



(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 I610229 B

(45)公告日：中華民國 107 (2018) 年 01 月 01 日

(21)申請案號：104138542

(22)申請日：中華民國 104 (2015) 年 11 月 20 日

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

G06F9/38 (2006.01)

(30)優先權：2014/12/23 美國

14/582,171

(71)申請人：英特爾股份有限公司 (美國) INTEL CORPORATION (US)

美國

(72)發明人：歐德亞麥德維爾 艾蒙斯特阿法 OULD-AHMED-VALL, ELMOUSTAPHA (MR) ;
 艾斯帕薩 羅傑 ESPASA, ROGER (ES) ; 吉倫范朵斯 大衛 GUILLEN FANDOS,
 DAVID (ES) ; 桑契斯 弗 SANCHEZ, F. JESUS (ES) ; 索羅 吉勒姆 SOLE,
 GUILLEM (ES)

(74)代理人：林志剛

(56)參考文獻：

TW 201337749A

TW 201339964A

US 7873812B2

US 2014/0006753A1

US 2014/0129801A1

審查人員：李國福

申請專利範圍項數：20 項 圖式數：15 共 96 頁

(54)名稱

用於向量廣播及互斥或和邏輯指令的設備與方法

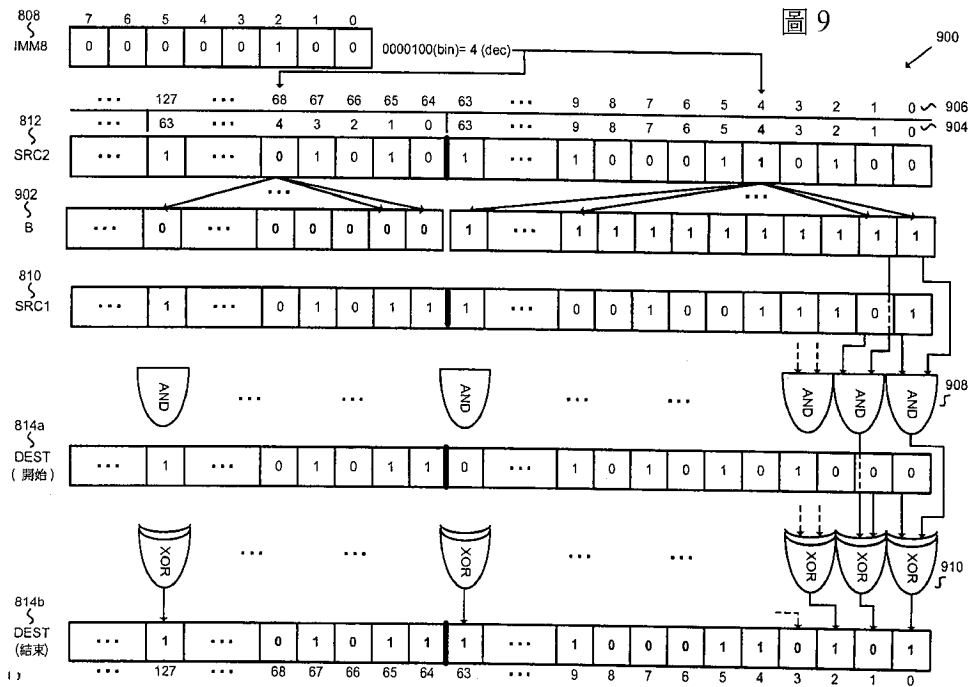
APPARATUS AND METHOD FOR VECTOR BROADCAST AND XORAND LOGICAL
 INSTRUCTION

(57)摘要

本案描述一種用以執行向量廣播與互斥或和邏輯指令的設備與方法。例如，處理器的一實施例包含：擷取邏輯，用以自記憶體擷取一表示目的緊縮資料運算元、第一源緊縮資料運算元、第二源緊縮資料運算元、及立即運算元的指令；及執行邏輯，用以：根據對應於該立即值的位置，決定在該第二源緊縮資料運算元中的位元；執行該第一源緊縮資料運算元與該所決定位元間的按位元 (bitwise) 和運算，以產生中間結果；執行該目的緊縮資料運算元與該中間結果間的按位元互斥或運算，以產生最終結果；並儲存該最終結果於該目的緊縮資料運算元所指示的儲存位置中。

An apparatus and method are described for performing a vector broadcast and XORAND logical instruction. For example, one embodiment of a processor comprises: fetch logic to fetch an instruction from memory indicating a destination packed data operand, a first source packed data operand, a second source packed data operand, and an immediate operand, and execution logic to determine a bit in the second source packed data operand based a position corresponding to the immediate value, perform a bitwise AND between the first source packed data operand and the determined bit to generate an intermediate result, perform a bitwise XOR between the destination packed data operand and the intermediate result to generate a final result, and store the final result in a storage location indicated by the destination packed data operand.

指定代表圖：



- 符號簡單說明：
- 808 . . . 立即運算元
 - 810 . . . 第一源緊縮資料運算元
 - 812 . . . 第二源緊縮資料運算元
 - 814 . . . 目的緊縮資料運算元
 - 900 . . . 邏輯
 - 902 . . . 暫時向量
 - 908 . . . 按位元和
 - 910 . . . 按位元互斥或

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)

用於向量廣播及互斥或和邏輯指令的設備與方法

Apparatus and method for vector broadcast and xorand logical instruction

【技術領域】

[0001] 本發明的實施例大致關係於電腦系統的領域。更明確地說，本發明的實施例關係於用以在電腦處理器內執行向量廣播與互斥或和（XORAND）邏輯指令的設備與方法。

【先前技術】

[0002] 某些類型的應用程式經常需要相同運算被執行於大數量的資料項上（稱為“資料並列”）。單指令多資料（SIMD）表示一類型的指令，其使得處理器對多重資料項執行運算。SIMD 技術特別適用於可以將在暫存器中的多數位元邏輯地分割成若干固定大小資料元件的處理器，各個資料元件代表分開的值。例如，在 256 位元暫存器中的位元可以被指明為予以運算為四個分開的 64 位元緊縮（packed）資料元件（四-字（Q）大小資料元件）、八個分開的 32 位元緊縮資料元件（雙字（D）大小資料元件）、十六個分開的 16 位元緊縮資料元件（字（W）大小資料元件）、或三十二個分開的 8 位元資料元件（位元

組（B）大小資料元件）的源運算元。此類型的資料被稱為“緊縮”資料類型或“向量”資料類型，及此資料類型的運算元被稱為緊縮資料運算元或向量運算元。換句話說，緊縮資料項或向量表示一順序的緊縮資料元件，及緊縮資料運算元或向量運算元為 SIMD 指令（也稱為緊縮資料指令或向量指令）的源或目的運算元。

[0003] 例如為具有包含 x86、MMX™、串流 SIMD 擴充（SSE）、SSE2、SSE3、SSE4.1 與 SSE4.2 指令的指令集的英特爾®Core™ 處理器所使用的 SIMD 技術已經在應用程式效能上完成顯著的改良。另一組稱為先進向量擴充（AVX）（AVX1 與 AVX2）並使用向量擴充（VEX）編碼設計的 SIMD 擴充已經公開（見，例如 2011 年十月的英特爾®64 與 IA-32 架構軟體開發者手冊；及見 2011 年六月的英特爾®先進向量擴充規劃參考書）。這些 AVX 擴充已經進一步提議使用擴充向量擴充（EVEX）編碼設計，來擴充以支援 512 位元暫存器（AVX-512）。

[0004] 在將布林（位元）矩陣乘以布林向量時仍會有挑戰存在。在很多情況下，現行實施方式可能需要對矩陣的元件重覆布林加法（XOR）運算。這浪費大量的處理器循環並且布林矩陣乘法運算也很慢。因此，如果此位元矩陣乘法運算係以減少不必要計算的方式加以實施，則可以獲得效率上的增加。

【圖式簡單說明】

[0005] 圖 1A 為依據本發明實施例之例示有序管線與例示暫存器重命名無序發出/執行管線的方塊圖；

[0006] 圖 1B 為依據本發明實施例之予以包含在處理器內的有序架構核心與例示暫存器重命名無序發出/執行架構核心的例示實施例的方塊圖；

[0007] 圖 2 為依據本發明實施例之具有積體記憶體控制器與圖形的單核處理器與多核心處理器的方塊圖；

[0008] 圖 3 例示依據本發明一實施例之系統的方塊圖；

[0009] 圖 4 例示依據本發明實施例之第二系統的方塊圖；

[0010] 圖 5 例示依據本發明實施例之第三系統的方塊圖；

[0011] 圖 6 例示依據本發明實施例之系統晶片（SoC）的方塊圖；

[0012] 圖 7 例示依據本發明實施例之對應於軟體指令轉換器的使用的方塊圖，以將在源指令集的二進制指令轉換為目標指令集的二進制指令；

[0013] 圖 8 例示系統 800 的方塊圖，其係可操作以執行向量廣播及 XORAND 邏輯指令的實施例；

[0014] 圖 9 例示依據本發明的一實施例之執行向量廣播與 XORAND 邏輯指令的邏輯；

[0015] 圖 10 為用於系統的方法 1000 的流程圖，其可操作以執行向量廣播與 XORAND 邏輯指令的實施例；

[0016] 圖 11 例示邏輯的例示虛擬碼，可操作以執行向量廣播與 XORAND 邏輯指令的實施例；

[0017] 圖 12A 與 12B 為方塊圖，例示依據本發明實施例之同屬向量友善指令格式與指令樣板；

[0018] 圖 13 依據本發明實施例的例示特定向量友善指令格式的方塊圖；

[0019] 圖 14 依據本發明的一實施例之暫存器架構的方塊圖；及

[0020] 圖 15A-B 例示更特定例示有序核心架構的方塊圖。

【發明內容與實施方式】

[例示處理器架構]

[0021] 圖 1A 為依據本發明實施例之例示有序擷取解碼退役管線與例示暫存器重命名無序發出/執行管線的方塊圖。圖 1B 為例示予以包含在依據本發明實施例之處理器中之有序擷取解碼退役核心的例示實施例與例示暫存器重命名無序發出/執行架構核心之方塊圖。圖 1A-B 的實線方塊例示管線與核心的有序部份，同時，選用加入的虛線方塊例示暫存器重命名無序發出/執行管線與核心。

[0022] 在圖 1A 中，處理器管線 100 包含擷取級 102、長度解碼級 104、解碼級 106、配置級 108、重命名級 110、排序（同時稱為分派或發出）級 112、暫存器讀取/記憶體讀取級 114、執行級 116、寫回/記憶體寫入級

118、例外處理級 122 及確定級 124。

[0023] 圖 1B 顯示處理器核心 190，包含有：耦接至執行引擎單元 150 的前端單元 130，並且兩者均耦接至記憶體單元 170。核心 190 可以為精簡指令集計算（RISC）核心、複雜指令集計算（CISC）核心、很長指令字（VLIW）核心或混合或交替核心類型。在另一選項中，核心 190 也可以是特殊用途核心，例如，網路或通訊核心、壓縮引擎、共處理器核心、通用計算圖型處理單元（GPGPU）核心、圖型核心或類似物。

[0024] 前端單元 130 包含耦接至指令快取單元 134 的分支預測單元 132，指令快取單元 134 係耦接至指令轉譯旁看緩衝器（TLB）136、轉譯旁看緩衝器 136 係耦接至指令擷取單元 138，指令擷取單元 138 耦接至解碼單元 140。解碼單元 140（或解碼器）可以解碼指令，並產生由原始指令所解碼或反映或推導出的一或更多微運算、微碼輸入項點、微指令、其他指令或其他控制信號作為輸出。解碼單元 140 可以使用各種不同機制加以實施。適當機制的例子包含但並不限於查看表、硬體實施方式、可程式邏輯陣列（PLA）、微碼唯讀記憶體（ROM）等等。在一實施例中，核心 190 包含微碼 ROM 或其他媒體，其儲存用於某些巨指令的微碼（例如，在解碼單元 140 或前端單元 130 內）。解碼單元 140 係耦接至在執行引擎單元 150 中的重命名/配置器單元 152。

[0025] 執行引擎單元 150 包含耦接至退役單元 154

與一組的一或更多排序單元 156 的重命名/配置器單元 152。排序單元 156 呈現任意數量的不同排序器，包含保留站、中央指令窗等等。排序單元 156 係耦接至實體暫存器檔案單元 158。各個實體暫存器檔案單元 158 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一或更多不同資料類型，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、狀態（例如，指令指標為下一個予以執行的指令的位址）等等。在一實施例中，實體暫存器檔案單元 158 包含向量暫存器單元、寫入遮罩暫存器單元及純量暫存器單元。這些暫存器單元可以提供架構向量暫存器、向量遮罩暫存器、及通用暫存器。實體暫存器檔案單元 158 係為退役單元 154 所重疊，以例示暫存器重命名及無序執行可以實行的各種方式（例如，使用重排序緩衝器與退役暫存器檔案；使用未來檔案，歷史緩衝器、及退役暫存器檔案；使用暫存器圖與暫存器池；等等）。退役單元 154 與實體暫存器檔案單元 158 係被耦接至執行叢集 160。執行叢集 160 包含一組之一或更多執行單元 162 及一組之一或更多記憶體存取單元 164。執行單元 162 可以執行各種運算（例如，移位、加法、減法、乘法）並針對各種類型之資料（例如，純量浮點、緊縮整數、緊縮浮點、向量整數、向量浮點）。雖然一些實施例可能包含若干專屬於特定功能或功能組的執行單元，但其他實施例也可以只包含一執行單元或都執行所有功能的多執行單元。排序單元 156、實體暫存器檔案單元 158、及

執行叢集 160 係被顯示為可能多數個，因為某些實施例建立用於某些類型資料/運算的分開管線（例如，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/或記憶體存取管線，其各個具有其本身排序單元、實體暫存器檔案單元、及/或執行叢集-及當在分開記憶體存取管線情況下，某些實施例係被實施為只有此管線的執行叢集具有記憶體存取單元 164）。也應了解的是，當使用分開管線時，這些管線的一或更多者可以為無序發出/執行及其他為有序。

[0026] 該組記憶體存取單元 164 係被耦接至記憶體單元 170，其係包含耦接至資料快取單元 174 的資料 TLB 單元 172，以及資料快取單元 174 係耦接至第二層（L2）快取單元 176。在例示實施例中，記憶體存取單元 164 可以包含負載單元、儲存位址單元、及儲存資料單元，其各個係耦接至在記憶體單元 170 中的資料 TLB 單元 172。指令快取單元 134 更進一步耦接至在記憶體單元 170 中的第二層（L2）快取單元 176。L2 快取單元 176 係耦接至一或更多其他層的快取並最後耦接至主記憶體。

[0027] 例如，例示暫存器重命名無序發出/執行核心架構可以如下實施管線 100：1）指令擷取 138 執行擷取與長度解碼級 102 及 104；2）解碼單元 140 執行解碼級 106；3）重命名/配置器單元 152 執行配置級 108 與重命名級 110；4）排序單元 156 執行排序級 112；5）實體暫存器檔案單元 158 與記憶體單元 170 執行暫存器讀取/記

記憶體讀取級 114；執行叢集 160 實行執行級 116；6）記憶體單元 170 與實體暫存器檔案單元 158 執行寫回/記憶體寫入級 118；7）各種單元可以涉及於例外處理級 122；及 8）退役單元 154 與實體暫存器檔案單元 158 執行確定級 124。

[0028] 核心 190 可以支援包含於此所述之指令之一或更多指令集（例如，x86 指令集（一些擴充已經加入至新版中）；美國加州的 Sunnyvale 的 MIPS 科技的 MIPS 指令集；美國加州的 Sunnyvale 的 ARM 控股的 ARM 指令集（有例如 NEON 的選用額外擴充）。在一實施例中，核心 190 包含邏輯，用以支援緊縮資料指令集擴充（例如，AVX1、AVX2 及/或如下所述之一些形式之同屬向量友善指令格式（U=0 及/或 U=1）），藉以允許為很多多媒體應用所使用的運算使用緊縮資料加以實行。

[0029] 應了解的是，核心可以支援多穿線（執行兩或更多並列組的運算或線程），並且可以以各種方式加以進行，包含時間分割（time sliced）多穿線、同時多穿線（其中單一實體核心提供邏輯核心，用於各個線程，實體核心被同時作多穿線）、或其組合（例如，在英特爾®超穿線（Hyperthreading）技術中的時間分割擷取及解碼及隨後同時多穿線）。

[0030] 雖然暫存器重命名係以無序執行的脈絡加以描述，但應了解的是，暫存器重命名也可以被用於有序架構中。雖然處理器的例示實施例也包含分開指令與資料快

取單元 134/174 與共享 L2 快取單元 176，但其他實施例也可以具有用於指令與資料的單一內部快取，例如第一層（L1）內部快取，或多層的內部快取。在一些實施例中，系統也可以包含內部快取及在核心及/或處理器外的外部快取的組合。或者，所有快取也可以在核心及/或處理器的外部。

[0031] 圖 2 為處理器 200 的方塊圖，其可以具有一個以上核心，並可以具有積體記憶體控制器，及可以具有依據本發明實施例之積體圖形。圖 2 的實線方塊例示處理器 200，其具有單一核心 202A、系統代理 210、一組一或更多的匯流排控制器單元 216，同選用加入的虛線方塊則例示另一處理器 200，其具有多數核心 202A-N、在系統代理單元 210 中的一組一或更多積體記憶體控制器單元 214、及特殊用途邏輯 208。

[0032] 因此，處理器 200 的不同實施方式可以包含：1）具有被整合有圖形及/或科學（通量）邏輯（其可以包含一或更多核心）的特殊用途邏輯 208 的 CPU，核心 202A-N 係為一或更多通用核心（例如，通用有序核心、通用無序核心、兩者的組合）；2）具有為大量特殊用途核心的核心 202A-N 的共處理器，其主要用於圖形與/或科學（通量）；及 3）具有為大量通用有序核心的核心 202A-N 的共處理器。因此，處理器 200 可以為通用處理器、共處理器或特殊用途處理器，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)、高通量多積體核心(MIC)共處理器(包含30或更多核心)、嵌式處理器或類似物。處理器可以實施在一或更多晶片上。處理器200可以使用若干製程技術,例如,Bi-CMOS、CMOS或NMOS之任一而為一或更多基板的一部份及/或可以實施在一或更多基板上。

[0033] 記憶體階層包含在核心內的一或更多層的快取、一組或一或更多共享快取單元206、及外部記憶體(未示出)耦接至該組積體記憶體控制器單元214。該組共享快取單元206可以包含一或更多中層快取,例如第2層(L2)、第3層(L3)、第4層(L4)或其他層的快取、最後層快取(LLC)及/或其組合。雖然在一實施例中,環為主互連單元212互連該積體圖形邏輯208、該組共享快取單元206及系統代理單元210/積體記憶體控制器單元214,但在其他實施例中也可以使用任意數量的已知技術,用以互連這些單元。在一實施例中,在一或更多快取單元206與核心202A-N間維持住連貫性(coherency)。

[0034] 在一些實施例中,一或更多核心202A-N係能夠多穿線。系統代理210包含這些元件,用以協調與操作核心202A-N。系統代理單元210可以包含例如電力控制單元(PCU)與顯示單元。PCU可以是或包含用以調節核心202A-N與積體圖形邏輯208的電力狀態所需的邏輯與元件。顯示單元係用以驅動一或更多外部連接顯示器。

[0035] 核心202A-N以架構指令集看來可以為同質或

異質；即，兩或更多核心 202A-N 可以執行相同指令集，而其他則可以只執行該指令集的一次組或不同的指令集。在一實施例中，核心 202A-N 為異質並可以包含如下所述之“小”核心與“大”核心。

[0036] 圖 3 至 6 為例示電腦架構的方塊圖。在膝上型電腦、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、開關、嵌式處理器、數位信號處理器（DSP）、圖形裝置、視訊遊戲裝置、機頂盒、微控制器、行動電話、攜帶式媒體播放器、手持裝置及各種其他電子裝置的領域中已知的其他系統設計與架構也是適用的。通常，能夠加入如於此所揭露之處理器及/或其他執行邏輯的各種系統或電子裝置也是適用的。

[0037] 現參考圖 3，顯示依據本發明一實施例之系統 300 的方塊圖。系統 300 可以包含一或更多處理器 310、315，其係耦接至控制器集線器 320。在一實施例中，控制器集線器 320 包含圖形記憶體控制器集線器（GMCH）390 與輸入/輸出集線器（IOH）350（可以是在分開的晶片上）；GMCH390 包含記憶體與圖形控制器，其上連接有記憶體 340 與共處理器 345；IOH350 係將輸入/輸出（I/O）裝置 360 耦接至 GMCH390。或者，記憶體與圖形控制器之一或兩者係被整合在處理器上（如於此所述），記憶體 340 與共處理器 345 係被直接耦接至處理器 310，及控制器集線器 320 與 IOH350 係在單一晶片上。

[0038] 額外處理器 315 的選用本質係被以虛線被描繪於圖 3 中。各個處理器 310、315 可以包含如此所述之一或更多處理核心並可以為一些版本的處理器 200。

[0039] 記憶體 340 可以例如為動態隨機存取記憶體 (DRAM)、相變記憶體 (PCM)、或兩者之組合。對於至少一實施例中，控制器集線器 320 與處理器 310、315 透過例如，前側匯流排 (FSB) 之多投匯流排、例如快速路徑 (QuickPath) 互連 (QPI) 的點對點介面，或類似連接 395 進行通訊。

[0040] 在一實施例中，共處理器 345 為一特殊用途處理器，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、嵌式處理器或類似物。在一實施例中，控制器集線器 320 可以包含積體圖形加速器。

[0041] 於實體資源 310、315 間以各種指標度量標準 (metrics of merit) 看來可以有各種不同，包含架構上、微架構上、熱學上、功率消耗特徵等等的不同。

[0042] 在一實施例中，處理器 310 執行控制一般類型的資料處理運算的指令。內藏在指令內的可以是共處理器指令。處理器 310 認出這些共處理器指令為應為附屬共處理器 345 所執行的類型。因此，處理器 310 發出這些共處理器指令 (或代表共處理器指令的控制信號) 於共處理器匯流排或其他互連上，至共處理器 345。共處理器 345 接受並執行所接收的共處理器指令。

[0043] 現參考圖 4，顯示依據本發明一實施例之第一更明確例示系統 400 的方塊圖。如於圖 4 所示，多處理器系統 400 為點對點互連系統，並包含第一處理器 470 與第二處理器 480，其係經由點對點互連 450 耦接。各個處理器 470 與 480 可以為一些版本的處理器 200。在本發明之一實施例中，處理器 470 與 480 分別為處理器 310 與 315，而共處理器 438 為共處理器 345。在另一實施例中，處理器 470 與 480 分別為處理器 310 與共處理器 345。

[0044] 處理器 470 與 480 係被顯示分別包含積體記憶體控制器（IMC）單元 472 與 482。處理器 470 同時也包含其匯流排控制器單元點對點（P-P）介面 476 與 478 作為其一部份；類似地，第二處理器 480 包含 P-P 介面 486 與 488。處理器 470、480 可以使用點對點（P-P）介面電路 478 與 488 透過 P-P 介面 450 交換資訊。如於圖 4 所示，IMC472 與 482 將處理器耦接至個別記憶體，即記憶體 432 與記憶體 434，其可以為本地附接至個別處理器的主記憶體的部份。

[0045] 處理器 470、480 可以各別使用點對點介面電路 476、494、486、498 透過個別 P-P 介面 452、454 與晶片組 490 交換資訊。晶片組 490 可以選用地經由高效介面 439 與共處理器 438 交換資訊。在一實施例中，共處理器 438 為特殊用途處理器，例如，高通量 MIC 處理器、網路或通訊處理器、壓縮引擎、圖型處理器、GPGPU、嵌式處

理器或類似物。

[0046] 共享快取（未示出）可以包含在處理器內或兩處理器外，但與處理器經由 P-P 互連連接，使得如果處理器被置入於低電力模式時，處理器之一或兩者的本地快取資訊可以儲存在共享快取內。

[0047] 晶片組 490 可以經由介面 496 耦接至第一匯流排 416。在一實施例，第一匯流排 416 可以為週邊元件互連（PCI）匯流排，或例如 PCI Express 匯流排之匯流排或另一第三代 I/O 互連匯流排，但本發明的範圍並不作此限定。

[0048] 如於圖 4 所示，各種 I/O 裝置 414 可以耦接至第一匯流排 416，及匯流排橋接器 418 將第一匯流排 416 耦接至第二匯流排 420。在一實施例中，一或更多額外處理器 415，例如，共處理器、高通量 MIC 處理器、GPGPU、加速器（例如圖形加速器或數位信號處理（DSP）單元）、場可程式閘陣列、或任何其他處理器係耦接至第一匯流排 416。在一實施例中，第二匯流排 420 可以為低接腳數（LPC）匯流排。各種可以耦接至第二匯流排 420 的裝置，包含有例如，鍵盤及/或滑鼠 422、通訊裝置 427 及儲存單元 428，例如磁碟機或其他大量儲存裝置，其在一實施例中可以包含指令/碼及資料 430。再者，音訊 I/O 424 可以耦接至第二匯流排 420。注意，其他架構也是可能的。例如，除了圖 4 的點對點架構之外，系統也可以實施多投匯流排或其他此架構。

[0049] 現參考圖 5，顯示依據本發明實施例之第二更特定例示系統 500 的方塊圖。在圖 4 及 5 中的類似元件係有類似元件符號，及圖 4 的某些態樣已經被由圖 5 中省略，以避免模糊了圖 5 的其他態樣。

[0050] 圖 5 例示處理器 470、480 可以分別包含積體記憶體及 I/O 控制邏輯（CL）472 與 482。因此，CL 472 與 482 包含積體記憶體控制器單元並包含 I/O 控制邏輯。圖 5 例示不但是記憶體 432、434 耦接至 CL 472、482，I/O 裝置 514 也耦接至控制邏輯 472、482。傳統 I/O 裝置 515 係耦接至晶片組 490。

[0051] 現參考圖 6，所示為依據本發明實施例之 SoC 600 的方塊圖。圖 2 中的類似元件有類似元件符號。同時，虛線方塊係為在更先進 SoC 上的選用特性。在圖 6 中，互連單元 602 係耦接至：應用處理器 610，其包含一組一或更多核心 202A-N 及共享快取單元 206；系統代理單元 210；匯流排控制器單元 216；積體記憶體控制器單元 214；一組或一或更多共處理器 620，其可以包含積體圖形邏輯、影像處理器、音訊處理器、及視訊處理器；靜態隨機存取記憶體（SRAM）單元 630；直接記憶體存取（DMA）單元 632；及顯示單元 640，用以耦接至一或更多外部顯示器。在一實施例中，共處理器 620 包含特殊用途處理器，例如網路或通訊處理器、壓縮引擎、GPGPU、高通量 MIC 處理器、嵌式處理器或類似物。

[0052] 於此所揭露之機制的實施例可以被實施於硬

體、軟體、韌體、或此等實施方式方法的組合。本發明之實施例可以被實施為電腦程式或執行於可程式系統上的程式碼，該可程式系統包含至少一處理器、儲存系統（包含揮發與非揮發記憶體及/或儲存元件）、至少一輸入裝置、及至少一輸出裝置。

[0053] 例如圖 4 所示之碼 430 的程式碼可以被應用至輸入指令，以執行於此所述之功能並產生輸出資訊。輸出資訊可以被以已知方式應用至一或更多輸出裝置。為了本案的目的，處理系統可以包含具有例如：數位信號處理器（DSP）、微控制器、特定應用積體電路（ASIC）、或微處理器的處理器之任意系統。

[0054] 程式碼可以實施為高階程序或物件導向程式語言，以與處理系統通訊。如果想要，程式碼也可以實施為組合語言或機器語言。事實上，於此所述之機制並不將範圍限制於任何特定程式語言。任何情況下，語言也可以是編譯或解譯語言。

[0055] 至少一實施例的一或更多態樣可以為儲存在機器可讀取媒體上的代表指令所實施，其代表在處理器內的各種邏輯，其當為機器所讀取時，使得機器製造邏輯以執行於此所述之技術。此稱為“IP 核心”的代表可以被儲存在有形機器可讀取媒體上並被供應至各客戶或製造設施，以載入實際完成該邏輯或處理器的該製造機器。

[0056] 此機器可讀取儲存媒體可以包含但並不限於非暫態有形配置的機器或裝置製造或形成的物體，包含儲

存媒體，例如硬碟、任何其他類型的碟片，包含軟碟、光碟、光碟唯讀記憶體（CD-ROM）、微型可寫光碟（CD-RW）、及磁光碟、半導體裝置，如唯讀記憶體（ROM）、隨機存取記憶體（RAM），例如動態隨機存取記憶體（DRAM）、靜態隨機存取記憶體（SRAM）、可抹除可程式唯讀記憶體（EPROM）、快閃記憶體、電氣可抹除可程式唯讀記憶體（EEPROM）、相變記憶體（PCM）、磁或光卡、或任何其他類型的適用以儲存電子指令的媒體。

[0057] 因此，本發明之實施例也包含非暫態有形機器可讀取媒體，包含有指令或含有設計資料，例如硬體描述語言（HDL），其界定於此所述之結構、電路、設備、處理器及/或系統特性。此等實施例也可以被稱為程式產品。

[0058] 在一些情況下，指令轉換器可以被用以將指令由源指令集轉換為目標指令集。例如，指令轉換器可以轉譯（例如，使用靜態二進制轉譯、動態二進制轉譯，包含動態編譯）、變體、仿真、或其他將一指令轉換為一或更多其他予以為核心所處理的指令。指令轉換器可以實施為軟體、硬體、韌體或其組合。指令轉換器可以為在處理器上、處理器外、或一部份在處理器內而一部份在處理器外。

[0059] 圖 7 為對比於軟體指令轉換器的使用的方塊圖，依據本發明實施例的將源指令集的二進制指令轉換為

目標指令集的二進制指令。在例示實施例中，指令轉換器為軟體指令轉換器，或者，指令轉換器也可以實施為軟體、韌體、硬體或其各種組合。圖 7 顯示高階語言 702 的程式，其可以使用 x86 編譯器 704 編譯，以產生 x86 二進制碼 706，其也可以為具有至少一 x86 指令集核心 716 的處理器所本地執行。具有至少一 x86 指令集核心 716 的處理器代表任何處理器，其可以執行與具有至少一 x86 指令集核心的英特爾處理器基本相同的功能，藉由相容地執行或處理（1）英特爾 x86 指令集核心的指令集的大部份或（2）針對執行於具有至少一 x86 指令集核心的英特爾處理器的應用程式的目標碼版本或其他軟體，以完成與具有至少一 x86 指令集核心的英特爾處理器基本相同的結果。x86 編譯器 704 代表一編譯器，其可操作以產生 x86 二進制碼 706（例如，目標碼），其可以有或沒有額外鏈結處理益可以被執行於具有至少一 x86 指令集核心 716 的處理器上。

[0060] 同樣地，圖 7 顯示高階語言 702 的程式，其可以使用另外指令集編譯器 708 編譯，以產生另一指令集二進制碼 710，其可以為沒有至少一 x86 指令集核心 714 的處理器所本地執行（例如，具有執行美國加州 Sunnyvale 的 MIPS 科技的 MIPS 指令集及/或執行美國加州 Sunnyvale 的 ARM 控股公司的 ARM 指令集的核心的處理器）。指令轉換器 712 係被用以將 x86 二進制碼 706 轉換為被沒有 x86 指令集核心 714 的處理器所本地執行的

碼。所轉換的碼似乎不會與另一指令集二進制碼 710 相同，因為能作此的指令轉換器很難完成；然而，所轉換碼將完成一般操作並可以由另一指令集作成指令。因此，指令轉換器 712 呈現軟體、韌體、硬體或其組合，其可以透過仿真、模擬或任何其他處理，允許沒有 x86 指令集處理器或核心的處理器或其他電子裝置執行 x86 二進制碼 706。

執行向量廣播與互斥或和邏輯指令的設備與方法

[0061] 如上所述，將布林（位元）矩陣乘以布林向量可能效率低。因此，想要將布林矩陣乘以向量的更有效率的方法。更明確地說，在一些實施例中，指令係被使用以執行矩陣乘法。該指令執行緊縮資料運算元的按位元和運算，其可以代表位元矩陣的行與被廣播至暫時向量的該位元向量的對應位元值。該位元向量可以為第二緊縮資料運算元所代表。按位元互斥或運算然後可以對此結果值與為第三緊縮資料運算元所表示的目的向量加以執行。當此指令針對該位元矩陣的所有行重覆時，目的向量累積所有運算的結果並呈現矩陣乘法的結果。注意，模組式算術係被用於位元向量與矩陣的矩陣運算，因此，一積運算變成按元位和，及總和運算變成按位元互斥或（即，XOR）。此一指令使一處理器，藉由消除在傳統指令集指令表（repertoire）中的額外邏輯指令的負擔，而有效地處理位元矩陣與位元向量的乘法。

[0062] 圖 8 為一方塊圖，其例示可以操作以執行向量廣播與 XORAND 邏輯指令的實施例的系統 800。在一些實施例中，系統 800 可以為通用處理器（例如在桌上型、膝上型或其他電腦中常用的類型）的一部份。或者，系統 800 可以為特殊用途處理器。適當特殊用途處理器的例子包含但並不限於加密處理器、網路處理器、通訊處理器、共處理器、圖形處理器、嵌式處理器、數位信號處理器（DSP）、及控制器（例如，微控制器），只提出幾個作為例子。處理器可以為任何各種複雜指令集計算（CISC）處理器、各種精簡指令集計算（RISC）處理器、各種很長指令字（VLIW）處理器、其各種混合或其他類型的處理器。

[0063] 在運算期間，系統 800 可以接收向量廣播與 XORAND 邏輯指令 802（以下稱為指令 802）的實施例。例如，指令 802 可以自指令擷取單元、指令佇列或類似物接收。指令 802 可以代表巨指令、組合語言指令、機器碼指令、或處理器的指令集的其他指令或控制信號。在一些實施例中，指令 802 可以明確指明（例如，透過一或更多欄位或一組位元），或表示（例如隱含表示）第一源緊縮資料運算元 810，並可以明確地指明或表示第二源緊縮資料運算元 812。指令 802 也可以明確地指明或表示目的緊縮資料運算元 814，並可以明確地指明或表示立即運算元 808。

[0064] 再次參考圖 8，系統 800 包含解碼單元或解碼

器 804。解碼單元可以接收與解碼包含指令 802 的指令。解碼單元可以輸出一或更多微指令、微運算、微碼輸入點、解碼指令或控制信號，或其他相對較低階指令或控制信號，其反映、表示指令 802 及/或由指令 802 導出。該一或更多相對較低階指令或控制信號可以透過一或更多相對較低階（例如電路級或硬體級）操作，實施相對較高階指令 802。在一些實施例中，解碼單元 804 可以包含：一或更多輸入結構（例如，輸入埠、輸入互連、輸入介面等），以接收指令 802；耦接至輸入結構的指令辨識邏輯，以接收與辨識指令 802；耦接至該辨識邏輯的解碼邏輯，以接收與解碼指令 802；及耦接至該解碼邏輯的一或更多輸出結構（例如，輸出埠、輸出互連、輸出介面等），以輸出一或更多對應較低階指令或控制信號。辨識邏輯與解碼邏輯可以使用各種不同機制實施，包含但並不限於微碼唯讀記憶體（ROM）、查看表、硬體實施法、可程式邏輯陣列（PLA）、及用以在本技藝中實施已知解碼單元的其他機制。在一些實施例中，解碼單元 804 可以與如圖 1 中所示之解碼單元 140 相同。

[0065] 系統 800 也可以包含一組暫存器。在一些實施例中，暫存器也可以包含通用暫存器，可操作以保持資料。雖然並不必要如此，但用語“通用”經常係用以表示可以在暫存器中儲存資料或位址的能力。各個通用暫存器也可以代表晶粒上儲存位置，其可操作以儲存資料。通用暫存器也可以代表可見架構暫存器（例如，架構暫存器檔）。

可見架構或架構暫存器可以為軟體及/或程式員所見及/或為指令所表示以識別運算元的暫存器。在給定微架構中，這些架構暫存器係對比於其他非架構或非可見架構暫存器（例如，暫時暫存器、重排序緩衝器、退役暫存器等等）。暫存器可以使用已知技術以不同方式實施於不同微架構中，並不限定於任何特別類型的電路。各種不同類型的暫存器係適用的。適當類型的暫存器的例子包含但並不限於專屬實體暫存器、使用暫存器重命名的動態配置實體暫存器，或其組合。

[0066] 在一些實施例中，第一源緊縮資料運算元 810 可以被儲存在第一通用暫存器中，第二源緊縮資料運算元 812 可以被儲存在第二通用暫存器中，目的緊縮資料運算元 814 可以被儲存在第三通用暫存器中。或者，記憶體位置、或其他儲存位置也可以被使用作為一或更多源運算元。例如，在一些實施例中，記憶體操作可能被用作為第二源緊縮資料運算元，但這並不是必然如此。

[0067] 執行單元 806 自解碼單元 804 接收控制信號並執行指令 802。執行單元 806 被指示以接收立即 8 位元值、第一源儲存位置、第二源儲存位置、及目的儲存位置。這些可以分別為立即運算元 808、第一源緊縮資料運算元 810、第二源緊縮資料運算元 812 與目的緊縮資料運算元 814 所表示。在一些實施例中，儲存位置表示暫存器，例如實體暫存器檔單元 158。在一些實施例中，儲存位置表示記憶體位置，例如，在如記憶體單元 170 之記憶

體單元內的位置。執行單元 806 的操作與功能可以進一步參考在圖 1 中之執行引擎單元 150 加以詳述。

[0068] 再次參考圖 8，執行單元 806 係耦接至解碼單元 804 與暫存器。例如，執行單元可以包含算術單元、算術邏輯單元、執行算術與邏輯運算的數位電路、包含乘法器與加法器的數位電路，或類似物。執行單元也接收一或更多解碼或轉換過的指令或控制信號，其代表指令 802 及/或由指令 802 所導出。執行單元也可以接收第一源緊縮資料運算元 810、第二源緊縮資料運算元 812、目的緊縮資料運算元 814、及立即運算元 808。在一些實施例中，立即運算元具有 8 位元值。在一些實施例中，第一源緊縮資料運算元 810、第二源緊縮資料運算元 812、及目的緊縮資料運算元 814 表示儲存位置，其具有 64 位元的倍數至 512 位元的值。執行單元係可回應於及/或由於指令 802 的結果（例如，回應於來自指令的直接或間接（例如，透過仿真）解碼的一或更多指令或控制信號）加以操作以儲存結果。

[0069] 在一些實施例中，在第一源緊縮資料運算元 810、第二源緊縮資料運算元 812 及目的緊縮資料運算元 814 中的緊縮資料元件（位元）為 64 位元緊縮資料元件（四倍字）。在此一實施例中，執行於各個 64 位元緊縮資料元件部上的運算係被重覆，及執行單元 806 可以以平行或串列方式對各個 64 位元緊縮資料元件部執行運算。在一些實施例中，為緊縮資料運算元所表示的值之長度可

以為長度 64 的很多倍數，並可以包含多數 64 位元部。

[0070] 由於指令 802 的結果，執行單元使用為立即運算元所表示的值作為索引位置，而決定在為第二源緊縮資料運算元 812 所表示之各個 64 位元緊縮資料元件部中的一位元。在一實施例中，立即運算元為 8 位元值，因此，可以代表由 0 至 63 的 64 索引位置，並可以表示在該第二源緊縮資料運算元之各個 64 位元緊縮資料元件部中的位元位置。

[0071] 一旦執行單元 806 決定在一或更多 64 位元緊縮資料元件部中的位元，在一些實施例中，該位元然後被廣播給各個 64 位元緊縮資料元件部的 64 位元的暫時向量陣列。換句話說，此一位元值係被重覆並被放置於此暫時向量陣列的所有 64 位元位置中。在一些實施例中，此暫時向量陣列可以為暫時或內部暫存器，其係並不能透過任何至系統的程式語言介面所存取者。

[0072] 對於各個 64 位元緊縮資料元件部，執行單元 806 然後採具有廣播位元的對應暫時向量陣列並執行此暫時向量陣列與為第一源緊縮資料運算元所示的對應 64 位元緊縮資料元件部的按位元和（AND）運算。換句話說，執行單元 806 採用所決定的位元並執行它與在第一源緊縮資料運算元中之對應位元值間的按位元和運算。

[0073] 執行單元 806 進一步採用此按位元和運算元的所得值（即，中間結果）並執行這些所得值與在目的緊縮資料運算元中之對應緊縮資料元件的按位元互斥或

(XOR) 運算。執行單元 806 然後採用由此按位元 XOR 運算所決定的值並將這些值儲存於目的緊縮資料運算元中的對應位置。

[0074] 上述之這些實施例允許系統 800 使用模組式算術，有效地將一位元矩陣乘以位元向量。在一些實施例中，一些或全部位元矩陣係被儲存於為第一源緊縮資料運算元所表示的儲存位置中，及位元向量係被儲存於第二緊縮資料運算元所表示的儲存位置中。在一些實施例中，位元矩陣係被儲存於列（即，在儲存位置中之位置 0 儲存該位元矩陣的第一列的第一元件、位置 1 係儲存該位元陣列的相同第一列的第二元件，等等）。在此安排下，位元矩陣應被轉置，使得在執行上述運算前，其係被以行儲存於儲存位置。當藉由設定在立即運算元與該位元矩陣的所有行中的不同值，執行單元 806 針對該位元向量的所有值重覆執行上述運算時，儲存在為目的緊縮資料運算元所表示的儲存位置中的值代表該位元矩陣與位元向量的矩陣乘法的結果。注意，在各個運算中之為立即運算元所表示的值應匹配為第一源緊縮資料運算元所表示的位元矩陣的行數，使得位元矩陣中的正確行被乘以在位元向量中的正確值（即，在位元向量中的位置 0 的值應被乘以在該位元矩陣的位置 0 的行）。為完成如此，為第一源緊縮資料運算元所表示的儲存位置或儲存位置位址可以針對指令 802 的每一遞迴加以改變。

[0075] 在一些實施例中，位元矩陣為 64×64 位元矩

陣。因為由第一源緊縮資料運算元所表示之儲存位置可能不能儲存位元矩陣的所有 64 行（例如，如果儲存位置為 512 位元暫存器），所以不同儲存位置也可以為第一源緊縮資料運算元所表示，以用於指令 802 的各個遞迴，以完成矩陣乘法運算。

[0076] 有關於上述實施例的進一步細節將如下參考圖 9 至 11 加以描述。

[0077] 執行單元與/或處理器可以包含特定或特殊邏輯（例如，電晶體、積體電路或其他硬體可能組合韌體（例如儲存在非揮發記憶體中的指令）及/或軟體），其可以操作以執行指令 802 及/或回應於指令 802 及/或指令 802 的結果而儲存結果（例如，回應於由指令 802 解碼或由指令 802 推導出的一或更多指令或控制信號）。在一些實施例中，執行單元可以包含：一或更多輸入結構（例如，輸入埠、輸入互連、輸入介面等等），以接收源運算元；耦接至輸入結構以接收並處理源運算元及產生所得運算元的電路或邏輯（例如乘法器及至少一加法器）；及耦接電路或邏輯以輸出所得運算元的一或更多輸出結構（例如，輸出埠、輸出互連、輸出介面等等）。

[0078] 為了避免模糊本說明，相當簡單地系統 800 已經被顯示並描述。在其他實施例中，系統 800 可以選用地包含其他已知處理器元件。此等元件的可能例子包含但並不限於指令擷取單元、指令與資料快取、第二或更高層快取、無序執行邏輯、指令排序單元、暫存器重命名單

元、退役單元、匯流排介面單元、指令及資料轉譯旁看緩衝器、預擷取緩衝器、微指令佇列、微指令序向器、其他包含在處理器中之各種元件、及其各種組合。此等元件的各種不同組合與架構也是適用的。實施例並不限於任何已知組合或架構。再者，實施例也可以包含在具有多數核心的處理器、邏輯處理器、或執行引擎中，其中的至少之一包含解碼單元與執行單元以執行指令 802 的實施例。

[0079] 圖 9 例示邏輯 900，用以依據本發明的一實施例，執行向量廣播與 XORAND 邏輯指令。在一些實施例中，執行單元 806 包含邏輯 900 以執行指令 802。在一些實施例中，指令 802 指明立即運算元 808 (IMM8)、第一源緊縮資料運算元 810 (SRC1)、第二源緊縮資料運算元 812 (SRC2)、及目的緊縮資料運算元 814 (DEST)。雖然在邏輯 900 中描繪的運算元表示特定二進制值，但這些值只包含作例示目的，在其他實施例中，運算元可以包含不同值。

[0080] 注意為 SRC1 810、SRC2 812、與 DEST 814 所指示的儲存位置可以各個儲存多個緊縮 64 位元值。在此一情況下，邏輯 900 表示這些 64 位元緊縮資料元件部可以各個以類似方式加以處理。在一些實施例中，各個部係與其他部並列處理。各個這些部的長度與區分 (demarcation) 係為在 904 的數字所表示。運算元值的總長度係為在 906 的數字所表示，在一實施例中，包含 512 位元 (即 8 個 64 位元緊縮資料元件被儲存在 512 位元向

量暫存器中）。

[0081] 在一實施例中，IMM8 808 為 8 位元值，其能指明由 0 至 63 的數字。在圖 9 的例示二進制值中，此值為“4”（即，二進制之 100b）。在邏輯 900 後的執行單元決定在對應於 IMM8 808 值的位元位置之 SRC2 812 的值。在圖 9 的例示值中，此對應於在 SRC2 812 的第一 64 位元部中的值“1”，及在 SRC2 的第二 64 位元部中的“0”。

[0082] 執行單元然後複製或廣播 SRC2 812 的各個選擇值 64 次至暫時向量 B 902。因此，對於 SRC2 的各個 64 位元部，具有相同長度的暫時向量 B 902 中的對應部係被散佈有 SRC2 812 的選擇值成為使用 IMM8 808 的索引值所選擇者。例如，在圖 9 中，第一個 64 位元部中的 SRC2 812 的選擇值為“1”。因此，在 B 902 的對應 64 位元部中的各個 64 值係被設定為“1”。在一些實施例中，暫時向量 B 902 可以為不能透過任何程式介面存取的隱藏式暫存器。在一些實施例中，暫時向量 B 902 可以被儲存在記憶體（例如，RAM）中、或快取中或其他儲存媒體中。

[0083] 雖然在圖 9 中，暫時向量 B 902 包含對應於 SRC2 812 的該 64 位元部的多數 64 位元部，但在一些實施例中，SRC2 的各個 64 位元部係被依序處理，因此，暫時向量 B 902 只包含單一 64 位元部。在其他實施例中，SRC2 812 的各個 64 位元部係被並列但分開處理，因此，一次可能存在暫時向量 B 902 的多個備份，但卻是彼此分開。

[0084] 在一些實施例中，當執行單元廣播 SRC2 812 中的單一選擇值至在暫時向量 B 902 中的 64 值時，需要少於 64 處理器週期或運算，並且，值係被以同時方式被置放於暫時向量 B 902 中。在一些實施例中，廣播係依據熟習於本技藝者所知的廣播方法加以執行。

[0085] 執行單元更進一步藉由執行暫時向量 B 902 的重覆值與 SRC1 810 的對應值的按位元和（AND）運算，來執行邏輯 900。

[0086] 執行單元更執行邏輯 900，以進行以上按位元和運算（在 908）的結果與如 DEST 814a 所示之 DEST 的原始值的按位元互斥或（XOR）運算。此按位元 XOR 結果然後被儲存回由 DEST 814b 所顯示的 DEST 所表示的儲存位置。例如，在位元位置 0（如 906 所表示），於暫時向量 B 902 的值“1”與 SRC1 810 的對應位元位置的值“1”間的按位元和運算為“1”，以及，此所得“1”值與在 DEST 814a 的對應位元位置中之“0”值間的按位元 XOR 運算為“1”。此“1”值係被儲存在由 DEST 所表示的儲存位置的對應位元位置，如所示係在 DEST 814b 的位元位置 0。

[0087] 在一些情況下，邏輯 900 可以用以有效地決定於大小 64×64 的位元矩陣與大小 64×1 的位元向量間的矩陣乘法的結果。在此一實施例中，位元向量的值係為 SRC2 812 所表示，及在位元矩陣的一或更多行中的值係為 SRC1 810 所表示。如果在儲存器中的位元矩陣係原先使用逐列（row by row）格式加以表示，則位元矩陣將首

先被執行單元所轉置為逐行（column by column）格式。對於位元矩陣的矩陣乘法係使用模組式算術加以表現。因此，乘法係為按位元和運算（AND）所表示，及加法係為按位元互斥或（XOR）運算所表示。使用邏輯 900 以將位元矩陣乘以位元向量的第一步為採用該位元矩陣的第一行並將之乘以在位元向量中的第一值。為此，IMM8 係被設定至十進制值“0”，以表示在該位元向量中的第一位元位置，其係為 SRC2 812 所表示。為 64 位元長的位元矩陣的第一行也為 SRC1 810 所表示。然後，進行在暫時向量 B 902 中的位元向量的第一位元與 SRC1 810 的值的按位元和（AND）表示該位元矩陣的第一行與該位元向量的第一位元的乘法。如圖 9 所示，按位元和（AND）運算係在 908 所示並如上述。

[0088] 為了適當完成矩陣乘法，上述運算被重覆用於位元矩陣的後續行與在位元向量中的對應位元，其中各個行與位元乘法產生所得 64 位元值。這些所得 64 位元值應被總和在一起，以到達單一 64 位元值。此單一 64 位元值代表位元矩陣與位元向量的矩陣乘法的結果。在位元矩陣運算中的總和係為按位元 XOR 運算所代表。因此，在邏輯 900 中的如 910 所示的按位元 XOR 運算將現行位元矩陣行與位元向量位元積加至執行總和，以用於整個乘法運算。

[0089] 在一些情況下，64×64 位元矩陣係以逐行被儲存於記憶體位置中。因此，對於指令的各個執行，為

SRC1 810 所表示的儲存位址可以向位元矩陣的下一行移位 64 位元。在其他情況下，SRC1 810 可以改變至在位元矩陣中儲存下一行的不同儲存位置。在一些其他情況下，代表位元向量的 SRC2 812 的各個 64 位元部可以由前一 64 位元部移開（或旋轉）1 位元，以及代表位元矩陣的 SRC1 810 的各個 64 位元部可以包含相同位元矩陣的連續行。這將允許位元矩陣的連續行被一次處理。在另一情況下，SRC1 與 SRC2 的各個 64 位元部可以代表不同組的位元矩陣與位元向量結合，允許同時進行多個位元矩陣與位元向量的乘法計算。

[0090] 雖然以上矩陣乘法例描繪 64 位元長的位元矩陣與位元向量的乘法，但在其他情況下，位元矩陣與位元向量也可以少於 64 位元長。

[0091] 圖 10 為一種用於可以操作以執行向量廣播與 XORAND 邏輯指令的實施例的系統的方法 1000 的流程圖。在各種實施例中，該方法可以為處理器、指令處理設備、或其他數位邏輯裝置所執行。在一些實施例中，圖 10 的運算及/或方法可以為圖 8 的處理器所執行及/或在圖 8 處理器之中執行。於此所述之用於圖 8 的處理器之組件、特性、及特定選用地應用細節也可以選用於圖 10 的運算及/或方法中。或者，圖 10 的運算及/或方法也可以為參考圖 1 至 8 所述之處理器或設備的類似或不同處理器或設備所執行及/或在其中。再者，圖 8 的處理器也可以執行與圖 10 中所執行的運算及/或方法相同、類似或不同。

[0092] 方法 1000 包含在方塊 1002，由記憶體擷取表示目的緊縮資料運算元、第一源緊縮資料運算元、第二源緊縮資料運算元與立即運算元的指令。在各種態樣中，該指令可以於處理器、指令處理設備或其一部份（例如，指令擷取單元、解碼單元、匯流排介面單元等等）所擷取及接收。在各種態樣中，指令可以由晶粒外源（例如，由記憶體、互連等）、或由晶粒內源（例如，指令快取、指令佇列等）所接收。

[0093] 在方塊 1004，指令被解碼。在一些實施例中，指令的解碼可以為例如圖 8 的解碼單元 804 的解碼單元所執行。

[0094] 在方塊 1006，方法 1000 包含根據對應於立即運算元的值的位置，決定在第二源緊縮資料運算元中的一位元。在一些實施例中，資料元件的決定係為例如圖 8 的執行單元 806 的執行單元所執行。

[0095] 在方塊 1008，方法 1000 包含儲存結果於為目的緊縮資料運算元所表示的儲存位置，其中結果係由執行目的緊縮資料運算元與中間結果間的按位元 XOR 運算加以決定，以及，其中該中間結果係由執行於第一源緊縮資料運算元與所決定位元間的按位元 AND 運算加以決定。

[0096] 所例示方法涉及架構運算（例如，這些可以由軟體觀點看出）。在其他實施例中，該方法可以任意包含一或更多微架構運算。例如，指令可以被擷取、解碼、無序排序，源運算元可以被存取，執行單元可以執行微架

構運算，以實施該指令，結果可以重新排回到程式順序等等。在一些實施例中，實施該指令的微架構運算可以選用地包含圖 1 至 7 與 12 至 15 中所述之任一運算。

[0097] 圖 11 例示用以可操作執行向量廣播與 XORAND 邏輯指令的實施例之邏輯的例示虛擬碼。在一些實施例中，此邏輯為邏輯 900。在此可以為運算碼“vxorandbcstq”1152 所代表的指令 802 可以指明如在 1154-1160 所示的各種運算。運算元 zmm0 1152 指明目的緊縮資料運算元。在一些實施例中，zmm0 1154 為 DEST 814。在一些實施例中，指令指明寫入遮罩 1162，在此例為“k1”。寫入遮罩的值可以表示執行為此虛擬碼所代表的邏輯的執行單元是否將值寫入至為目的緊縮資料運算元所表示的暫存器的指定部份。運算元 zmm1 1156 指明第一源緊縮資料運算元。在一些實施例中，此為 SRC1 810。運算元 zmm2 1158 指明第二源緊縮資料運算元。在一些實施例中，此為 SRC2 812。運算元 imm8 1160 指明立即運算元。在一些實施例中，imm8 1160 為 IMM8 808。

[0098] 線 1102 表示在一些實施例中，指令係相容於 128、256 及 512 的向量長度。K 長度表示 64 緊縮資料元件部的數量，二進制值的對應向量長度可以被分割的數量。如上所述，指令對 64 個緊縮資料元件部運算。

[0099] 在一些實施例中，指令的運算元指明表示可以儲存多達 512 位元的儲存位置的運算元，在此情況下，只有暫存器的一部份被用於指令的執行。在一些實施例

中，一或更多運算元可以表示記憶體儲存位置，而不是暫存器位置。

[0100] 在圖 11 中，具有等號符號的冒號符號（： $=$ 符號）表示在： $=$ 號的右側的值係被指定為在： $=$ 號左側的變數。

[0101] 在線 1104，迴路被設定以遞迴等於 K 長度的迴路次數。例如，如果向量長度為 128，則 K 長度將為 2，及迴路將遞迴兩次。在一些實施例中，如圖 11 所示，迴路變數為“j”。

[0102] 在線 1106，變數 i 被設定為 j 乘以 64。例如，當 j 為“2”時，變數 i 將會是“128”。

[0103] 在線 1108，可以被儲存在內部暫存器的暫時向量 B 的 64 位元係被設定為用於暫時向量 B 的 64 位元的 $SRC2[i+IMM8]$ 複製 64 次的值。在一些實施例中，暫時向量 B 為暫時向量 B 902。 $SRC2[i+IMM8]$ 代表在 $SRC2$ 中位置“ $i+IMM8$ ”的值。

[0104] 在一些實施例中，值 $SRC2[i+IMM8]$ 被設定至暫時值“ b ”，其可以為內部暫存器，及此暫時值 b 然後被複製或廣播至暫時向量 B 的值。

[0105] 在線 1112，對於現行被處理的 64 位元部的為線 1104 的迴路所表示的 $DEST$ 的 64 位元係與先前按位元 AND 運算的結果作 XOR 運算，並且，此結果被指定回到 $DEST$ 的相同 64 位元。

[0106] 在一些實施例中，在線 1112 的運算係被預估

是否指令 802 指明一寫入遮罩。如果指明一寫入遮罩，則如在線 1110 所示，在寫入遮罩中之位置 j 的位元應被設定為予以為執行單元 806 所執行之在線 1112 上的運算的值“1”。否則，將執行在線 1114-1118 的運算。

[0107] 如果在線 1110 的條件被決定為“0”或假，則執行線 1114。在一些實施例中，在線 1114，條件宣告檢查是否致能合併遮罩。在一些實施例中，合併遮罩係為旗標所表示。在一些實施例中，此旗標為“EVEX.z”。在一些實施例中，此旗標為在指令中之運算元（例如，“{z}”）所表示。合併遮罩給執行單元表示保留目的運算元的原始值，而不是以“0”來重寫這些值。如果合併遮罩為導通（on），則在 DEST 中的正被處理的該組 64 緊縮資料元件保持不變，如線 1116 所示。否則，如在線 1118 所示，這些值被以“0”置換（即，值“0”被儲存在為目的運算元所表示的暫存器的對應位置中）。

[0108] 在一些實施例中，在線 1120，未被處理為指令的一部份的 DEST 中的剩餘值，即超出向量長度所指定的值被零化（zeroed out）（即，值“0”係被儲存在為目的運算元所表示的暫存器的對應位置中）。

[0109] 雖然以上實施例係參考 512 位元寬的暫存器加以描述，但本發明之其他實施例中暫存器並不需為此長度，本發明可以以任何長度的暫存器加以實施。

例示指令格式

[0110] 於此所述之指令的實施例可以以不同格式加以實施。另外，例示系統、架構及管線係詳述如下。該（等）指令的實施例可以執行在此等系統、架構及管線上，但並不限於在此所詳述者。

[0111] 向量友善指令格式係為適用於向量指令的指令格式（例如，有幾欄位特定於向量運算）。雖然所述實施例中，可以透過向量友善指令格式而支援向量與純量運算，但其他實施例中也可以只使用向量來運算該向量友善格式。

[0112] 圖 12A-12B 為例示依據本發明實施例之同屬向量友善指令格式與其指令樣板的方塊圖。圖 12A 為一方塊圖，例示依據本發明實施例之同屬向量友善指令格式與其類型 A 指令樣板；而圖 12B 為一方塊圖，例示依據本發明實施例之同屬向量友善指令格式及其類型 B 指令樣板。明確地說，同屬向量友善指令格式 1200 被界定有類型 A 與類型 B 指令樣板，其兩者均不包含記憶體存取 1205 指令樣板與記憶體存取 1220 指令樣板。在向量友善指令格式脈絡中的用語“同屬”表示指令格式並不被綁至任何特定指令集。

[0113] 雖然本發明實施例將以向量友善指令格式支援以下的態樣加以描述，但其他實施例也可以支援具有更多、更少、或不同資料元件寬度（例如 128 位元（16 位元組）資料元件寬度）的更多、更少及/或不同向量運算元大小（例如 256 位元組向量運算元）的其他實施例。這

些態樣為：具有 32 位元（4 位元組）或 64 位元（8 位元組）資料元件寬（或大小）的 64 位元組向量運算元長度（或大小）（因此，64 位元組向量係由 16 個雙字大小元件或者 8 四倍字大小元件所構成）；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）的 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 8 位元（1 位元組）資料元件寬度（或大小）的 32 位元組向量運算元長度（或大小）；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 8 位元（1 位元組）資料元件寬度（或大小）的 16 位元組向量運算元長度（或大小）。

[0114] 在圖 12A 中的類型 A 指令樣板包含：1）在無記憶體存取 1205 內的指令樣板中，其中顯示沒有記憶體存取，全捨入控制類型運算 1210 指令樣板；及無記憶體存取，資料轉換類型運算 1215 指令樣板；及 2）在記憶體存取 1220 內的指令樣板中，其中顯示記憶體存取，時間 1225 指令樣板；及記憶體存取非時間 1230 指令樣板。在圖 12B 中的類型 B 指令樣板包含：1）在無記憶體存取 1205 內的指令樣板中，其中顯示無記憶體存取，寫入遮罩控制部份捨入控制類型運算 1212 指令樣板；及無記憶體存取，寫入遮罩控制 *vsize* 類型運算 1217 指令樣板；及 2）在記憶體存取 1220 指令樣板內，顯示記憶體存取，寫入遮罩控制 1227 指令樣板。

[0115] 同屬向量友善指令格式 1200 包含在圖 12A-12B 所列的順序中的下列以下欄位。

[0116] 格式欄 1240-在此欄中之特定值（指令格式識別值）獨特地指明該向量友善指令格式，及在該指令流中的向量友善指令格式中的指令的發生。因此，此欄為選用的，對於只有同屬向量友善指令格式的指令集是不必要的。

[0117] 基礎運算欄 1242-其內容區別不同基礎運算。

[0118] 暫存器索引欄 1244-其內容直接或透過位址產生指明源及目的運算元在暫存器或記憶體中的位置。這些包含足夠數量的位元，以由 $P \times Q$ （例如， 32×512 、 16×128 、 32×1024 、 64×1024 ）暫存器檔案中選出 N 個暫存器。雖然在一實施例中， N 可以是多達 3 個源及一個目的暫存器，但其他實施例中，也可以支援更多或更少源與目的暫存器（例如，可以支援多達兩源，其中兩者之一也可以作動為目的地，也可以支援多達三個源，其中之一也可以作動為目的地，也可以支援多達兩個源及一目的地）。

[0119] 修飾符欄 1246-其內容區別在同屬向量指令格式中指明記憶體存取與未指明記憶體存取的指令的發生；即，在無記憶體存取 1205 指令樣板與記憶體存取 1220 指令樣板間。記憶體存取運算讀取及/或寫入至記憶體階層（在一些情況下，使用暫存器中的值，指明源與/或目的位址），而無記憶體存取運算則否（例如，源與目的為暫存器）。雖然在一實施例中，此欄也可以於三個不同方式

間作選擇，以執行記憶體位址計算，但在其他實施例中，也可以支援更多、更少或不同方式，以執行記憶體位址計算。

[0120] 擴增運算欄 1250-其內容區分除了基礎運算外，各種不同運算中的哪個將被執行。此欄為上下文特定。在本發明的一實施例中，此欄被分為類型欄 1268、 α 欄 1252 與 β 欄 1254。擴增運算欄 1250 允許共同群組內的運算可以於單一指令中執行，而不是分成 2、3 或 4 指令執行。

[0121] 縮放欄 1260-其內容允許用於記憶體位址產生（例如，用於使用 $2^{\text{縮放}} * \text{索引} + \text{基礎的位址產生}$ ）的索引欄的內容縮放。

[0122] 位移欄 1262A-其內容係用以作為記憶體位址產生的一部份（例如，用於使用 $2^{\text{縮放}} * \text{索引} + \text{基礎} + \text{位移的位址產生}$ ）。

[0123] 位移因數欄 1262B（注意，位移欄 1262A 的直接並列於位移因數欄 1262B 上表示一或另一個被使用）-其內容係被使用作為位址產生的一部份；其指明一位移因數，其予以縮放記憶體存取的大小（N）-其中 N 為在記憶體存取中的位元組數量（例如，對於使用 $2^{\text{縮放}} * \text{索引} + \text{基礎} + \text{縮放位移的位址產生}$ ）。冗餘低階位元被忽略，因此，位移因數欄的內容被乘以記憶體運算元總尺寸（N），以產生被用以計算有效位址的最終位移。N 的值被處理器硬體根據全運算碼欄 1274（於此所述）及資料

調處欄 1254C 的執行時間加以決定。位移欄 1262A 與位移因數欄 1262B 為選用，在意義上，它們並未被用於無記憶體存取 1205 指令樣板及/或不同實施例也可以只實施兩者之一或都不實施。

[0124] 資料元件寬度欄 1264-其內容區分若干資料元件寬度中之哪一個將被使用（在一些實施例中，全部指令；在其他實施例中，只有部份的指令）。此欄為選用，其意義在於如果只有一資料元件寬度被支援及/或使用部份態樣的運算碼支援資料元件寬度，則其並不需要。

[0125] 寫入遮罩欄 1270-其內容以每資料元件位置為基礎地控制在目的向量運算元中的資料元件位置是否反映基礎運算與擴增運算的結果。類型 A 指令樣板支援合併寫入遮罩，而類型 B 指令樣板支援合併與歸零寫入遮罩。當合併時，向量遮罩允許在目的中的任何組元件在任何（為基礎運算與擴增運算所指明的）運算被執行期間被保護不受到更新；在另一實施例中，保留目的的各個元件的舊值，其中對應遮罩位元具有 0。相反地，當歸零向量遮罩時，允許任何在目的的任意組元件在任何（為基礎運算與擴增運算所指明的）運算被執行期間被歸零；在一實施例中，當對應遮罩位元具有 0 值時，目的的元件被設定為 0。此功能的一次組為控制被執行的運算的向量長度（即，被修改元件的第一元件至最後元件的元件間隔）的能力；然而，並不必要這些元件被修改為連續。因此，寫入遮罩欄 1270 允許部份向量運算，包含載入、儲存、算

術、邏輯等等。雖然本發明的實施例係被描述，其中寫入遮罩欄 1270 內容選擇包含予以使用的寫入遮罩的若干寫入遮罩暫存器之一（因此，寫入遮罩欄 1270 的內容間接指明予以執行的遮罩），而其他實施例或者或另外地允許遮罩寫入欄 1270 的內容被直接指明予以執行的遮罩。

[0126] 立即欄 1272-其內容允許立即值的規格。此欄係為選用的，其意義在於其並不會出現在同屬向量友善格式的實施方式中，其並不支援立即值並且也不會出現在不使用立即值的指令中。

[0127] 類型欄 1268-其內容區分不同類型的指令。參考圖 12A-B，此欄的內容於類型 A 與類型 B 指令間作選擇。在圖 12A-B 中，圓角方塊係被使用以指明一特定值出現在一欄中（例如，用於類型欄 1268 的類型 A 1268A 與類型 B 1268B 分別在圖 12A-B 中）。

類型 A 的指令樣板

[0128] 於類型 A 的非記憶體存取 1205 指令樣板中， α 欄 1252 係被解譯為 RS 欄 1252A，其內容區分不同擴增運算類型（例如，捨入 1252A.1 與資料轉換 1252A.2 係分別特定用於無記憶體存取，捨入類型 1210 與無記憶體存取，資料轉換類型運算 1215 指令樣板）的哪一個將被執行，而 β 欄 1254 區分所指定類型的哪一運算將被執行。在無記憶體存取 1205 指令樣板中，並未出現縮放欄 1260、位移欄 1262A、與位移縮放欄 1262B。

無記憶體存取指令樣板-全捨入控制類型運算

[0129] 在無記憶體存取全捨入控制類型運算 1210 指令樣板中， β 欄 1254 被解譯為捨入控制欄 1254A，其內容提供靜態捨入。雖然在本發明所述實施例中，捨入控制欄 1254A 包含抑制全浮點例外（SAE）欄 1256 與捨入運算控制欄 1258，但其他實施例也可以支援，可以將這些概念編碼至相同欄或只有這些概念/欄之一或另一（例如，可以只有捨入運算控制欄 1258）。

[0130] SAE 欄 1256-其內容區分是否去能例外事件報告；當 SAE 欄 1256 的內容指示抑制被致能時，給定指令並不報告任何類型的浮點例外旗標並且不會提出任何浮點例外處置器。

[0131] 捨入運算控制欄 1258-其內容區分一群捨入運算中的哪一個被執行（例如，捨進、捨去、捨向零及捨至最近值）。因此，捨入運算控制欄 1258 允許每一指令為主地改變捨入模式。本發明的一實施例中，其中處理器包含控制暫存器，用以指明捨入模式，捨入運算控制欄 1250 內容置換該暫存器值。

無記憶體存取指令樣板-資料轉換類型運算

[0132] 在無記憶體存取資料轉換類型運算 1215 指令樣板中， β 欄 1254 被解譯為資料轉換欄 1254B，其內容區分若干資料轉換中的哪一個將被執行（例如，無資料轉

換、拌和（swizzle）、廣播）。

[0132] 於類型 A 的記憶體存取 1220 指令樣板中， α 欄 1252 係被解譯為驅逐暗示欄 1252B，其內容區分該等驅逐暗示中的哪一個被使用（在圖 12A 中，時間欄 1252B.1 與非時間欄 1252B.2 分別指明用於記憶體存取時間 1225 指令樣板與記憶體存取非時間 1230 指令樣板），同時， β 欄 1254 係被解譯為資料調處欄 1254C，其內容區分若干資料調處運算中之哪一個（也稱為基元）被執行（例如，無調處；廣播；源的上轉換；及目的的下轉換）。記憶體存取 1220 指令樣板包含縮放欄 1260，及選用地位移欄 1262A 或位移縮放欄 1262B。

[0133] 向量記憶體指令以轉換支援執行來自記憶體的向量載入與向量儲存至記憶體。如同一般向量指令，向量記憶體指令以按資料元件的方式傳送資料進/出記憶體，實際被傳送的元件係為被選擇為寫入遮罩的向量遮罩的內容所表示。

記憶體存取指令樣板-時間

[0134] 時間資料為似乎將被足夠快再次使用，以由快取獲利的資料。然而，這是一暗示，並且，不同處理器可以將之以不同方式實施，包含整個忽略該暗示。

記憶體存取指令樣板-非時間

[0135] 非時間資料為資料將不會被足夠快再次使用

以由第一層快取進行快取而獲利並應被給定驅逐用的優先權。然而，這是一暗示，並且，不同處理器可以將之以不同方式實施，包含整個忽略該暗示。

類型 B 的指令樣板

[0136] 在類型 B 的指令樣板中， α 欄 1252 被解譯為寫入遮罩控制（Z）欄 1252C，其內容區分是否為寫入遮罩欄 1270 所控制的寫入遮罩應被合併或歸零。

[0137] 在類型 B 的非記憶體存取 1205 指令樣板中， β 欄 1254 的部份係被解譯為 RL 欄 1257A，其內容區分不同擴增運算類型（例如，捨入 1257A.1 及向量長度（VSIZE）1257A.2 分別指明用於無記憶體存取寫入遮罩控制，部份捨入控制類型運算 1212 指令樣板與無記憶體存取寫入遮罩控制 VSIZE 類型運算 1217 指令樣板）的哪一類型被執行，而 β 欄 1254 的其他部份則區分指定類型的運算的哪一個將被執行。在無記憶體存取 1205 指令樣板中，並未出現縮放欄 1260、位移欄 1262A、及位移縮放欄 1262B。

[0138] 在無記憶體存取寫入遮罩控制部份捨入控制類型運算 1210 指令樣板中， β 欄 1254 的剩餘部份被解譯為捨入運算欄 1259A 及例外事件報告被去能（給定指令並未報告任何類型的浮點例外旗標並且未提出任何浮點例外處置器）。

[0139] 捨入運算控制欄 1259A-如同捨入運算控制欄

1258，其內容區分一群捨入運算中的哪一個被執行（例如，捨進、捨去、捨向零、及捨至最近值）。因此，捨入運算控制欄 1259A 允許以每指令為基礎地改變捨入模式。在本發明之處理器包含指明捨入模式的控制暫存器的一實施例中，捨入運算控制欄 1250 的內容置換該暫存器值。

[0140] 在無記憶體存取寫入遮罩控制 VSIZE 類型運算 1217 指令樣板中， β 欄 1254 的剩餘部份被解譯為向量長度欄 1259B，其內容區分若干資料向量長度中的哪一個將被執行（例如，128、256、或 512 位元組）。

[0141] 在類型 B 的記憶體存取 1220 指令樣板中， β 欄 1254 的部份被解譯為廣播欄 1257B，其內容區分是否將執行廣播類型資料調處運算，而 β 欄 1254 的剩餘部份被解譯為向量長度欄 1259B。記憶體存取 1220 指令樣板包含縮放欄 1260，及選用地位移欄 1262A 或位移縮放欄 1262B。

[0142] 有關同屬向量友善指令格式 1200，全運算碼欄 1274 被顯示為包含格式欄 1240、基礎運算欄 1242、及資料元件寬度欄 1264。雖然一實施例係被顯示，其中全運算碼欄 1274 包含所有這些欄，但全運算碼欄 1274 也可以在包含較並不支援全部欄的實施例中的較所有欄為少的欄。全運算碼欄 1274 提供運算碼（opcode）。

[0143] 擴增運算欄 1250、資料元件寬度欄 1264、及寫入遮罩欄 1270 允許這些特性在同屬向量友善指令格式中，以每指令為基礎地被指明。

[0144] 寫入遮罩欄與資料元件寬度欄的組合建立分類的指令，其允許該遮罩根據不同資料元件寬度加以應用。

[0145] 在類型 A 與類型 B 內找到的各種指令樣板係有利於不同狀況。在本發明之一些實施例中，不同處理器或在處理器內的不同核心可以只支援類型 A、或類型 B、或兩者都支援。例如，用於通用計算的高效通用無序核心可以只支援類型 B，主要用於圖形及/或科學（通量）計算的核心可以只支援類型 A，及用於兩者的核心可以支援兩者（當然，一核心可以具有來自兩類型的樣板與指令的部份混合，但並不是所有來自兩類型的樣板與指令都在本發明的範圍內）。同時，單一處理器可以包含多數核心，所有這些核心支援相同類型或其中不同核心支援不同類型。例如，在具有分開圖形與通用核心的處理器中，主要想要用於圖形及/或科學計算的圖形核心之一可以只支援類型 A，而一或更多通用核心可以為高效通用核心，其具有無序執行及暫存器重命名，用於只支援類型 B 的通用計算。沒有分開圖形核心的另一處理器可以包含一或更多通用有序或無序核心，其支援類型 A 與類型 B 兩者。當然，在本發明的不同實施例中，來自一類型的特性也可以實施在其他類型中。以高階語言撰寫的程式將（例如，以時間編譯或靜態編譯）置放為各種不同可執行形式，包含：1）一形式，其只具有用於執行的目標處理器所支援的類型的指令；或 2）一形式，其具有使用所有類型的指

令的不同組合所寫的替代常式並具有控制流程碼，其根據現正執行該碼的處理器所支援的指令選擇要執行之常式。

[0146] 圖 13A-D 為方塊圖，例示依據本發明實施例之例示特定向量友善指令格式。圖 13 顯示特定向量友善指令格式 1300，其係特定於在其指明欄的位置、大小、解譯、及順序，以及，這些欄的一部份的值。特定向量友善指令格式 1300 可以用以擴充 x86 指令集，因此，部份這些欄係類似或相同於在現行 x86 指令集與其擴充（例如，AVX）所用的欄。此格式保持相容於現行 x86 指令集與擴充的前置編碼欄、真運算碼位元組欄、MOD R/M 欄、SIB 欄、位移欄、及立即欄。來自圖 12 的欄中也顯示有來自圖 13 的欄。

[0147] 應了解的是，雖然本發明的實施例係參考以例示目的之同屬向量友善指令格式 1200 的脈絡中的特定向量友善指令格式 1300 加以描述，但本發明並不限於該特定向量指令格式 1300，除非特別主張之外。例如，同屬向量友善指令格式 1200 想出用於各欄的各種可能大小，而特定向量友善指令格式 1300 係被顯示以具有特定大小的欄。例如特定例子中，雖然資料元件寬度欄 1264 被顯示為在特定向量友善指令格式 1300 中的一位元欄，但本發明並不作此限定（即，同屬向量友善指令格式 1200 想要其他大小的資料元件寬度欄 1264）。

[0148] 同屬向量友善指令格式 1200 包含以下以圖 13A 所示之順序列出的以下欄。

EVEX 前置（位元組 0-3）1302-係被編碼為四位元組格式。

[0149] 格式欄 1240（EVEX 位元組 0，位元[7：0]）-第一位元組（EVEX 位元組 0）為格式欄 1240 及其包含 0x62（用於區分在本發明一實施例中的向量友善指令格式的特有值）。

[0150] 第二至第四位元組（EVEX 位元組 1-3）包含提供特定能力的若干位元欄。

[0151] REX 欄 1305（EVEX 位元組 1，位元[7-5]）-由 EVEX.R 位元欄（EVEX 位元組 1，位元[7] -R）、EVEX.X 位元欄（EVEX 位元組 1，位元[6] -X）、及 1257BEX 位元組 1，位元[5] -B）構成。EVEX.R、EVEX.X 與 EVEX.B 位元欄提供與對應 VEX 位元欄相同的功能，並且，使用 1 的補數格式加以編碼，即 ZMM0 被編碼為 1211B、ZMM15 被編碼為 0000B。如同於本技藝所知，指令的其他欄編碼暫存器索引的較低三個位元（rrr，xxx 與 bbb），使得 Rrrr、Xxxx、及 Bbbb 可以藉由加入 EVEX.R、EVEX.X 與 EVEX.B 加以形成。

[0152] REX'欄 1210—此為 REX'欄 1210 的第一部份並為 EVEX.R'的位元欄（EVEX 位元組 1，位元[4] -R'），其係被使用以編碼擴充 32 暫存器組的上 16 或下 16 個。在本發明的一實施例中，此位元以及以下所表示的其他位元係被儲存為位元反轉格式，以區分（已知 x86 的 32 位元模式）BOUND 指令，其真運算碼位元組為

62，但在 MOD R/M 欄（下述）中並不接受在 MOD 欄中的 11 的值；本發明的其他實施例並不儲存此及反轉格式的以下的其他表示位元。1 的值係被用以編碼下 16 個暫存器。換句話說，R'Rrrr 係藉由組合 EVEX.R'、EVEX.R 與來自其他欄的其他 RRR 加以形成。

[0153] 運算碼圖欄 1315（EVEX 位元組 1，位元[3：0] -mmmm）－其內容編碼暗示領前運算碼位元組（0F、0F38 或 0F3）。

[0154] 資料元件寬度欄 1264（EVEX 位元組 2，位元[7] -W）-以標記 EVEX.W 表示。EVEX.W 被用以定義資料類型的粒度（大小）（32 位元資料元件或 64 位元資料元件）。

[0155] EVEX.vvvv1320（EVEX 位元組 2，位元[6：3] -vvvv）-EVEX.vvvv 的角色可以包含如下：1）EVEX.vvvv 編碼以反轉（1 的補數）形式指明的第一源暫存器運算元，並對具有 2 或更多源運算的指令有效；2）EVEX.vvvv 編碼為某些向量位移的 1 的補數所指明的目的暫存器運算元；或 3）EVEX.vvvv 並不編碼任何運算元，該欄係被保留並應包含 1211b。因此，EVEX.vvvv 欄 1320 編碼儲存為反轉（1 的補數）格式的第一源暫存器說明符（specifier）的 4 個低階位元。取決於指令，額外不同 EVEX 位元欄也可以使用以擴充說明符的大小至 32 暫存器。

[0156] EVEX.U 1268 類型欄（EVEX 位元組 2，位元

[2] -U) — 如果 EVEX.U=0，它表示類型 A 或 EVEX.U0；如果 EVEX.U=1，它表示類型 B 或 EVEX.U1。

[0157] 前置編碼欄 1325 (EVEX 位元組 2，位元[1:0] -pp) — 提供額外位元給基礎運算欄。除了提供 EVEX 前置格式的傳統 SSE 指令的支援外，此也有利於緊縮 SIMD 前置（而不必一位元組來表達 SIMD 前置，EVEX 前置只需 2 位元）。在一實施例中，為了支援使用傳統格式及 EVEX 前置格式，的 SIMD 前置（66H，F2H，F3H）的傳統 SSE 指令，這些傳統 SIMD 前置係被編碼入 SIMD 前置編碼欄；及在被提供給解碼器的 PLA（使得 PLA 可以執行這些傳統指令的傳統與 EVEX 格式，不必修改）之前，這些傳統 SIMD 前置在執行時間被擴充入傳統 SIMD 前置。雖然較新的指令可能直接使用 EVEX 前置編碼欄的內容作為運算碼擴充，但某些實施例中，為了相容性可能以類似方式擴充，但允許這些傳統 SIMD 前置指明不同意義。其他實施例也可以重新設計 PLA 以支援 2 位元 SIMD 前置編碼，因而不需要擴充。

[0158] α 欄 1252 (EVEX 位元組 3，位元[7] -EH；同時也稱為 EVEX.EH、EVEX.rs、EVEX.RL、EVEX.寫入遮罩控制，及 EVEX.N；同時也被顯示為 α) — 如前述，此欄為特定上下文。

[0159] β 欄 1254 (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；也被顯示為 $\beta \beta \beta$) — 如前所

述，此欄為上下文特定。

[0160] REX'欄 1210 – 此為 REX'欄的剩餘並為 EVEX.V'位元欄（EVEX 位元組 3，位元[3] -V'），其係可以用以編碼擴充 32 暫存器組的上面 16 個或下面 16 個。此位元係儲存為位元反轉格式。1 的值係被用以編碼下面 16 個暫存器。換句話說，V'VVVV 係由組合 EVEX.V'與 EVEX.vvvv 所形成。

[0161] 寫入遮罩欄 1270（EVEX 位元組 3，位元[2：0] -kkk）– 其內容指明在如前所述中之寫入遮罩暫存器中的暫存器的索引。在本發明之一實施例中，特定值 EVEX.kkk=000 具有暗示無寫入遮罩被用於該特定指令的特殊行為（這可以以各種方式加以實施，包含使用寫入遮罩硬體接線至所有硬體或旁路遮罩硬體的硬體）。

[0162] 真運算碼欄 1330（位元組 4）也稱為運算碼位元組。運算碼的部份係被指明於此欄中。

[0163] MOD R/M 欄 1340（位元組 5）包含 MOD 欄 1342、Reg 欄 1344、及 R/M 欄 1346。如前所述，MOD 欄 1342 的內容區分記憶體存取與非記憶體存取運算。Reg 欄 1344 的角色可以被總結為兩種情況：編碼目的暫存器運算元或源暫存器運算元，或被處理為運算碼擴充並不被用以編碼任何指令運算元。R/M 欄 1346 的角色可以包含如下：編碼參考記憶體位址的指令運算元，或編碼目的暫存器運算元或源暫存器運算元。

[0164] 縮放索引基礎（SIB）位元組（位元組 6）–

如前所述，縮放欄 1250 的內容被用以記憶體位址產生。SIB.xxx1354 與 SIB.bbb 1356-這些欄的內容已經先前參考暫存器索引 Xxxx 與 Bbbb 加以表示。

[0165] 位移欄 1262A (位元組 7-10) — 當 MOD 欄 1342 包含 10 時，位元組 7-10 為位移欄 1262A，其與傳統 32 位元位移 (disp32) 一樣地動作並以位元組粒度加以動作。

[0166] 位移因數欄 1262B (位元組 7) — 當 MOD 欄 1342 包含 01 時，位元組 7 為位移因數欄 1262B。此欄的位置係與傳統 x86 指令集的 8 位元位移 (disp8) 相同，並以位元組粒度動作。因為 disp8 為帶符號擴充，所以，其也可以只定址於 -128 及 127 位元組偏移間；以 64 位元組快取線看來，disp8 使用可以只被設定至四個真實有用值 -128、-64、0、及 64 的 8 位元；因為經常需要更大範圍，所以，disp32 被使用；然而，disp32 需要 4 位元組。不同於 disp8 與 disp32，位移因數欄 1262B 係為 disp8 的重新解釋；當使用位移因數欄 1262B 時，實際位移係為位移因數欄乘以記憶體運算元存取的大小 (N) 的內容所決定。此類型位移被稱為 $\text{disp8} * N$ 。這降低平均指令長度 (單一位元組用於位移，但具有大許多的範圍)。此壓縮位址係根據有效位移係為記憶體存取的粒度的倍數的假設，因此，位址偏移的冗餘低階位元並不需要被編碼。換句話說，位移因數欄 1262B 替代傳統 x86 指令集 8 位元位移。因此，位移因數欄 1262B 被以與 x86 指令集 8 位元位

移（因此，在 ModRM/SIB 編碼規則中沒有改變），以只有 disp8 過載至 disp8*N 的例外的相同方式編碼。換句話說，在編碼規則或編碼長度上並沒有改變，只有在以硬體解釋位移值時有改變（其需要以記憶體運算元的大小來縮放該位移，以取得按位元組位址偏移）。

[0167] 立即欄 1272 如同前述操作。

全運算碼欄

[0168] 圖 13B 為方塊圖，例示依據本發明實施例之作成全運算碼欄 1274 的特定向量友善指令格式 1300 的欄位。明確地說，全運算碼欄 1274 包含格式欄 1240、基礎運算欄 1242、及資料元件寬度（W）欄 1264。基礎運算欄 1242 包含前置編碼欄 1325、運算碼圖欄 1315 及真運算碼欄 1330。

暫存器索引欄

[0169] 圖 13C 為方塊圖，例示依據本發明實施例之作成暫存器索引欄 1244 的特定向量友善指令格式 1300 的欄位。明確地說，暫存器索引欄 1244 包含 REX 欄 1305、REX' 欄 1310、MODR/M.reg 欄 1344、MODR/M.r/m 欄 1346、VVVV 欄 1320、xxx 欄 1354、及 bbb 欄 1356。

擴增運算欄

[0170] 圖 13D 為方塊圖，例示依據本發明實施例之

作成擴增運算欄 1250 的特定向量友善指令格式 1300 的欄位。當類型 (U) 欄 1268 包含 0 時，其表示 EVEX.U0 (類型 A 1268A)；當其包含 1 時，其表示 EVEX.U1 (類型 B 1268B)。當 U=0 與 MOD 欄 1342 包含 11 時 (表示無記憶體存取運算)，則 α 欄 1252 (EVEX 位元組 3，位元[7] -EH) 被解譯為 rs 欄 1252A。當 rs 欄 1252A 包含 1 (捨入 1252A.1)， β 欄 1254 (EVEX 位元組 3，位元[6:4] -SSS) 係被解譯為捨入控制欄 1254A。捨入控制欄 1254A 包含一位元 SAE 欄 1256 及兩位元捨入運算欄 1258。當 rs 欄 1252A 包含 0 (資料轉換 1252A.2) 時， β 欄 1254 (EVEX 位元組 3，位元[6:4] -SSS) 係被解譯為三位元資料轉換欄 1254B。當 U=0 及 MOD 欄 1342 包含 00，01 或 10 時 (表示記憶體存取運算)，則 α 欄 1252 (EVEX 位元組 3，位元[7] -EH) 被解譯為驅逐暗示 (EH) 欄 1252B 及 β 欄 1254 (EVEX 位元組 3，位元[6:4] -SSS) 係被解譯為三位元資料調處欄 1254C。

[0171] 當 U=1 時， α 欄 1252 (EVEX 位元組 3，位元[7] -EH) 係被解譯為寫入遮罩控制 (Z) 欄 1252C。當 U=1 及 MOD 欄 1342 包含 11 (表示無記憶體存取運算)，則 β 欄 1254 的部份 (EVEX 位元組 3，位元[4] -S₀) 係被解譯為 RL 欄 1257A；當其包含 1 (捨入 1257A.1) 時， β 欄 1254 (EVEX 位元組 3，位元[6-5] -S₂₋₁) 的剩餘部份被解譯為捨入運算欄 1259A，而當 RL 欄 1257A 包含 0 (VSIZE 1257.A2) 時， β 欄 1254 (EVEX

位元組 3，位元[6-5] -S₂₋₁) 的剩餘部份被解譯為向量長度欄 1259B (EVEX 位元組 3，位元[6-5] -L₁₋₀)。當 U=1 及 MOD 欄 1342 包含 00，01 或 10 (表示記憶體存取運算) 時，β 欄 1254 (EVEX 位元組 3，位元[6：4] -SSS) 係被解譯為向量長度欄 1259B (EVEX 位元組 3，位元[6-5] -L₁₋₀) 及廣播欄 1257B (EVEX 位元組 3，位元[4] -B)。

[0172] 圖 14 為依據本發明一實施例之暫存器架構 1400 的方塊圖。在所示實施例中，其中有 512 位元寬的 32 個向量暫存器 1410，這些暫存器係被稱為 zmm0 至 zmm31。下 16 個 zmm 暫存器的下階 256 位元被疊於暫存器 ymm0-16 上。下 16 個 zmm 暫存器的下階 128 位元 (ymm 暫存器的下階 128 位元) 係被重疊於暫存器 xmm0-15 上。這些特定向量友善指令格式 1300 運算於這些重疊暫存器檔，如下表所示。

可調向量長度	類型	運算	暫存器
不包含向量長度欄 1259B 的指令樣板	A (圖 12A；U=0)	1210，1215， 1225，1230	zmm 暫存器 (向量長度為 64 位元組)
	B (圖 12B；U=1)	1212	zmm 暫存器 (向量長度為 64 位元組)
包含向量長度欄 1259B 的指令樣板	B (圖 12B；U=1)	1217，1227	取決於向量長度欄 1259B 之 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)

[0173] 換句話說，向量長度欄 1259B 於最大長度與一或更多其他較短長度間作較選擇，其中各個此較短長度為前一長度的長度之一半；及沒有向量長度欄 1259B 的指令樣板運算於最大向量長度。再者，在一實施例中，特定向量友善指令格式 1300 的類型 B 指令樣板運算於緊縮或純量單/雙倍精度浮點資料及緊縮或純量整數資料。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係被保持與指令之前或歸零之前相同。

[0174] 寫入遮罩暫存器 1415— 在所示實施例中，有 8 個寫入遮罩暫存器（`k0` 至 `k7`），各個大小為 64 位元。在另一實施例中，寫入遮罩暫存器 1415 的大小為 16 位元。如前所述，在本發明之一實施例中，向量遮罩暫存器 `k0` 不能被使用作為寫入遮罩；當通常表示 `k0` 的編碼被使用為寫入遮罩時，其選擇 `0xFFFF` 的硬體接線寫入遮罩，有效地去能用於該指令的寫入遮罩。

[0175] 通用暫存器 1425— 在所示的實施例中，有十六個 64 位元通用暫存器，其被與現行 `x86` 定址模式一起使用以定址記憶體運算元。這些暫存器係以名稱 `RAX`、`RBX`、`RCX`、`RDX`、`RBP`、`RSI`、`RDI`、`RSP`、及 `R8` 至 `R15` 表示。

[0176] 純量浮點堆疊暫存器檔（`x87` 堆疊）1445，其上假化 `MMX` 緊縮整數平暫存器檔 1450— 在所示實施例

中，x87 堆疊為八元件堆疊，用以使用 x87 指令集擴充對 32/64/80 位元浮點資料，執行純量浮點運算；同時，MMX 暫存器係被使用以對 64 位元緊縮整數資料執行運算，及對執行於 MMX 與 XMM 暫存器間的一些運算保持運算元。

[0177] 本發明之其他實施例中也可以使用較寬或較窄暫存器。另外，本發明之其他實施例也可以使用更多、更少或不同暫存器檔與暫存器。

[0178] 圖 15A-B 例示更特定例示有序核心架構的方塊圖，該核心可以為在晶片中的（包含相同類型及/或不同類型的其他核心的）幾個邏輯方塊之一。取決於應用而定，邏輯方塊透過高頻寬互連網路（例如，環狀網路）與一些固定功能邏輯、記憶體 I/O 介面、及其他必要 I/O 邏輯通訊。

[0179] 圖 15A 為依據本發明實施例之單一處理器核心的方塊圖，其連接至晶粒上互連網路 1502 及該第二層（L2）快取 1504 的本地次組。在一實施例中，指令解碼器 1500 支援具有緊縮資料指令集擴充的 x86 指令集。L1 快取 1506 允許低潛候存取該快取記憶體為該純量與向量單元。雖然在一實施例中（以簡化設計），純量單元 1508 與向量單元 1510 使用分開暫存器組（分別為純量暫存器 1512 與向量暫存器 1514）及轉換於其間的資料係被寫入記憶體，然後被由第一層（L1）快取 1506 讀回，本發明之其他實施例可以使用不同方法（例如，使用單一暫

存器集或包含一通訊路徑，其允許資料被傳送於兩暫存器檔之間，而不必被寫與讀回）。

[0180] L2 快取 1504 的本地次組係為全局 L2 快取的一部份，其被分割為分開的本地次組，每一處理器核心一本地次組。各個處理器核心具有至 L2 快取 1504 的本身本地次組的直接存取路徑。為處理器核心所讀取的資料係被儲存於其 L2 快取次組 1504 並可以並列於其他處理器核心存取其本身本地 L2 快取次組作快速地存取。為處理器核心所寫入的資料係被儲存於其本身 L2 快取次組 1504 並如果有必要時，被由其他次組清除（flush）。環狀網路確保共享資料的連貫性。環狀網路係為雙向的並允許例如處理器核心的代理、L2 快取及其他邏輯方塊彼此在該晶片內通訊。各個環狀資料路徑每方向為 1012 位元寬。

[0181] 圖 15B 為依據本發明實施例之圖 15A 中的處理器核心的部份放大圖。圖 15B 包含 L1 快取 1504 的 L1 資料快取 1506A 部，及有關向量單元 1510 與向量暫存器 1514 的更多細節。明確地說，向量單元 1510 為 16 寬向量處理單元（VPU）（見 16 寬 ALU 1528），其執行整數、單精度浮點及雙倍精度浮指令之一或多者。VPU 支援與拌和單元 1520 拌和暫存器輸入，以數值轉換單元 1522A-B 的數值轉換、及在記憶體輸入上以複製單元 1524 的複製。寫入遮罩暫存器 1526 允許預測結果向量寫入。

[0182] 本發明實施例也可以包含各種已經說明過的步驟。該等步驟也可以實施為機器可執行指令，其可以被

用以使通用或特殊用途處理器執行這些步驟。或者，這些步驟也可以為特定硬體組件所執行，該等組件包含用以執行這些步驟的硬體接線邏輯，或者，已程式的電腦組件與客製硬體組件的任意組合。

[0183] 如此所述，指令可以被表示為例如特定應用積體電路（ASIC）的硬體的特定架構，其被組態以執行某些運算或具有預定功能或儲存被實施於非暫態電腦可讀取媒體中的記憶體中的軟體指令。因此，於圖中所示的技術可以使用儲存於並被執行於一或更多電子裝置（例如，端站、網路元件等）的碼與資料而被實施。此等電子裝置使用電腦機器可讀取媒體儲存及（內部地及/或透過網路與其他電子裝置）互通碼與資料，機器可讀取媒體係例如非暫態電腦機器可讀取儲存媒體（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）及暫態電腦機器可讀取通訊媒體（例如，電、光、聲或其他形式的傳遞信號－例如載波、紅外線信號、數位信號等）。另外，此等電子裝置典型包含一組一或更多處理器，耦接至一或更多其他組件，例如，一或更多儲存裝置（非暫態機器可讀取儲存媒體）、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。該組處理器與其他組件的耦接典型係透過一或更多匯流排與橋接器（也稱為匯流排控制器）。儲存裝置與承載網路話務的信號分別代表一或更多機器可讀取儲存媒體及機器可讀取通訊媒體。因此，給定電子裝置的儲存裝置典型儲

存碼及/或資料，用以被執行於該電子裝置的該組一或更多處理器上。當然，本發明之實施例的一或更多部份可以使用軟體、韌體與/或硬體的不同組合加以實施。在此詳細說明中，為了解釋為目的，更多特定細節係被說明，以提供對本發明的完整了解。然而，對於本技藝者明顯地，本發明可以在沒有這些特定細節的一部份下加以實施。在某些例子中，已知結構與功能並未被特別詳細描述，以免模糊了本發明的標的。因此，本發明的範圍與精神係由以下的申請專利範圍所判斷。

[0184] 本發明之實施例包含具有一處理器，其包含：擷取邏輯，用以自記憶體擷取表示目的緊縮資料運算元、第一源緊縮資料運算元、第二源緊縮資料運算元及立即值的指令；及執行邏輯，以根據對應於立即值的位置，決定在第二源緊縮資料運算元中的位元；執行於第一源緊縮資料運算元與該所決定位元間的按位元和運算，以產生中間結果；執行目的緊縮資料運算元與該中間結果間的按位元互斥或運算，以產生最終結果；及儲存最終結果於目的緊縮資料運算元所表示的儲存位置。

[0185] 另一實施例包含，其中為了執行於第一源緊縮資料運算元與所決定位元間的按位元和運算，該執行邏輯進一步被組態以執行於第一源緊縮資料運算元與暫時向量間的按位元和運算，其中所決定的位元的值係予以被一或更多次廣播至該暫時向量。

[0186] 另一實施例包含，其中為目的緊縮資料運算

元、第一源緊縮資料運算元與第二源緊縮資料運算元所表示的儲存位置係予以在分開的 64 位元部中處理，其中該處理器係予以對各個該 64 位元部執行相同邏輯。

[0187] 另一實施例包含，其中該指令更包含寫入遮罩運算元，及其中該執行邏輯用以更進一步反應於決定該寫入遮罩運算元表示寫入遮罩係設定用於目的緊縮資料運算元的該等 64 位元部之一，而設定為目的緊縮資料運算元所表示的儲存位置中的 64 位元部中之一的值為零。

[0188] 另一實施例包含，其中為目的緊縮資料運算元、第一源緊縮資料運算元、及第二源緊縮資料運算元所表示的儲存位置係為暫存器及記憶體位置之至少一者。

[0189] 另一實施例包含，其中為目的緊縮資料運算元、第一源緊縮資料運算元、及第二源緊縮資料運算元所表示的儲存位置係為 512 位元長的暫存器。

[0190] 另一實施例包含，其中立即值為 8 位元長。

[0191] 另一實施例包含，其中該指令被使用以執行於位元矩陣與位元向量間的位元矩陣乘法運算，其中該位元矩陣的一或更多行係被儲存在為第一源緊縮資料運算元所表示的儲存位置，及其中該位元向量的值係被儲存在第二源緊縮資料運算元所表示的儲存位置中。

[0192] 本發明的實施例包含，其中該位元矩陣被轉置，使得該位元矩陣的一或更多行係被逐行地儲存在第一源緊縮資料運算元所表示的儲存位置中。

[0193] 另一實施例包含，其中為目的緊縮資料運算

元所表示的儲存位置包含當指令被執行為用於該位元矩陣的各個行時的該位元矩陣與該位元向量間的位元矩陣乘法運算的結果，其中各個指令的執行，該立即值指明對應於該位元矩陣的被處理的行數的該位元向量中的位置的值。

[0194] 本發明實施例包含電腦處理器中之方法，其包含：自記憶體中擷取表示目的緊縮資料運算元、第一源緊縮資料運算元、第二源緊縮資料運算元、及立即值的指令；根據對應於立即值的位置，決定第二源緊縮資料運算元中的位元；執行於第一源緊縮資料運算元與所決定位元間的按位元和運算，以產生中間結果；執行於目的緊縮資料運算元與中間結果間的按位元互斥或運算，以產生最終結果；及儲存最終結果於目的緊縮資料運算元所表示的儲存位置中。

[0195] 另外實施例包含，其中於第一源緊縮資料運算元與所決定位元間執行按位元和運算更包含執行於第一源緊縮資料運算元與暫時向量間的按位元和運算，其中該所決定位元的值係予以被一次或多次廣播至該暫時向量。

[0196] 另外實施例包含，其中為目的緊縮資料運算元、第一源緊縮資料運算元、及第二源緊縮資料運算元所表示的儲存位置係被以分開的 64 位元部處理，其中該處理器對於各個 64 位元部執行相同邏輯。

[0197] 另外實施例包含，其中該指令更包含寫入遮罩運算元，及其中該方法更包含回應於決定該寫入遮罩運算元表示寫入遮罩係被設定用於目的緊縮資料運算元中的

64 位元部之一，設定為目的緊縮資料運算元所表示的儲存位置中的該 64 位元部之一的值為零。

[0198] 另一實施例包含，其中為目的緊縮資料運算元、第一源緊縮資料運算元、及第二源緊縮資料運算元所表示的儲存位置係為暫存器與記憶體位置之至少之一。

[0199] 另一實施例包含，其中為目的緊縮資料運算元、第一源緊縮資料運算元、及第二源緊縮資料運算元所表示的儲存位置係為 512 位元長的暫存器。

[0200] 另一實施例包含，其中立即值為 8 位元長。

[0201] 另一實施例包含，其中指令被用以執行於位元矩陣與位元向量間的位元矩陣乘法運算，其中該位元矩陣的一或更多行係被儲存於為第一源緊縮資料運算元所表示的儲存位置中，及其中該位元向量的值係被儲存在第二源緊縮資料運算元所表示的儲存位置中。

[0202] 本發明的一實施例包含，其中該位元矩陣係被轉置，使得該位元矩陣的一或更多行係逐行地儲存在第一源緊縮資料運算元所表示的儲存位置。

[0203] 另一實施例包含，其中為目的緊縮資料運算元所表示的儲存位置包含當該指令被執行用於該位元矩陣的各個行時，該位元矩陣與該位元向量間的位元矩陣乘法運算的結果，其中對於指令的各個執行，該立即值指明對應於該位元矩陣中被處理的行數的在該位元向量中的位置之值。

[0204] 雖然本發明已經以幾個實施例加以描述，熟

習於本技藝者將認知本發明將不限於所述之實施例，也可以實施為在隨附申請專利範圍的精神與範圍內的修改與替換。因此，本說明係視為例示性而非限定性。

【符號說明】

[0205]

100：處理器管線

102：擷取級

104：長度解碼級

106：解碼級

108：配置級

110：重命名級

112：排序級

114：暫存器讀取/記憶體讀取級

116：執行級

118：寫回/記憶體寫入級

122：例外處理級

124：確定級

130：前端單元

131：指令擷取

132：分支預測單元

134：指令快取單元

136：轉譯旁看緩衝器

140：解碼單元

- 150：執行引擎單元
- 151：實體暫存器檔案單元
- 152：重命名/配置器單元
- 154：退役單元
- 156：排序單元
- 160：執行叢集
- 162：執行單元
- 164：記憶體存取單元
- 170：記憶體單元
- 172：資料 TLB 單元
- 174：資料快取單元
- 176：L2 快取單元
- 190：核心
- 200：處理器
- 202A-N：核心
- 204A-N：快取單元
- 206：共享快取單元
- 208：特殊用途邏輯
- 210：系統代理單元
- 212：環為主互連單元
- 214：積體記憶體控制器單元
- 216：匯流排控制器單元
- 300：系統
- 310：處理器

- 315：處理器
- 320：控制器集線器
- 340：記憶體
- 345：共處理器
- 350：輸入/輸出集線器
- 360：輸入/輸出裝置
- 390：圖形記憶體控制器集線器
- 395：連接
- 400：多處理器系統
- 414：I/O 裝置
- 415：處理器
- 416：匯流排
- 418：匯流排橋接器
- 420：匯流排
- 422：鍵盤/滑鼠
- 427：通訊裝置
- 428：儲存單元
- 430：碼與資料
- 432：記憶體
- 434：記憶體
- 438：共處理器
- 439：高效介面
- 450：P-P 介面
- 452：P-P 介面

454 : P-P 介面
470 : 處理器
472 : IMC
476 : P-P 介面
478 : P-P 介面
480 : 處理器
482 : IMC
486 : P-P 介面
488 : P-P 介面
490 : 晶片組
492 : 介面
494 : P-P 介面
496 : 介面
498 : P-P 介面
500 : 系統
514 : I/O 裝置
515 : 傳統 I/O 裝置
600 : 系統晶片
602 : 互連單元
610 : 應用處理器
620 : 共處理器
630 : SRAM 單元
632 : DMA 單元
640 : 顯示單元

702：高階語言
 704：x86 編譯器
 706：x86 二進制碼
 708：指令集編譯器
 710：指令集二進制碼
 712：指令轉換器
 714：無 x86 指令集核心的處理器
 716：有 x86 指令集核心的處理器
 800：系統
 802：指令
 804：解碼單元
 806：執行單元
 808：立即運算元
 810：第一源緊縮資料運算元
 812：第二源緊縮資料運算元
 814：目的緊縮資料運算元
 900：邏輯
 902：暫時向量
 908：按位元和
 910：按位元互斥或
 1200：同屬向量友善指令格式
 1205：無記憶體存取
 1210：全捨去控制類型運算
 1215：資料轉換類型運算

- 1220：記憶體存取
- 1225：記憶體存取時間指令樣板
- 1230：記憶體存取非時間指令樣板
- 1240：格式欄
- 1242：基礎運算欄
- 1244：暫存器索引欄
- 1246：修飾符欄
- 1250：擴增運算欄
- 1252： α 欄
- 1254： β 欄
- 1260：縮放欄
- 1262A：位移欄
- 1262B：位移因數欄
- 1268：類型欄
- 1270：寫入遮罩欄
- 1272：立即欄
- 1274：全運算碼欄
- 1256：SAE 欄
- 1258：捨入運算控制欄
- 1252A：RS 欄
- 1252A.1：捨去
- 1252A.2：資料轉換
- 1254A：捨去控制欄
- 1254B：資料轉換欄

- 1254C：資料調處欄
- 1257A：RL 欄
- 1257A.1：捨去
- 1257A.2：向量長度
- 1259A：捨去運算控制欄
- 1259B：向量長度欄
- 1257B：廣播欄
- 1300：特定向量友善指令格式
- 1305：REX 欄
- 1315：運算碼圖欄
- 1320：EVEX.vvvv
- 1325：前置編碼欄
- 1330：真運算碼欄
- 1340：MOD R/M 欄
- 1342：MOD 欄
- 1344：Reg 欄
- 1346：R/M 欄
- 1354：SIB.xxx
- 1356：SIB.bbb
- 1400：暫存器架構
- 1410：向量暫存器
- 1415：寫入遮罩暫存器
- 1425：通用暫存器
- 1445：純量浮點堆疊暫存器檔

1450：MMX 緊縮整數平暫存器欄

1500：指令解碼

1502：環型網路

1504：第 2 層快取的本地次組

1506：L1 快取

1508：純量單元

1510：向量單元

1512：純量暫存器

1514：向量暫存器

1506A：L1 資料快取

1520：拌和

1522A、B：數值轉換

1524：複製

1526：寫入遮罩暫存器

1528：ALU

發明摘要

※申請案號：104138542

※申請日：104 年 11 月 20 日

※IPC 分類：G06F 9/30 (2006.01)
G06F 9/38 (2006.01)

【發明名稱】(中文/英文)

用於向量廣播及互斥或和邏輯指令的設備與方法

Apparatus and method for vector broadcast and xorand logical instruction

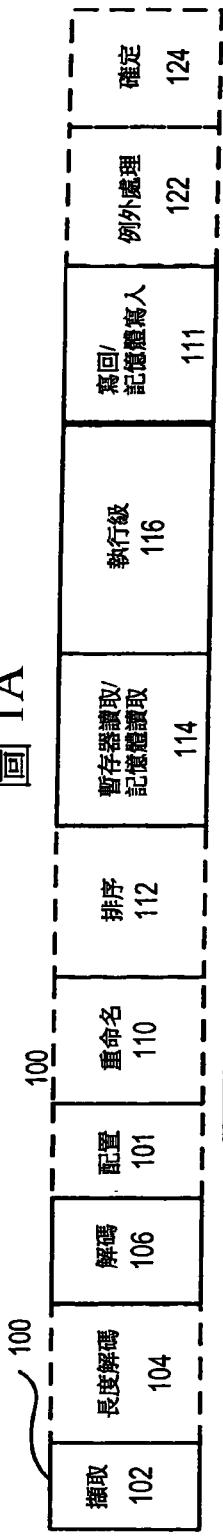
【中文】

本案描述一種用以執行向量廣播與互斥或和邏輯指令的設備與方法。例如，處理器的一實施例包含：擷取邏輯，用以自記憶體擷取一表示目的緊縮資料運算元、第一源緊縮資料運算元、第二源緊縮資料運算元、及立即運算元的指令；及執行邏輯，用以：根據對應於該立即值的位置，決定在該第二源緊縮資料運算元中的位元；執行該第一源緊縮資料運算元與該所決定位元間的按位元（bitwise）和運算，以產生中間結果；執行該目的緊縮資料運算元與該中間結果間的按位元互斥或運算，以產生最終結果；並儲存該最終結果於該目的緊縮資料運算元所指示的儲存位置中。

【 英文 】

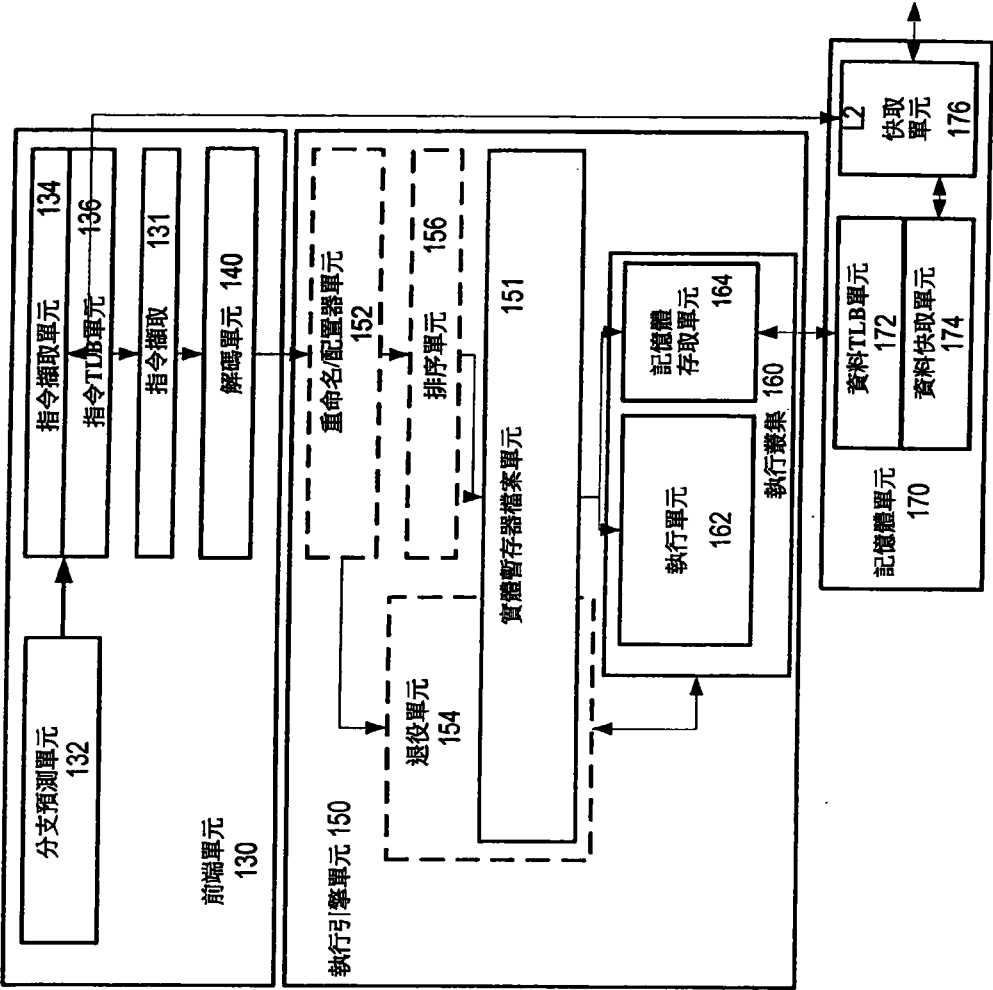
An apparatus and method are described for performing a vector broadcast and XORAND logical instruction. For example, one embodiment of a processor comprises: fetch logic to fetch an instruction from memory indicating a destination packed data operand, a first source packed data operand, a second source packed data operand, and an immediate operand, and execution logic to determine a bit in the second source packed data operand based a position corresponding to the immediate value, perform a bitwise AND between the first source packed data operand and the determined bit to generate an intermediate result, perform a bitwise XOR between the destination packed data operand and the intermediate result to generate a final result, and store the final result in a storage location indicated by the destination packed data operand.

圖 1A



圖式

圖 1B



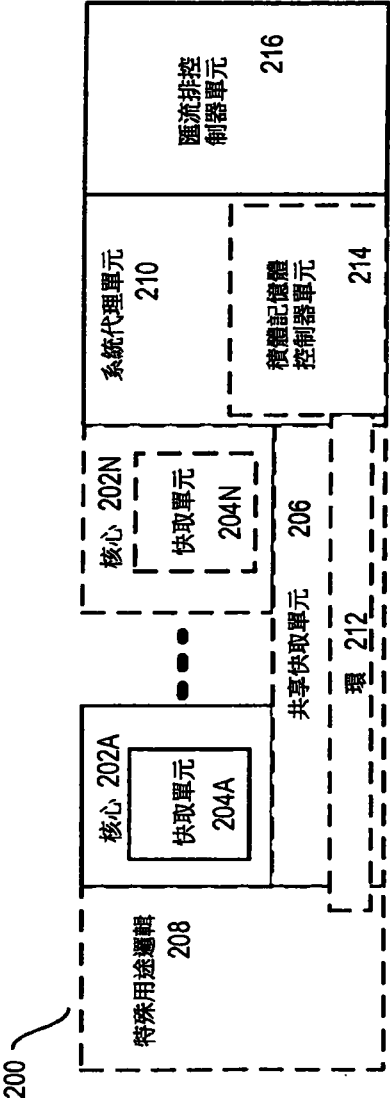


圖 2

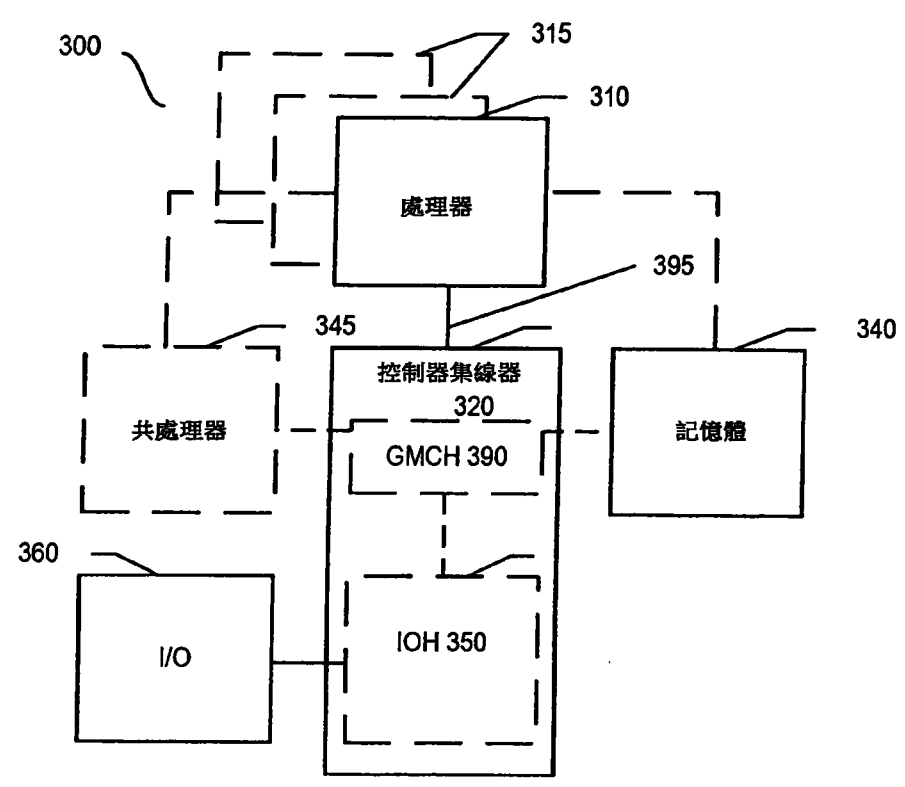


圖 3

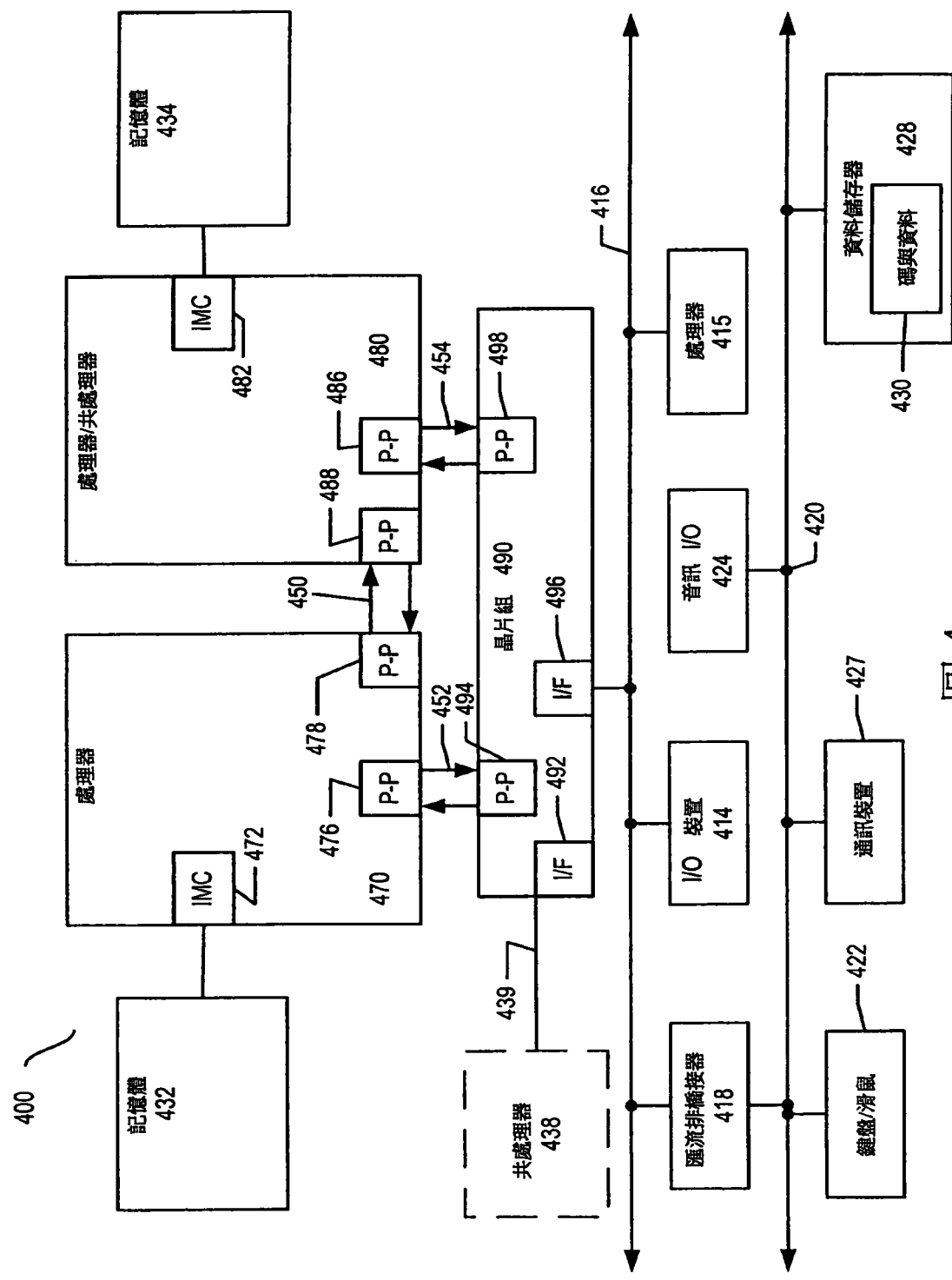


圖 4

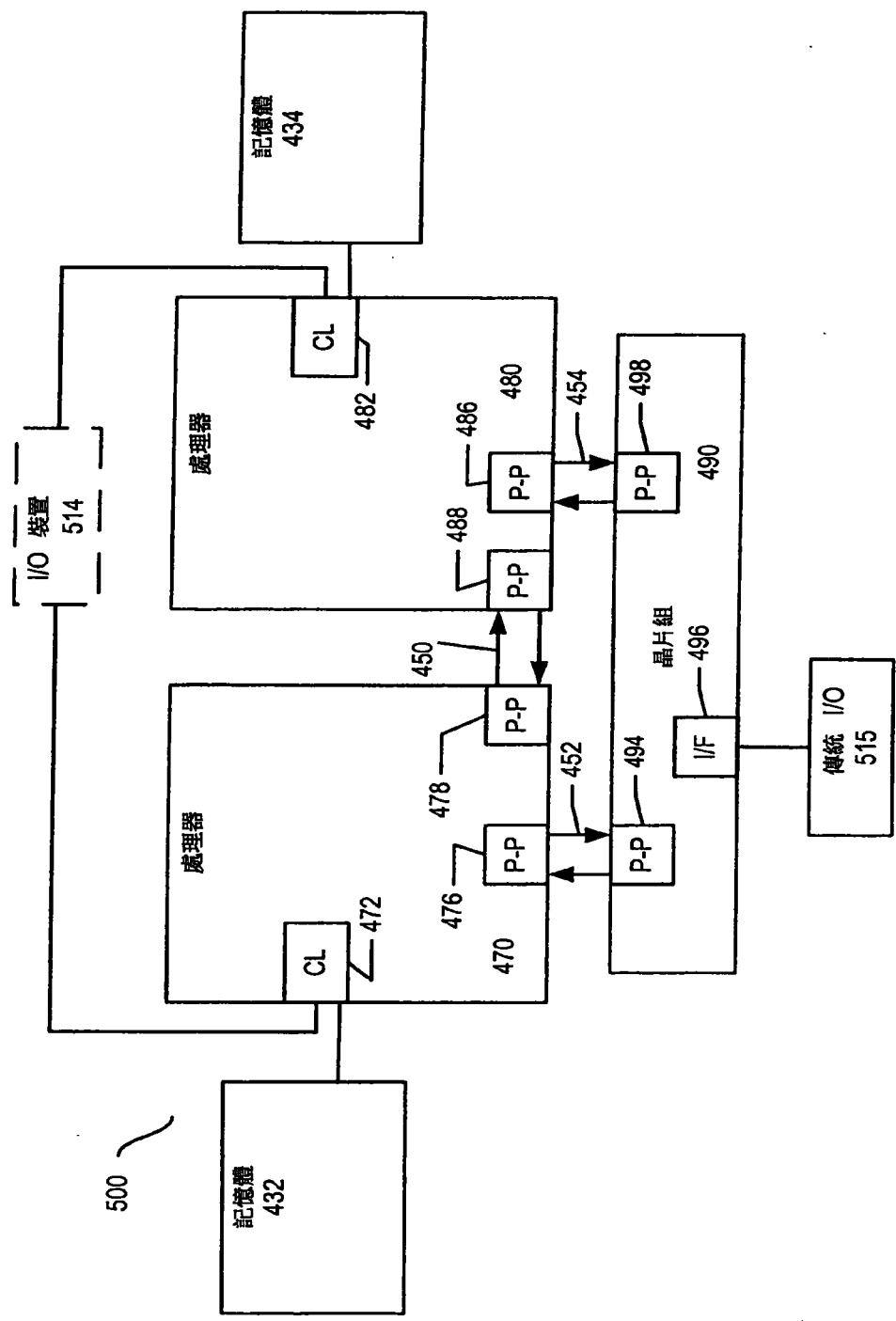


圖 5

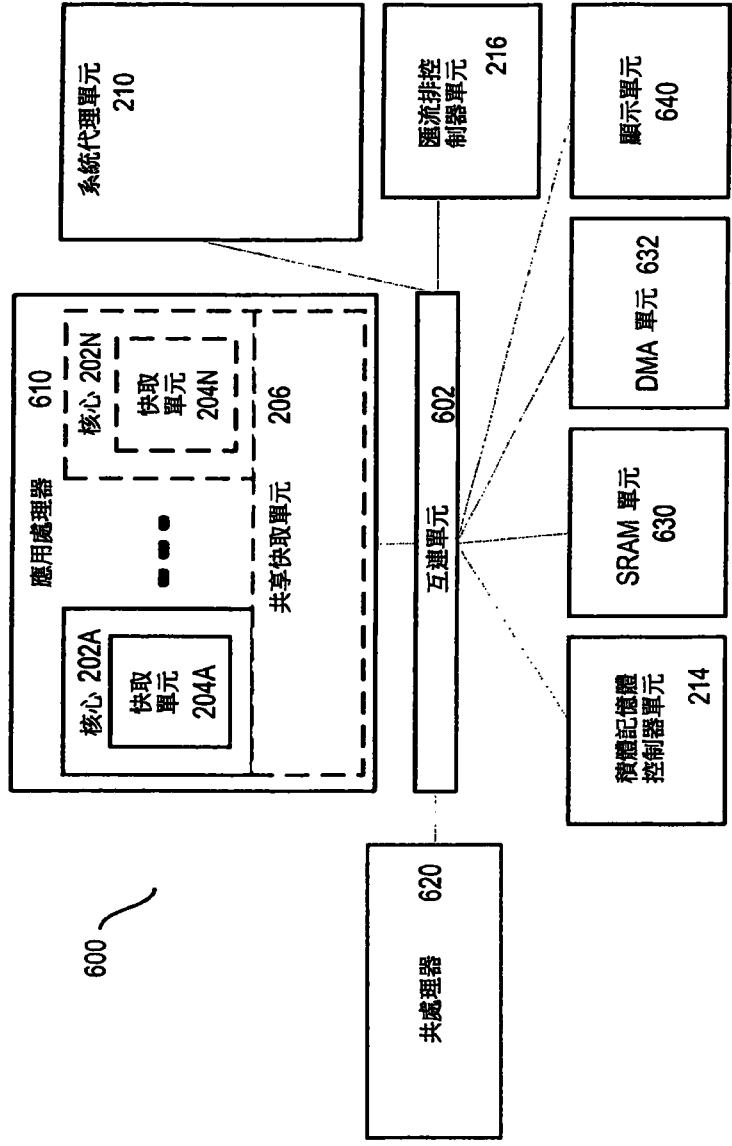


圖 6

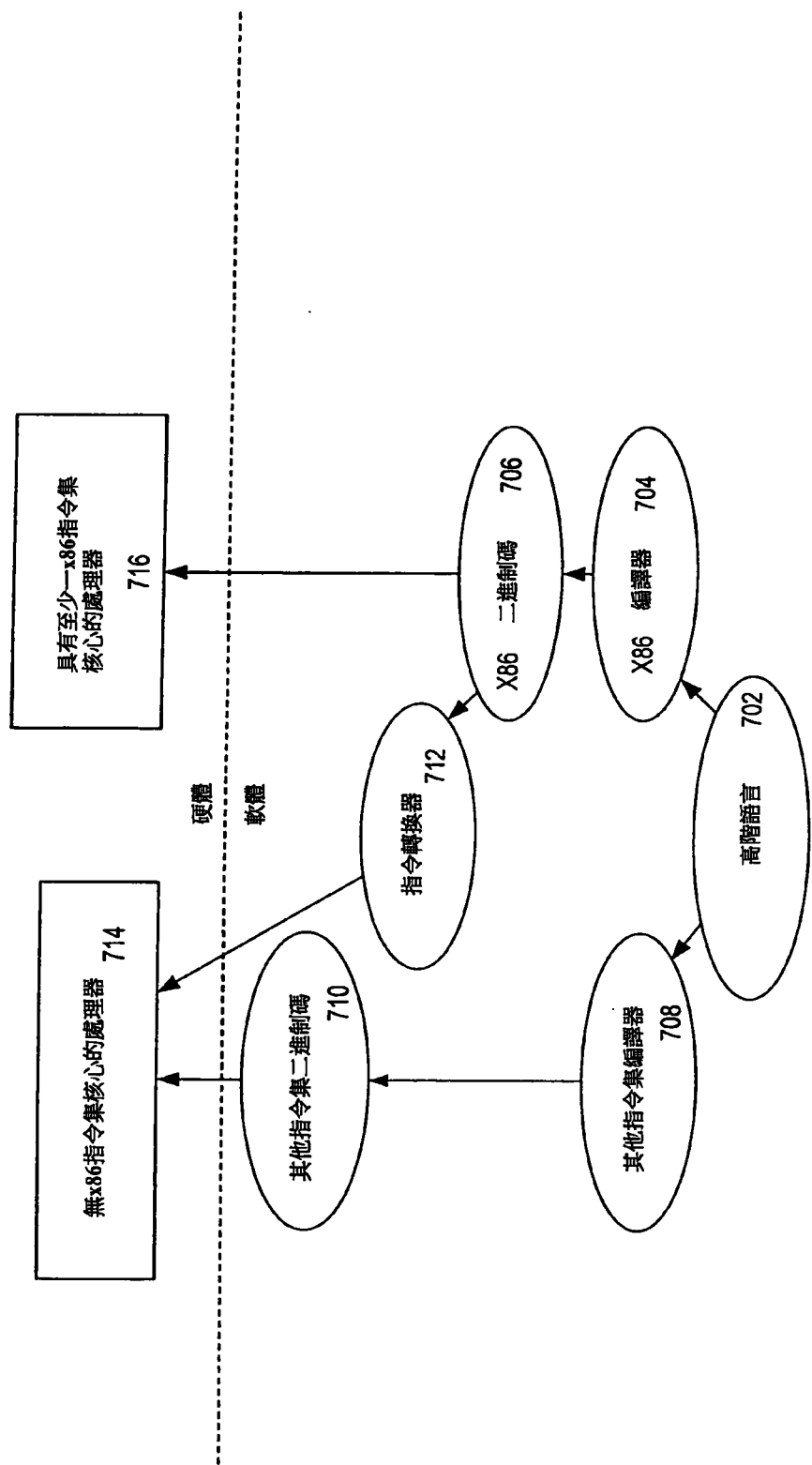


圖 7

800 →

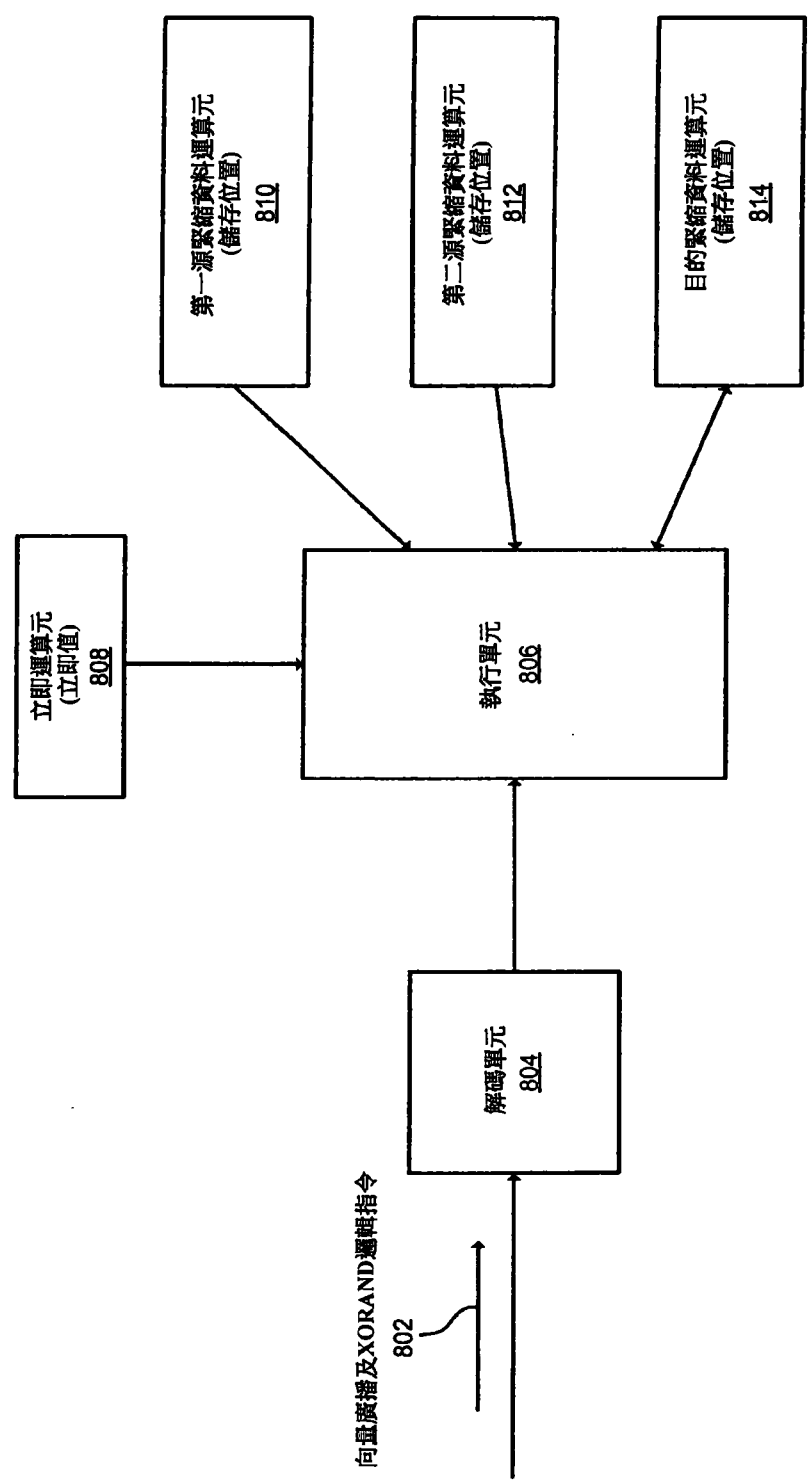


圖 8

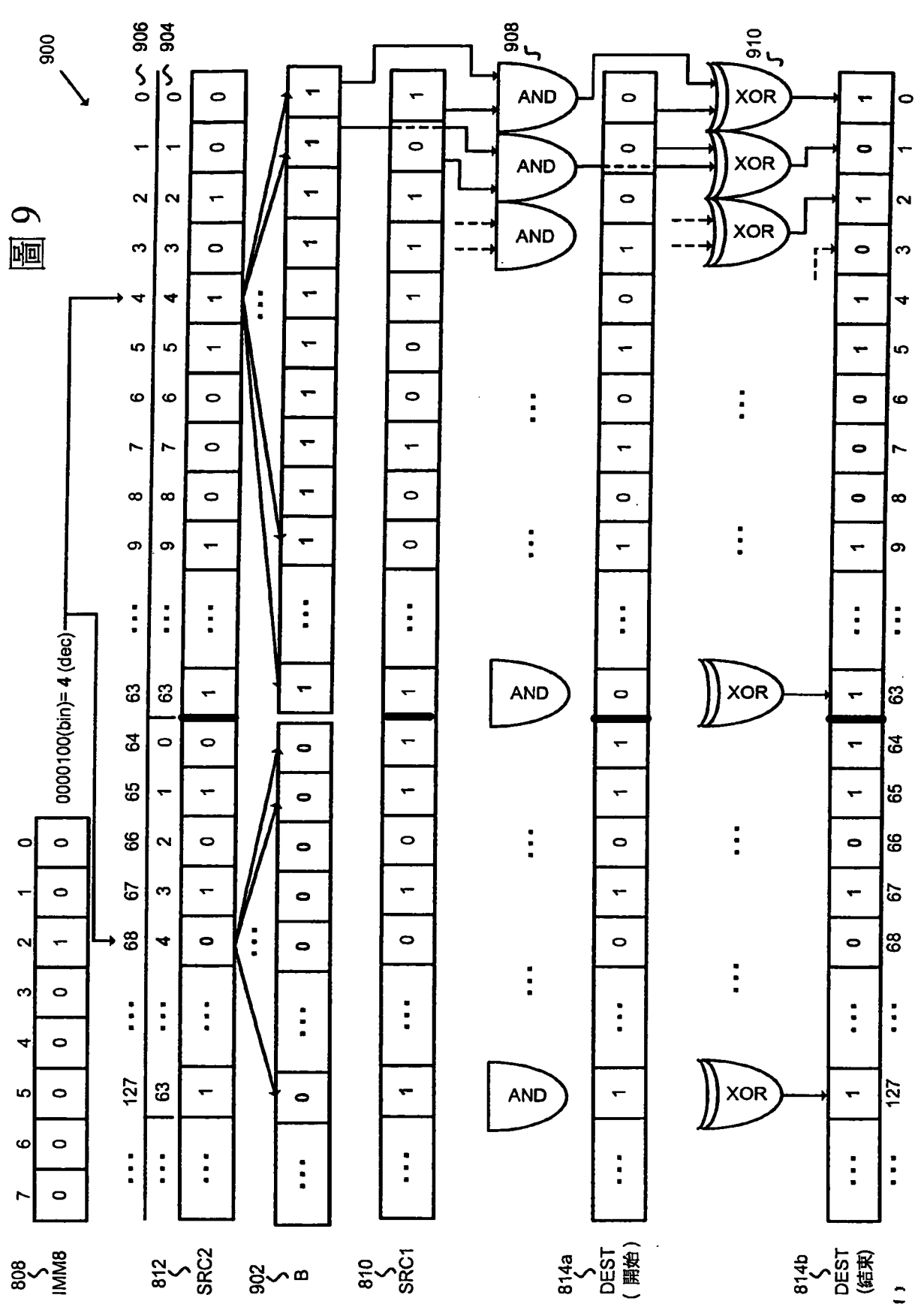


圖 9

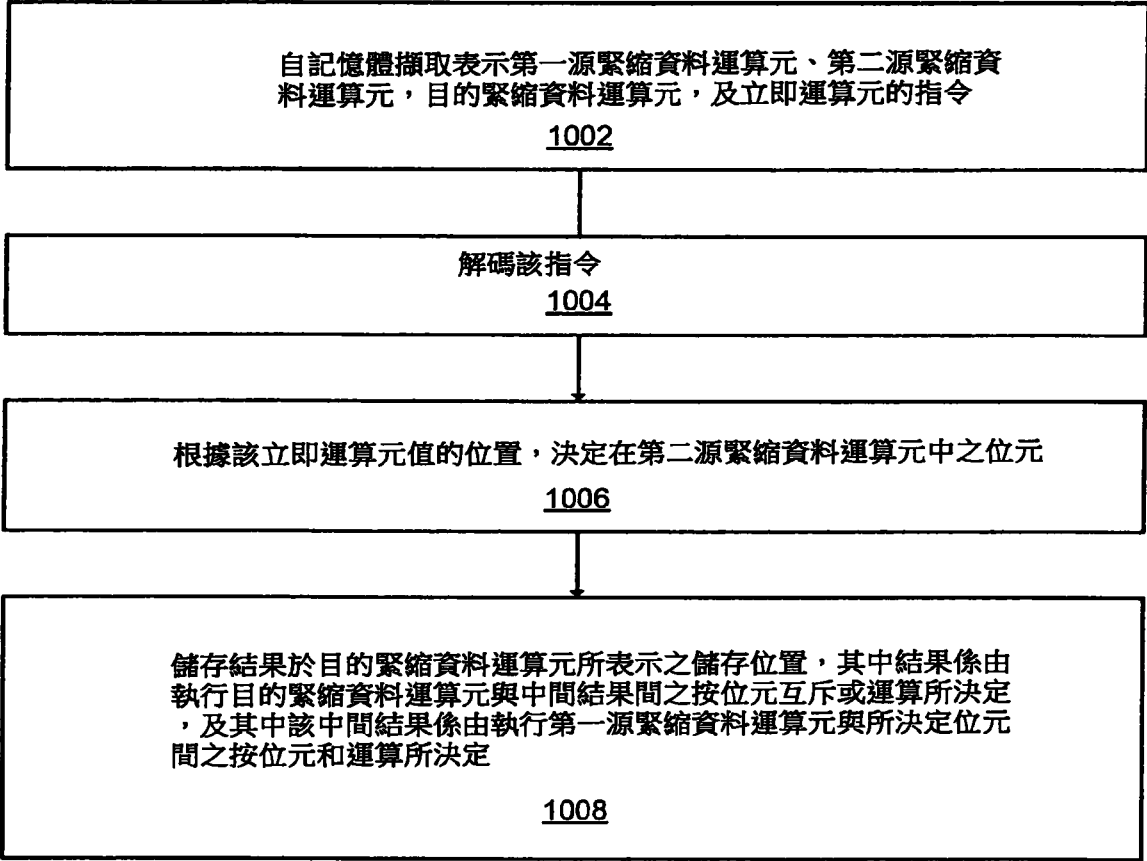


圖 10

	1152	1154	1162	1156	1158	1160
	↓	↓	↓	↓	↓	↓
	v xorandbcstq zmm0 {k1}, zmm1, zmm2, imm8					
1102	→ (KL, VL) = (2, 128), (4, 256), (8, 512)					
1104	→ FOR j := 0 to KL-1					
1106	→ i := j * 64					
1108	→ B[63:0] := SRC2[i+IMM8] <i>replicated 64 times</i>					
1110	→ IF k1[j]					
1112	→ DEST[63+i:i] := DEST[63+i:i] ^ (SRC1[63+i:i] & B)					
	ELSE					
1114	→ IF <i>*merging-masking*</i> /* merging-masking					
1116	→ THEN <i>*DEST[63+i:i] remains unchanged*</i> /* DEST[63+i:i] remains unchanged					
	ELSE					
1118	→ DEST[63+i:i] = 0 /* zero masking					
	FI					
	ENDFOR					
1120	→ DEST[MAX_VL-1:VL] := 0					

圖 11

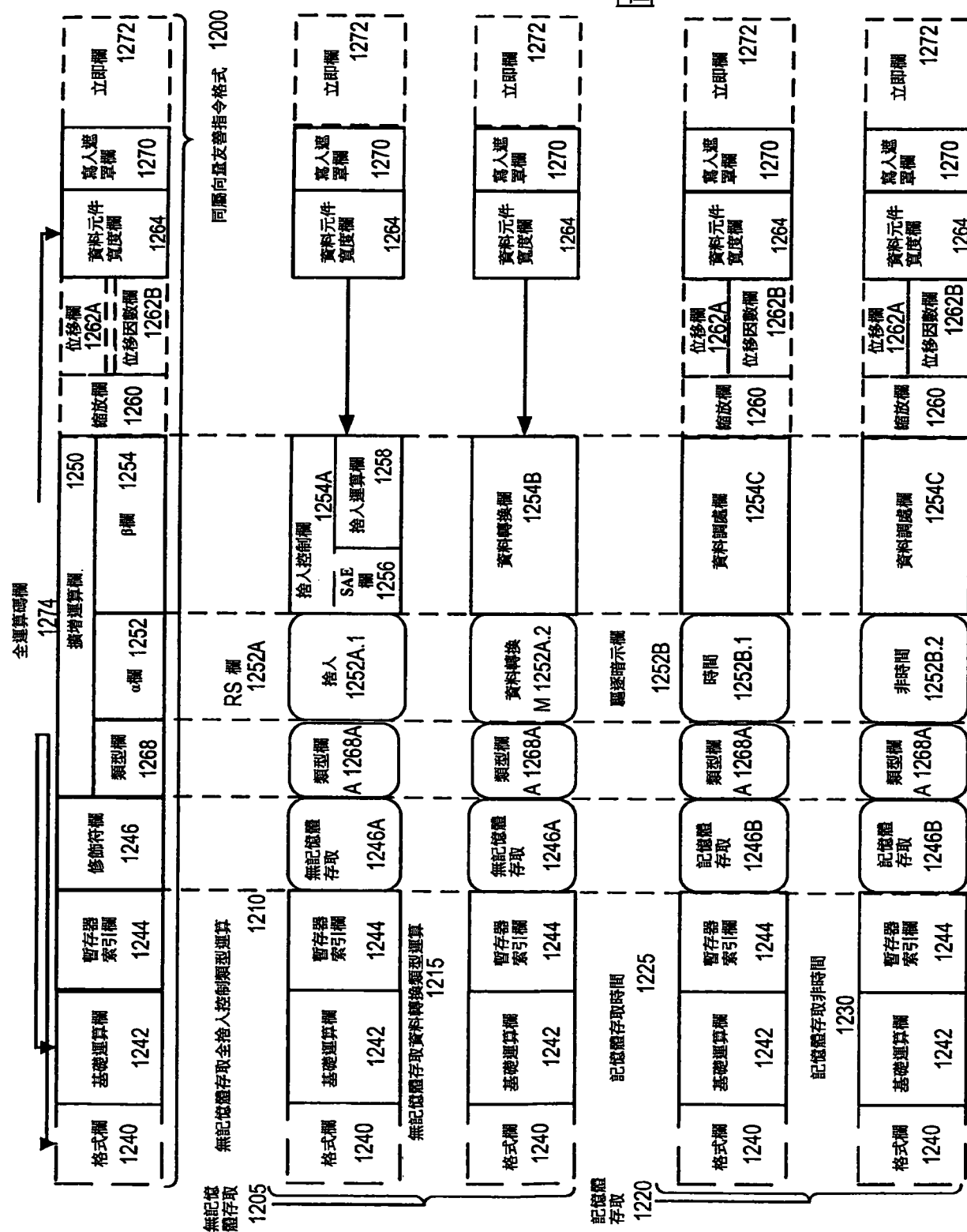


圖 12A

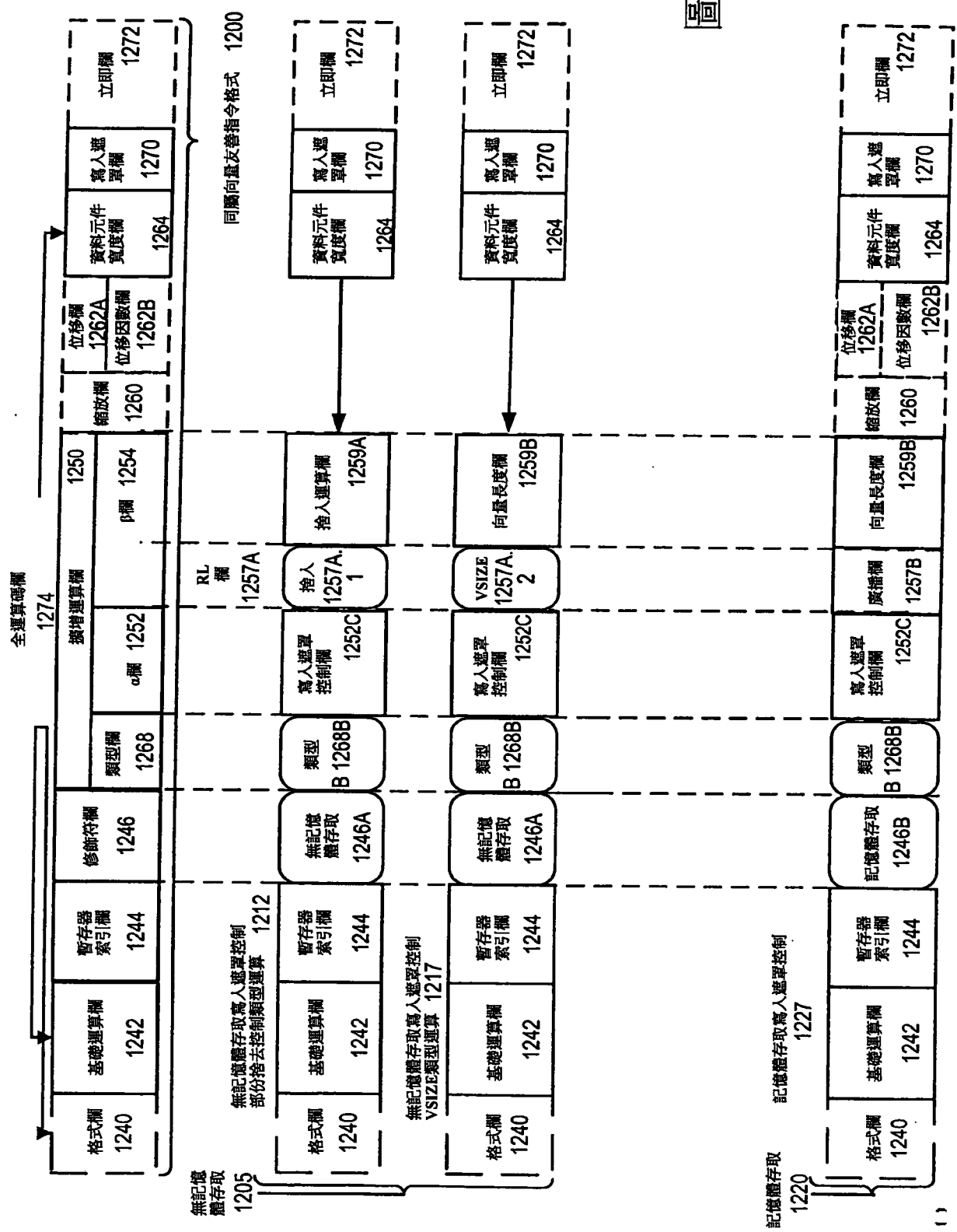


圖 13A

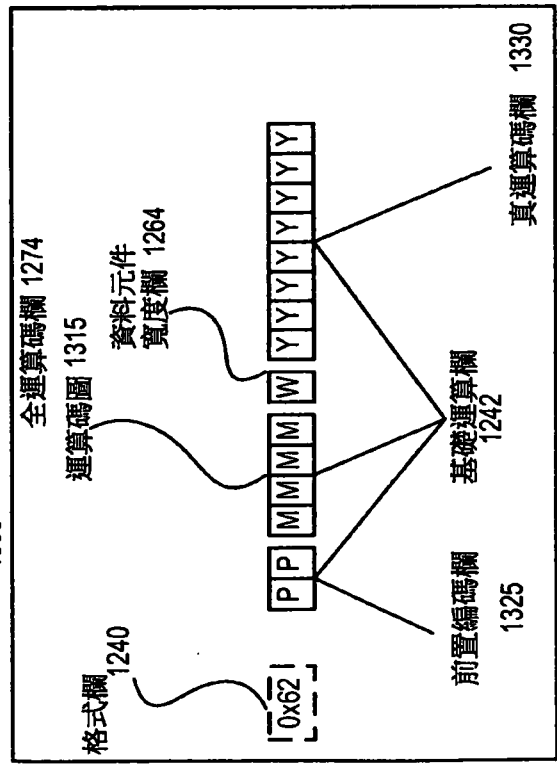
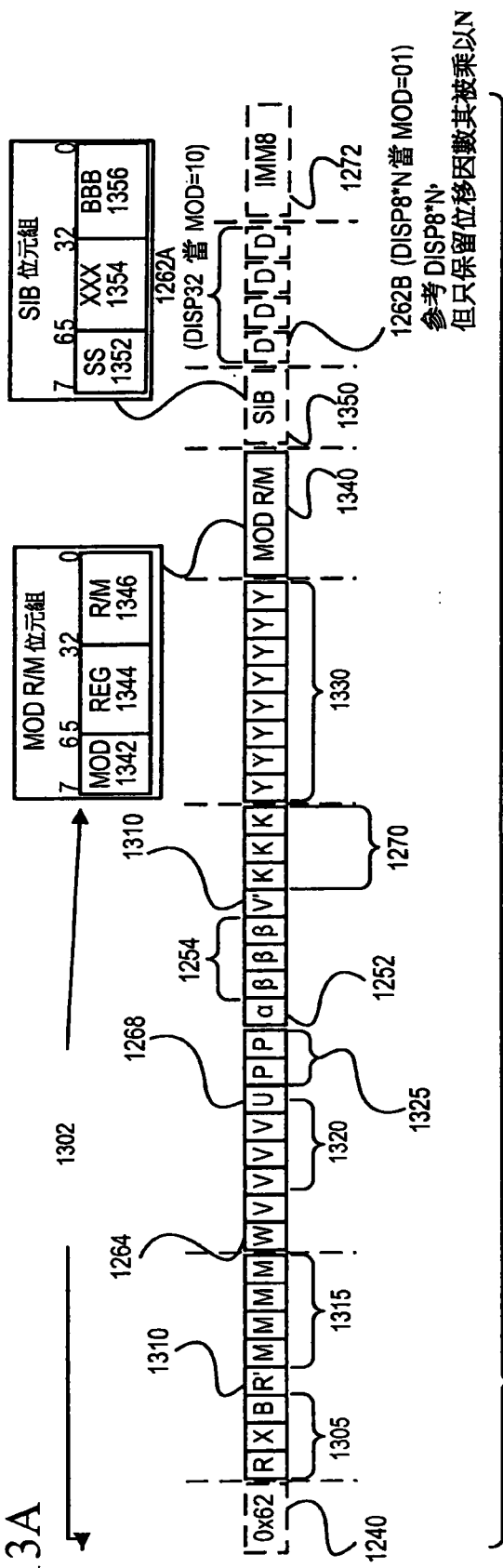


圖 13B

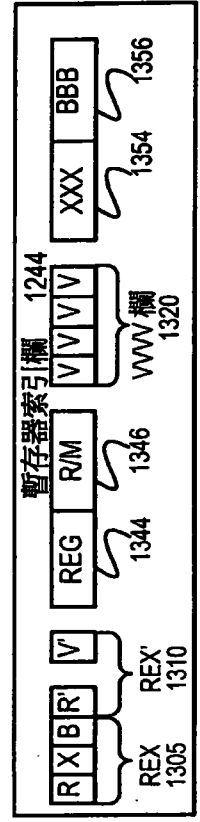


圖 13C

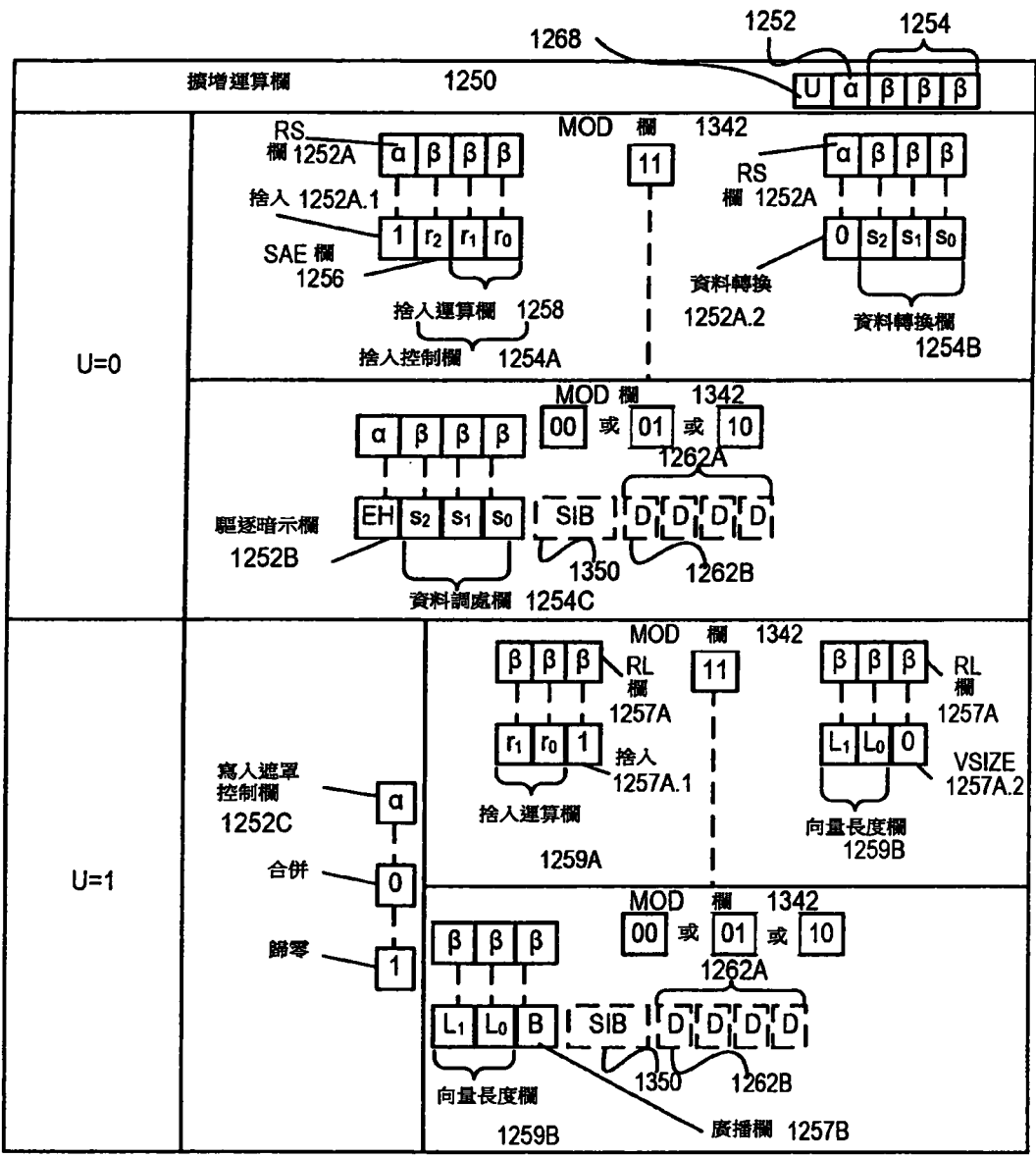


圖 13D

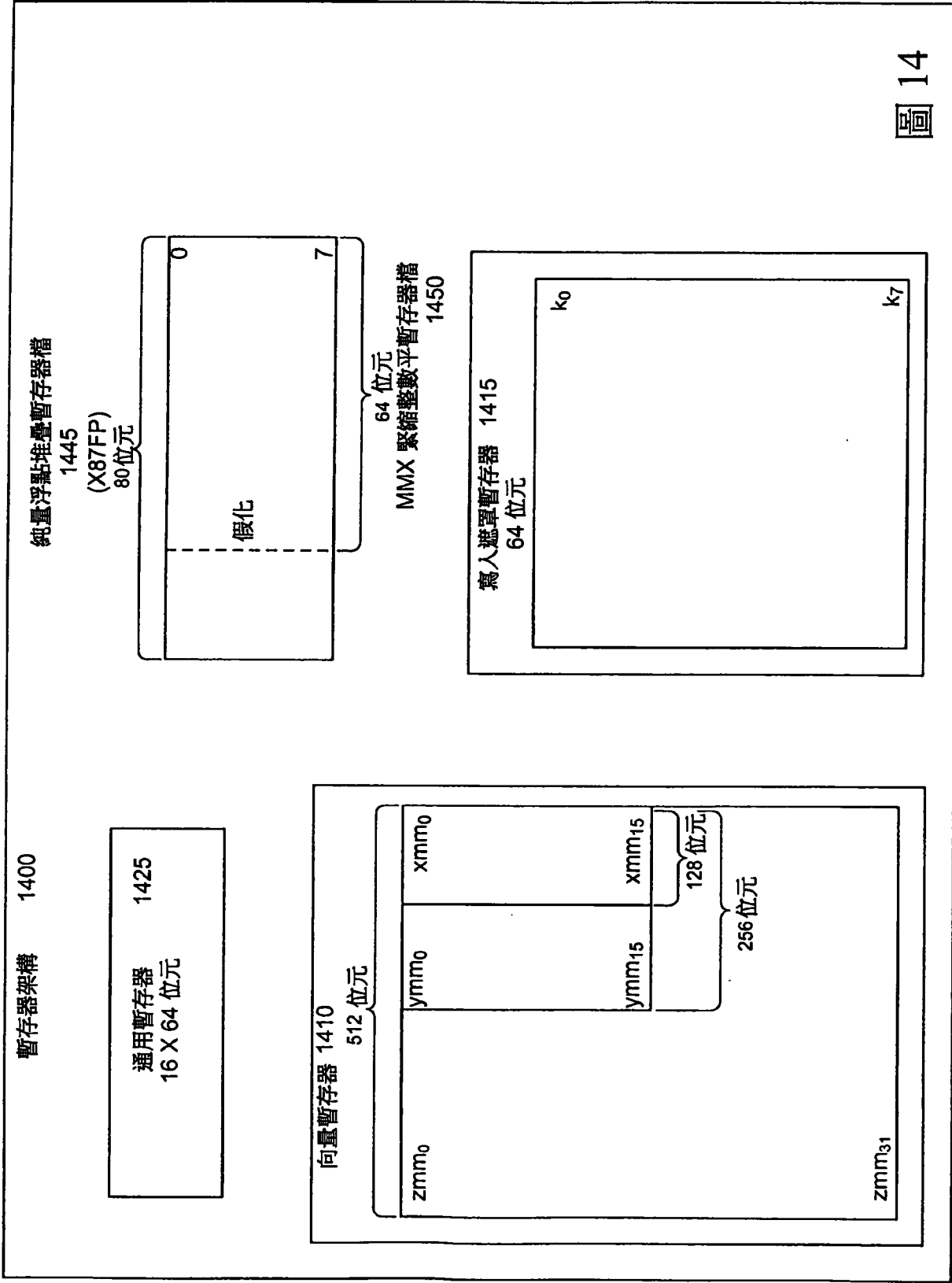


圖 14

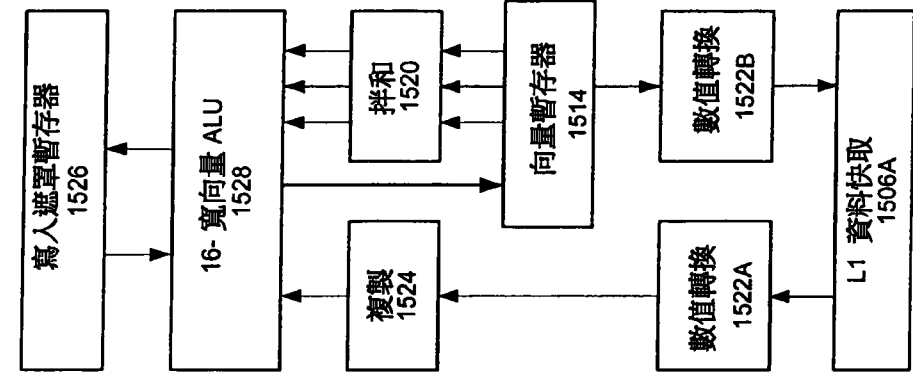


圖 15A

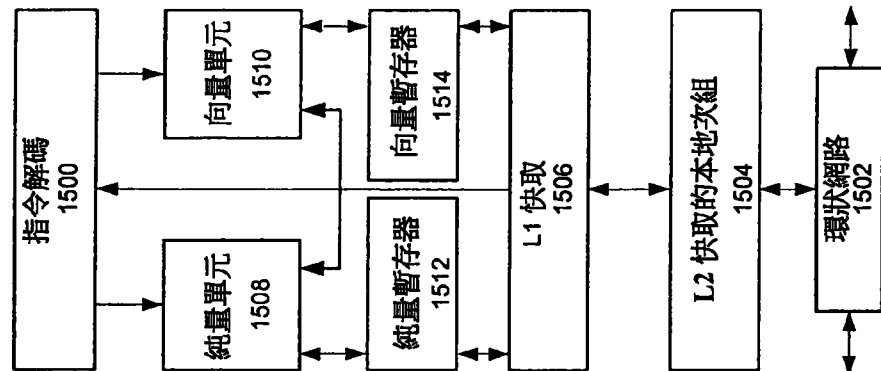


圖 15B

【代表圖】

【本案指定代表圖】：第(9)圖。

【本代表圖之符號簡單說明】：

808：立即運算元

810：第一源緊縮資料運算元

812：第二源緊縮資料運算元

814：目的緊縮資料運算元

900：邏輯

902：暫時向量

908：按位元和

910：按位元互斥或

【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，包含：

解碼電路，用以解碼來自記憶體中表示目的緊縮資料運算元、第一源緊縮資料運算元、第二源緊縮資料運算元、及立即值的指令；及

執行電路，用以執行該所解碼的指令，以：

根據對應於該立即值的位置，決定在該第二源緊縮資料運算元中的位元；

執行該第一源緊縮資料運算元與該所決定的位元間的按位元和運算，以產生中間結果；

執行該目的緊縮資料運算元與該中間結果間的按位元互斥或運算，以產生最終結果；及

儲存該最終結果於為該目的緊縮資料運算元所表示的儲存位置中。

2. 如申請專利範圍第 1 項之處理器，其中為了執行於該第一源緊縮資料運算元與該所決定的位元間的該按位元和運算，該執行電路更進一步用以執行於該第一源緊縮資料運算元與暫時向量間的按位元和運算，其中該所決定位元的值係予以被一次或多次廣播至該暫時向量。

3. 如申請專利範圍第 1 項之處理器，其中為該目的緊縮資料運算元、該第一源緊縮資料運算元、及該第二源緊縮資料運算元所表示的該等儲存位置係予以被處理於分開的 64 位元部中。

4. 如申請專利範圍第 3 項之處理器，其中該指令更

包含寫入遮罩運算元，及其中該執行電路更用以回應於決定該寫入遮罩運算元表示寫入遮罩係設定用於該目的緊縮資料運算元中的該等 64 位元部中之一，而進一步設定為該目的緊縮資料運算元所表示的該儲存位置中的該等 64 位元部之該一的值為零。

5. 如申請專利範圍第 1 項之處理器，其中為該目的緊縮資料運算元、該第一源緊縮資料運算元與該第二源緊縮資料運算元所表示的該等儲存位置係為暫存器與記憶體位置之至少之一。

6. 如申請專利範圍第 5 項之處理器，其中為該目的緊縮資料運算元、該第一源緊縮資料運算元與該第二源緊縮資料運算元所表示的該等儲存位置係為 512 位元長的暫存器。

7. 如申請專利範圍第 5 項之處理器，其中該立即值為 8 位元長。

8. 如申請專利範圍第 1 項之處理器，其中該指令係被用以執行於位元矩陣與位元向量間的位元矩陣乘法運算，其中該位元矩陣的一或更多行係被儲存於為該第一源緊縮資料運算元所表示的儲存位置中，及其中該位元向量的值係被儲存於為該第二源緊縮資料運算元所表示的該儲存位置中。

9. 如申請專利範圍第 8 項之處理器，其中該位元矩陣係被轉置，使得該位元矩陣的該一或更多行係被逐行地儲存於該第一源緊縮資料運算元所表示的該儲存位置中。

10. 如申請專利範圍第 9 項之處理器，其中為該目的緊縮資料運算元所表示的該儲存位置包含當該指令係被執行用於該位元矩陣的各個所述行時，該位元矩陣與該位元向量間的該位元矩陣乘法運算的結果，其中對於該指令的各個執行，該立即值指明一值，其表示該位元向量中對應於該被處理的位元矩陣的行數的位置。

11. 一種在電腦處理器中的方法，包含：

解碼來自記憶體之表示目的緊縮資料運算元、第一源緊縮資料運算元、第二源緊縮資料運算元與立即運算元的指令；及

藉由以下執行所解碼的指令：

根據對應於該立即值的位置，決定在該第二源緊縮資料運算元中的位元，

執行於該第一源緊縮資料運算元與該所決定的位元間的按位元和運算，以產生中間結果，

執行於該目的緊縮資料運算元與該中間結果間的按位元互斥或運算，以產生最終結果，及

儲存該最終結果於該目的緊縮資料運算元所表示的儲存位置中。

12. 如申請專利範圍第 11 項之方法，其中該執行於該第一源緊縮資料運算元與該所決定的位元間的按位元和運算更包含執行該第一源緊縮資料運算元與暫時向量間的按位元和運算，其中該所決定的位元的值係予以被一次或多次廣播至該暫時向量。

13. 如申請專利範圍第 11 項之方法，其中為該目的緊縮資料運算元、該第一源緊縮資料運算元與該第二源緊縮資料運算元所表示的儲存位置係予以被處理於分開的 64 位元部中。

14. 如申請專利範圍第 13 項之方法，其中該指令更包含寫入遮罩運算元，及其中該方法更包含回應於決定該寫入遮罩運算元表示寫入遮罩係被設定用於該目的緊縮資料運算元中的該等 64 位元部中之一，設定為該目的緊縮資料運算元所表示的該儲存位置中的該等 64 位元部之該一的值為零。

15. 如申請專利範圍第 11 項之方法，其中為該目的緊縮資料運算元、該第一源緊縮資料運算元與該第二源緊縮資料運算元所表示的儲存位置係為暫存器與記憶體位置之至少之一。

16. 如申請專利範圍第 15 項之方法，其中為該目的緊縮資料運算元、該第一源緊縮資料運算元與該第二源緊縮資料運算元所表示的儲存位置係為 512 位元長的暫存器。

17. 如申請專利範圍第 15 項之方法，其中該立即為 8 位元長。

18. 如申請專利範圍第 11 項之方法，其中該指令係用以執行於位元矩陣與位元向量間的位元矩陣乘法運算，其中該位元矩陣的一或更多行係被儲存於為該第一源緊縮資料運算元所表示的該儲存位置中，及其中該位元向量的

值係被儲存於為該第二源緊縮資料運算元所表示的該儲存位置中。

19. 如申請專利範圍第 18 項之方法，其中該位元矩陣係被轉置，使得該位元矩陣的該一或更多行係被逐行地儲存於為該第一源緊縮資料運算元所表示的該儲存位置中。

20. 如申請專利範圍第 19 項之方法，其中為該目的緊縮資料運算元所表示之該儲存位置包含當該指令被執行用於該位元矩陣的各個所述行時的該位元矩陣與該位元向量間的該位元矩陣乘法運算的結果，其中對於該指令的各個執行，該立即值指明一值，其表示該位元向量中對應於被處理的該位元矩陣的行數的位置。