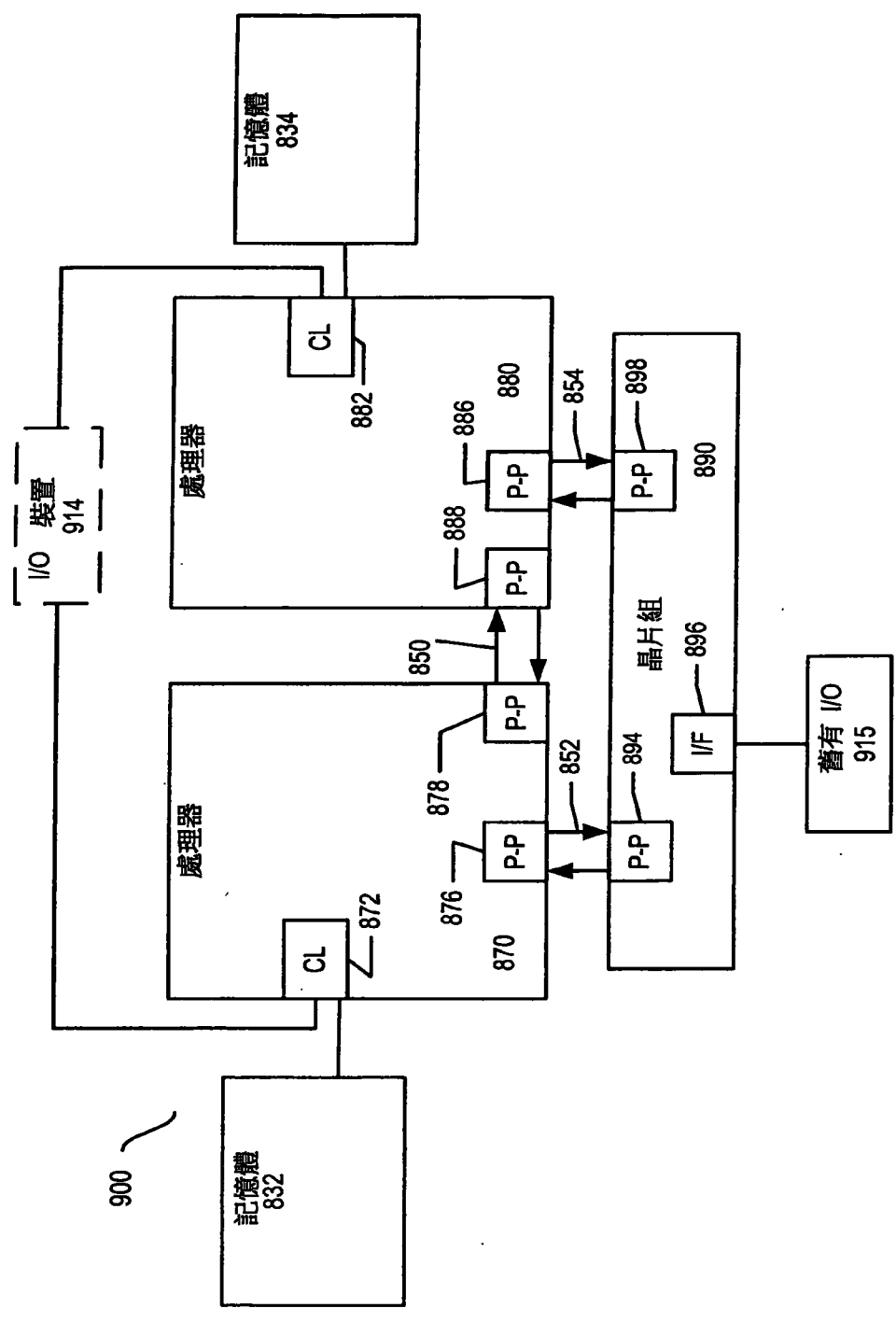
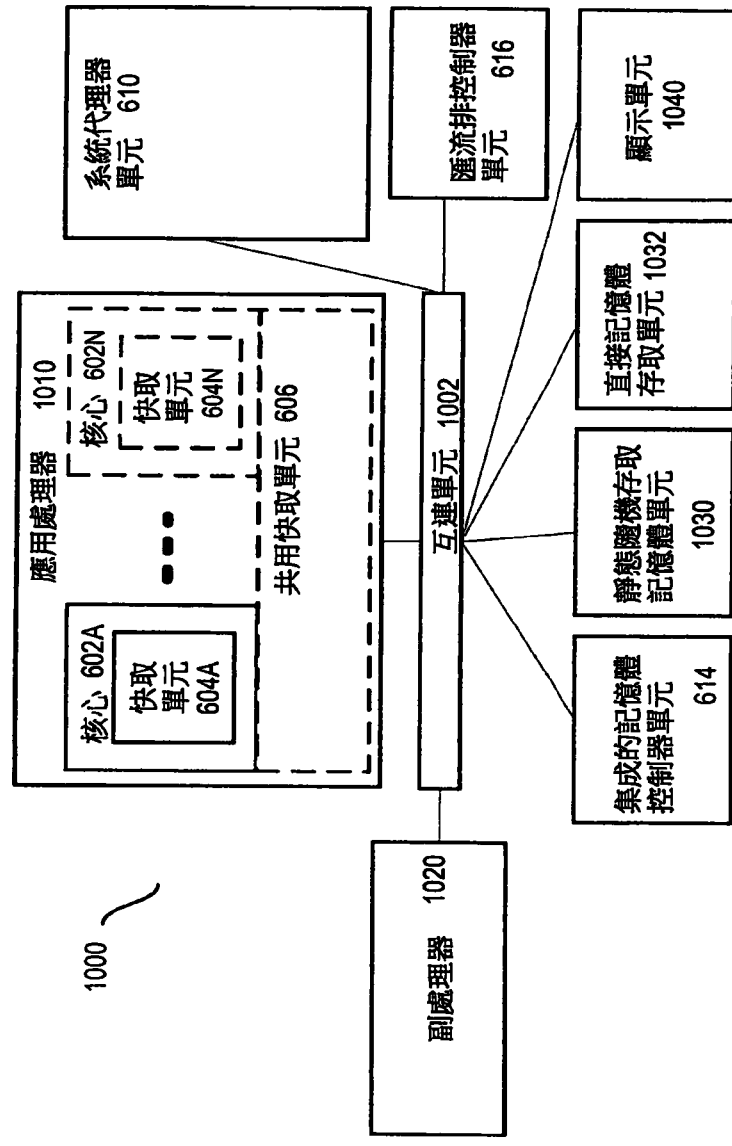


圖 9





1000

圖 10

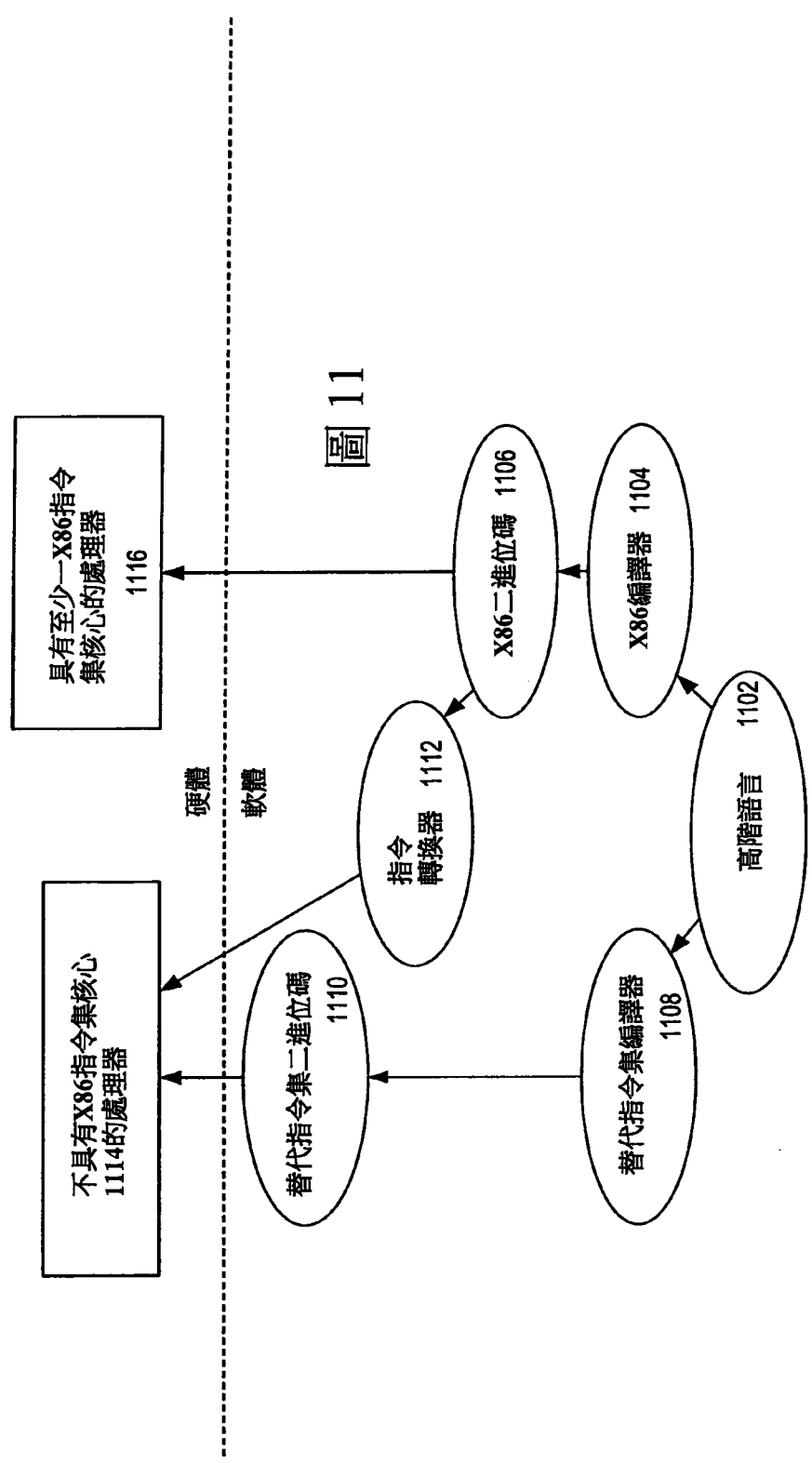


圖 11

圖 12

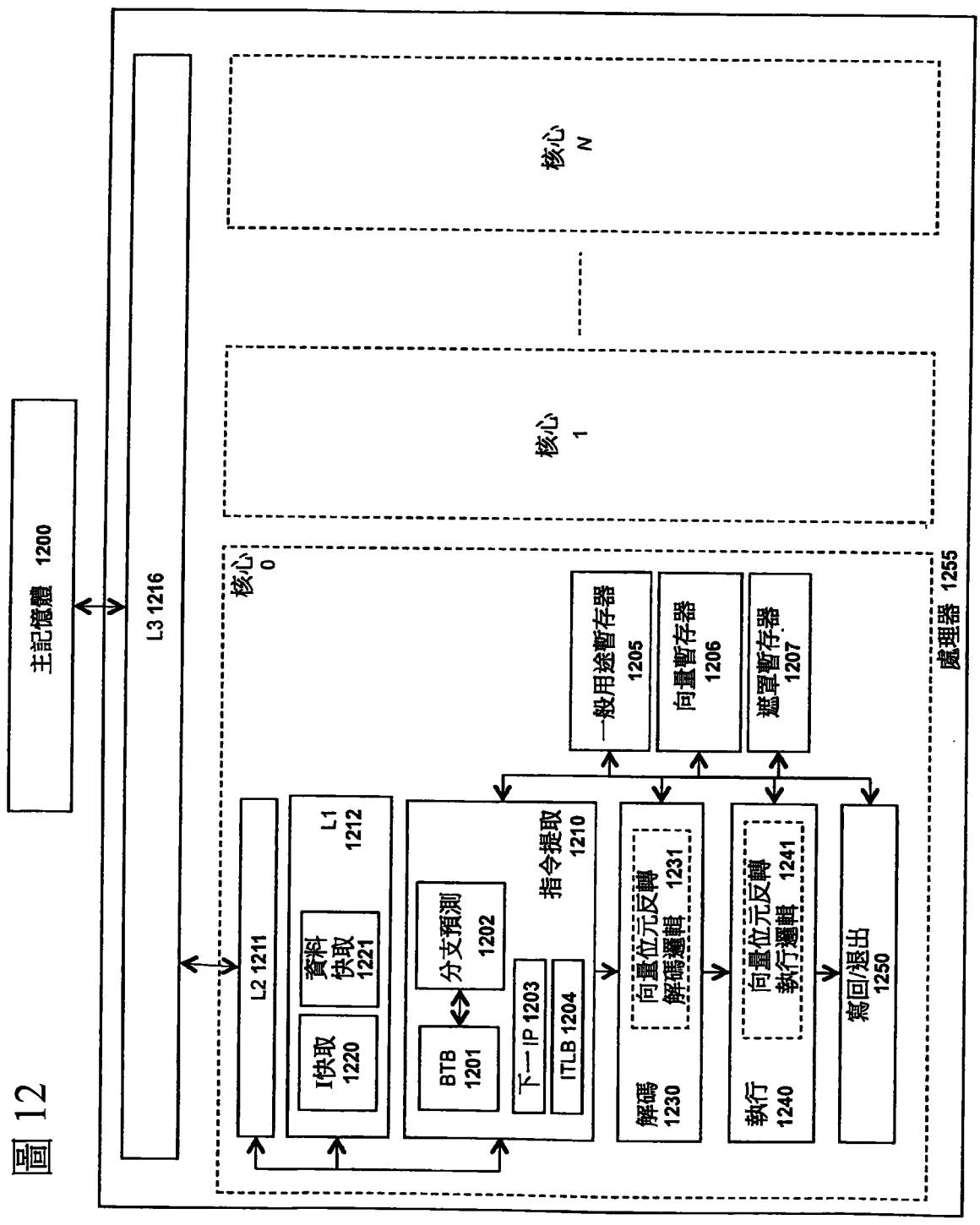


圖 13

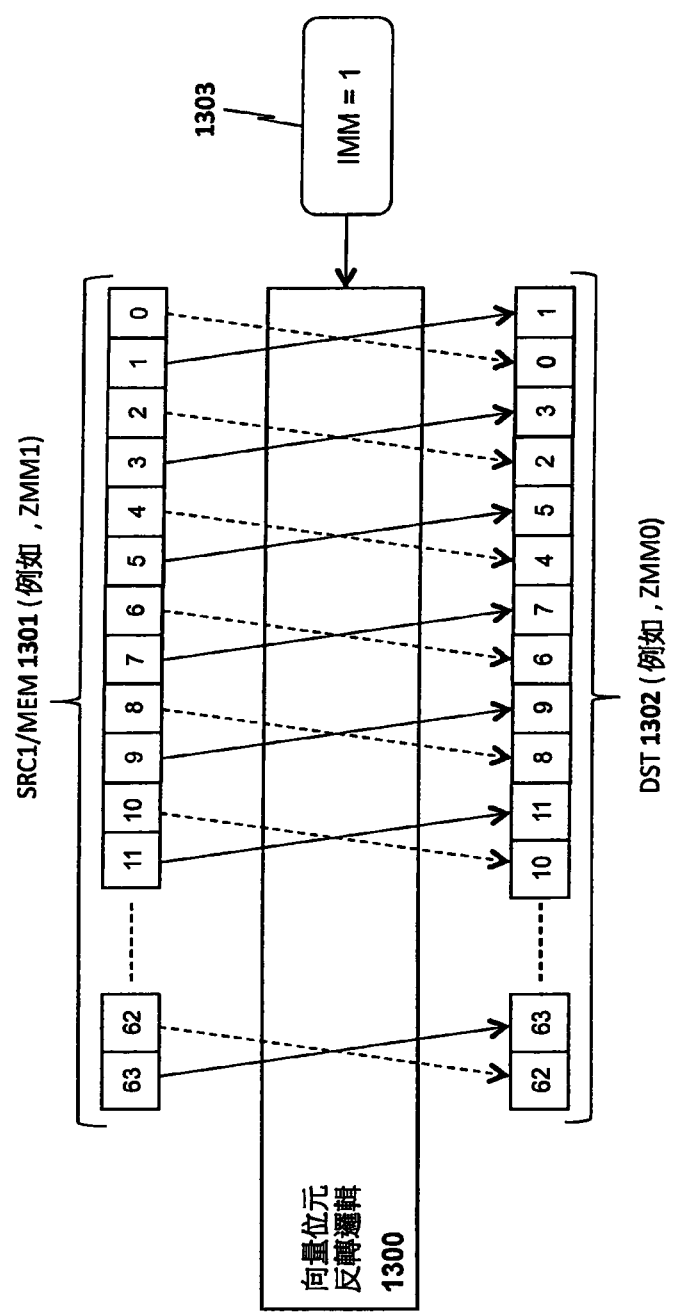


圖 14

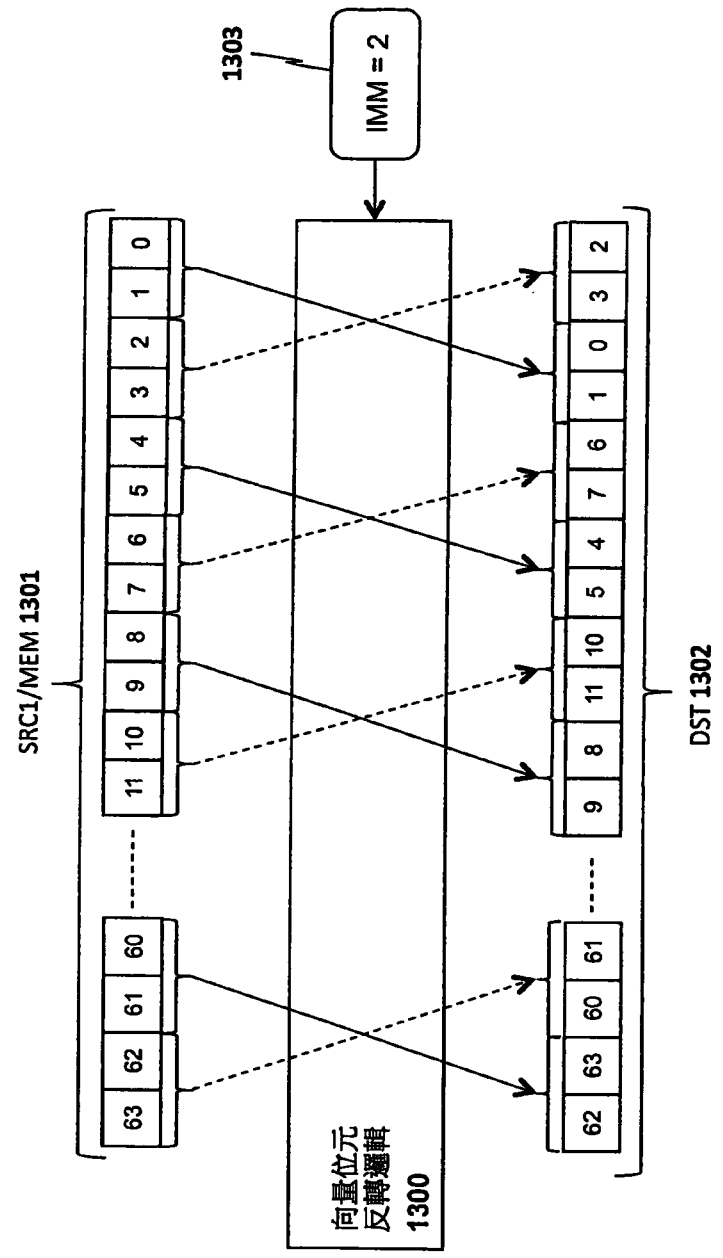


圖 15

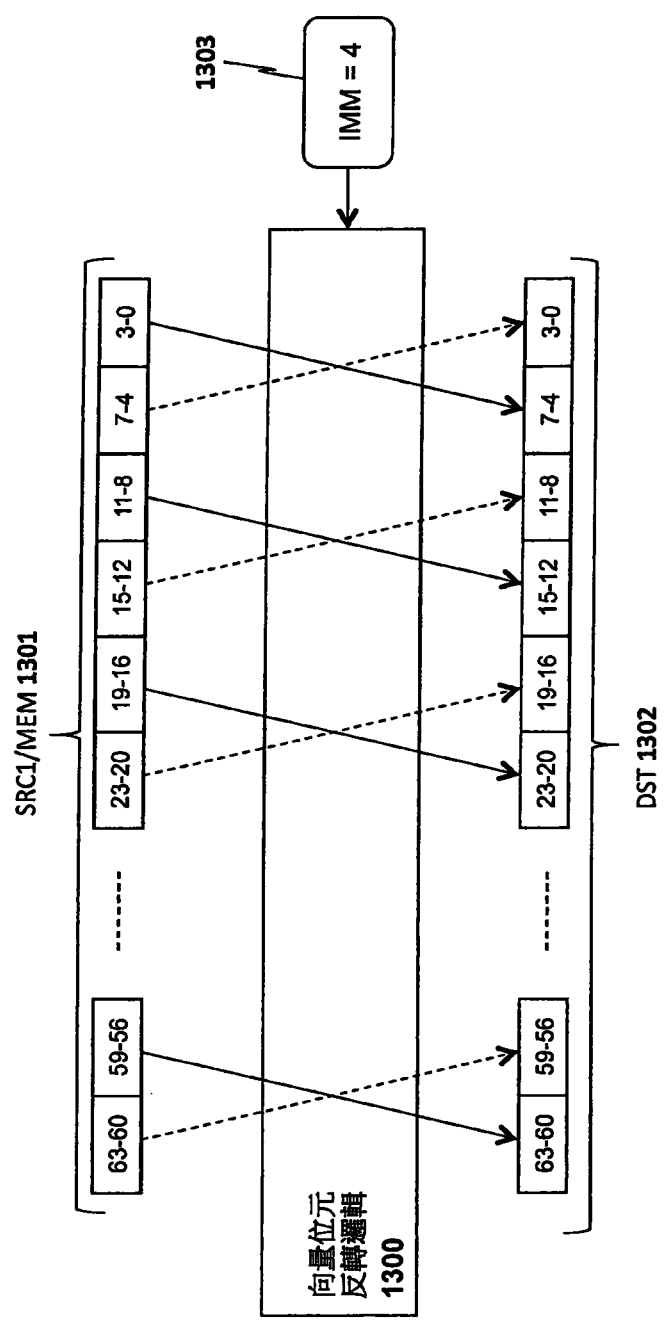


圖 16

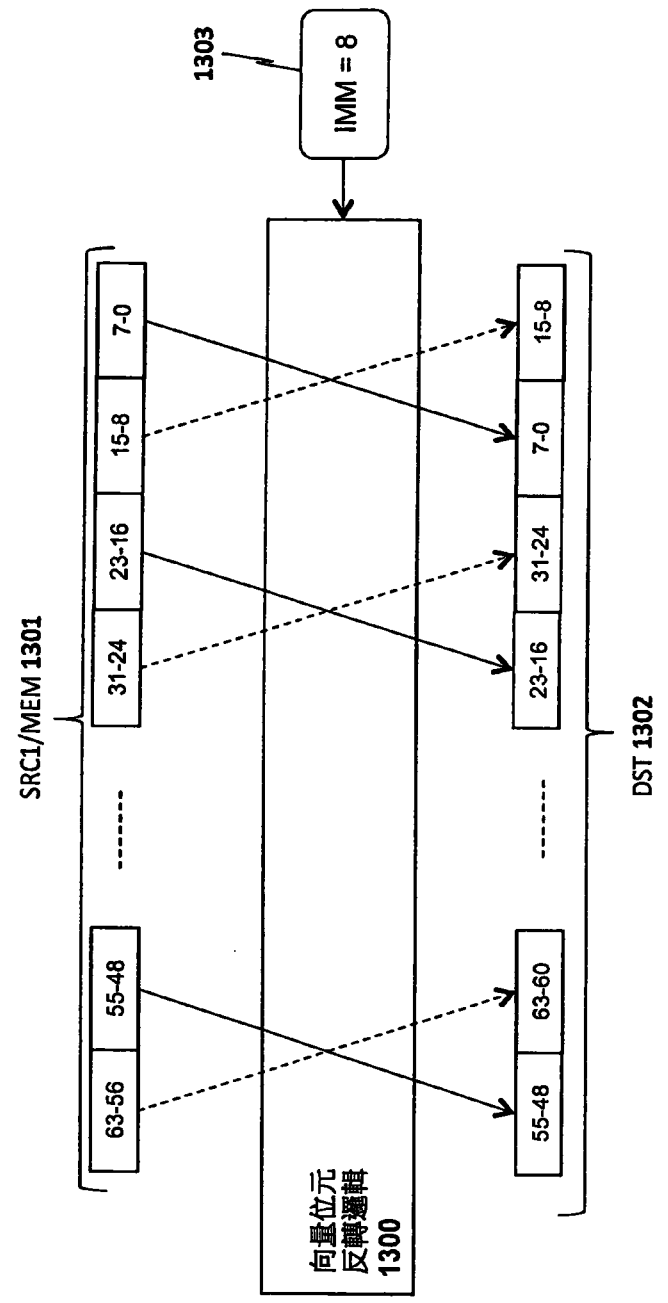


圖 17

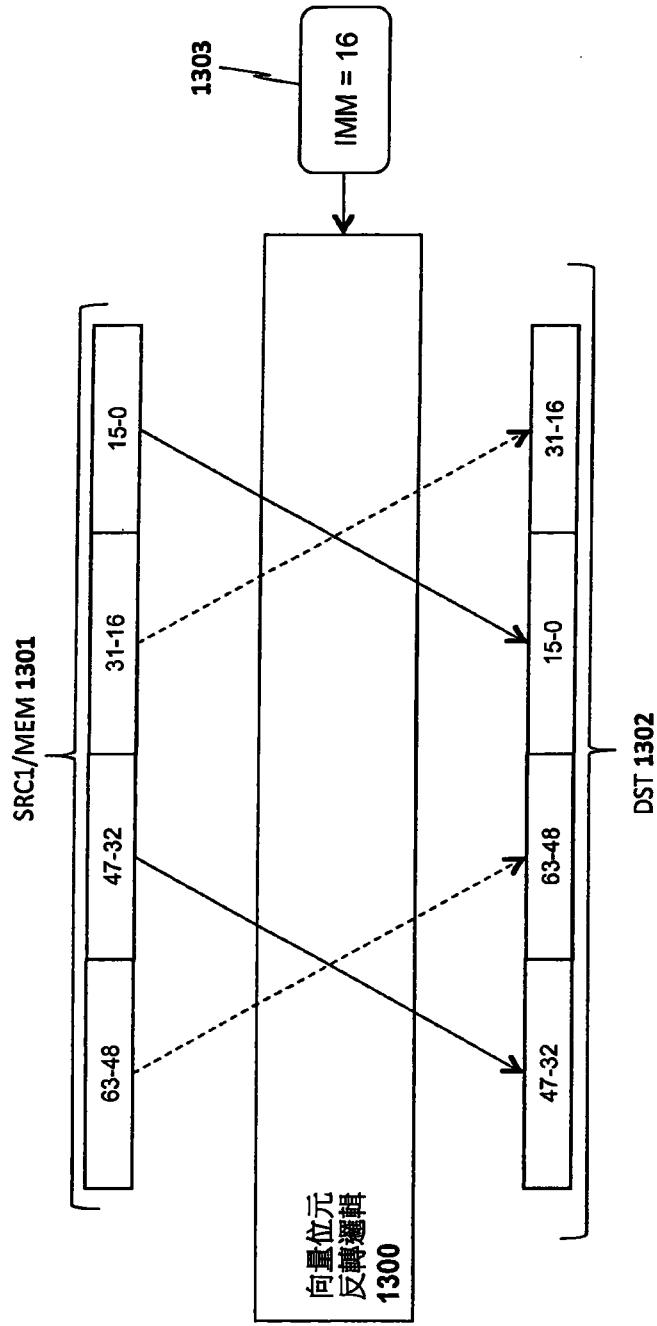


圖 18

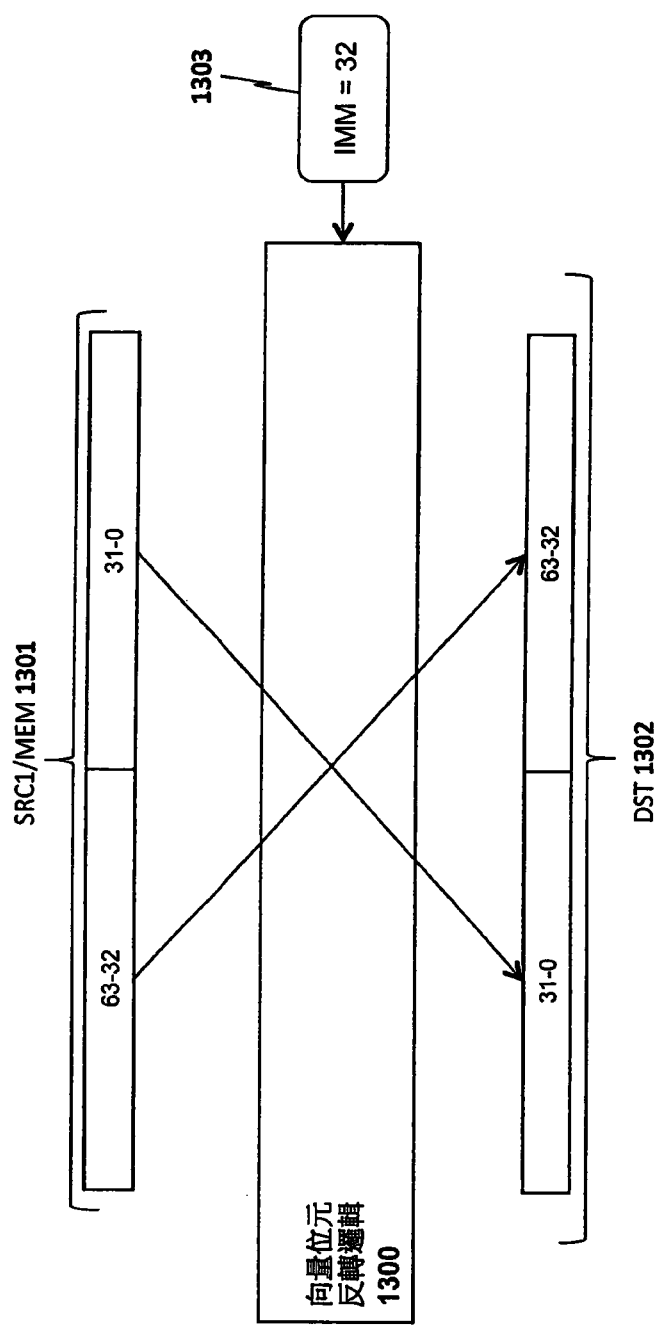
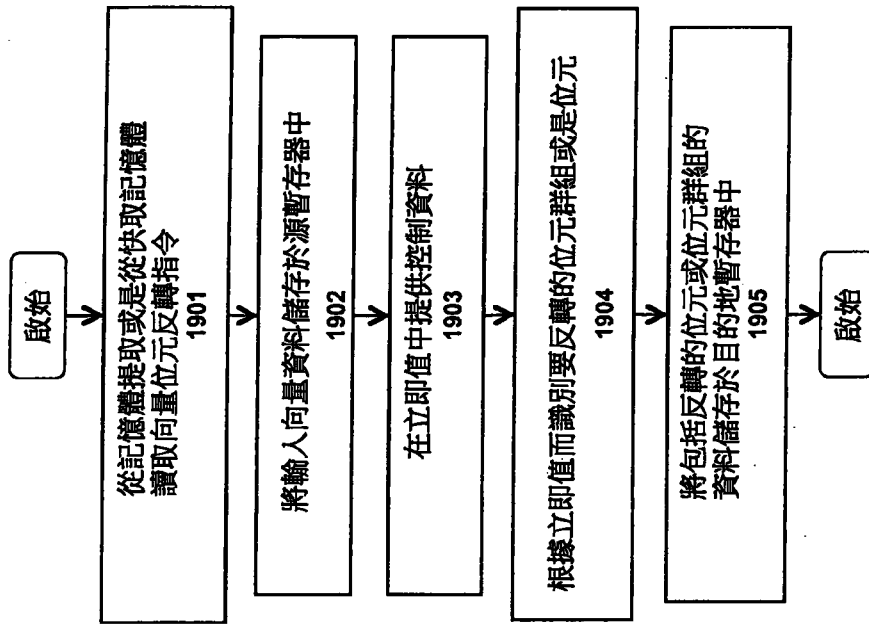


圖 19



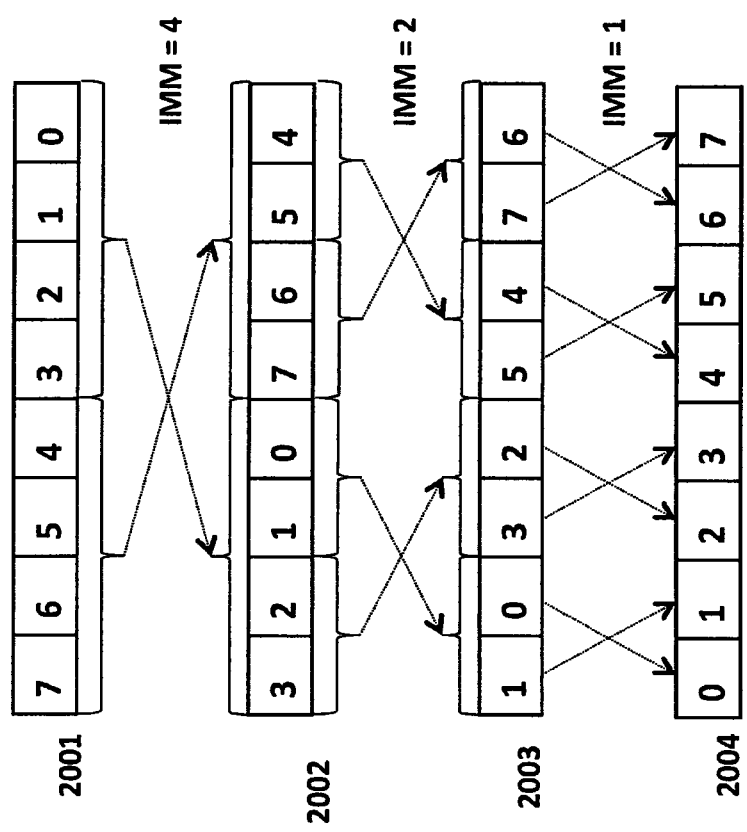


圖 20

圖 21

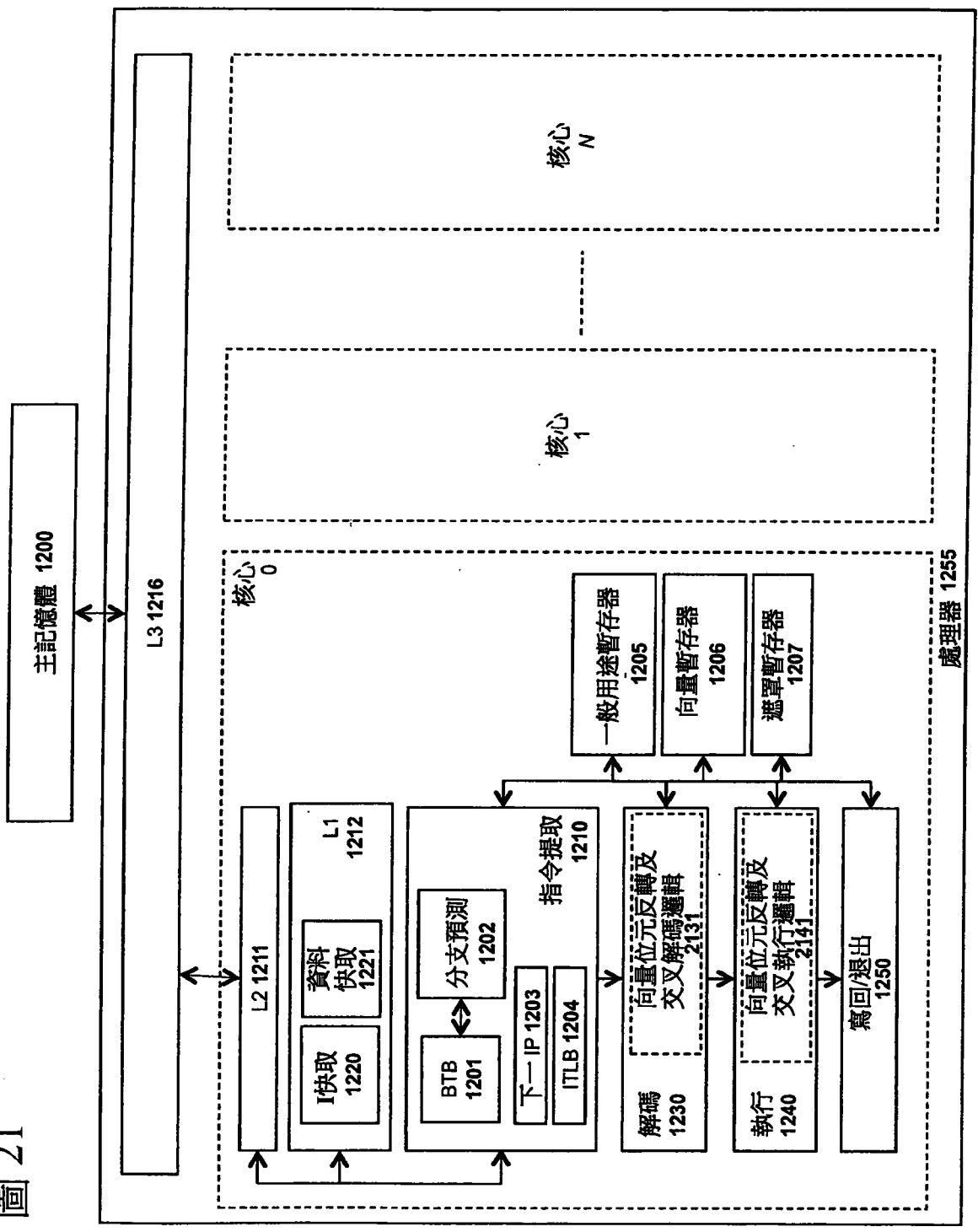


圖 22

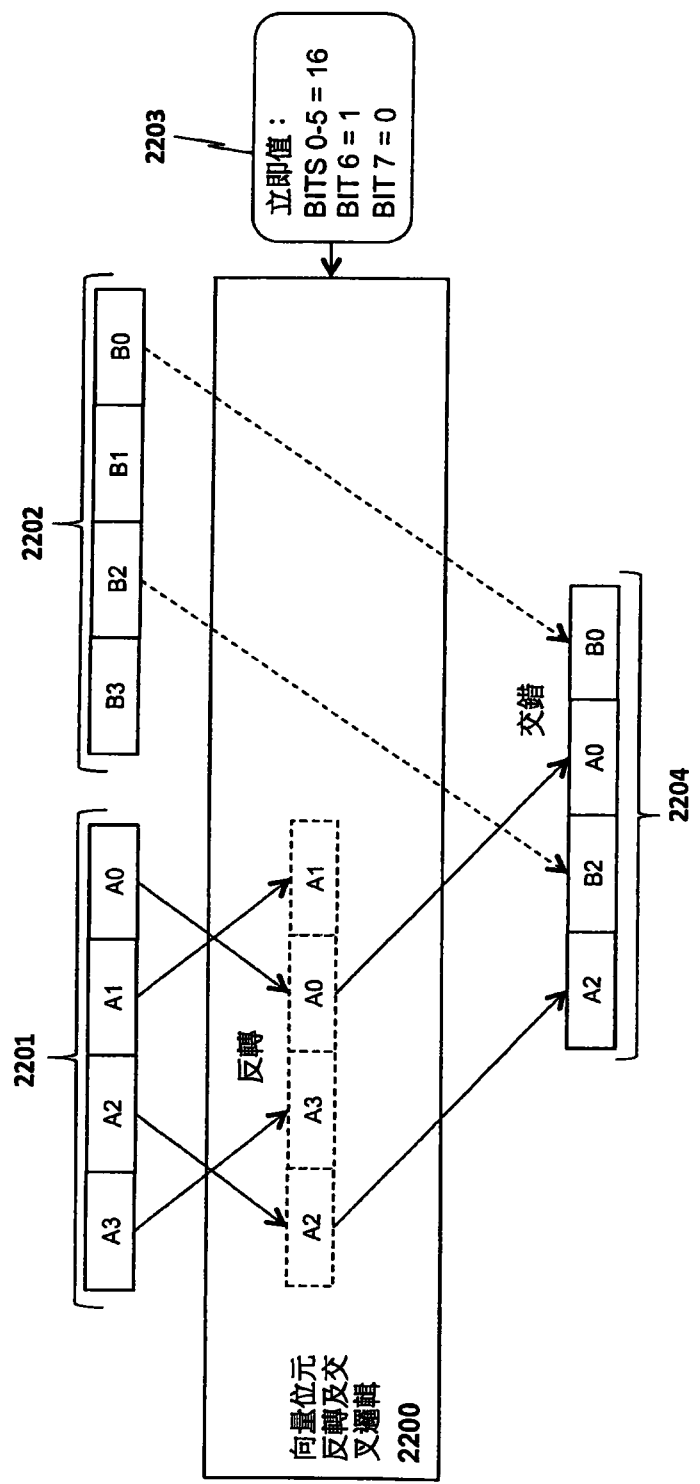


圖 23

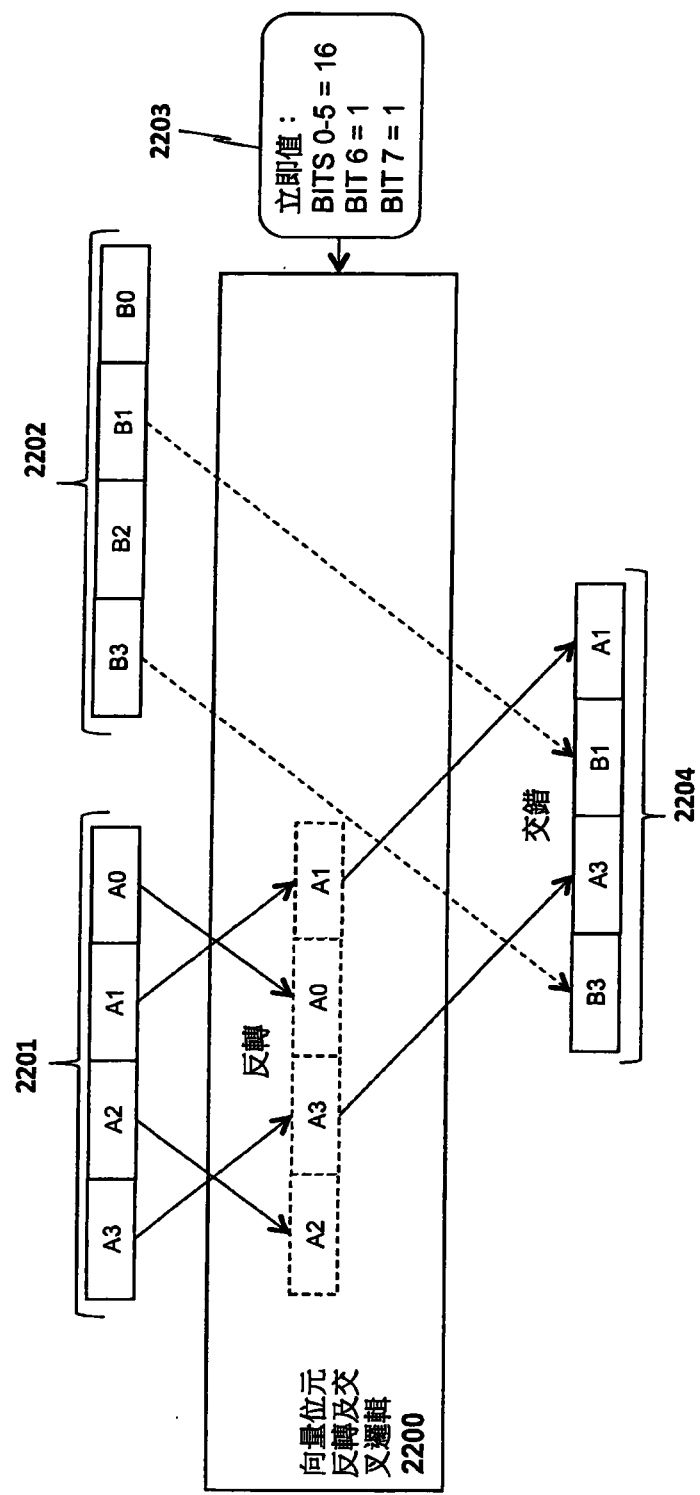
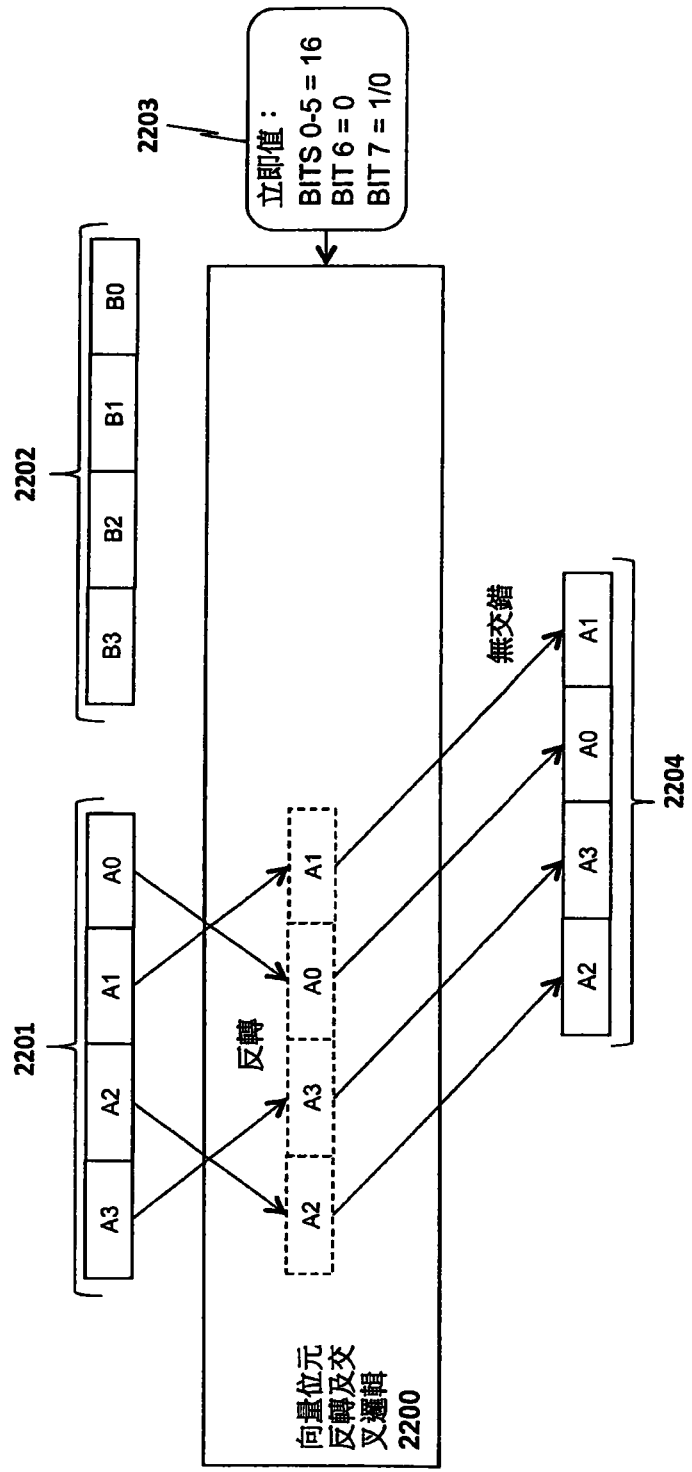


圖 24



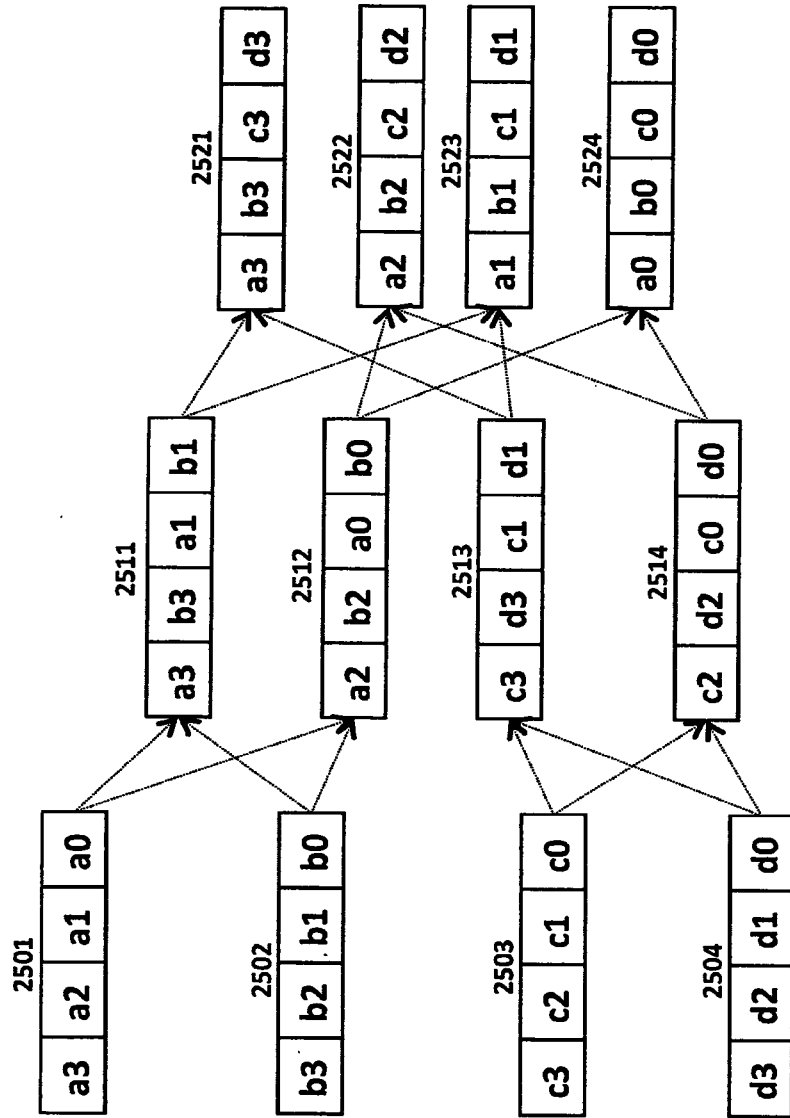


圖 25

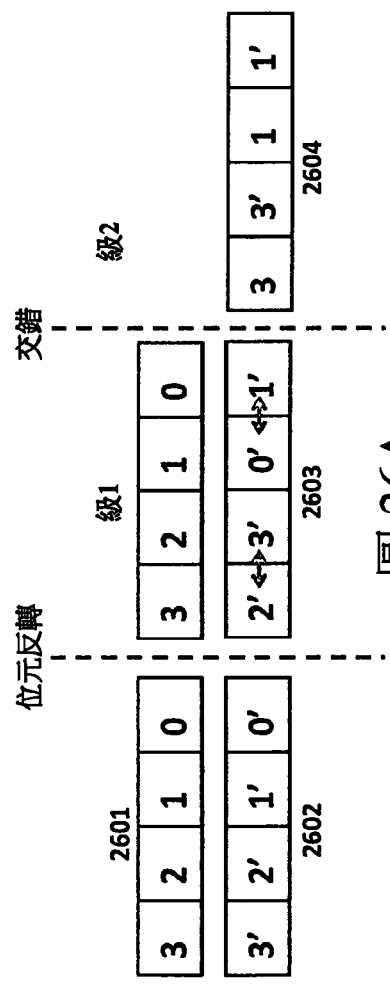


圖 26A

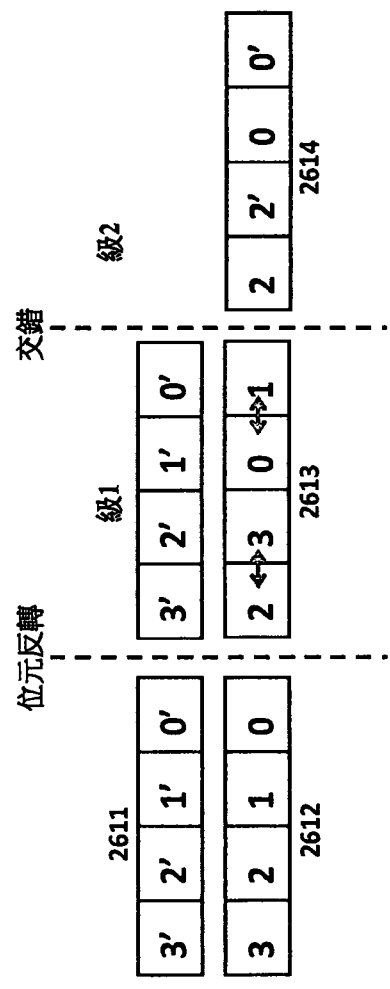
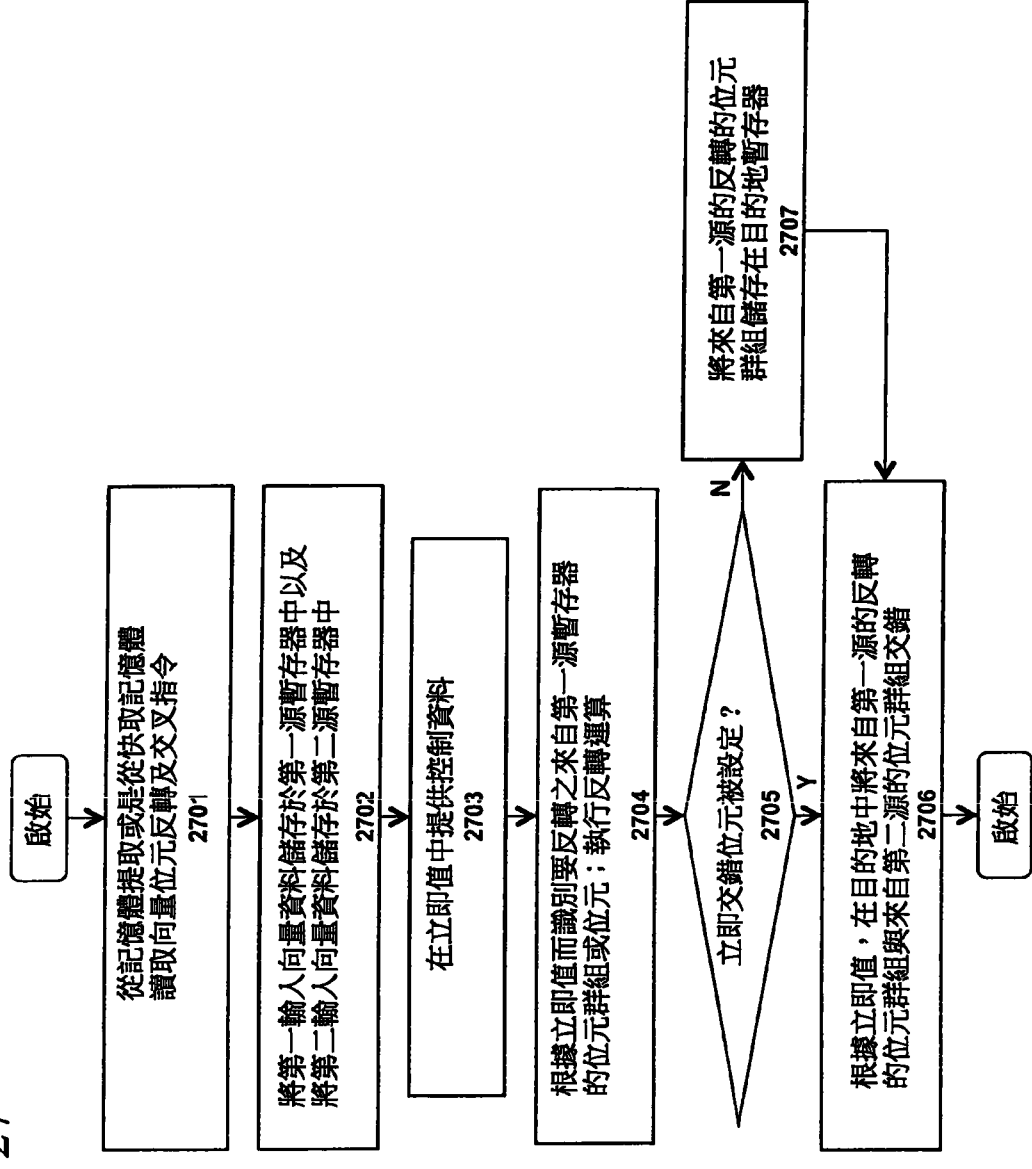


圖 26B

圖 27



【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	1220：I 快取
1221：資料快取	1201：BTB
1202：分支預測	1203：下一 IP
1204：ITLB	1210：指令提取
1230：解碼	1231：向量位元反轉解碼邏輯
1240：執行	1241：向量位元反轉執行邏輯
1250：寫回/退出	1205：一般用途暫存器
1206：向量暫存器	1207：遮罩暫存器
1255：處理器	1211：L1 快取記憶體
1212：L2 快取記憶體	1216：L3 快取記憶體

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：無

申請專利範圍

1. 一種處理器，包括：

第一源向量暫存器，用以儲存第一複數個源位元群組，其中，用於該些位元群組的大小會於指令的立即值中指定；

第二源向量，用以儲存第二複數個源位元群組；

向量位元反轉及交叉邏輯，用以從該立即值決定位元群組大小以及用以基於所決定的該位元群組大小反轉該第一源向量暫存器內鄰接的位元群組的位置，以產生一組反轉的位元群組，其中，該向量位元反轉及交叉邏輯是用以額外地將該組反轉的位元群組與該第二複數個位元群組交錯；以及

目的地向量暫存器，用以儲存與該第一複數個位元群組交錯之該些反轉的位元群組。

2. 如申請專利範圍第 1 項之處理器，其中，由於該交錯的結果，該些反轉的位元群組中的一半被選取而儲存在該目的地中，以及，該第二複數個位元群組中的一半被選取而儲存在該目的地中。

3. 如申請專利範圍第 2 項之處理器，其中，該些反轉的位元群組的該被選取的一半將根據該立即值而插入於該目的地向量暫存器內的偶數位置或奇數位置，以及，其中，該第二複數個位元群組的該被選取的一半將被儲存在相對於該些反轉的位元群組的該些位置之交錯位置內。

4. 如申請專利範圍第 1 項之處理器，其中，該向量

位元反轉及交叉邏輯包括一或更多多工器以將來自該源向量暫存器的該些位元群組反轉以及根據該立即值而在該目的地向量暫存器中將該些反轉的位元群組與來自該第二複數個位元群組的位元群組交錯。

5. 如申請專利範圍第 1 項之處理器，其中，該些位元群組的該大小是選自 1 位元、2 位元、4 位元、8 位元、16 位元、及 32 位元組成的族群。

6. 如申請專利範圍第 1 項之處理器，其中，該源向量暫存器及該目的地向量暫存器包括複數個 512 位元向量暫存器，各該 512 位元向量暫存器具有 64 位元資料元件，以及，其中，各位元群組包含在該 64 位元資料元件中之一者內。

7. 如申請專利範圍第 6 項之處理器，其中，該向量位元反轉及交叉邏輯係用以從該立即值決定位元群組大小以及用以回應地反轉用於該源向量暫存器的多個 64 位元資料元件的鄰接位元群組的位置。

8. 如申請專利範圍第 1 項之處理器，其中，該向量位元反轉及交叉邏輯係用以藉由首先反轉在最高等級的粒度的資料元件內鄰接的位元群組的位置，以及接著連續地降低該粒度直到該位元群組大小包括單一位元為止，而執行該資料元件內的所有位元的反轉。

9. 如申請專利範圍第 8 項之處理器，其中，該資料元件是 64 位元，以及，其中，用於反轉鄰接位元群組的位置之該最高等級的粒度包括 32 位元的位元群組大小，

用於反轉鄰接位元群組的位置之下一被選取的位元群組大小包括 16 位元，用於反轉鄰接位元群組的位置之下一被選取的位元群組大小包括 8 位元，用於反轉鄰接位元群組的位置之下一被選取的位元群組大小包括 4 位元，用於反轉鄰接位元群組的位置之下一被選取的位元群組大小包括 2 位元，以及，用於反轉鄰接位元群組的位置之最後被選取的位元群組大小包括 1 位元。

10. 如申請專利範圍第 1 項之處理器，又包括：

指令提取單元，用以從記憶體或快取記憶體提取向量位元反轉指令，該位元反轉指令具有與其相關連的立即值，該向量位元反轉及交叉邏輯用以處理該向量位元反轉指令以從該立即值決定該位元群組大小、以及用以回應地反轉該源向量暫存器內鄰接的位元群組的位置而產生一組反轉的位元群組。

11. 如申請專利範圍第 10 項之處理器，其中，該向量位元反轉及交叉邏輯包括向量位元反轉解碼組件用以將該向量位元反轉指令解碼而產生解碼的向量位元反轉指令、以及向量位元反轉執行組件用以執行該解碼的向量位元反轉指令。

12. 如申請專利範圍第 11 項之處理器，其中，該解碼的向量位元反轉指令包括複數個微運算。

13. 如申請專利範圍第 1 項之處理器，其中，該向量位元反轉及交叉邏輯是用以藉由產生多組反轉的位元群組及將該多組反轉的位元群組與包含來自該第二複數個位元

群組的位元群組之額外的多組位元群組交錯，而執行數學函數。

14. 如申請專利範圍第 13 項之處理器，其中，該額外的多組位元群組中的至少一些包括複數位元群組，該些反轉的位元群組是從該複數位元群組產生的，該複數位元群組包括來自該第二複數個位元群組的位元群組。

15. 如申請專利範圍第 13 項之處理器，其中，該數學函數包括轉置運算或是傾斜原始運算。

16. 一種用於執行向量位元反轉和交叉的方法，包括：

將第一複數個源位元群組儲存在第一源向量暫存器中，其中，用於該些位元群組的大小會於指令的立即值中指定；

將第二複數個源位元群組儲存在第二源向量中；

從該立即值決定位元群組大小以及基於所決定的該位元群組大小反轉該第一源向量暫存器內鄰接的位元群組的位置，以產生一組反轉的位元群組；

將該組反轉的位元群組與該第二複數個位元群組交錯；以及

將與該第一複數個位元群組交錯之該些反轉的位元群組儲存在目的地向量暫存器內。

17. 如申請專利範圍第 16 項之方法，其中，由於該交錯的結果，該些反轉的位元群組中的一半被選取而儲存在該目的地中，以及，該第二複數個位元群組中的一半被

選取而儲存在該目的地中。

18. 如申請專利範圍第 17 項之方法，其中，該些反轉的位元群組的該被選取的一半會根據該立即值而插入於該目的地向量暫存器內的偶數位置或奇數位置，以及，其中，該第二複數個位元群組的該被選取的一半將被儲存在相對於該些反轉的位元群組的該些位置之交錯位置內。

19. 如申請專利範圍第 16 項之方法，其中，該向量位元反轉及交叉邏輯包括一或更多多工器以將來自該源向量暫存器的該些位元群組反轉以及根據該立即值而在該目的地向量暫存器中將該些反轉的位元群組與來自該第二複數個位元群組的位元群組交錯。

20. 如申請專利範圍第 16 項之方法，其中，該些位元群組的該大小是選自 1 位元、2 位元、4 位元、8 位元、16 位元、及 32 位元組成的族群。

21. 如申請專利範圍第 16 項之方法，其中，該源向量暫存器及該目的地向量暫存器包括複數個 512 位元向量暫存器，各該 512 位元向量暫存器具有 64 位元資料元件，以及，其中，各位元群組包含在該 64 位元資料元件中之一者內。

22. 如申請專利範圍第 21 項之方法，又包括：從該立即值決定位元群組大小以及用以回應地反轉用於該源向量暫存器的多個 64 位元資料元件的鄰接位元群組的位置。

23. 如申請專利範圍第 16 項之方法，又包括：藉由

首先反轉在最高等級的粒度的資料元件內鄰接的位元群組的位置、然後連續地降低該粒度直到該位元群組大小包括單一位元為止，以執行該資料元件內的所有位元的反轉。

24. 如申請專利範圍第 23 項之方法，其中，該資料元件是 64 位元，以及，其中，用於反轉鄰接位元群組的位置之該最高等級的粒度包括 32 位元的位元群組大小，用於反轉鄰接位元群組的位置之下一被選取的位元群組大小包括 16 位元，用於反轉鄰接位元群組的位置之下一被選取的位元群組大小包括 8 位元，用於反轉鄰接位元群組的位置之下一被選取的位元群組大小包括 4 位元，用於反轉鄰接位元群組的位置之下一被選取的位元群組大小包括 2 位元，以及，用於反轉鄰接位元群組的位置之最後被選取的位元群組大小包括 1 位元。

25. 一種用於執行向量位元反轉和交叉的系統，包括：

記憶體，用以儲存程式碼及資料；

快取階層，包括多個快取階層以根據指定的快取管理策略而快取該程式碼及資料；

輸入裝置，用以從使用者接收輸入；

處理器，用以執行該程式碼及處理該資料以回應來自該使用者的輸入，該處理器包括：

第一源向量暫存器，用以儲存第一複數個源位元群組，其中，用於該些位元群組的大小將於指令的立即值中被指定；

第二源向量，用以儲存第二複數個源位元群組；

向量位元反轉及交叉邏輯，用以從該立即值決定位元群組大小以及用以基於所決定的該位元群組大小反轉該第一源向量暫存器內鄰接的位元群組的位置，以產生一組反轉的位元群組，其中，該向量位元反轉及交叉邏輯是用以額外地將該組反轉的位元群組與該第二複數個位元群組交錯；以及

目的地向量暫存器，用以儲存與該第一複數個位元群組交錯之該些反轉的位元群組。