



(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 I506546 B

(45)公告日：中華民國 104 (2015) 年 11 月 01 日

(21)申請案號：103135824

(22)申請日：中華民國 100 (2011) 年 12 月 07 日

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

(30)優先權：2011/04/01 美國

61/471,043

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

(72)發明人：瓦倫泰 羅柏 VALENTINE, ROBERT (US)；聖安德里安 耶穌 SAN ADRIAN, JESUS CORBAL (ES)；桑斯 羅傑 SANS, ROGER ESPASA (ES)；凱文 羅勃 CAVIN, ROBERT D. (US)；托爾 布萊特 TOLL, BRET (US)；都蘭 山堤雅哥 DURAN, SANTIAGO GALAN (ES)；威德梅爾 傑佛瑞 WIEDEMEIER, JEFFREY G. (US)；山姆卓拉 沙海 SAMUDRALA, SRIDHAR (US)；吉卡 密林德 GIRKAR, MILIND B. (US)；葛羅秋斯基 愛德華 GROCHOWSKI, EDWARD T. (US)；賀爾 強納森 HALL, JONATHAN C. (US)；布瑞福 丹尼斯 BRADFORD, DENNIS R. (US)；歐德亞麥德維爾 艾蒙斯特阿法 OULD-AHMED-VALL, ELMOUSTAPHA (MR)；雅伯 詹姆士 ABEL, JAMES C. (US)；查尼 馬克 CHARNEY, MARK J. (US)；艾柏罕 西仕 ABRAHAM, SETH (US)；賽爾 蘇利曼 SAIR, SULEYMAN (TR)；佛西斯 安卓 FORSYTH, ANDREW T. (GB)；吳 麗莎 WU, LISA (US)；楊特 查爾斯 YOUNT, CHARLES (US)

(74)代理人：林志剛

(56)參考文獻：

US 2010/0191939A1

US 2010/0217854A1

審查人員：高嘉男

申請專利範圍項數：55 項 圖式數：30 共 173 頁

(54)名稱

向量親合的指令格式及其執行

VECTOR FRIENDLY INSTRUCTION FORMAT AND EXECUTION THEREOF

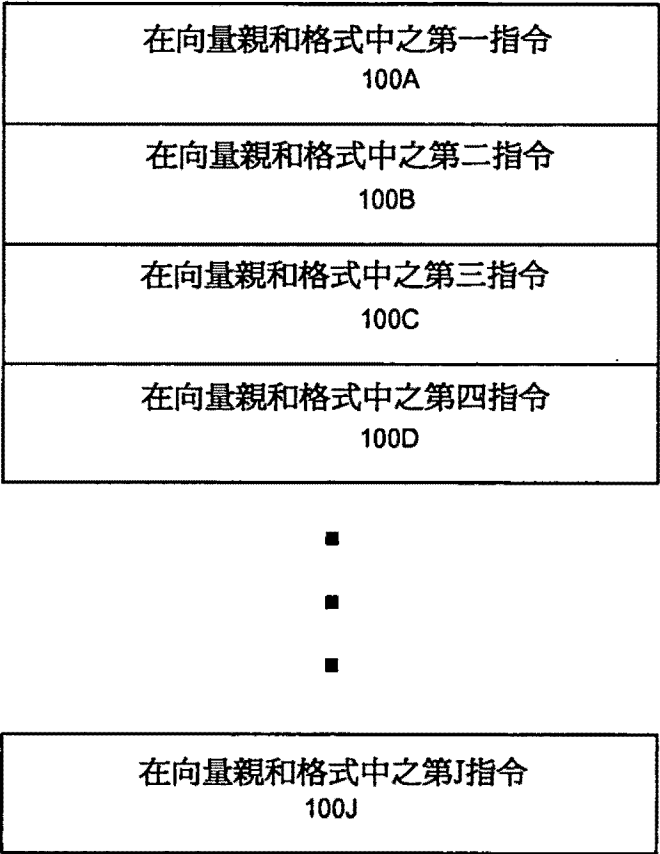
(57)摘要

本文揭露向量親和的指令格式及其執行。根據本發明之一實施例，處理器組態成執行指令集。指令集包括向量親和的指令格式。向量親和的指令格式具有複數欄位，包括基礎運算欄位、修飾符欄位、擴充運算欄位、及資料元件寬度欄位，其中第一指令格式透過在基礎運算欄位、修飾符欄位、阿爾發欄位、貝他欄位、及資料元件寬度欄位中放置不同值來支援基礎運算的不同版本及不同擴充運算，且其中在指令流中在第一指令格式中的指令之每一個出現上，在基礎運算欄位、修飾符欄位、阿爾發欄位、貝他欄位、及資料元件寬度欄位的每一者中僅可放置該些不同值的一個。

A vector friendly instruction format and execution thereof. According to one embodiment of the invention, a processor is configured to execute an instruction set. The instruction set includes a vector friendly instruction format. The vector friendly instruction format has a plurality of fields including a base operation field, a modifier field, an augmentation operation field, and a data element width field, wherein the first

instruction format supports different versions of base operations and different augmentation operations through placement of different values in the base operation field, the modifier field, the alpha field, the beta field, and the data element width field, and wherein only one of the different values may be placed in each of the base operation field, the modifier field, the alpha field, the beta field, and the data element width field on each occurrence of an instruction in the first instruction format in instruction streams.

第1A圖



100A . . . 在向量親和格式中之第一指令
100B . . . 在向量親和格式中之第二指令
100C . . . 在向量親和格式中之第三指令
100D . . . 在向量親和格式中之第四指令
100J . . . 在向量親和格式中之第J指令

發明摘要

※申請案號：103135824

(由100/45056分割)

※申請日：100年12月07日

※IPC分類：G06F 9/30 (2006.01)

【發明名稱】(中文/英文)

向量親合的指令格式及其執行

Vector friendly instruction format and execution thereof

● 【中文】

本文揭露向量親和的指令格式及其執行。根據本發明之一實施例，處理器組態成執行指令集。指令集包括向量親和的指令格式。向量親和的指令格式具有複數欄位，包括基礎運算欄位、修飾符欄位、擴充運算欄位、及資料元件寬度欄位，其中第一指令格式透過在基礎運算欄位、修飾符欄位、阿爾發欄位、貝他欄位、及資料元件寬度欄位中放置不同值來支援基礎運算的不同版本及不同擴充運算，且其中在指令流中在第一指令格式中的指令之每一個出現上，在基礎運算欄位、修飾符欄位、阿爾發欄位、貝他欄位、及資料元件寬度欄位的每一者中僅可放置該些不同值的一個。

【 英文 】

A vector friendly instruction format and execution thereof. According to one embodiment of the invention, a processor is configured to execute an instruction set. The instruction set includes a vector friendly instruction format. The vector friendly instruction format has a plurality of fields including a base operation field, a modifier field, an augmentation operation field, and a data element width field, wherein the first instruction format supports different versions of base operations and different augmentation operations through placement of different values in the base operation field, the modifier field, the alpha field, the beta field, and the data element width field, and wherein only one of the different values may be placed in each of the base operation field, the modifier field, the alpha field, the beta field, and the data element width field on each occurrence of an instruction in the first instruction format in instruction streams.

【代表圖】

【本案指定代表圖】：第(1A)圖。

【本代表圖之符號簡單說明】：

100A：在向量親和格式中之第一指令

100B：在向量親和格式中之第二指令

100C：在向量親和格式中之第三指令

100D：在向量親和格式中之第四指令

100J：在向量親和格式中之第 J 指令

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：無

發明專利說明書

(本說明書格式、順序，請勿任意更動)

【發明名稱】(中文/英文)

向量親合的指令格式及其執行

Vector friendly instruction format and execution thereof

【技術領域】

本發明主要有關於電腦的領域；且詳言之，關於由處理器所支援的指令集。

【先前技術】

指令集，或指令集架構（ISA），為有關於編程之電腦架構的部分，包括原生資料類型、指令、暫存器架構、定址模式、記憶體架構、中斷及異常處置、及外部輸入及輸出（I/O）。應注意到指令一詞在此一般係指巨集指令（亦即，提供給處理器供執行之指令），相對於係處理器之解碼器解碼巨集指令之結果的微指令或微運算。

指令集架構與微架構不同，微架構係用來實現指令集之處理器設計技術集。具有不同微架構的處理器可共享一個共同的指令集。例如，Intel Pentium 4處理器、Intel Core處理器、及美國加州的Sunnyvale的Advanced Micro Devices公司之處理器實現幾乎相同版本的x86指令集（較新版本有增添一些延伸），但具有不同的內部設計。例如，可使用眾所皆知的技術在不同微架構中以不同方式實

現ISA的相同暫存器架構，包括專屬實體暫存器、使用暫存器重命名機制的一或更多動態分配的實體暫存器（例如，使用暫存器別名表（RAT）重排序緩衝器（ROB）及引退暫存器檔，如在美國專利案號5,446,912中所述；使用多地圖及暫存器池，如在美國專利案號5,207,132中所述）等等。除非另有所指，片語暫存器架構、暫存器檔、及暫存器在此用來指軟體/程序設計師可見者及指令指明暫存器的方式。當需要區別時，形容詞邏輯、架構、或軟體可見的將用來表示暫存器架構中的暫存器/檔，而不同形容詞用來標指在給定微架構中的暫存器（例如，實體暫存器、重排序緩衝器、引退暫存器、暫存器池）。

一指令集包括一或更多指令格式。一給定指令格式界定各種欄位（位元數量、位元位置）以尤其指明將履行之運算及其上將履行運算之（諸）運算元。進一步透過指令模板（或子格式）之界定分解一些指令格式。例如，一給定指令格式的指令模板可界定成具有指令格式之欄位的不同子集（所包括的欄位通常在相同順序中，但至少一些具有不同位元位置，因為包括較少欄位）及/或界定成具有不同解譯的給定欄位。一給定指令係使用一給定指令格式（且若有界定，在那個指令格式的諸指令模板的一給定者中）來加以表示並指明運算及運算元。指令流為指令之一特定序列，其中在該序列中的每一個指令係在指令格式（且若有界定，在那個指令格式的諸指令模板的一給定者）中之一個指令的出現。

科學、金融、自動向量化通用、RMS（辨認、挖掘、及合成）/視覺及多媒體應用（例如，2D/3D圖形、影像處理、視頻壓縮/解壓縮、語音辨認演算法及音頻操縱）經常需要在大量資料物件上履行相同的運算（在此稱為「資料平行性」）。單指令多資料（SIMD）意指一種令處理器在多個資料物件上履行相同運算之指令。SIMD技術特別適合可將位元邏輯地分成若干固定大小資料元件（其之每一個代表一個分別的值）的處理器。例如，可將64位元暫存器中的位元指明為將被運算為四個分別的16位元資料元件（其之每一個代表一個分別的16位元值）之來源運算元。此類型的資料稱為分包（packed）資料類型或向量資料類型，且此資料類型之運算元稱為分包資料運算元或向量運算元。換言之，分包資料物件或向量意指分包資料元件之一序列；且分包資料運算元或向量運算元為SIMD指令之來源或目的地運算元（亦稱為分包資料指令或向量指令）。

舉例來說，一種SIMD指令指明以垂直方式在兩個來源向量運算元上履行單一向量運算來產生相同大小、具有相同數量的資料元件、及在相同資料元件順序中的目的地向量運算元（亦稱為結果向量運算元）。在來源向量運算元中的資料元件稱為來源資料元件，而在目的地向量運算元中的資料元件稱為目的地或結果資料元件。這些來源向量運算元為相同大小且含有相同寬度的資料元件，並因此它們含有相同數量的資料元件。在這兩個來源向量運算元

中在相同位元位置中的來源資料元件形成資料元件對（也稱為相應資料元件）。分別在這些來源資料元件對之每一者上履行由SIMD指令所指的運算，以產生對應數量的結果資料元件，並因此每一對來源資料元件具有一個相應的結果資料元件。由於運算為垂直，且由於結果向量運算元和來源向量運算元為相同大小、具有相同數量的資料元件、且以相同資料元件順序儲存結果資料元件，因此結果資料元件係在結果向量運算元之和來源向量運算元中其相對應的來源資料元件相同的位元位置中。除了此示範類型的SIMD指令外，有各式各樣其他類型的SIMD指令（例如，僅有一個或有超過兩個來源向量運算元；以水平方式運算；產生不同大小、具有不同資料元件大小、及/或具有不同資料元件順序之結果向量運算元）。應了解的是，目的地向量運算元（或目的地運算元）一詞界定為履行由一指令所指之運算的直接結果，包括目的地運算元在一位置（由那個指令所指之暫存器或在記憶體位址）之儲存，使得其可被另一指令存取作為來源運算元（藉由另一指令指明相同位置）。

SIMD技術，比如具有包括x86、MMX™、串流SIMD延伸（SSE）、SSE2、SSE3、SSE4.1、及SSE4.2指令之指令集的由Intel®Core™處理器所採用者，已使應用性能上獲得顯著改善（Core™及MMX™為美國加州Santa Clara的英特爾公司之註冊商標）。已公開一組額外的未來SIMD延伸，稱為先進向量延伸（AVX）並使用VEX編碼方案。

【圖式簡單說明】

藉由參照說明及用來繪示本發明之實施例的附圖可最佳了解本發明。在圖中：

第1A圖為繪示根據本發明之一實施例的僅具有在向量親和指令格式中之指令的指令流之區塊圖；

第1B圖為繪示根據本發明之一實施例的具有在多個指令格式中之指令的指令流之區塊圖；

第2A圖為繪示根據本發明之一實施例的泛型向量親和指令格式及其之類別A指令模板的區塊圖；

第2B圖為繪示根據本發明之一實施例的泛型向量親和指令格式及其之類別B指令模板的區塊圖；

第3A圖為繪示根據本發明之一實施例之一示範特定向量親和指令格式的區塊圖；

第3B圖為繪示根據本發明之一實施例的構成全運算碼欄位274之特定向量親和指令格式300的欄位之區塊圖；

第3C圖為繪示根據本發明之一實施例的構成暫存器索引欄位244之特定向量親和指令格式300的欄位之區塊圖；

第3D圖為繪示根據本發明之一實施例的構成擴充運算欄位250之特定向量親和指令格式300的欄位之區塊圖；

第4A圖為顯示根據本發明之一實施例的向量親和指令格式的一些欄位的相互關係之流程圖的一部分；

第4B圖為顯示根據本發明之一實施例的向量親和指令格式的一些欄位的相互關係之流程圖的第二部分；

第4C圖為顯示根據本發明之一實施例的向量親和指令格式的一些欄位的相互關係之流程圖的第三部分；

第4D圖為顯示根據本發明之一實施例的向量親和指令格式的一些欄位的相互關係之流程圖的其餘部分；

第4E圖為根據本發明之一實施例的區塊415A至H之各者的放大圖；

第5圖繪示根據本發明之一實施例的暫存器架構500之區塊圖；

第6A圖為繪示根據本發明之實施例的無記憶體存取類型運算之暫存器索引欄位244的流程圖；

第6B圖為繪示根據本發明之實施例的記憶體存取類型運算之暫存器索引欄位244、標度欄位260、位移欄位262A、及位移因子欄位262B之使用的流程圖；

第6C圖為繪示根據本發明之實施例的在disp8、disp32、及定標位移之變異間的差別之表；

第7A圖為繪示根據本發明之實施例的可由捨入控制欄位254A指明的該群可能的運算之表；

第7B圖為繪示根據本發明之實施例的可由捨入控制欄位259A指明的該群可能的運算之表；

第8A圖為繪示根據本發明之實施例的當資料元件寬度為64位元時可由資料變換欄位254B指明之一群可能的資料變換運算的表；

第8B圖為繪示根據本發明之實施例的當資料元件寬度為32位元時可由資料變換欄位254B指明之一群可能的資料

變換運算的表；

第9圖為繪示根據本發明之實施例的交叉乘積拌和815的區塊圖；

第10A圖為繪示根據本發明之實施例的跨四元件封包廣播一元件820之區塊圖；

第10B圖為繪示根據本發明之實施例的針對32位元資料元件寬度的1元件粒度之廣播的區塊圖；

第10C圖為繪示根據本發明之實施例的針對32位元資料元件寬度的4元件粒度之廣播的區塊圖；

第11A圖為繪示根據本發明之實施例的可由運算碼映圖欄位指明之該群可能的運算碼映圖的表；

第11B圖為繪示根據本發明之實施例的可由運算碼映圖欄位指明之該群可能的前綴編碼的表；

第12A圖為繪示根據本發明之實施例的可分別由資料操縱欄位254C及廣播欄位257B所指明之針對其中資料元件寬度為64位元的載入/運算整數之可能的資料操縱運算群的表；

第12B圖為繪示根據本發明之實施例的可分別由資料操縱欄位254C及廣播欄位257B所指明之針對其中資料元件寬度為32位元的載入/運算整數之可能的資料操縱運算群的表；

第12C圖為繪示根據本發明之實施例的可分別由資料操縱欄位254C及廣播欄位257B所指明之針對其中資料元件寬度為64位元的載入/運算浮點之可能的資料操縱運算

群的表；

第12D圖為繪示根據本發明之實施例的可分別由資料操縱欄位254C及廣播欄位257B所指明之針對其中資料元件寬度為32位元的載入/運算浮點之可能的資料操縱運算群的表；

第13A圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為64位元的載入整數之可能的資料操縱運算群的表；

第13B圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為32位元的載入整數之可能的資料操縱運算群的表；

第13C圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為64位元的載入浮點之可能的資料操縱運算群的表；

第13D圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為32位元的載入浮點之可能的資料操縱運算群的表；

第14A圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為64位元的儲存整數之可能的資料操縱運算群的表；

第14B圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為32位元的儲存整數之可能的資料操縱運算群的表；

第14C圖為繪示根據本發明之實施例的可由資料操縱

欄位 254C 所指明之針對其中資料元件寬度為 64 位元的儲存浮點之可能的資料操縱運算群的表；

第 14D 圖為繪示根據本發明之實施例的可由資料操縱欄位 254C 所指明之針對其中資料元件寬度為 32 位元的儲存浮點之可能的資料操縱運算群的表；

第 15A 圖為繪示根據本發明之實施例的可由資料操縱欄位 254C 所指明之針對其中資料元件寬度為 32 位元的載入圖形之可能的資料操縱運算群的表；

第 15B 圖為繪示根據本發明之實施例的可由資料操縱欄位 254C 所指明之針對其中資料元件寬度為 32 位元的載入分包圖形之可能的資料操縱運算群的表；

第 15C 圖為繪示根據本發明之實施例的可由資料操縱欄位 254C 所指明之針對其中資料元件寬度為 32 位元的儲存圖形之可能的資料操縱運算群的表；

第 16A 圖為繪示根據本發明之實施例的一示範運算 1600 的區塊圖，其使用寫入遮罩暫存器 K1 中之寫入遮罩來合併，其中資料元件寬度為 32 位元且其中第二來源及目的地為相同；

第 16B 圖為繪示根據本發明之實施例的一示範運算 1625 之區塊圖，其使用皆為一的固線遮罩（由指明寫入遮罩暫存器 K0 的指令使用該固線遮罩）來合併，其中資料元件寬度為 32 位元且其中第二來源及目的地為相同；

第 16C 圖為繪示根據本發明之實施例的針對 8、16、32、及 64 位元資料元件寬度之寫入遮罩暫存器中之位元對

一個512位元向量之資料元件位置的相應性；

第16D圖為繪示根據本發明之實施例的一示範運算1660的區塊圖，其使用寫入遮罩暫存器K1中之寫入遮罩來合併，其中資料元件寬度為32位元且其中第二來源及目的地為不同；

第16E圖為繪示根據本發明之實施例的一示範運算1666的區塊圖，其使用寫入遮罩暫存器K1中之寫入遮罩來歸零，其中資料元件寬度為32位元且其中第二來源及目的地為不同；

第17A圖繪示根據本發明之實施例的來自一示範特定向量親和指令格式的欄位之子集；

第17B圖繪示根據本發明之實施例的來自編碼於第17A圖之特定向量親和指令格式中之一示範特定向量親和指令的欄位之子集；

第17C圖繪示根據本發明之實施例的來自編碼於第17A圖之特定向量親和指令格式中之一示範特定向量親和指令的欄位之子集；

第17D圖繪示根據本發明之實施例的來自編碼於第17A圖之特定向量親和指令格式中之一示範特定向量親和指令的欄位之子集；

第18A圖繪示根據本發明之實施例的來自一示範特定向量親和指令格式的欄位之子集；

第18B圖繪示根據本發明之實施例的來自編碼於第18A圖之特定向量親和指令格式中之一示範特定向量親和

指令的欄位之子集；

第 18C 圖繪示根據本發明之實施例的來自編碼於第 18A 圖之特定向量親和指令格式中之一示範特定向量親和指令的欄位之子集；

第 18D 圖繪示根據本發明之實施例的來自編碼於第 18A 圖之特定向量親和指令格式中之一示範特定向量親和指令的欄位之子集；

第 18E 圖繪示根據本發明之實施例的來自編碼於第 18A 圖之特定向量親和指令格式中之一示範特定向量親和指令的欄位之子集；

第 18F 圖繪示根據本發明之實施例的來自編碼於第 18A 圖之特定向量親和指令格式中之一示範特定向量親和指令的欄位之子集；

第 19A 圖為繪示根據本發明之實施例的類別 A 之無記憶體存取全捨入控制類型運算 210 指令模板的哪個欄位係利用於第一示範處理器管線的不同階段中之區塊圖；

第 19B 圖為繪示根據本發明之實施例的類別 A 之無記憶體存取資料變換類型運算 215 指令模板的哪個欄位係利用於第一示範處理器管線的不同階段中之區塊圖；

第 19C 圖為繪示根據本發明之實施例的類別 A 之記憶體存取 220 指令模板的哪個欄位係利用於第一示範處理器管線的不同階段中之區塊圖；

第 20A 圖為繪示根據本發明之實施例的類別 A 之無記憶體存取全捨入控制類型運算 210 指令模板的哪個欄位係

利用於第二示範處理器管線的不同階段中之區塊圖；

第20B圖為繪示根據本發明之實施例的類別A之無記憶體存取資料變換類型運算215指令模板的哪個欄位係利用於第二示範處理器管線的不同階段中之區塊圖；

第20C圖為繪示根據本發明之實施例的類別A之記憶體存取220指令模板的哪個欄位係利用於第二示範處理器管線的不同階段中之區塊圖；

第21A圖為繪示根據本發明之實施例的類別A之無記憶體存取全捨入控制類型運算210指令模板的哪個欄位係利用於第三示範處理器管線的不同階段中之區塊圖；

第21B圖為繪示根據本發明之實施例的類別A之無記憶體存取資料變換類型運算215指令模板的哪個欄位係利用於第三示範處理器管線的不同階段中之區塊圖；

第21C圖為繪示根據本發明之實施例的類別A之記憶體存取220指令模板的哪個欄位係利用於第三示範處理器管線的不同階段中之區塊圖；

第22A圖為繪示根據本發明之實施例的類別A之無記憶體存取全捨入控制類型運算210指令模板的哪個欄位係利用於第四示範處理器管線的不同階段中之區塊圖；

第22B圖為繪示根據本發明之實施例的類別A之無記憶體存取資料變換類型運算215指令模板的哪個欄位係利用於第四示範處理器管線的不同階段中之區塊圖；

第22C圖為繪示根據本發明之實施例的類別A之記憶體存取220指令模板的哪個欄位係利用於第四示範處理器

管線的不同階段中之區塊圖；

第23A圖為根據本發明之實施例的單一CPU核心連同其至晶粒上互連網路2302的連結及其之二階（L2）快取2304的本地子集之區塊圖；

第23B圖為根據本發明之實施例的第23A圖中的CPU核心的一部分之放大圖；

第24圖為繪示根據本發明之實施例的一示範無序處理器架構之區塊圖；

第25圖為根據本發明之一實施例的系統2500的區塊圖；

第26圖為根據本發明之一實施例的第二系統2600之區塊圖；

第27圖為根據本發明之一實施例的第三系統2700之區塊圖；

第28圖為根據本發明之一實施例的SoC 2800之區塊圖；

第29圖為根據本發明之一實施例的具有整合記憶體控制器及圖形之單核心處理器及多核心處理器2900的區塊圖；及

第30圖為根據本發明之實施例的對比使用軟體指令轉換器來將來源指令集中的二元指令轉換成目的地指令集中之二元指令的區塊圖。

【發明內容及實施方式】

在下列說明中，提出比如邏輯實作、運算碼、指明運算元之方式、資源劃分/共享/複製實作、系統組件的類型及相互關係、及邏輯劃分/整合選擇的諸多特定細節以提供本發明之更詳盡的理解。然而，熟悉此技藝人士可認知可在無這種特定細節下實行本發明。在其他例子中，控制結構、閘級電路、及完全的軟體指令序列並為詳細顯示以不混淆本發明。此技藝中具有通常知識者，以所包括的說明，將能夠實現適當功能而不用過度實驗。

也應當認知在整份說明書中對於例如「一實施例 (one embodiment 或 an embodiment)」或「一或更多實施例」的參照意指特定特徵可包括在本發明之實施例的實行中，但每一個實施例不一定都得包括該特定特徵。類似地，應當認知在說明中有時將各種特徵集結在一起於單一實施例、圖示、或其之說明中，以便於揭露之流暢並輔助各個發明態樣的理解。此外，當連同一實施例敘述一特定特徵、結構、或特性時，連同無論是否明確敘述的其他實施例實現這種特徵、結構、或特性應係在熟悉此技藝人士的知識內。然而，此揭露方法不應詮釋為反映本發明需要比每一個申請專利範圍中所明確敘述更多的特徵之意圖。更確切地，如下列申請專利範圍反映出來，發明態樣係在於單一揭露實施例之所有特徵更少中。因此，在實施方式後之申請專利範圍在此明確併入到實施方式中，每一個申請專利範圍獨立成立為本發明的一分別實施例。

在下列說明及申請專利範圍中，可能使用術語「耦合

的」及「連接的」，還有其衍生詞。應了解的是這些術語非意圖為彼此的同義詞。「耦合的」用來表示兩或更多元件，其可或可不彼此直接實體或電性接觸，彼此合作或互動。「連接的」用來表示彼此耦合的兩或更多元件之間的通訊之建立。

將參照區塊圖之示範實施例敘述流程圖的操作。然而，應了解的是可藉由除了參照區塊圖所述的那些以外之本發明的實施例履行流程圖的操作，且參照區塊圖所述的實施例可履行與參照流程圖所述的那些不同之操作。

為了便於理解，在圖中已用虛線來代表某元件之可選本質（例如，不被本發明的一給定實作支援的特徵；被一給定實作支援的特徵但用於某些而非其他情況）。

向量親和指令格式－第1至16圖

向量親和指令格式為適合用於向量指令的指令格式（例如，有某些針對向量運算之欄位）。雖敘述其中透過向量親和指令格式支援向量及純量運算兩者的實施例，替代實施例僅使用向量運算向量親和指令格式。

指令格式數量－第1A及1B圖

一個指令格式－第1A圖

第1A圖為繪示根據本發明之一實施例的僅具有在向量親和指令格式中之指令的指令流之區塊圖。該指令流包括皆在向量親和格式中之一序列的J個指令100A至100J。在

本發明的一實施例中，處理器僅支援向量指令格式並可執行此指令流。

多個指令格式－第1B圖

第1B圖為繪示根據本發明之一實施例的具有在多個指令格式中之指令的指令流之區塊圖。在向量親和指令格式、第二格式、或第三格式中表示該指令流中的每一個指令。該指令流包括J個指令110A至110J。在本發明的一實施例中，處理器支援多個指令格式（包括第1B圖中所示的格式）並可執行第1A及1B圖兩者中之指令流。

示範泛型向量親和指令格式－第2A及2B圖

第2A及2B圖為繪示根據本發明之一實施例的泛型向量親和指令格式及其之指令模板之區塊圖。第2A圖為繪示根據本發明之一實施例的泛型向量親和指令格式及其之類別A指令模板之區塊圖；而第2B圖為繪示根據本發明之一實施例的泛型向量親和指令格式及其之類別B指令模板之區塊圖。詳言之，泛型向量親和指令格式200界定有類別A及類別B指令模板，這兩者都包括無記憶體存取205指令模板及記憶體存取220指令模板。在向量親和指令格式的上下文中的泛型一詞意指不綁到任何特定指令集之指令格式。雖將敘述一些實施例，其中向量親和指令格式中的指令在取源自暫存器（無記憶體存取205指令模板）或暫存器/記憶體（記憶體存取220指令模板）之向量上運算，但

本發明之替代實施例可僅支援這些之一。並且，雖將敘述一些實施例，其中在向量指令格式中有載入及儲存指令，但替代實施例取代或額外地具有將向量移動到暫存器中或移出暫存器（例如，從記憶體到暫存器中、從暫存器到記憶體中、在暫存器之間）的在不同指令格式中的指令。此外，雖將敘述本發明實施例支援兩種類別的指令模板，但替代實施例可支援這些之僅一種或超過兩種。

雖將敘述一些實施例，其中向量親和指令格式支援下列：具有32位元（4位元組）或64位元（8位元組）的資料元件寬度（或大小）之64位元組向量運算元長度（或大小）（並因此，64位元組向量由16個雙字大小元件或者8個四字大小元件所構成）；具有16位元（2位元組）或8位元（1位元組）的資料元件寬度（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8位元（1位元組）的資料元件寬度（或大小）之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）的資料元件寬度（或大小）之16位元組向量運算元長度（或大小）；替代實施例可支援具有更多、更少、及/或不同的資料元件寬度（例如，128位元（16位元組）資料元件寬度）之更多、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）。

第2A圖中之類別A指令模板包括：1）在無記憶體存

取205指令模板內顯示無記憶體存取全捨入控制類型運算210指令模板及無記憶體存取資料變換類型運算215指令模板；及2）在記憶體存取220指令模板內顯示記憶體存取時間性225指令模板及記憶體存取非時間性230指令模板。第2B圖中之類別B指令模板包括：1）在無記憶體存取205指令模板內顯示無記憶體存取寫入遮罩控制部分捨入控制類型運算212指令模板及無記憶體存取寫入遮罩控制vsize類型運算217指令模板；及2）在記憶體存取220指令模板內顯示記憶體存取寫入遮罩控制227指令模板。

格式

泛型向量親和指令格式200包括以第2A及2B圖中所示的順序於下列出之以下欄位。

格式欄位240－在此欄位中之特定值（指令格式識別符值）獨特識別向量親和指令格式，並且因此在指令流中的向量親和指令格式中之指令的出現。因此，格式欄位240的內容區別在第一指令格式中之指令的出現與在其他指令格式中之指令的出現，藉此允許將向量親和指令格式引進到具有其他指令格式的指令集中。依此，此欄位為可選，意思係針對僅具有泛型向量親和指令格式的指令集而言其為非必要。

基礎運算欄位242－其之內容區別不同的基礎運算。將於本文稍後敘述，基礎運算欄位242可包括運算碼欄位及/或為運算碼欄位之一部分。

暫存器索引欄位 244 – 其之內容，直接或透過位址產生，指明來源和目的地運算元的位置，無論係在暫存器中或在記憶體中。這些包括足夠數量的位元以從 $P \times Q$ （例如 32×512 ）暫存器檔選擇 N 暫存器。雖在一實施例中 N 可為高達三個來源及一個目的地暫存器，替代實施例可支援更多或更少的來源和目的地暫存器（例如，可支援高達兩個來源，其中這些來源之一也充當目的地；可支援高達三個來源，其中這些來源之一也充當目的地；可支援高達兩個來源及一個目的地）。雖在一實施例中 $P=32$ ，替代實施例可支援更多或更少的暫存器（例如 16）。雖在一實施例中 $Q=512$ 位元，替代實施例可支援更多或更少的位元（例如，128 或 1024）。

修飾符欄位 246 – 其之內容區別在泛型向量指令格式中指明記憶體存取與不指明的指令之出現；亦即，區別無記憶體存取 205 指令模板及記憶體存取 220 指令模板。記憶體存取運算讀取及/或寫入到記憶體階級（在一些情況中使用暫存器中的值來指明來源及/或目的地位址），而無記憶體存取運算則不（例如，來源及目的地為暫存器）。雖在一實施例中，此欄位亦在履行記憶體位址計算的三個不同方式之間做選擇，替代實施例可支援履行記憶體位址計算的更多、更少、或不同方式。

擴充運算欄位 250 – 其之內容區別除了基礎運算之外還要履行各種不同運算的哪一個。此欄位為上下文特定。在本發明之一實施例中，此欄位分成類別欄位 268、阿爾

發（alpha）欄位252、及貝他（beta）欄位254。擴充運算欄位允許在單一指令而非2、3、或4個指令中履行共同運算群。以下為使用擴充欄位250來減少所需指令數量的指令之一些範例（在本文稍後詳細敘述其之命名）。

先前指令序列	根據本發明之一實施例的指令序列
vaddps ymm0, ymm1, ymm2	vaddps zmm0, zmm1, zmm2
vpshufd ymm2, ymm2, 0x55 vaddps ymm0, ymm1, ymm2	vaddps zmm0, zmm1, zmm2 {bbbb}
vpmovsxbd ymm2, [rax] vcvtdq2ps ymm2, ymm2 vaddps ymm0, ymm1, ymm2	vaddps zmm0, zmm1, [rax]{sint8}
vpmovsxbd ymm3, [rax] vcvtdq2ps ymm3, ymm3 vaddps ymm4, ymm2, ymm3 vblendvps ymm1, ymm5, ymm1, ymm4	vaddps zmm1 {k5}, zmm2, [rax]{sint8}
vmaskmovps ymm1, ymm7, [rbx] vbroadcastss ymm0, [rax] vaddps ymm2, ymm0, ymm1 vblendvps ymm2, ymm2, ymm1, ymm7	vmovaps zmm1 {k7}, [rbx] vaddps zmm2 {k7}{z}, zmm1, [rax]{1toN}

其中[rax]為用於位址產生之基礎指標，且其中{}表示資料操縱欄位所指明之轉換運算（容後詳述）。

標度欄位260－其之內容允許索引欄位的內容針對記憶體位址產生而定標（例如，針對使用 $2^{\text{標度}} \times \text{索引} + \text{基礎}$ 之

位址產生）。

位移欄位 262A – 其之內容用為記憶體位址產生之一部分（例如，針對使用 $2^{\text{標度}} * \text{索引} + \text{基礎} + \text{位移}$ 之位址產生）。

位移因子欄位 262B（注意到位移欄位 262A 並列於位移因子欄位 262B 正上方指示使用其中之一）– 其之內容用為位址產生之一部分；其指明將被記憶體存取的大小（N）所定標之位移因子 – 其中 N 為記憶體存取中之位元組的數量（例如，針對使用 $2^{\text{標度}} * \text{索引} + \text{基礎} + \text{經定標位移}$ 之位址產生）。忽略冗餘低階位元並因此，將位移因子欄位的內容乘以記憶體運算元總大小（N）以產生用來計算有效位址之最終位移。N 的值係由處理器硬體在運行時間基於諸如全運算碼欄位 274（容後敘述）及資料操縱欄位 254C（容後敘述）等欄位而定。位移欄位 262A 及位移因子欄位 262B 為可選，意思係針對無記憶體存取 205 指令模板不使用它們及 / 或不同實施例可實現僅兩者之一或都沒有。

資料元件寬度欄位 264 – 其之內容區別若干資料元件寬度中何者將被使用（在一些實施例中針對所有指令；在其他實施例中僅指令之一些）。此欄位為可選，意思係若僅支援一個資料元件寬度則無需其及 / 或使用運算碼之某些態樣來支援資料元件寬度。

寫入遮罩欄位 270 – 其之內容控制，在每個資料元件位置的基礎上，目的地向量運算元中的資料元件位置是否

反映基礎運算及擴充運算的結果。類別A指令模板支援合併－寫入遮蓋，而類別B指令模板支援合併－及歸零－寫入遮蓋。當合併時，向量遮罩允許在（由基礎運算及擴充運算所指）任何運算的執行期間保護目的地中的任何元件集不被更新；在其他實施例中，保留相應遮罩位元具有0的目的地之各元件之舊值。相反地，當歸零時，向量遮罩允許在（由基礎運算及擴充運算所指）任何運算的執行期間歸零目的地中的任何元件集；在一實施例中，當相應遮罩位元具有0值時，目的地的元件設定成0。此功能的子集為控制被履行之運算的向量長度之能力（亦即，元件跨距被修改，從第一到最後一個）；然而，被修改的元件不需為連續的。因此，寫入遮罩欄位270允許部分向量運算，包括載入、儲存、算術、邏輯等等。並且，此遮蓋可用於錯誤抑制（亦即，藉由遮蓋目的地之資料元件位置來防止接收可能/將導致錯誤之任何運算的結果－例如，假設在記憶體中之一向量跨頁邊界且第一頁但非第二頁會導致頁錯誤，若以寫入遮罩遮蓋位在第一頁上的該向量之所有資料元件，則可忽略該頁錯誤）。此外，寫入遮罩允許含有某類型的條件語句之「向量化迴路」。雖敘述本發明之一些實施例，其中寫入遮罩欄位270的內容選擇含有將使用的寫入遮罩的若干寫入遮罩暫存器之一（並因此寫入遮罩欄位270的內容間接識別將履行的遮蓋），替代實施例取代或額外地允許寫入遮罩欄位270的內容直接指明將履行的遮蓋。另外，歸零允許性能改善，當1）將暫存器重命

名用於其之目的地運算元也並非係來源的指令（也稱為非三元指令）上時，因為在暫存器重命名管線階段中目的地不再為隱含的來源（當前目的地的任何資料元件無需被複製到重命名的目的地暫存器或以某種方式與運算一起帶著，因為並非為運算結果的任何資料元件（任何被遮蓋的資料元件）將會被歸零）；及2）在寫回階段中時，因為寫入零。

立即（immediate）欄位272－其之內容允許立即的指明。此欄位為可選，意思係在不支援立即的泛型向量親和格式的實作中其不會存在且在不使用立即的指令中其不會存在。

指令模板類別選擇

類別欄位268－其之內容在不同類別的指令之間做區別。參照第2A及2B圖，此欄位的內容在類別A及類別B之間做選擇。在第2A及2B圖中，圓角正方形用來指示特定值存在於欄位中（例如，在第2A及2B圖中針對類別欄位268分別為類別A 268A及類別B 268B）。

類別A之無記憶體存取指令模板

在類別A之無記憶體存取205指令模板的情況中，將阿爾發欄位252解譯為RS欄位252A，其之內容區別將履行不同擴充運算類型的哪一個（例如，捨入252A.1及資料變換252A.2分別指明無記憶體存取全捨入控制類型運算210及

無記憶體存取資料變換類型運算215指令模板)，而貝他欄位254區別將履行所指類型的運算之哪一個。在第2圖中，圓角區塊用來指示存在特定值（例如，在修飾符欄位246中之無記憶體存取246A；阿爾發欄位252/rs欄位252A的捨入252A.1及資料變換252A.2）。在無記憶體存取205指令模板中，標度欄位260、位移欄位262A、及位移因子欄位262B不存在。

無記憶體存取指令模板－全捨入控制類型運算

在無記憶體存取全捨入控制類型運算210指令模板中，將貝他欄位254解譯為捨入控制類型欄位254A，其之（諸）內容提供靜態捨入。雖在本發明的所述實施例中，捨入控制類型欄位254A包括抑制所有浮點異常（SAE）欄位256及捨入運算欄位258，替代實施例可支援將這兩個概念編碼到相同欄位中或僅有這些概念/欄位之一或另一（例如，可僅具有捨入運算欄位258）。

SAE欄位256－其之內容區別是否禁能異常事件通報；當SAE欄位256的內容指示致能抑制時，給定指令不會通報任何種類的浮點異常旗標並不會產生任何浮點異常處置器。

捨入運算欄位258－其之內容區別將履行一群捨入運算的哪一者（例如，向上捨入、向下捨入、朝零捨入、及朝最近捨入）。因此，捨入運算欄位258允許在每個指令的基礎上改變捨入模式，且因此當需要時會特別有用。在

其中處理器包括指明捨入模式的控制暫存器的本發明之一實施例中，捨入運算欄位258的內容凌駕於那個暫存器值之上（能夠選擇捨入模式而無需在此種控制暫存器上履行保存－修改－恢復很有益處）。

無記憶體存取指令模板－資料變換類型運算

在無記憶體存取資料變換類型運算215指令模板中，將貝他欄位254解譯為資料變換欄位254B，其之內容區別將履行若干資料變換之哪一個（例如，無資料變換、拌和（swizzle）、廣播）。

類別A之記憶體存取指令模板

在類別A的記憶體存取220指令模板中，將阿爾發欄位252解譯為驅逐提示欄位252B，其之內容區別將使用驅逐提示的哪一個（在第2A圖中，針對記憶體存取時間性225指令模板及記憶體存取非時間性230指令模板分別指明時間性252B.1及非時間性252B.2），而將貝他欄位254解譯為資料操縱欄位254C，其之內容區別將履行若干資料操縱運算（亦稱為基元）的哪一個（例如，無操縱；廣播；來源之向上轉換；目的地之向下轉換）。記憶體存取220指令模板包括標度欄位260，且可選地位移欄位262A或位移因子欄位262B。

向量記憶體指令履行從記憶體之向量載入及至記憶體的向量儲存，具有轉換支援。針對正常向量指令，向量記

記憶體指令以資料元件方式往返記憶體傳送資料，其中由選擇為寫入遮罩的向量遮罩之內容決定實際傳送的元件。在第2A圖中，圓角正方形用來指示特定值存在於欄位中（例如，針對修飾符欄位246之記憶體存取246B；阿爾發欄位252/驅逐提示欄位252B的時間性252B.1及非時間性252B.2）。

記憶體存取指令模板－時間性

時間性資料為可能很快又被重用而受益於快取之資料。然而，這係一個提示，且不同處理器可以不同方式實現該者，包括完全忽略提示。

記憶體存取指令模板－非時間性

非時間性資料為不可能很快又被重用而受益於第一階快取之資料且應給予驅逐之優先。然而，這係一個提示，且不同處理器可以不同方式實現該者，包括完全忽略提示。

類別B之指令模板

在類別B的指令模板之情況中，將阿爾發欄位252解譯為寫入遮罩控制（Z）欄位252C，其之內容區別由寫入遮罩欄位270所控制之寫入遮蓋是否應為合併或歸零。

類別B之無記憶體存取指令模板

在類別B的無記憶體存取205指令模板之情況中，將貝他欄位254的一部分解譯為RL欄位257A，其之內容區別將履行不同擴充運算類型的哪一個（例如，針對無記憶體存取寫入遮罩控制部分捨入控制類型運算212指令模板及無記憶體存取寫入遮罩控制VSIZE類型運算217指令模板分別指明捨入257A.1及向量長度（VSIZE）257A.2），而貝他欄位254的其餘部分則區別將履行所指明之類型的運算的哪一個。在第2圖中，圓角區塊用來指示存在特定值（例如，在修飾符欄位246中之無記憶體存取246A；針對RL欄位257A的捨入257A.1及VSIZE 257A.2）。在無記憶體存取205指令模板中，標度欄位260、位移欄位262A、及位移因子欄位262B不存在。

無記憶體存取指令模板－寫入遮罩控制部分捨入控制類型運算

在無記憶體存取寫入遮罩控制部分捨入控制類型運算212指令模板中，將貝他欄位254的其餘部分解譯為捨入運算欄位259A並禁能異常事件通報（給定指令不會通報任何種類的浮點異常旗標並不會產生任何浮點異常處置器）。

捨入運算欄位259A－如同捨入運算欄位258，其之內容區別將履行一群捨入運算的哪一者（例如，向上捨入、向下捨入、朝零捨入、及朝最近捨入）。因此，捨入運算欄位259A允許在每個指令的基礎上改變捨入模式，且因此當需要時會特別有用。在其中處理器包括指明捨入模式的

控制暫存器的本發明之一實施例中，捨入運算欄位259A的內容凌駕於那個暫存器值之上（能夠選擇捨入模式而無需在這種控制暫存器上履行保存－修改－恢復很有益處）。

無記憶體存取指令模板－寫入遮罩控制VSIZE類型運算

在無記憶體存取寫入遮罩控制VSIZE類型運算217指令模板中，將貝他欄位254的其餘部分解譯為向量長度欄位259B，其之內容區別將履行若干資料向量長度之哪一個（例如，128、256、或512位元）。

類別B之記憶體存取指令模板

在類別B的記憶體存取220指令模板的情況中，將貝他欄位254的一部分解譯為廣播欄位257B，其之內容區別是否將履行廣播類型資料操縱運算，而將貝他欄位254的其餘部分解譯為向量長度欄位259B。記憶體存取220指令模板包括標度欄位260，且可選地位移欄位262A或位移因子欄位262B。

有關於欄位之額外註解

關於泛型向量親和指令格式200，顯示全運算碼欄位274，包括格式欄位240、基礎運算欄位242、及資料元件寬度欄位264。雖顯示一實施例，其中全運算碼欄位274包括所有這些欄位，在不支援它們全部的實施例中，全運算碼欄位274包括少於這些欄位的全部。全運算碼欄位274提

供運算碼。

擴充運算欄位250、資料元件寬度欄位264、及寫入遮罩欄位270允許在泛型向量親和指令格式中在每個指令基礎上指明這些特徵。

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

指令格式需要相對小量的位元，因其基於其他欄位的內容針對不同目的重用不同欄位。例如，一個觀點係修飾符欄位的內容在第2A及2B圖上的無記憶體存取205指令模板與在第2A及2B圖上的記憶體存取220指令模板之間做選擇；而類別欄位268的內容在那些記憶體存取220指令模板內之第2A圖的指令模板210/215與第2B圖的指令模板212/217之間做選擇；且同時類別欄位268的內容在那些記憶體存取220指令模板內之第2A圖的指令模板225/230與第2B圖的指令模板227之間做選擇。從另一觀點來看，類別欄位268的內容在分別第2A及2B圖的類別A及類別B指令模板之間做選擇；而修飾符欄位的內容在那些類別A指令模板內在第2A圖的指令模板205與220之間做選擇；且同時修飾符欄位的內容在那些類別B指令模板內在第2B圖的指令模板205與220之間做選擇。在類別欄位的內容指示類別A指令模板的情況中，修飾符欄位246的內容選擇阿爾發欄位252的解譯（在rs欄位252A及驅逐提示（EH）欄位252B之間）。以相關的方式，修飾符欄位246及類別欄位268的內容選擇阿爾發欄位是否解譯為rs欄位252A、EH欄位

252B、或寫入遮罩控制（Z）欄位252C。在類別及修飾符欄位指示類別A無記憶體存取運算的情況中，擴充欄位的貝他欄位之解譯會隨rs欄位的內容而變；同時在類別及修飾符欄位指示類別B無記憶體存取運算的情況中，貝他欄位之解譯取決於RL欄位的內容。在類別及修飾符欄位指示類別A記憶體存取運算的情況中，擴充欄位的貝他欄位之解譯會隨基礎運算欄位的內容而變；同時在類別及修飾符欄位指示類別B記憶體存取運算的情況中，擴充欄位的貝他欄位之廣播欄位257B的解譯會隨基礎運算欄位的內容而變。因此，基礎運算欄位、修飾符欄位、及擴充運算欄位的組合允許指明甚至更廣泛種類的擴充運算。

在類別A及類別B內可見的各種指令模板在不同情況中有益。當爲了性能的緣故而希望歸零寫入遮蓋或較小向量長度，類別B有用。例如，歸零允許避免當使用重命名時假相依性，因爲不再需要與目的地人爲合併；舉另一例而言，當以向量遮罩仿真較短的向量大小時，向量長度控制減輕儲存－載入轉送問題。當希望：1）允許浮點異常（亦即，當SAE欄位的內容指示否）同時使用捨入模式控制；2）能夠使用向上轉換、拌和、互換、及/或向下轉換；3）在圖形資料類型上運算時，類別A有用。例如，向上轉換、拌和、互換、向下轉換、及圖形資料類型減少當與在不同格式中的來源處理時所需指令的數量；舉另一例而言，允許異常的能力提供和直接捨入模式的完全IEEE符合。

示範特定向量親和指令格式－第3A至3D圖

第3A圖為繪示根據本發明的一實施例之一示範特定向量親和指令格式的區塊圖。第3A圖顯示特定向量親和指令格式300，其為特定的，意思係其指明位置、大小、解譯、及欄位順序，還有那些欄位之一些的值。特定向量親和指令格式300可用來延伸x86指令集，且因此欄位之一些與現有x86指令集和其延伸（例如，AVX）中所用的那些類似或相同。此格式與具有延伸的現有x86指令集之前綴編碼欄位、真實運算碼位元組欄位、MOD R/M欄位、SIB欄位、位移欄位、及立即欄位一致。繪示映射第3A圖的欄位到其之中的第2圖的欄位。

應了解的是為了便於說明雖本發明之實施例在泛型向量親和指令格式200的上下文中參照特定向量親和指令格式300，本發明不限於特定向量親和指令格式300，除了在主張專利範圍之處。例如，泛型向量親和指令格式200考量各種欄位之各種可能大小，而特定向量親和指令格式300顯示成具有特定大小的欄位。舉特定範例來說，雖在特定向量親和指令格式300中資料元件寬度欄位264繪示為一位元欄位，本發明不如此受限（亦即，泛型向量親和指令格式200考量資料元件寬度欄位264的其他大小）。

格式－第3A圖

泛型向量親和指令格式200包括以第3A圖中所示的順

序於下列出的以下欄位。

EVEX前綴（位元組0-3）

EVEX前綴302係以四位元組形式加以編碼。

格式欄位240（EVEX位元組0，位元[7:0]）－第一位元組（EVEX位元組0）為格式欄位240，且其含有0x62（在本發明之一實施例中用於區別向量親和指令格式的獨特值）。

第二至第四位元組（EVEX位元組1至3）包括提供特定能力之若干位元欄位。

REX欄位305（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257BEX位元組1，位元[5]－B）所構成。EVEX.R、EVEX.X、EVEX.B位元欄位與相應VEX位元欄位提供相同的功能，且使用1的補數形式來加以編碼，亦即，將ZMM0編碼成1111B，ZMM15編碼成0000B。指令之其他欄位編碼暫存器索引的較低三個位元，如此技藝中已知（rrr、xxx、及bbb），所以可藉由添加EVEX.R、EVEX.X、及EVEX.B來形成Rrrr、Xxxx、Bbbb。

REX'欄位310－此為REX'欄位310之第一部分且為用來編碼延伸32暫存器集之上16或下16的EVEX.R'位元欄位（EVEX位元組1，位元[4]－R'）。在本發明之一實施例中，此位元，還有於下指示之其他者，係在位元反置格式

中儲存以和BOUND指令做區別（在眾所皆知的x86 32位元模式中），BOUND指令的真實運算碼位元組為62，但在MOD R/M欄位（下述）中不接受MOD欄位中之11的值；本發明的替代實施例不在位元反置格式中儲存此位元還有其他下列所指之位元。1的值用來編碼較低16個暫存器。換言之，藉由結合EVEX.R'、EVEX.R、及來自其他欄位的其他RRR形成R'Rrrr。

運算碼映圖欄位315（EVEX位元組1，位元[3:0]—mmmm）—其之內容編碼隱含領先的運算碼位元組（0F、0F 38、或0F 3A）。

資料元件寬度欄位264（EVEX位元組2，位元[7]—W）—由符號EVEX.W表示。EVEX.W用來界定資料類型的粒度（大小）（32位元資料元件或64位元資料元件）。

EVEX.vvvv 320（EVEX位元組2，位元[6:3]—vvvv）—EVEX.vvvv的角色可包括下列者：1）EVEX.vvvv編碼第一來源暫存器運算元，在反置（1的補數）形式中指明且針對2或更多來源運算元的指令有效；2）EVEX.vvvv編碼目的地暫存器運算元，針對某些向量移動在1的補數格式中指明；或者3）EVEX.vvvv不編碼任何運算元，保留該欄位並應含有1111b。因此，EVEX.vvvv欄位320編碼在反置（1的補數）格式中儲存之第一來源暫存器說明符的4個低階位元。根據指令，使用一額外的不同EVEX位元欄位來延伸說明符大小到32個暫存器。

EVEX.U 268類別欄位（EVEX位元組2，位元[2]—

U) – 若 $\text{EVEX.U} = 0$ ，其指示類別 A 或 EVEX.U0 ；若 $\text{EVEX.U} = 1$ ，其指示類別 B 或 EVEX.U1 。

前綴編碼欄位 325 (EVEX 位元組 2，位元 [1:0] – pp) – 為基礎運算欄位提供額外位元。除了在 EVEX 前綴格式中提供舊有 SSE 指令之支援外，此還具有密化 SIMD 前綴的益處（不像需要一位元組來表示 SIMD 前綴，EVEX 前綴僅需 2 個位元）。在一實施例中，欲在舊有格式及 EVEX 前綴格式兩者中支援使用 SIMD 前綴 (66H、F2H、F3H) 的舊有 SSE 指令，將這些舊有 SIMD 前綴編碼到 SIMD 前綴編碼欄位中；並在運行時間擴展到舊有 SIMD 前綴中，之後才提供到解碼器的 PLA（所以 PLA 可執行這些舊有指令之舊有及 EVEX 格式兩者而無修改）。雖較新的指令可直接使用 EVEX 前綴編碼欄位的內容作為運算元延伸，某些實施例為了一致而以類似方式擴展，但允許這些舊有 SIMD 前綴指明不同意思。一替代實施例可重設計 PLA 來支援 2 位元 SIMD 前綴編碼，並因此不需擴展。

阿爾發欄位 252 (EVEX 位元組 3，位元 [7] – EH；亦稱為 EVEX.EH 、 EVEX.rs 、 EVEX.RL 、 EVEX. 寫入遮罩控制、及 EVEX.N ；亦以 α 繪示) – 如前述，此欄位為上下文特定的。在本文稍後提供額外說明。

貝他欄位 254 (EVEX 位元組 3，位元 [6:4] – SSS；亦稱為 EVEX.s_{2-0} 、 EVEX.r_{2-0} 、 EVEX.rr1 、 EVEX.LL0 、及 EVEX.LLB ；亦以 $\beta\beta\beta$ 繪示) – 如前述，此欄位為上下文特定的。在本文稍後提供額外說明。

REX'欄位310－此為REX'欄位之其餘者且為可用來編碼延伸32暫存器集之上16或下16的EVEX.V'位元欄位（EVEX位元組3，位元[3]－V'）。在位元反置格式中儲存此位元。1的值用來編碼下16個暫存器。換言之，藉由結合EVEX.V'及EVEX.vvvv來形成V'VVVV。

寫入遮罩欄位270（EVEX位元組3，位元[2:0]－kkk）－其之內容如前述般指明寫入遮罩暫存器中之一個暫存器的索引。在本發明之一實施例中，特定值EVEX.kkk=000具有特殊行為，意味著針對特定指令不使用寫入遮罩（這可以各種方式實現，包括使用固線到全部一的寫入遮罩或旁通遮蓋硬體的硬體）。

真實運算碼欄位330（位元組4）

此亦稱為運算碼位元組。在此欄位中指明運算碼的一部份。

MOD R/M欄位340（位元組5）

修飾符欄位246（MODR/M.MOD，位元[7-6]－MOD欄位342）－如前述，MOD欄位342的內容在記憶體存取及無記憶體存取運算之間做區別。在本文稍後中進一步敘述此欄位。

MODR/M.reg欄位344，位元[5-3]－ModR/M.reg欄位的角色可總結於兩種情況：ModR/M.reg編碼目的地暫存器運算元或來源暫存器運算元，或將ModR/M.reg看待成運算

碼延伸並不用來編碼任何指令運算元。

MODR/M.r/m 欄位 346，位元 [2-0] – ModR/M.r/m 欄位的角色可包括下列：ModR/M.r/m 編碼參照記憶體位址之指令運算元，或 ModR/M.r/m 編碼目的地暫存器運算元或來源暫存器運算元。

標度、索引、基礎（SIB）位元組（位元組 6）

標度欄位 260（SIB.SS，位元 [7-6]） – 如前述，標度欄位 260 的內容用於記憶體位址產生。在本文稍後中進一步敘述此欄位。

SIB.xxx 354（位元 [5-3]）及 SIB.bbb 356（位元 [2-0]） – 已先前相關於暫存器索引 Xxxx 及 Bbbb 提及這些欄位的內容。

（諸）位移位元組（位元組 7 或位元組 7 至 10）

位移欄位 262A（位元組 7 至 10） – 當 MOD 欄位 342 含有 10 時，位元組 7 至 10 為位移欄位 262A，且和舊有 32 位元位移（disp32）相同作用且在位元組粒度作用。

位移因子欄位 262B（位元組 7） – 當 MOD 欄位 342 含有 01 時，位元組 7 為位移因子欄位 262B。此欄位的位置和舊有 x86 指令集 8 位元位移（disp8）相同，其在位元組粒度作用。由於 disp8 為正負號延伸，其僅可定址於 -128 及 127 位元組偏移之間；就 64 位元組快取線而論，disp8 使用 8 位元，其僅可被設定成四個真正有用的值 -128、-64、

0、及 64；由於時常需要更大範圍，使用 disp32 ；然而， disp32 需 4 位元組。相較於 disp8 及 disp32 ，位移因子欄位 262B 係 disp8 的重新解譯；當使用位移因子欄位 262B 時，由位移因子欄位乘上記憶體運算元存取的大小（ N ）來判定實際的位移。這種位移稱為 $\text{disp8} * N$ 。這減少平均指令長度（針對位移使用單一位元組但有大上許多的範圍）。這種壓縮的位移係建立在有效位移為記憶體存取之粒度的倍數之假設上，且因此，不需編碼位址偏移的冗餘低階位元。換言之，該位移因子欄位 262B 取代舊有 x86 指令集 8 位元位移。因此，以與 x86 指令集 8 位元位移之相同方式編碼位移因子欄位 262B（所以在 ModRM/SIB 編碼規則中無改變），除了 disp8 超載至 $\text{disp8} * N$ 。換言之，在編碼規則或編碼長度中無改變，而僅在硬體對位移值的解譯中有變（其需以記憶體運算元之大小定標位移來獲得逐位元組之位址偏移）。

立即

立即欄位 272 如前述般操作。

全運算碼欄位 — 第 3B 圖

第 3B 圖為繪示根據本發明之一實施例的構成全運算碼欄位 274 之特定向量親和指令格式 300 的欄位之區塊圖。詳言之，全運算碼欄位 274 包括格式欄位 240、基礎運算欄位 242、及資料元件寬度（ W ）欄位 264。基礎運算欄位 242

包括前綴編碼欄位325、運算碼映圖欄位315、及真實運算碼欄位330。

暫存器索引欄位－第3C圖

第3C圖為繪示根據本發明之一實施例的構成暫存器索引欄位244之特定向量親和指令格式300的欄位之區塊圖。詳言之，暫存器索引欄位244包括REX欄位305、REX'欄位310、MODR/M.reg欄位344、MODR/M.r/m欄位346、VVVV欄位320、xxx欄位354、及bbb欄位356。

擴充運算欄位－第3D圖

第3D圖為繪示根據本發明之一實施例的構成擴充運算欄位250之特定向量親和指令格式300的欄位之區塊圖。當類別（U）欄位268含有0時，其代表EVEX.U0（類別A 268A）；當其含有1時，其代表EVEX.U1（類別B 268B）。當U=0且MOD 342欄位含有11時（代表無記憶體存取運算），將阿爾發欄位252（EVEX位元組3，位元[7]－EH）解譯為rs欄位252A。當rs欄位252A含有1（捨入252A.1）時，將貝他欄位254（EVEX位元組3，位元[6:4]－SSS）解譯為捨入控制欄位254A。捨入控制欄位254A包括一位元SAE欄位256及兩位元捨入運算欄位258。當rs欄位252A含有0時（資料變換252A.2），將貝他欄位254（EVEX位元組3，位元[6:4]－SSS）解譯為三位元資料變換欄位254B。當U=0且MOD 342欄位含有00、01、或10時

（代表記憶體存取運算），將阿爾發欄位 252（EVEX 位元組 3，位元 [7]－EH）解譯為驅逐提示（EH）欄位 252B 並將貝他欄位 254（EVEX 位元組 3，位元 [6:4]－SSS）解譯為三位元資料操縱欄位 254C。

當 $U=1$ 時，將阿爾發欄位 252（EVEX 位元組 3，位元 [7]－EH）解譯為寫入遮罩控制（Z）欄位 252C。當 $U=1$ 且 MOD 342 欄位含有 11 時（代表無記憶體存取運算），將貝他欄位 254 的一部分（EVEX 位元組 3，位元 [4]－ S_0 ）解譯為 RL 欄位 257A；當其含有 1（捨入 257A.1）時，將貝他欄位 254 的其餘部分（EVEX 位元組 3，位元 [6-5]－ S_{2-1} ）解譯為捨入運算欄位 259A，而當 RL 欄位 257A 含有 0（VSIZE 257A.2）時，將貝他欄位 254 的其餘部分（EVEX 位元組 3，位元 [6-5]－ S_{2-1} ）解譯為向量長度欄位 259B（EVEX 位元組 3，位元 [6-5]－ L_{1-0} ）。當 $U=1$ 且 MOD 342 欄位含有 00、01、或 10 時（代表記憶體存取運算），將貝他欄位 254（EVEX 位元組 3，位元 [6:4]－SSS）解譯為向量長度欄位 259B（EVEX 位元組 3，位元 [6-5]－ L_{1-0} ）及廣播欄位 257B（EVEX 位元組 3，位元 [4]－B）。

一些額外的點

向量格式延伸暫存器的數量到 32（REX'）。

非破壞性來源暫存器編碼（適用於三及四運算元語義）：此為指令語義中的第一來源運算元。其係由符號 EVEX.vvvv 表示。使用 1 的補數形式（反置形式）編碼此

欄位，亦即ZMM0編碼為1111B，ZMM15編碼為0000B。注意到需要EVEX中的一額外位元欄位來延伸來源到32個暫存器。

EVEX.W界定指令之資料類型大小（32位元或64位元）。

32個延伸暫存器集編碼：EVEX前綴提供額外位元欄位來以下列專用位元欄位編碼每來源之32個暫存器：EVEX.R'及EVEX.V'（連同暫存器－暫存器格式之EVEX.X）。

SIMD前綴的密化：舊有SSE指令實際上使用SIMD前綴（66H、F2H、F3H）作為運算碼延伸欄位。EVEX前綴編碼允許使用512位元向量長度的這種舊有SSE指令之功能能力。

兩位元組和三位元組運算碼之密化：較近來引進的舊有SSE指令採用兩及三位元組運算碼。一或兩個領先位元組為：0FH及0FH 3AH/0FH 38H。也可將一位元組逸出（escape）（0FH）及兩位元組逸出（0FH 3AH/0FH 38H）解譯為運算碼延伸欄位。EVEX.mmm欄位提供密化以允許許多舊有指令被編碼而無一貫的位元組序列0FH、0FH 3AH、0FH 38H。

顯示向量親和指令格式的一些欄位的相互關係之示範流程圖－第4A至4E圖

第4A至4D圖繪示根據本發明之一實施例的向量親和

指令格式的一些欄位的相互關係之流程圖；而第4E圖為根據本發明之一實施例的區塊415A至H之各者的放大圖。在區塊400中，判定初始欄位的值是否指示向量親和指令格式（例如，0x62）。若否，控制進至區塊402，其中根據指令集之其他格式之一來處置該指令。若是，控制進至區塊492。

在區塊492中，判定類別（U）欄位的內容是否指示類別A或類別B指令模板。在類別A的情況中，控制進至兩個分別的區塊：區塊404A及490。否則，控制通過圓形B至第4C圖上的兩個分別的區塊：區塊404B及493。

在區塊404A中，判定修飾符欄位的內容是否指示無記憶存取運算或記憶存取運算。在無記憶存取運算的情況中（例如MOD欄位342 = 11），控制進至區塊406及408。在記憶存取運算的情況中（例如MOD欄位342 = 00、01、或10），控制進至區塊422、區塊430、及區塊440A的每一者（在第4B圖上通過圓形A）。

標為阿爾發欄位252之圓角框涵蓋區塊408及區塊422，因為它們代表阿爾發欄位252的不同解譯。詳言之，區塊408代表阿爾發欄位252作為rs欄位252A的解譯，而區塊422代表阿爾發欄位252作為驅逐提示欄位252B的解譯。

在區塊406中，如第6A圖中所示般使用暫存器索引欄位244的內容。

在區塊408中，判定rs欄位252A的內容是否指示捨入類型運算（例如，rs欄位252A = 1）或資料變換類型運算

（例如，rs欄位252A= 0）。在前者，控制進至區塊410、區塊412A、及區塊414的各者。在後者情況中，控制進至區塊416。

標為貝他（捨入控制）欄位254A之圓角框涵蓋區塊410及區塊412A。區塊410繪示有關於SAE欄位256的內容之決定（是否抑制浮點異常），而區塊412A繪示基於捨入運算欄位258的內容之決定（區別一群可能的捨入運算之一）。在第7A圖中繪示在區塊410及區塊412A中所做的決定。

區塊414、416、442、448、454、460、468及474皆繪示關於資料元件寬度（w）欄位264之決定。如第4圖中所示，資料元件寬度欄位264為第3A圖之特定向量親和指令格式300中的1位元欄位。依此，這些區塊決定資料元件寬度是否為64位元（例如，1）或32位元（例如，0）。有關於區塊414，此決定標記流程之此分支的結束。相反地，針對64位元及32位元，控制從區塊416分別進至區塊418或區塊420。

標為貝他（資料變換）欄位254B之圓角框涵蓋區塊418及區塊420；並因此代表貝他欄位254解譯為資料變換欄位254B的情況。在區塊418及420中，資料變換欄位254B的內容用來區別將履行若干資料變換運算之哪一個。在第8A及8B圖中分別顯示針對區塊418及區塊420之一群可能的資料變換運算。

在區塊422中，驅逐提示欄位252B的內容用來區別應

使用該群可能的驅逐提示選項的哪一個。第4圖繪示從特定向量親和指令格式300之1位元驅逐提示欄位252B的使用。詳言之，驅逐提示選項為非時間性（1）及時間性（0）。此標記流程圖之此分支的結束。

在區塊430中，如第6B圖中所示般使用暫存器索引欄位244、標度欄位260、及位移欄位262A或位移因子欄位262B的內容。此標記流程圖之此分支的結束。

在區塊440A中，基礎運算欄位242的內容用來區別將履行一群不同記憶體存取運算之哪一者。下表繪示根據本發明之一實施例的被支援記憶體存取運算群，還有針對各者自區塊440A之控制流程。本發明之替代實施例可支援更多、更少、或不同的記憶體存取運算。

記憶體存取運算類型	區塊
載入/運算整數（load/op int）	442
載入/運算浮點（load/op fp）	448
載入整數（load int）	454
載入浮點（load fp）	460
儲存整數（store int）	468
儲存浮點（store fp）	474
載入圖形（load gr）	480
載入分包圖形（load p.gr）	482
儲存圖形（store gr）	484

如前述，區塊442、448、454、460、468、及474基於資料元件寬度判定控制流程中的改變；在下表中繪示控制流程。

區塊	64位元	32位元
442	如第12A圖中所示之444A	如第12B圖中所示之446A
448	如第12C圖中所示之450A	如第12D圖中所示之452A
454	如第13A圖中所示之456	如第13B圖中所示之458
460	如第13C圖中所示之462	如第13D圖中所示之464
468	如第14A圖中所示之470	如第14B圖中所示之472
474	如第14C圖中所示之476	如第14D圖中所示之478

類似地，在第15A、15B、及15C圖中分別繪示區塊480、482、及484之決定。標為貝他（資料操縱）欄位254C之圓角框涵蓋區塊444A、446A、450A、452A、456、458、462、464、470、472、476、478、480、482、及484；藉此繪示資料操縱欄位254C的內容區別將履行該群可能的資料操縱運算之哪一個。

在區塊490中，寫入遮罩（k）欄位270的內容及資料元件寬度（w）欄位264的內容用來判定將在運算中使用的寫入遮罩。第4圖繪示實施例，其中有八個寫入遮罩暫存器且暫存器000指示不應使用任何寫入遮罩。在寫入遮罩欄位270的內容指示非000的情況中，控制進至第16A至16D圖。

在區塊404B中，判定修飾符欄位的內容是否指示無記憶體存取運算或記憶體存取運算。在無記憶體存取運算的情況中（例如，MOD欄位342 = 11），控制進至區塊406（在第4A圖上通過圓形E）及495。在記憶體存取運算的情況中（例如，MOD欄位342 = 00、01、或10），控制進

至區塊498區塊430（在第4A圖上通過圓形D）、及區塊440B（在第4D圖上通過圓形C）的每一者。

標為貝他欄位254的一部分之圓角框涵蓋區塊495、區塊412B、及區塊498，因為它們代表貝他欄位254之一部分的不同解譯。詳言之，區塊495代表貝他欄位254的一部分作為RL欄位257A的解譯；而在第4D圖上標為廣播欄位257B的圓角框代表貝他欄位254的此部分作為廣播欄位257B的解譯。

在區塊495中，判定RL欄位257A的內容是否指示捨入類型運算（例如，RL欄位257A = 1）或向量長度類型運算（例如，RL欄位257A = 0）。在前者，控制進至區塊412B及區塊415A的各者。在後者情況中，控制進至區塊498及區塊415B的各者。

區塊412B繪示基於捨入運算欄位259A的內容之決定（區別該群可能的捨入運算之一）。在第7B圖中繪示在區塊412B中做出的決定。

區塊415A-H皆繪示有關於在其上運算之資料元件的寬度之決定。如所示，類別B之所支援的資料元件（當U = 1）為64位元、32位元、16位元、及8位元。履行這些區塊的示範方式在本文中稍後參照第4E圖加以說明。區塊415A-B分別標記流程圖之這些分支的結束。有關於區塊415A，以虛線顯示到16位元及8位元資料元件寬度的線，因為在本發明之一實施例中不支援這些；更確切地，若無在16位元或8位元資料元件上之類別B的記憶體存取類型運

算，則預期RL欄位257A的內容為0，且因此導致控制從區塊495流至區塊415B及498（換言之，部分捨入不可用）。

在區塊498中，向量長度（LL）欄位259B的內容用來判定將被運算之向量的大小。第4圖繪示其中支援下列者的實施例：1）128位元（00）、2）256位元（01）、512位元（10）、及1024位元（11）。

在區塊440B中，基礎運算欄位242的內容用來區別將履行一群不同記憶體存取運算之哪一者。下表繪示根據本發明之一實施例的被支援的記憶體存取運算群，還有針對各者自區塊440B之控制流程。本發明之替代實施例可支援更多、更少、或不同的記憶體存取運算。

記憶體存取運算類型	區塊
載入/運算整數（load/op int）	415C
載入/運算浮點（load/op fp）	415D
載入整數（load int）	415E
載入浮點（load fp）	415F
儲存整數（store int）	415G
儲存浮點（store fp）	415H

如前述，區塊415C-H基於資料元件寬度判定控制流程中的改變；在下表中繪示控制流程。

區塊	64位元	32位元	16位元	8位元
415C	如第12A圖中所 示之444B	如第12B圖中所 示之446B	流程的分支結束	流程的分支結束
415D	如第12C圖中所 示之450B	如第12D圖中所 示之452B	流程的分支結束	流程的分支結束
415E	流程的分支結束	流程的分支結束	流程的分支結束	流程的分支結束
415F	流程的分支結束	流程的分支結束	流程的分支結束	流程的分支結束
415G	流程的分支結束	流程的分支結束	流程的分支結束	流程的分支結束
415H	流程的分支結束	流程的分支結束	流程的分支結束	流程的分支結束

標為廣播欄位257B之圓角框涵蓋區塊444B、446B、450B、及452B；藉此繪示廣播欄位257B的內容區別是否將履行廣播運算。如所示，本發明之一實施例允許廣播（b）欄位257B的內容選擇是否針對64位元及32位元的資料元件寬度履行廣播運算，這對16位元和8位元資料元件寬度並非為選項；更確切地，若有在16位元或8位元資料元件上運算之類別B的記憶體存取類型運算，則預期廣播（B）欄位257B的內容為0。

在區塊493中，阿爾發欄位252之內容（寫入遮罩控制（Z）欄位252C之內容）、寫入遮罩（k）欄位270之內容、及資料元件寬度的判定用來判定將履行的寫入遮罩運算（合併或歸零）及運算中將使用的寫入遮罩。在本發明之一些實施例中，在履行儲存之記憶體存取運算上預期阿爾發欄位252（寫入遮罩控制，（Z欄位252C））為零（針對零遮蓋）。以和區塊415相同的方式進行資料元件寬度的判定。第4圖繪示實施例，其中有八個寫入遮罩暫

存器且暫存器000指示不應使用任何寫入遮罩。在寫入遮罩欄位270的內容指示非000的情況中，控制進至第16D至E圖。

第4E圖為根據本發明之一實施例的區塊415A-H的各者之放大圖。詳言之，繪示單一流程415，其代表區塊415A-H的各者之流程。在區塊417A中，真實運算碼欄位330的內容之一些或全部係用來在兩組資料元件寬度之間做選擇：第一組417A.1（例如，包括64位元及32位元）及第二組417A.2（例如，16位元及8位元）。雖如區塊417B中所示針對第一組417A.1係基於資料元件寬度（w）欄位264判定資料元件，在第二組417A.2內，判定資料元件寬度有兩種方式：417A.2.2（基於僅真實運算碼欄位330）及417A.2.1（如區塊417C中所示基於資料元件寬度（w）欄位264）。如第4圖中所示，資料元件寬度欄位264為第3A圖的特定向量親和指令格式300中之1位元欄位。因此，這些區塊417B決定資料元件寬度是否為64位元（例如，1）或32位元（例如，0）；而區塊417C決定資料元件寬度是否為16位元（例如，1）或8位元（例如，0）。雖第4E圖繪示真實運算碼欄位417A在判定資料元件寬度上的牽涉，可實現替代實施例以使用僅w欄位（例如，具有一位元w欄位並僅支援兩種資料元件大小；具有兩位元w欄位並支援四種資料元件大小）。

雖已參照第4圖敘述本發明之實施例，替代實施例可使用不同流程。例如，如以區塊480、482、及484所示，

支援僅一種資料元件寬度的運算無需有資料元件寬度決定（例如，區塊442）且無需兩個貝他欄位決定（例如，區塊444A及446A）；一替代實施例可針對所有這種運算僅支援一種資料元件寬度，在所有類型的運算上支援資料元件寬度兩者（其在載入圖形、載入分包圖形、及儲存圖形運算上會需要資料元件寬度及額外貝他欄位決定），或在其他運算的一些上不支援不同的資料元件寬度（例如，針對載入/運算之運算不支援不同資料元件寬度）。在類似上下文中，替代實施例在無記憶體存取捨入類型運算及無記憶體存取資料變換類型運算的一或更多者上可不支援不同資料元件寬度（在前者中，區塊414及415A將不存在；在後者中，區塊415B將不存在，同時區塊416將不存在且區塊418及420可合併）。舉另一例來說，本發明之不同實施例：可不包括類別（U）欄位268並僅支援類別A或B指令模板之一；可包括SAE欄位256而不包括捨入運算欄位258；可不包括捨入運算欄位259A；可不包括驅逐提示欄位252B；在類別A及B指令模板之一或兩者中可不包括捨入類型運算；可不包括資料變換類型運算；在無記憶體存取205及記憶體存取220之一或兩者中可不包括向量長度欄位259B；僅支援載入/運算及載入運算之一或另一者；可不包括遮罩寫入欄位270；可不包括寫入遮罩控制（Z）欄位252C；及/或可不包括向量長度欄位259B。

示範暫存器架構－第5圖

第5圖繪示根據本發明之一實施例的暫存器架構500之區塊圖。於下列出暫存器架構之暫存器檔及暫存器：

向量暫存器檔510－在所示實施例中，有為512位元寬之32個向量暫存器；這些暫存器稱為zmm0至zmm31。下16個zmm暫存器的低階256位元覆蓋在暫存器ymm0-16上。下16個zmm暫存器的低階128位元（ymm暫存器之低階128位元）覆蓋在暫存器xmm0-15上。特定向量親和指令格式300在如下表中所示之這些被覆蓋暫存器檔上運算。

可調向量長度	類別	運算	暫存器
不包括向量長度欄位259B的指令模板	A(第2A圖；U=0)	210、215、225、230	zmm暫存器（向量長度為64位元組）
	B(第2B圖；U=1)	212	zmm暫存器（向量長度為64位元組）
包括向量長度欄位259B的指令模板	B(第2B圖；U=1)	217、227	取決於向量長度欄位259B，zmm、ymm、或xmm暫存器（向量長度為64位元組、32位元組、或16位元組）

換言之，向量長度欄位259B在最大長度及一或更多其他較短長度之間做選擇，其中每一個這種較短長度為前一長度之半長；且無向量長度欄位259B之指令模板在最大向量長度上運算。此外，在一實施例中，特定向量親和指令格式300之類別B指令模板在分包或純量單/雙精準浮點資料及分包或純量整數資料上運算。純量運算為在zmm/ymm/xmm暫存器中的最低階資料元件位置上履行之

運算；較高階資料元件位置取決於實施例被保留成與在指令前一般相同或歸零。

寫入遮罩暫存器 515 – 在所示實施例中，有8個寫入遮罩暫存器（k0至k7），各具有64位元之大小。如前述，在本發明之一實施例中，向量遮罩暫存器k0不可用為寫入遮罩；當正常會指示k0的編碼係用於寫入遮罩時，其選擇0xFFFF之固線寫入遮罩，實際上針對那個指令禁能寫入遮蓋。

多媒體延伸控制狀態暫存器（MXCSR）520 – 在所示實施例中，此32位元暫存器提供浮點運算中所用之狀態及控制位元。

通用暫存器 525 – 在所示實施例中，有十六個64位元的通用暫存器，其連同現有x86定址模式一起用來定址記憶體運算元。以名稱RAX、RBX、RCX、RDX、RBP、RSI、RDI、RSP、及R8至R15參照這些暫存器。

延伸旗標（EFLAGS）暫存器 530 – 在所示實施例中，此32位元暫存器用來記錄許多指令的結果。

浮點控制字（FCW）暫存器 540 及浮點狀態字（FSW）暫存器 530 – 在所示實施例中，這些暫存器由x87指令集延伸用來設定捨入模式、異常遮罩及旗標（在FCW的情況中），並追蹤異常（在FSW的情況中）。

純量浮點堆疊暫存器檔（x87堆疊）545，其上別名有MMX分包整數扁暫存器檔 550 – 在所示實施例中，該x87堆疊為8元件堆疊，用來在32/64/80位元浮點資料上使用

x87指令集延伸來履行純量浮點運算；而MMX暫存器用來在64位元分包整數資料上履行運算，還有針對在MMX與XMM暫存器之間履行的一些運算保持運算元。

段暫存器555－在所示實施例中，有六個16位元暫存器，用來儲存用於分段位址產生之資料。

RIP暫存器565－在所示實施例中，此64位元暫存器儲存指令指標。

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

暫存器索引欄位、標度欄位、位移欄位、及位移因子欄位 流程－第6A-6C圖

修飾符欄位 = 無記憶體存取－第6A圖

第6A圖為繪示根據本發明之實施例的無記憶體存取類型運算之暫存器索引欄位244的流程圖。第6A圖以橢圓形600開始，其指示根據mod欄位342（=11）履行暫存器至暫存器定址。從區塊600，控制進至區塊605。

在區塊605中，從暫存器索引欄位244選擇位元來定址暫存器。關於特定向量親和指令格式300，具有延伸之現有x86指令集允許基於REX欄位305、reg欄位344、r/m欄位346、VVVV欄位320、xxx欄位354、及bbb欄位356之廣泛各式各樣的暫存器定址選項。REX'欄位310延伸這些選項。從區塊605，控制進至區塊610。

在區塊 610 中，選擇暫存器 A（例如，zmm19）且控制進至區塊 615。在區塊 615 中，選擇暫存器 B（例如，zmm5）且控制進至區塊 620。在區塊 625 中，選擇暫存器 C（例如，zmm7）。暫存器 A 可為來源運算元暫存器；暫存器 B 可為來源運算元暫存器、目的地運算元暫存器、或來源/目的地運算元暫存器；且暫存器 C 可為來源運算元暫存器、目的地運算元暫存器、或來源/目的地運算元暫存器。

修飾符欄位 = 記憶體存取 – 第 6B 圖

第 6B 圖為繪示根據本發明之實施例的記憶體存取類型運算之暫存器索引欄位 244、標度欄位 260、位移欄位 262A、及位移因子欄位 262B 之使用的流程圖。第 6B 圖以橢圓形 630 開始，其指示暫存器 – 記憶體定址（mod 欄位 342 = 00、01、或 10）。從區塊 630，控制進至區塊 635。

在區塊 635 中，從暫存器索引欄位選擇位元來定址暫存器且控制進至區塊 640。

在區塊 640 中，選擇暫存器 A（例如，zmm19）且控制可選地進至區塊 645。在區塊 645 中，選擇暫存器 B（例如，zmm31）且控制進至區塊 650。在不使用區塊 645 的情況下，控制直接從區塊 640 進至區塊 650。

在區塊 650 中，REX 欄位 305、REX' 欄位 310、mod r/m 欄位 340、SIB 欄位 350、及位移欄位 262A 或位移因子欄位 262B 的內容用來定址記憶體；詳言之，從 REX 欄位 305 及

SIB欄位350拉出索引及基礎，同時從SIB欄位350拉出標度欄位260（ss欄位352）的內容。從區塊650，控制進至區塊660。

在區塊660中，判定記憶體存取模式（例如，基於mod欄位342的內容）。在記憶體存取模式為無位移模式的情況中（mod欄位342 = 00），控制進至區塊665，其中如下般產生位址： $2^{ss} \times \text{索引} + \text{基礎}$ 。

在記憶體存取模式為無定標位移模式的情況中（mod欄位342 = 10），控制進至區塊670，其中如下般產生位址： $2^{ss} \times \text{索引} + \text{基礎} + \text{disp32}$ 。在記憶體存取模式為定標位移模式的情況中（mod欄位342 = 01），控制進至區塊675，其中如下般產生位址： $2^{ss} \times \text{索引} + \text{基礎} + \text{定標位移}$ ；其中定標位移（ $\text{disp8} \times n$ ）= 位移因子欄位262B的內容乘以記憶存取大小（N），其中N取決於諸如全運算碼欄位274及資料操縱欄位254C等欄位之內容。

定標位移－第6C圖

第6C圖為繪示根據本發明之實施例的在disp8、disp32、及定標位移之變異間的差別之表。該表之行爲：
 1) 「位元組」，其指示以位元組增額位址；2) 「disp8欄位」，其為用來儲存從-128到127之1位元組欄位；3) 「disp32欄位」，其為用來儲存從 -2^{31} 到 $2^{31}-1$ 之4位元組欄位；4) 「disp32*N欄位」，其為用來儲存從-128到127之1位元組欄位，該行具有「N=1」、「N=2」、「N=64」的

子行。

在「位元組」行中列中的值隨著行往下而增加。第二行、第三行、及各個子行在列中包括黑色點，表示可由那個欄位產生位址。應注意到disp8欄位、disp32欄位、及在N=1之處在其範圍中每一位元組有一個黑色點，代表這些欄位以位元組粒度增額。相反地，N=2行以兩位元組增額且因此在其範圍內僅每隔一位元組有一個黑色點；依此，其相較於disp8欄位具有較廣的範圍但較粗的粒度，而同時其需disp32欄位之位元組的四分之一。N=64行以64位元組增額且因此在其範圍內僅每第64個位元組有一個黑色點；依此，其相較於disp8欄位及N=2具有較廣的範圍但較粗的粒度，而同時其也需disp32欄位之位元組的四分之一。

捨入欄位表－第7A-B圖

第7A圖為繪示根據本發明之實施例的可由捨入控制欄位254A指明的該群可能的運算之表。第7A圖顯示含有貝他欄位254（其充當捨入控制欄位254A且分成SAE欄位256及捨入運算欄位258）之可能內容的第一行。

類似地，第7B圖為繪示根據本發明之實施例的可由捨入控制欄位259A指明的該群可能的運算之表。在類別B指令模板的情況中，沒有SAE欄位256且浮點異常抑制總為現行。

注意到在一些指令已經允許經由立即位元靜態地指明

捨入模式的一實施例中，立即位元比捨入運算欄位 258 及 259A 優先。

資料類型

下表列出本文所用之示範資料類型（在 Microsoft® 的 DirectX®10 中敘述其之一些（見 Microsoft®, DirectX®, Data Conversion Rules (Aug. 17, 2010)））：

FLOAT10	10位元浮點數（無正負符號）
FLOAT11	11位元浮點數（無正負符號）
FLOAT16	16位元浮點數
FLOAT16RZ	具有朝零捨入之捨入模式（RZ）之16位元浮點數
FLOAT32	32位元浮點數
FLOAT64	64位元浮點數
UINT8	映射至範圍[0, 255]中之一個整數的8位元值
UINT16	映射至範圍[0, 65535]中之一個整數的16位元值
UINT32	映射至範圍[0, $2^{32} - 1$]中之一個整數的32位元值
UINT64	映射至一個整數的64位元值
SINT8	映射至範圍[-128, 127]中之一個整數的8位元值
SINT16	映射至範圍[-32768, 32768]中之一個整數的16位元值
SINT32	映射至範圍[-2^{31} , $2^{31} - 1$]中之一個整數的32位元值
SINT64	映射至一個整數的64位元值
UNORM2	映射至範圍[0, 1]中之一個浮點數的2位元值
UNORM8	映射至範圍[0, 1]中之一個浮點數的8位元值
UNORM10	映射至範圍[0, 1]中之一個浮點數的10位元值
UNORM16	映射至範圍[0, 1]中之一個浮點數的16位元值
SNORM8	映射至範圍[-1, 1]中之一個浮點數的8位元值
SNORM16	映射至範圍[-1, 1]中之一個浮點數的16位元值
SRGB8	通過伽馬（gamma）函數（一般實現為查詢表）映射至範圍[0, 1]中之一個浮點數的8位元值

UNORM指示無正負符號的歸一整數，意指針對一 n 位元數，所有的0意指 $0.0f$ ，且所有的1意指 $1.0f$ 。表示從 $0.0f$ 到 $1.0f$ 之均勻間隔的浮點值序列，例如2位元 UNORM 表示 $0.0f$ 、 $1/3$ 、 $2/3$ 、 $1.0f$ 。

SNORM指示有正負符號的歸一整數，意指針對一 n 位元2的補數，最大值意指 $1.0f$ （例如，5位元值01111映射至 $1.0f$ ），且最小值意指 $-1.0f$ （例如，5位元值10000映射至 $-1.0f$ ）。另外，次小數映射至 $-1.0f$ （例如，5位元值10001映射至 $-1.0f$ ）。 $-1.0f$ 因此有兩個整數表示。 $0.0f$ 有單一個表示，且 $1.0f$ 有單一個表示。這導致針對在範圍（ $-1.0f...0.0f$ ）中之均勻間隔的浮點值之一組整數表示，及針對在範圍（ $0.0f...1.0f$ ）中之數的一互補組表示。

如前述，SIMD技術特別適用於可將暫存器中的位元邏輯分成若干固定/大小不同的資料元件之處理器，其之每一者可代表一分別的值。這種資料稱為分包資料類型或向量資料類型，且這種資料類型的運算元稱為分包資料運算元或向量運算元。通常，向量運算元之資料元件為相同之資料類型；一給定資料元件的資料類型稱為資料元件資料類型。在所有的資料元件之資料元件資料類型為相同的情況中，則向量運算元可稱為屬於那個資料類型（例如，在一向量運算元之全部資料元件為32位元浮點資料元件資料類型的情況中，則該向量運算元可稱為32位元浮點向量運算元）。

敘述本發明之實施例，其支援單值資料元件資料類型及多值資料元件資料類型。單值資料元件資料類型在每一個資料元件中儲存單一值；用於本發明之一些實施例中之單值資料元件資料類型的範例為32位元浮點、64位元浮點、32位元無正負符號整數、64位元無正負符號整數、32位元有正負符號整數、及64位元有正負符號整數。多值資料元件資料類型在每一個資料元件位置中儲存一封包（packet），其中含有多個值；用於本發明之一些實施例中之多值資料元件資料類型的範例為下述之分包圖形資料元件資料類型：

UNORM10A10B10C2D：三個UNORM10值及兩個UNORM2值之32位元封包，以最後2b（10b）欄位在32b欄位的最高有效位元中開始（例如，unorm2D[31-30]浮點10C[29-20]浮點10B[20-10]浮點10A[9-0]，其中D-A代表槽位置且其前方的名稱或數字代表格式）。

FLOAT11A11B10C：兩個FLOAT11值及一個FLOAT10值之32位元封包，以最後一個位在高階位元中開始（例如，浮點10C[31-32]浮點10B[21-11]浮點10A[10-0]）。

應注意到雖於上之多值資料元件資料類型的一封包中的不同值係由不同位元數量表示，替代實施例可具有不同的組態（例如，由不同位元數量表示更多值，由相同位元數量表示所有的值）。

雖敘述支援單值資料元件資料類型及多值資料元件資料類型的實施例，替代實施例可支援兩者之一。另外，雖

敘述本發明之實施例利用某些資料類型，本發明之替代實施例可利用更多、更少、或不同的資料類型。

資料變換欄位表 – 第8A及8B圖

第8A-8B圖為繪示根據本發明之實施例的可由資料變換欄位指明之可能的資料變換運算群的表。在這兩個表中之第一行繪示資料變換欄位254B的內容之可能的值；第二行為功能；且第三行為用法。

資料元件大小欄位 = 64位元 – 第8A圖

第8A圖為繪示根據本發明之實施例的當資料元件寬度為64位元時可由資料變換欄位指明之一群可能的資料變換運算的表。此表稱為64位元暫存器SwizzUpConv拌和基元且其為區塊418的表示。注意：dcba標示形成來源中之一個256位元區塊的64位元元件（其中「a」最小有效且「d」為最大有效），所以aaaa意指來源中之256位元區塊的最小有效元件係複製到目的地中相同256位元區塊的所有四個元件；接著針對來源及目的地中的兩個256位元區塊重複所示型樣。符號「hgfe dcba」用來標示完全來源暫存器，其中「a」為最小有效元件且「h」為最大有效元件。然而，由於每一個256位元區塊針對暫存器拌和履行相同的置換，僅繪示最小有效區塊。

資料元件大小欄位 = 32位元 – 第8B圖

第8B圖為繪示根據本發明之實施例的當資料元件寬度為32位元時可由資料變換欄位指明之一群可能的資料變換運算的表。此表稱為32位元暫存器SwizzUpConv拌和基元且其為區塊420的表示。注意：dcba標示形成來源中之一個128位元區塊的32位元元件（其中「a」最小有效且「d」為最大有效），所以aaaa意指來源中之128位元區塊的最小有效元件係複製到目的地中相同128位元區塊的所有四個元件；接著針對來源及目的地中的全部四個128位元區塊重複所示型樣。片語「ponm lkji hgfe dcba」用來標示完全來源暫存器，其中「a」為最小有效元件且「p」為最大有效元件。然而，由於每一個128位元區塊針對暫存器拌和履行相同的，僅顯示最小有效區塊。

第8B圖呼叫出兩個示範運算以進一步繪示第8A-8B圖中所示之所有運算的意義：繪示在第9圖中之交叉乘積拌和815及第10A圖中之跨四元件封包廣播一元件820。

示範拌和運算－第9圖

第9圖為繪示根據本發明之實施例的交叉乘積拌和815的區塊圖。第9圖顯示來源運算元900及目的地運算元910，兩者皆為512位元寬並分成連續的128區塊（稱為封包位置3-0），其中每一個區塊分成四個32位元的資料元件（例如，在來源運算元900中之封包位置0的內容為D0 C0 B0 A0，而在目的地運算元910中之封包位置0的內容為D0 A0 C0 B0）。

示範廣播運算－第10A-10C圖

第10A圖為繪示根據本發明之實施例的跨四元件封包廣播一元件820之區塊圖。第10A圖顯示來源運算元1000及目的地運算元1010，兩者皆為512位元寬並分成連續的128區塊（稱為封包位置3-0），其中每一個區塊分成四個32位元的資料元件（例如，在來源運算元1000中之封包位置0的內容為D0 C0 B0 A0，而在目的地運算元1010中之封包位置0的內容為A0 A0 A0 A0；來源運算元1000中之封包位置1的內容為D1 C1 B1 A1，而在目的地運算元1010中之封包位置1的內容為A1 A1 A1 A1）。

雖第10A圖為無記憶體存取運算之一示範廣播，第10B-10C圖為記憶體存取運算之示範廣播。當來源記憶體運算元含有少於總元件數量時，其可廣播（重複）以形成有效來源運算元之完全元件數量（針對32位元指令為16；針對64位元指令為8）。於第12A-12D圖中參照這些類型的廣播運算。有兩種廣播粒度：

1元件粒度，其中來源記憶體運算元之1元件廣播16次以形成完全16元件有效來源運算元（針對32位元指令），或8次以形成完全8元件有效來源運算元（針對64位元指令）。第10B圖為繪示根據本發明之實施例的針對32位元資料元件寬度的1元件粒度之廣播的區塊圖。在第12B圖中將運算之範例標為1210。第10B圖顯示源自具有一個32位元資料元件（A0）的記憶體之來源運算元1020，及為512

位元寬且含有十六個32位元資料元件的目的地運算元1030（在目的地運算元1030中之所有資料元件都為A0）。1元件廣播對混合向量及純量指令的指令有用，其中來源之一跨不同運算為共同的。

4元件粒度，其中來源記憶體運算元之4元件廣播4次以形成完全16元件有效來源運算元（針對32位元指令），或2次以形成完全8元件有效來源運算元（針對64位元指令）。第10C圖為繪示根據本發明之實施例的針對32位元資料元件寬度的4元件粒度之廣播的區塊圖。在第12B圖中將運算之範例標為1220。第10C圖顯示源自具有四個32位元資料元件（D0 C0 B0 A0）的記憶體之來源運算元1040，及目的地運算元1050，其為512位元寬並分成連續的128區塊（稱為封包位置3-0），其中每一個區塊分成四個32位元的資料元件（例如，在目的地運算元1050之封包位置3-0的各者中之內容為D0 C0 B0 A0。4至16廣播對AOS（結構陣列）來源碼很有用，其中在分包值的一陣列（像顏色分量RGBA）上履行計算；在此情況中，當跨一向量指令的不同運算使用一共同封包時，4至16有利（16元件的向量被視為各4元件之4封包的陣列）。

基礎運算欄位表－第11A及11B圖

運算碼映圖欄位－第11A圖

第11A圖為繪示根據本發明之實施例的可由運算碼映圖欄位指明之一群可能的運算碼映圖的表。第一行繪示運

算碼映圖欄位 315 之內容的可能值；第二行繪示隱含的領先運算碼位元組；且第三行繪示是否有立即值。

前綴編碼欄位－第 11B 圖

第 11B 圖為繪示根據本發明之實施例的可由運算碼映圖欄位指明之一群可能的前綴編碼的表。第一行繪示前綴編碼欄位 325 之內容的可能值；第二行繪示那個前綴之意義。

資料操縱欄位表－第 12-15 圖

第 12-15 圖為繪示根據本發明之實施例的可分別由資料操縱欄位 254C 指明之可能的資料操縱運算及廣播運算群的表，且，針對第 12A-D 圖廣播欄位 257B。表中之第一行繪示資料操縱欄位 254C 之內容的可能值；第二行繪示功能；且第三行繪示用法。

針對載入/運算之資料操縱欄位表－第 12A-12D 圖

第 12A-12D 圖為繪示根據本發明之實施例的可分別由資料操縱欄位 254C 及廣播欄位 257B 所指明之針對載入/運算指令之可能的資料操縱運算及廣播運算群的表。在第 3A-D 圖中之示範特定向量親和指令格式的情況中，資料操縱欄位 254C 為三位元欄位且廣播欄位 257B 為一位元欄位。在所示實施例中，廣播欄位 257B 的內容在第 12A-D 圖中可見之表中的前兩列之間做選擇；換言之，其之內容在

資料操縱欄位254C中之000與001的等效者之間做選擇。這在第12A-D圖中使用僅包括表之前兩列的括號來加以繪示。

載入/運算整數及資料元件大小欄位 = 64位元 – 第12A圖

第12A圖為繪示根據本發明之實施例的可由資料操縱欄位254C及廣播欄位257B所指明之針對其中資料元件寬度為64位元的載入/運算整數之可能的資料操縱運算群的表。此表稱為64位元整數載入－運算SwizzUpConv_{i64}（四字）拌和/轉換基元並且其為區塊444A及區塊444B的表示。

載入/運算整數及資料元件大小欄位 = 32位元 – 第12B圖

第12B圖為繪示根據本發明之實施例的可由資料操縱欄位254C及廣播欄位257B所指明之針對其中資料元件寬度為32位元的載入/運算整數之可能的資料操縱運算群的表。此表稱為32位元整數載入－運算SwizzUpConv_{i32}拌和/轉換基元並且其為區塊446A及區塊446B的表示。

載入/運算浮點及資料元件大小欄位 = 64位元 – 第12C圖

第12C圖為繪示根據本發明之實施例的可由資料操縱欄位254C及廣播欄位257B所指明之針對其中資料元件寬度為64位元的載入/運算浮點之可能的資料操縱運算群的表。此表稱為64位元浮點載入－運算SwizzUpConv_{i64}拌和/

轉換基元並且其為區塊450A及區塊450B的表示。

載入/運算浮點及資料元件大小欄位 = 32位元 – 第12D圖

第12D圖為繪示根據本發明之實施例的可由資料操縱欄位254C及廣播欄位257B所指明之針對其中資料元件寬度為32位元的載入/運算浮點之可能的資料操縱運算群的表。此表稱為32位元浮點載入－運算SwizzUpConv_{i32}拌和/轉換基元並且其為區塊452及區塊452B的表示。

針對載入之資料操縱欄位表 – 第13A-13D圖

第13A-13D圖為繪示根據本發明之實施例的可由資料操縱欄位所指明之針對載入指令之可能的資料操縱運算群的表。

載入整數及資料元件大小欄位 = 64位元 – 第13A圖

第13A圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為64位元的載入整數之可能的資料操縱運算群的表。此表稱為UpConv_{i64}並且其為區塊456的表示。

載入整數及資料元件大小欄位 = 32位元 – 第13B圖

第13B圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為32位元的載入整數之可能的資料操縱運算群的表。此表稱為UpConv_{i32}

並且其為區塊458的表示。

載入浮點及資料元件大小欄位 = 64位元 – 第13C圖

第13C圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為64位元的載入浮點之可能的資料操縱運算群的表。此表稱為UpConv_{f64}並且其為區塊462的表示。

載入浮點及資料元件大小欄位 = 32位元 – 第13D圖

第13D圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為32位元的載入浮點之可能的資料操縱運算群的表。此表稱為UpConv_{f32}並且其為區塊464的表示。

額外點

在第13A-13D圖（載入/運算表）的每一圖中所指明之可能的資料操縱運算群為相應第12A-12D圖中的那些之子集（載入表）。詳言之，該子集不包括廣播運算。如此做係因為在全運算碼欄位274中的某些值（例如，指明收集或廣播運算的那些）不能與資料操縱欄位254C中所指明之廣播一起使用，並因此在全運算碼欄位274中的這種值僅可與第12A-12D圖（載入表）之載入一起使用。以更具體範例說明，若在全運算碼欄位274有指明廣播運算的值，資料操縱欄位254C不能也指示廣播運算。雖本發明之某些

實施例以分別的載入/運算及載入表包括該分別的載入/運算及載入運算，替代實施例不需有此強制機制（例如，它們可僅支援載入/運算；它們可僅支援載入；它們可判定全運算碼欄位274中的廣播使資料操縱欄位254C中的廣播被忽略）。

針對儲存之資料操縱欄位表－第14A-14D圖

第14A-14D圖為繪示根據本發明之實施例的可由資料操縱欄位所指明之針對儲存指令之可能的資料操縱運算群的表。

儲存整數及資料元件大小欄位 = 64位元－第14A圖

第14A圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為64位元的儲存整數之可能的資料操縱運算群的表。此表稱為DownConv_{i64}並且其為區塊470的表示。

儲存整數及資料元件大小欄位 = 32位元－第14B圖

第14B圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為32位元的儲存整數之可能的資料操縱運算群的表。此表稱為DownConv_{i32}並且其為區塊472的表示。

儲存浮點及資料元件大小欄位 = 64位元－第14C圖

第14C圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為64位元的儲存浮點之可能的資料操縱運算群的表。此表稱為 DownConv_{f64} 並且其為區塊476的表示。

儲存浮點及資料元件大小欄位 = 32位元 – 第14D圖

第14D圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為32位元的儲存浮點之可能的資料操縱運算群的表。此表稱為 DownConv_{f32} 並且其為區塊478的表示。

針對圖形資料類型之資料操縱欄位表 – 第15A-15C圖

第15A-15C圖為繪示根據本發明之實施例的可由資料操縱欄位所指明之針對在圖形資料類型上運算的指令之可能的資料操縱運算群的表。

載入圖形 – 第15A圖

第15A圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為32位元的載入圖形之可能的資料操縱運算群的表。此表稱為 UpConv_{g32} 並且其為區塊480的表示。

載入分包圖形 – 第15B圖

第15B圖為繪示根據本發明之實施例的可由資料操縱

欄位254C所指明之針對其中資料元件寬度為32位元的載入分包圖形之可能的資料操縱運算群的表。此表稱為 UpConv_{pg32} 並且其為區塊482的表示。

儲存圖形－第15C圖

第15C圖為繪示根據本發明之實施例的可由資料操縱欄位254C所指明之針對其中資料元件寬度為32位元的儲存圖形之可能的資料操縱運算群的表。此表稱為 DownConv_{g32} 並且其為區塊484的表示。

寫入遮罩欄位－第16A-D圖

第16A-16B圖繪示根據本發明之實施例的以不同寫入遮罩且第二來源及目的地為相同下履行的兩個合併運算。第16A圖為繪示根據本發明之實施例的一示範運算1600的區塊圖，其使用寫入遮罩暫存器K1中之寫入遮罩來合併，其中資料元件寬度為32位元且其中第二來源及目的地為相同。第16A圖顯示來源運算元1605；來源/目的地運算元1610；遮罩暫存器K1的內容1615（其中下16位元包括一與零之混合）；及目的地運算元1620。在寫入遮罩暫存器K1中之下16位元位置的各者相應於資料元件位置之一（K1[0]相應於資料元件位置0；K1[1]相應於資料元件位置1；及以此類推）。針對在目的地運算元1620中之每一個資料元件位置，其含有在來源/目的地運算元1610中之那個資料元件位置的內容或運算（繪示為加法）之結果，

分別取決於遮罩暫存器K1中之相應位元位置為零或一。在其他實施例中，以第二來源運算元取代來源/目的地運算元1610。在那些實施例中，目的地運算元1620在其中遮罩暫存器K1的相應位元位置為零（若有的話）之資料元件位置的那些中含有來自運算以前目的地運算元1620之內容，並且在其中遮罩暫存器K1的相應位元位置為1（若有的話）之資料元件位置的那些中含有運算的結果。

如前所述，本發明之一實施例使用K0來指示無遮蓋應履行。第16B圖為繪示根據本發明之實施例的一示範運算1625之區塊圖，其使用皆為一的固線遮罩（由指明寫入遮罩暫存器k0的指令使用該固線寫入遮罩）來合併，其中資料元件寬度為32位元且其中第二來源及目的地為相同。第16B圖與第16A圖相同，除了以固線遮罩1630取代K1 1615，並以目的地運算元1635取代目的地運算元1620。固線遮罩1630皆為一，並因此目的地運算元1635含有代表運算結果之資料元件。

第16C圖為繪示根據本發明之實施例的針對8、16、32、及64位元資料元件寬度之寫入遮罩暫存器中之位元對一個512位元向量之資料元件位置的相應性。詳言之，繪示一個64位元暫存器K_N 1640，其中當資料元件寬度為8位元時使用所有64位元；當資料元件寬度為16位元時使用最小有效32位元；當資料元件寬度為32位元時使用最小有效16位元；且當資料元件寬度為64位元時使用最小有效8位元。針對一個256位元向量，當資料元件寬度為8位元時僅

使用最小有效32位元；當資料元件寬度為16位元時使用最小有效16位元；當資料元件寬度為32位元時使用最小有效8位元；且當資料元件寬度為64位元時使用最小有效4位元。針對一個128位元向量，當資料元件寬度為8位元時僅使用最小有效16位元；當資料元件寬度為16位元時使用最小有效8位元；當資料元件寬度為32位元時使用最小有效2位元；且當資料元件寬度為64位元時使用最小有效2位元。

給定遮罩暫存器的值可設定為向量比較指令之直接結果；從GP暫存器傳送而來；或計算為兩個遮罩之間的邏輯運算的直接結果。

第16D圖為繪示根據本發明之實施例的一示範運算1660的區塊圖，其使用寫入遮罩暫存器K1中之寫入遮罩來合併，其中資料元件寬度為32位元且其中第二來源及目的地為不同。

第16E圖為繪示根據本發明之實施例的一示範運算1666的區塊圖，其使用寫入遮罩暫存器K1中之寫入遮罩來歸零，其中資料元件寬度為32位元且其中第二來源及目的地為不同。雖僅相關於其中目的地與來源不同的運算繪示歸零運算，歸零也可用於其中第二來源與目的地為相同的情況。

示範模板及指令－第17-18圖

提供下列符號作為第17-18圖之引言。

運算元符號

符號	意義
zmm1	在指令之引數1欄位中之向量暫存器運算元。64位元組的向量暫存器為：zmm0至zmm31
zmm2	在指令之引數2欄位中之向量暫存器運算元。64位元組的向量暫存器為：zmm0至zmm31
zmm3	在指令之引數3欄位中之向量暫存器運算元。64位元組的向量暫存器為：zmm0至zmm31
$S_{\beta 32}(\text{zmm}/\text{m})$	向量浮點32位元拌和/轉換。
$S_{\beta 64}(\text{zmm}/\text{m})$	向量浮點64位元拌和/轉換。
$S_{i32}(\text{zmm}/\text{m})$	向量整數32位元拌和/轉換。
$S_{i64}(\text{zmm}/\text{m})$	向量整數64位元拌和/轉換。
$U_{\beta 32}(\text{m})$	浮點32位元載入向上轉換。
$U_{g32}(\text{m})$	圖形浮點32位元載入向上轉換。
$U_{pg32}(\text{m})$	分包圖形浮點32位元載入向上轉換。
$U_{i32}(\text{m})$	整數32位元載入向上轉換。
$U_{\beta 64}(\text{m})$	浮點64位元載入向上轉換。
$U_{i64}(\text{m})$	整數64位元載入向上轉換。
$D_{\beta 32}(\text{zmm})$	浮點32位元儲存向下轉換。
$D_{g32}(\text{zmm})$	圖形浮點32位元儲存向下轉換。
$D_{i32}(\text{zmm})$	整數32位元儲存向下轉換。
$D_{\beta 64}(\text{zmm})$	浮點64位元儲存下轉換。
$D_{i64}(\text{zmm})$	整數64位元儲存向下轉換。
m	記憶體運算元。
m_t	可能有EH提示屬性之記憶體運算元。
mv_t	可能有EH提示屬性之向量記憶體運算元。使用ModRM及VSIB位元組編碼此記憶體運算元。可將其看成一組指標，其中每一指標等於基礎+V索引[i]×標度
effective_address	當處理記憶體運算元時用來標示全有效位址。
imm8	立即位元組值。
SRC[a-b]	來自範圍從LSB b到MSB a之運算元的位元欄位。

向量運算元值符號

符號	意義
$zmm1[i+31:i]$	位在引數1向量運算元的位元 <i>i</i> 與位元 <i>i+31</i> 之間的元件之值
$zmm2[i+31:i]$	位在引數2向量運算元的位元 <i>i</i> 與位元 <i>i+31</i> 之間的元件之值
$k1[i]$	指明向量遮罩暫存器k1中之第 <i>i</i> 個位元
$\{k1\}$	與合併行為一起使用之指令的寫入遮罩欄位中的遮罩暫存器運算元。 64位元遮罩暫存器為：k0至k7
$\{k1\}\{z\}$	與歸零行為一起使用之指令的寫入遮罩欄位中的遮罩暫存器運算元。 64位元遮罩暫存器為：k0至k7

SwizzUpConv、FullUpConv及DownConv函數轉換

使用之拌和/轉換	用於運算敘述中之函數
$S_{\beta 32}(zmm/m)$	SwizzUpConvLoad $_{\beta 32}(zmm/m)$
$S_{\beta 64}(zmm/m)$	SwizzUpConvLoad $_{\beta 64}(zmm/m)$
$S_{i32}(zmm/m)$	SwizzUpConvLoad $_{i32}(zmm/m)$
$S_{i64}(zmm/m)$	SwizzUpConvLoad $_{i64}(zmm/m)$
$U_{\beta 32}(m)$	UpConvLoad $_{\beta 32}(m)$
$U_{g32}(m)$	UpConvLoad $_{g32}(m)$
$U_{pg32}(m)$	UpConvLoad $_{pg32}(m)$
$U_{i32}(m)$	UpConvLoad $_{i32}(m)$
$U_{\beta 64}(m)$	UpConvLoad $_{\beta 64}(m)$
$U_{i64}(m)$	UpConvLoad $_{i64}(m)$
$D_{\beta 32}(zmm)$	DownConvStore $_{\beta 32}(zmm)$ or DownConvStore $_{\beta 32}(zmm[xx:yy])$
$D_{g32}(zmm)$	DownConvStore $_{g32}(zmm)$ or DownConvStore $_{g32}(zmm[xx:yy])$
$D_{i32}(zmm)$	DownConvStore $_{i32}(zmm)$ or DownConvStore $_{i32}(zmm[xx:yy])$
$D_{\beta 64}(zmm)$	DownConvStore $_{\beta 64}(zmm)$ or DownConvStore $_{\beta 64}(zmm[xx:yy])$
$D_{i64}(zmm)$	DownConvStore $_{i64}(zmm)$ or DownConvStore $_{i64}(zmm[xx:yy])$

針對EVEX.U0之示範指令編碼－第17A-D圖

第17A圖繪示根據本發明之實施例的來自一示範特定向量親和指令格式的欄位之子集。詳言之，第17A圖顯示EVEX前綴302、真實運算碼欄位330、及MOD R/M欄位340。在此實施例中，格式欄位240含有0x62來指示指令格式為特定向量親和指令格式。

第17B-17D圖各繪示根據本發明之實施例的來自第17A圖之特定向量親和指令格式中編碼的一示範特定向量親和指令之欄位的子集。在第17B-17D圖的說明中，敘述一些欄位的特定使用來示範針對VADDPS指令的各種示範組態之那些欄位的可能編碼。在第17B-17D圖的每一圖中，格式欄位240含有0x62來指示在特定向量親和指令格式中編碼指令且真實運算碼欄位330含有VADDPS運算碼。第17B-17D圖的每一圖各繪示根據本發明之實施例的在EVEX.U0類別中之VADDPS指令之一種編碼；第17B圖及第17C圖各繪示在無記憶體存取205指令模板中之VADDPS的EVEX.U0編碼，同時第17D圖繪示在記憶體存取220指令模板中之VADDPS的EVEX.U0編碼。VADDPS指令將來自第一暫存器或記憶體運算元（例如，zmm3）的分包單精準浮點值加到第二暫存器（例如，zmm2）並根據寫入遮罩（例如，k1）儲存結果於第三暫存器中（例如，zmm1）。取決於指令之編碼，此指令允許各種捨入運算、資料變換運算、或資料操縱運算。可藉由下列指令助記符來描述該指令：EVEX.U0.NDS.512.0F 58 /r

VADDPS zmm1 {k1}, zmm2, S_{f32}(zmm3/mV) {eh}。

第17B圖繪示在無記憶體存取全捨入控制類型運算210指令模板中之VADDPS指令的一種編碼。資料元件寬度欄位264為0以指示32位元資料元件寬度。類別欄位268（亦即，EVEX.U）設定成0以指示EVEX.U0類別。阿爾發欄位252被解譯為RS欄位252A（亦即，EVEX.rs）並設定成1（亦即，RS欄位252A.1）來選擇捨入控制類型運算。由於阿爾發欄位252充當RS欄位252A.1，貝他欄位254被解譯為捨入運算欄位258（亦即，EVEX.r₂₋₀）。詳言之，EVEX.r₂被解譯為SAE欄位256，而EVEX.r₁₋₀充當捨入控制欄位254A。修飾符欄位246（亦即，MODR/M.MOD 342）設定成11以指示無記憶體存取（亦即暫存器zmm3為第一來源運算元而非記憶體運算元）。

第17C圖繪示在無記憶體存取資料變換類型運算215指令模板中之VADDPS指令的一種編碼。第17C圖中之編碼與第17B圖相同，除了阿爾發欄位252及貝他欄位254之外。阿爾發欄位252被解譯為RS欄位252A（亦即，EVEX.rs）並設定成0（亦即，RS欄位252A.2）來選擇資料變換類型運算。由於阿爾發欄位252充當RS欄位252A.2，貝他欄位254被解譯為資料變換欄位254B（亦即，EVEX.s₂₋₀）。

第17D圖繪示在記憶體存取220指令模板中之VADDPS指令的一種編碼。資料元件寬度欄位264為0以指示32位元資料元件寬度。類別欄位268（亦即，EVEX.U）設定成0

以指示 EVEX.U0 類別。阿爾發欄位 252 被解譯為驅逐提示欄位 252B（亦即，EVEX.EH）。貝他欄位 254 被解譯為資料操縱欄位 254C（亦即，EVEX.s₂₋₀）。修飾符欄位 246（亦即，MODR/M.MOD 342）設定成 00、01、或 10 以指示第一來源運算元為記憶體運算元；這在第 17D 圖顯示成 11（亦即除了 11 外的任何輸入）。

針對 EVEX.U1 之示範指令編碼 – 第 18A-18F 圖

第 18A 圖繪示根據本發明之實施例的來自一示範特定向量親和指令格式的欄位之子集。詳言之，第 18A 圖顯示 EVEX 前綴 302、真實運算碼欄位 330、及 MOD R/M 欄位 340。在此實施例中，格式欄位 240 含有 0x62 來指示指令格式為特定向量親和指令格式。

第 18B-18F 圖各繪示根據本發明之實施例的來自第 18A 圖之特定向量親和指令格式中編碼的一示範特定向量親和指令之欄位的子集。在第 18B-18F 圖的說明中，敘述一些欄位的特定使用來示範針對 VADDPS 指令的各種示範組態之那些欄位的可能編碼。在第 18B-18F 圖的每一圖中，格式欄位 240 含有 0x62 來指示在向量親和指令格式中編碼指令且真實運算碼欄位 330 含有 VADDPS 運算碼。第 18B-18F 圖的每一圖各繪示根據本發明之實施例的在 EVEX.U1 類別中之 VADDPS 指令之一種編碼；第 18B-18E 圖各繪示在無記憶體存取 205 指令模板中之 VADDPS 的 EVEX.U1 編碼，同時第 18F 圖繪示在記憶體存取 220 指令模

板中之 VADDPS 的 EVEX.U1 編碼。

第 18B 圖繪示在無記憶體存取寫入遮罩控制部分捨入控制類型運算 212 指令模板中之 VADDPS 指令的一種編碼。資料元件寬度欄位 264 為 0 以指示 32 位元資料元件寬度。類別欄位 268（亦即，EVEX.U）設定成 1 以指示 EVEX.U1 類別。阿爾發欄位 252 被解譯為寫入遮罩控制欄位 252C（在合併或歸零寫入遮罩間選擇）。貝他欄位 254 之最小有效位元被解譯為 RL 欄位 257A 並設定成 1 以指示部分捨入類型運算（亦即捨入 257A.1）。貝他欄位 254 之兩個最大有效位元被解譯為捨入運算欄位 259A。修飾符欄位 246（亦即，MODR/M.MOD 342）設定成 11 以指示無記憶體存取（亦即暫存器 zmm3 為第一來源運算元而非記憶體運算元）。在此編碼中，VADDPS 指令將來自第一暫存器（例如，zmm3）的分包單精準浮點值加到第二暫存器（例如，zmm2）並根據寫入遮罩（例如，k1）儲存經捨入結果於第三暫存器中（例如，zmm1）。可藉由下列助記符來描述此：針對歸零寫入遮蓋 EVEX.U1.NDS.512.0F.W0 58 /r VADDPS zmm1 {k1} {z}, zmm2, zmm3 {er}，且針對合併寫入遮蓋為相同但無 {z}。雖在此段中下示之其他助記符皆包括 {z}，應了解的是也可類似地有相同但無 {z} 之助記符。

第 18C-18E 圖各繪示在無記憶體存取寫入遮罩控制 VSIZE 類型運算 217 指令模板中之 VADDPS 指令的一種編碼。第 18C-18E 圖的編碼與第 17B 圖相同，除了貝他欄位

外。在第18C-18E圖的每一圖中，貝他欄位254之最小有效位元被解譯為RL欄位257A並設定成0以指示VSIZE類型運算257A.2。貝他欄位254之兩個最大有效位元被解譯為向量長度欄位259B。

在第18C圖中，向量長度欄位259B設定成10以指示512位元的向量大小。在第18D圖中，向量長度欄位259B設定成01以指示256位元的向量大小。在第18E圖中，向量長度欄位259B設定成00以指示128位元的向量大小。在此編碼中，VADDPS指令將來自第一暫存器（例如，zmm3）的分包單精準浮點值加到第二暫存器（例如，zmm2）並根據寫入遮罩（例如，k1）儲存結果於第三暫存器中（例如，zmm1）。可藉由下列助記符來描述第18C圖：
EVEX.U1.NDS.512.0F.W0 58 /r VADDPS zmm1 {k1} {z}, zmm2, zmm3。可藉由下列助記符來描述第18D圖：
EVEX.U1.NDS.256.0F.W0 58 /r VADDPS ymm1 {k1} {z}, ymm2, ymm3。可藉由下列助記符來描述第18E圖：
EVEX.U1.NDS.128.0F.W0 58 /r VADDPS xmm1 {k1} {z}, xmm2, xmm3。

第18F圖繪示在記憶體存取寫入遮罩控制227指令模板中之VADDPS指令的一種編碼。資料元件寬度欄位264為0以指示32位元資料元件寬度。類別欄位268（亦即，EVEX.U）設定成1以指示EVEX.U1類別。阿爾發欄位252被解譯為寫入遮罩控制欄位252C（在合併或歸零寫入遮罩間選擇）。貝他欄位254之最小有效位元被解譯為廣播欄

位 257B。貝他欄位 254 之兩個最大有效位元被解譯為向量長度欄位 259B。修飾符欄位 246（亦即，MODR/M.MOD 342）設定成 00、01、或 10 以指示第一來源運算元為記憶體運算元；這在第 18F 圖顯示成 $\bar{11}$ （亦即除了 11 外的任何輸入）。在此編碼中，VADDPS 指令將來自記憶體運算元的分包單精準浮點值（其可在被載入後廣播）加到第一暫存器（例如，zmm2）並根據寫入遮罩（例如，k1）儲存該結果於第二暫存器中（例如，zmm1）。當向量長度欄位指示 512 位元的向量，可藉由下列助記符來描述此：EVEX.U1.NDS.512.0F.W0 58 /r VADDPS zmm1 {k1} {z}, zmm2, B₃₂(mV)。當向量長度欄位指示 256 位元的向量，可藉由下列助記符來描述此：EVEX.U1.NDS.256.0F.W0 58 /r VADDPS ymm1 {k1} {z}, ymm2, B₃₂(mV)。當向量長度欄位指示 128 位元的向量，可藉由下列助記符來描述此：EVEX.U1.NDS.128.0F.W0 58 /r VADDPS xmm1 {k1} {z}, xmm2, B₃₂(mV)。

示範位移 8*N 值

在本發明之一實施例中，有關於 U=0（類別 A），下表顯示在記憶體中所存取之向量（或元件）的大小且，類似地，壓縮位移之位移因子（disp8*N）。注意到一些指令作用在元件粒度而非在記憶體級之全向量粒度，並因此應使用下表中之「元件級」行。功能行的標示（例如 U/S_{i64}）代表由基礎運算欄位所指明之記憶體存取類型

（例如，U/Si代表載入整數及載入/運算整數）及資料元件寬度（例如₆₄為64位元資料元件寬度）。在此行中的值為第3圖的實施例中之資料操縱欄位254C的可能值。參照第4B圖，顯示各種記憶體存取類型流動（在一些情況中經由資料元件寬度決定）至其資料操縱第12A-15C圖；各個表12A-15C驅動N的值之選擇，並因此適當地置於行2及3上。舉例而言，載入/運算整數64位元資料元件寬度記憶體存取運算流至第12A圖，在該處資料操縱欄位254C的內容用來選擇資料操縱運算（如第12A圖中所示）及N之值（如下所示）兩者。舉另一例而言，載入整數64位元資料元件寬度記憶體存取運算（其指示基礎運算欄位242中之廣播）流至第13A圖，在該處資料操縱欄位254C的內容用來選擇資料操縱運算（如第13A圖中所示，其不包括廣播資料變換）及N之值（如下所示）兩者。因此，第二行用於其之基礎運算欄位242不指明廣播或元件級記憶體存取的指令；第三行的第一子行用於其之基礎運算欄位242指明廣播但不指明元件級記憶體存取之指令；且第三行的第二子行用於其之基礎運算欄位242指明廣播或元件級記憶體存取之指令。

SwizzleUpConvert_{i64}及 UpConvert_{i64}

功能S/U _{i64}	第12A圖	第13A圖	
	無基礎運算欄位指明之廣播或元件級記憶體存取	由基礎運算欄位指明之4到16廣播	由基礎運算欄位指明之1到16廣播或元件級記憶體存取
000	64	32	8
001	8	NA	NA
010	32	NA	NA
011	NA	NA	NA
100	NA	NA	NA
101	NA	NA	NA
110	NA	NA	NA
111	NA	NA	NA

SwizzleUpConvert_{i32}及 UpConvert_{i32}

功能S/U _{i32}	第12B圖	第13B圖	
	無基礎運算欄位指明之廣播或元件級記憶體存取	由基礎運算欄位指明之4到16廣播	由基礎運算欄位指明之1到16廣播或元件級記憶體存取
000	64	16	4
001	4	NA	NA
010	16	NA	NA
011	NA	NA	NA
100	16	4	1
101	16	4	1
110	32	8	2
111	32	8	2

SwizzleUpConvert_{f64}及 UpConvert_{f64}

功能S/U _{f64}	第12C圖	第13C圖	
	無基礎運算欄位指明之廣播或元件級記憶體存取	由基礎運算欄位指明之4到16廣播	由基礎運算欄位指明之1到16廣播或元件級記憶體存取
000	64	32	8
001	8	NA	NA
010	32	NA	NA
011	NA	NA	NA
100	NA	NA	NA
101	NA	NA	NA
110	NA	NA	NA
111	NA	NA	NA

SwizzleUpConvert_{f32}及 UpConvert_{f32}

功能S/U _{f32}	第12D圖	第13D圖	
	無基礎運算欄位指明之廣播或元件級記憶體存取	由基礎運算欄位指明之4到16廣播	由基礎運算欄位指明之1到16廣播或元件級記憶體存取
000	64	16	4
001	4	NA	NA
010	16	NA	NA
011	32	8	2
100	16	4	1
101	16	4	1
110	32	8	2
111	32	8	2

Down Conversion_{i64}

功能D _{i64}	第14A圖	第14A圖	
	由基礎運算欄位指明的 正常儲存	不適用	由基礎運算欄位指明之 元件級記憶體存取
000	64	NA	8
001	NA	NA	NA
010	NA	NA	NA
011	NA	NA	NA
100	NA	NA	NA
101	NA	NA	NA
110	NA	NA	NA
111	NA	NA	NA

Down Conversion_{i32}

功能D _{i32}	第14B圖	第14B圖	
	由基礎運算欄位指明的 正常儲存	不適用	由基礎運算欄位指明之 元件級記憶體存取
000	64	NA	4
001	NA	NA	NA
010	NA	NA	NA
011	NA	NA	NA
100	16	NA	1
101	16	NA	1
110	32	NA	2
111	32	NA	2

Down Conversion_{f64}

功能 D _{f64}	第 14C 圖	第 14C 圖	
	由基礎運算欄位指明的 正常儲存	不適用	由基礎運算欄位指明之 元件級記憶體存取
000	64	NA	8
001	NA	NA	NA
010	NA	NA	NA
011	NA	NA	NA
100	NA	NA	NA
101	NA	NA	NA
110	NA	NA	NA
111	NA	NA	NA

Down Conversion_{f32}

功能 D _{f32}	第 14D 圖	第 14D 圖	
	由基礎運算欄位指明的 正常儲存	不適用	由基礎運算欄位指明之 元件級記憶體存取
000	64	NA	4
001	NA	NA	NA
010	NA	NA	NA
011	32	NA	2
100	16	NA	1
101	16	NA	1
110	32	NA	2
111	32	NA	2

UpConvert_{g32}

功能 U _{g32}	第 15A 圖	第 15A 圖	
	無基礎運算欄位指明之廣播或元件級記憶體存取	由基礎運算欄位指明之 4 到 16 廣播	由基礎運算欄位指明之 1 到 16 廣播或元件級記憶體存取
000	64	16	4
001	NA	NA	NA
010	16	4	1
011	NA	NA	NA
100	16	4	1
101	16	4	1
110	32	8	2
111	32	8	2

UpConvert_{pg32}

功能 U _{pg32}	第 15B 圖	第 15B 圖	
	無基礎運算欄位指明之廣播或元件級記憶體存取	由基礎運算欄位指明之 4 到 16 廣播	由基礎運算欄位指明之 1 到 16 廣播或元件級記憶體存取
000	64	16	4
001	64	16	4
010	64	16	4
011	64	16	4
100	64	16	4
101	64	16	4
110	64	16	4
111	64	16	4

Down Conversion_{g32}

功能 D _{g32}	第 15C 圖	第 15C 圖	
	由基礎運算欄位指明 的正常儲存	不適用	由基礎運算欄位指明 之元件級記憶體存取
000	64	NA	4
001	NA	NA	NA
010	NA	NA	NA
011	32	NA	2
100	16	NA	1
101	16	NA	1
110	32	NA	2
111	32	NA	2

在本發明之一實施例中，有關於U=1（類別B），各種指令具有使用壓縮位移的能力，藉由使用disp8連同由針對不同類型的指令之向量長度（由向量長度欄位259B所決定）、廣播欄位257B之值、及資料元件寬度（由如第4E圖中所描述之真實運算碼欄位330及/或資料位元寬度欄位264）而定之記憶體存取尺寸N。一般而言，記憶體存取尺寸N相應於記憶體輸入中之位元組的數量（例如，當存取全512位元記憶體向量時為64）。在本發明之一實施例中，於下面第一表解釋於下第二表中所用的一些術語，且於下面第二表提供針對各種類型的指令之N的值。在下表中之元組（Tuple）為記憶體中之分包結構。

Full	讀取全向量。接受廣播（載入－運算）。例如，VADDPS xmm1, xmm2, xmm3/B(mem)
FullMem	讀取全向量。不接受廣播（載入－運算）。例如，VMOVAPS xmm1, m512
Scalar	從記憶體讀取單一元件以進行純量運算：VADDSS xmm1, xmm2, m32
Tuple1	從記憶體讀取單一元件。例如，VBROADCASTSS xmm1, m32
Tuple2	從記憶體讀取僅 2 個元件。例如，VBROADCASTF32X2 xmm1, m64
Tuple4	從記憶體讀取僅 4 個元件。例如，VBROADCASTF32X4 xmm1, m128
Tuple8	從記憶體讀取僅 8 個元件。例如，VBROADCASTF32X8 xmm1, m256
Half	從記憶體讀取僅總元件之一半，例如 VCVTSS2PD xmm1, B(mem) //僅讀取 8 SP 輸入值以取得 8 DP 結果
HalfMem	同上，但僅記憶體（不接受廣播）。例如，VPMOVBXWB xmm1, mem //僅讀取 32 位元組輸入值以取得 32 字結果
QuarterMem	從記憶體讀取僅總元件之四分之一。例如，VPMOVBXWB xmm1, mem //僅讀取 16 位元組輸入值以取得 16 雙字結果
EighthMem	從記憶體讀取僅總元件之四分之一。例如，VPMOVBXWB xmm1, mem //僅讀取 8 位元組輸入值以取得 8 四字結果
Mem128	從記憶體讀取僅 m128 位元的一個封包。其不允許廣播。

Disp8*N 格式	廣播	輸入大小	廣播	N(disp8 *N)128 位元	N(disp8 *N)256 位元	N(disp8 *N)512 位元	註解
Full	0	32 位元	無	16	32	64	載入/運算
	1	32 位元	(1 至 X)	4	4	4	
	0	64 位元	無	16	32	64	
	1	64 位元	(1 至 X)	8	8	8	
FullMem	0	不適用	無	16	32	64	載入/儲存+子雙字
	1		(1 至 X)	不適用	不適用	不適用	
Tuple1/Scalar	0	8 位元	無	1	1	1	廣播/額外 ct/插入(1 元件) + 收集/分散 + 純量
	1	8 位元	不適用	不適用	不適用	不適用	
	0	16 位元	無	2	2	2	
	1	16 位元	不適用	不適用	不適用	不適用	
	0	32 位元	無	4	4	4	
	1	32 位元	不適用	不適用	不適用	不適用	
	0	64 位元	無	8	8	8	
	1	64 位元	不適用	不適用	不適用	不適用	
Tuple2	0	32 位元	無	8	8	8	廣播(2 元件)
	1	32 位元	不適用	不適用	不適用	不適用	
	0	64 位元	無	不適用	16	16	
	1	64 位元	不適用	不適用	不適用	不適用	
Tuple4	0	32 位元	無	不適用	16	16	廣播(4 元件)
	1	32 位元	不適用	不適用	不適用	不適用	
	0	64 位元	無	不適用	不適用	32	
	1	64 位元	不適用	不適用	不適用	不適用	
Tuple8	0	32 位元	無	不適用	不適用	32	廣播(8 元件)
	1	32 位元	不適用	不適用	不適用	不適用	
	0	64 位元	不適用	不適用	不適用	不適用	
	1	64 位元	不適用	不適用	不適用	不適用	
Half	0	32 位元	無	8	16	32	載入運算(半記憶體大小)
	1	32 位元	(1 至 X)	4	4	4	
	0	64 位元	不適用	不適用	不適用	不適用	
	1	64 位元	不適用	不適用	不適用	不適用	
HalfMem	0	不適用	無	8	16	32	載入/儲存 + 子雙字(半記憶體大小)
	1		不適用	不適用	不適用	不適用	
QuarterMem	0	不適用	無	4	8	16	載入/儲存 + 子雙字(四分之一記憶體大小)
	1		不適用	不適用	不適用	不適用	
EighthMem	0	不適用	無	2	4	8	載入/儲存 + 子雙字(八分之一記憶體大小)
	1		不適用	不適用	不適用	不適用	
Mem128	0	不適用	無	16	16	16	具有自記憶體之移動計數之移動
	1		不適用	不適用	不適用	不適用	

示範管線－第19-22圖

第19-22圖為繪示根據本發明之實施例在四個示範處理器管線的不同階段中使用第2A圖中之指令模板的哪些欄位的區塊圖。應注意到在所需理解程度，所示管線階段及其功能為眾所皆知。第19-22圖之每一者包括A、B、及C圖，分別繪示無記憶體存取全捨入控制類型運算210、無記憶體存取資料變換類型運算215指令模板、及記憶體存取225/230指令模板。雖第19-22圖之每一者顯示一個不同的示範管線，在A-C圖的每一圖中針對每一個圖號顯示相同管線。例如，第19A圖顯示無記憶體存取全捨入控制類型運算210及第一示範指令管線；第19B圖顯示無記憶體存取資料變換類型運算215指令模板及和第19A圖中相同的示範管線；而第20A圖顯示無記憶體存取全捨入控制類型運算210及第二示範處理器管線。

第19-22圖分別繪示處理器管線1900、處理器管線2000、處理器管線2100、及處理器管線2200。在跨不同示範管線中管線階段名稱相同的情況中，為了便於理解而使用相同參考符號；然而，這不意味著跨不同示範管線的相同名稱管線階段為相同，只是它們履行類似的操作（雖其可包括更多或更少子操作）。

示範泛型管線－第19圖

處理器管線1900代表泛型處理器管線，並因此其包括提取階段1910、解碼階段1920、暫存器讀取/記憶體讀取

階段1930、資料變換階段1940、執行階段1950、及寫回/記憶體寫入階段1960。

從指令模板到處理器管線階段的括號及箭頭線繪示由管線階段的不同者所利用之欄位。例如，在第19A圖中，解碼階段1920利用所有的欄位；暫存器讀取/記憶體讀取階段1930利用暫存器索引欄位244；執行階段1950利用rs欄位252A（捨入252A.1）、SAE欄位256、捨入運算欄位258、及資料元件寬度欄位264；也由寫回/記憶體寫入階段1960利用資料元件寬度欄位264；且由執行階段1950或寫回/記憶體寫入階段1960使用寫入遮罩欄位270（在兩個不同階級中寫入遮罩欄位270之可選的使用代表寫入遮罩欄位可禁能在執行階段1950中在被遮蓋資料元件上之運算的執行（藉此防止那些資料元件位置在寫回/記憶體寫入階段1960中被更新），或執行階段1950可履行運算並在寫回/記憶體寫入階段1960期間施加寫入遮罩以防止被遮蓋之資料元件位置的更新）。

應注意到箭頭線不一定代表由不同欄位所利用之唯一階段，但的確代表哪個欄位會最有可能有最大影響。在A及B圖之間，將注意到主要差別在於執行階段1950將擴充運算欄位250用於捨入運算；資料變換階段1940將擴充運算欄位250用於資料變換類型運算；且從資料元件寬度欄位264到執行階段1950的線移到資料變換階段1940。第19C圖顯示基礎運算欄位242取代地指向暫存器讀取/記憶體讀取階段1930；由暫存器讀取/記憶體讀取階段1930利用擴

充運算欄位 250 的 EH 欄位 252B；取決於記憶體讀取或記憶體寫入運算，由暫存器讀取/記憶體讀取階段 1930 或寫回/記憶體寫入階段 1960 可選地利用標度欄位 260、位移欄位 262A/位移因子欄位 262B、寫入遮罩欄位 270、及資料元件寬度欄位 264。由於眾所皆知可利用立即欄位 272 之管線階段，不表示那個欄位的映射以不混淆本發明。

示範有序管線 – 第 20 圖

處理器管線 2000 表示有序處理器管線並具有和處理器管線 1900 相同名稱的管線階段，但具有插在提取階段 1910 與解碼階段 1920 之間的長度解碼階段 2012。

第 20A-20C 圖之映射與第 19A-19C 圖基本上相同。

第一示範無序管線 – 第 21 圖

處理器管線 2100 表示第一示範無序管線，具有和處理器管線 2000 相同名稱的管線階段，但亦具有下列者：1) 插在解碼階段 1920 與暫存器讀取/記憶體讀取階段 1930 之間的分配階段 2122、重命名階段 2124、及排程階段 2126；及 2) 添加在寫回/記憶體寫入階段 1960 之後的重排序緩衝器 (rob) 讀取階段 2162、異常處置階段 2164、及確定階段 2166。

在第 21A-21C 圖中，映射大致和第 20A-20C 圖中相同，但有下列異常：1) 由重命名階段 2124 利用暫存器索引欄位 244 及修飾符欄位 246；2) 僅在第 21A 圖中，也由異

常處置階段2164可選地使用寫入遮罩欄位270以抑制被遮蓋資料元件位置上的異常；及3)僅在第21A圖中，取決於是否將抑制浮點異常，由執行階段1950及異常處置階段2164可選地使用SAE欄位256。

第二示範無序管線－第22圖

處理器管線2200表示第二示範無序管線，其具有和處理器管線2100相同名稱的處理器管線階段，除了已合併資料變換及執行階段以形成執行/資料變換階段2245。

在第22A-22C圖中的映射和第21A-21C圖中的那些基本上相同，除了之前分別指向資料變換階段1940及執行階段1950的映射取代地指向執行/資料變換階段2245。

在示範管線上之類別B指令模板

下表繪示根據本發明之實施例如何修改第19-22圖以遷就第2B圖中之指令模板的欄位。

欄位	圖	管線階段
寫入遮罩控制 (Z) 欄位 252C	19A-22C	寫回/記憶體寫入 1960 及/或執行 1950/2245
	21A-22C	重命名 2124
捨入 257A.1	19A、20A、21A	執行 1950
	22A	執行/資料變換 2245
	21A、22A	異常處置 2164
捨入運算欄位 259A	19A、20A、21A	執行 1950
	22A	執行/資料變換 2245
VSIZE 欄位 257A.2	19B、20B、21B、 22B	解碼 1920
	21B、22B	重命名 2124
	19B、20B、21B、 22B	暫存器讀取/記憶體讀取 1930
	19B、20B、21B 22B	執行 1950
	22B	執行/資料變換 2245
	19B、20B、21B、 22B	寫回/記憶體寫入 1960
向量長度欄位 259B	19B-C、20B-C、 21B-C、22B-C	解碼 1920
	21B-C、22B-C	重命名 2124
	19B-C、20B-C、 21B-C、22B-C	暫存器讀取/記憶體讀取 1930
	19B-C、20B-C、 21B-C	執行 1950
	22B-C	執行/資料變換 2245
	19B-C、20B-C、 21B-C、22B-C	寫回/記憶體寫入 1960
廣播欄位 257B	19C、20C、21C	資料變換 1940
	22C	執行/資料變換 2245
	19C、20C、21C、 22C	記憶體讀取 1930

解碼階段1920

各種不同的眾所皆知解碼單元可用於解碼階段1920中。例如，解碼單元可解碼巨集指令成為單寬微指令。舉另一例而言，解碼單元可解碼一些巨集指令成為單寬微指令，但其他成為多寬微指令。舉尤其適用於無序處理器管線的另一例而言，解碼單元可解碼每一個巨集指令成為一或更多微碼，其中可無序地發出並執行每一個微碼。

應注意到可以一或更多解碼器實現解碼單元，且可以可編程邏輯陣列（PLA）實現每一個解碼器，如為此技藝中眾所皆知。舉例而言，一給定解碼單元可：1）具有引導邏輯以指引不同巨集指令到不同解碼器；2）第一解碼器，其可解碼指令集之一子集（但比第二、第三、及第四解碼器解碼指令集之較多）並一次產生兩個微碼；3）第二、第三、及第四解碼器，其可各解碼僅整個指令集之一子集並一次產生僅一個微碼；4）微序列器ROM，其可解碼僅整個指令集之一子集並一次產生四個微碼；及5）由解碼器及微序列器ROM饋送之多工邏輯，其判定誰的輸出將供應至微碼佇列。解碼器單元之其他實施例可具有更多或更少的解碼器，其解碼更多或更少指令及指令子集。例如，一實施例可具有各可一次產生兩個微碼之第二、第三、及第四解碼器；並可包括一次產生八個微碼之微序列器ROM。

示範處理器架構－第23-24圖

示範有序處理器架構－第23A-23B圖

第23A-B圖繪示示範有序處理器架構之區塊圖。圍繞以寬向量處理器（VPU）擴充之有序CPU核心的多個實例設計此示範實施例。取決於確切的應用，核心透過高頻寬互連網路通訊與一些固定功能邏輯、記憶體I/O介面、及其他必要的I/O邏輯。例如，作為獨立GPU的此實施例之一實現通常可包括PCIe匯流排。

第23A圖為根據本發明之實施例的單一CPU核心連同其至晶粒上互連網路2302的連結及其之二階（L2）快取2304的本地子集之區塊圖。指令解碼器2300支援x86指令集，具有包括特定向量親和指令格式300之延伸。雖在本發明之一實施例（為了簡化設計）中純量單元2308及向量單元2310使用分別的暫存器集（分別為純量暫存器2312及向量暫存器2314）並將在其之間傳送的資料寫入記憶體並接著從一階（L1）快取2306回讀進來，本發明之替代實施例可使用不同的方式（例如，使用單一暫存器集或包括允許資料在兩個暫存器檔之間傳送而不被寫入及讀回的通訊路徑）。

L1快取2306允許低潛伏存取以快取記憶體到純量及向量單元中。連同在向量親和指令格式中之載入－運算指令，這意指L1快取2306可某程度地被看待為延伸的暫存器檔。這明顯改善許多演算法的性能，尤其係驅逐提示欄位252B。

L2快取2304的本地子集為總體L2快取之一部分，該總

體 L2 快取分成分別的本地子集，每一個 CPU 核心一個。每一個 CPU 具有至其本身的 L2 快取 2304 之本地子集的直接存取路徑。由 CPU 核心讀取的資料係儲存在其之 L2 快取 2304 並可迅速加以存取，與存取其本身的本地 L2 快取子集之其他 CPU 平行。由 CPU 核心寫入的資料係儲存在其本身的 L2 快取子集 2304 中並若有需要的話從其他子集清除。此環狀網路確保共享資料的同調。

第 23B 圖為根據本發明之實施例之第 23A 圖中的 CPU 核心的一部分之放大圖。第 23B 圖包括 L1 快取 2306 之 L1 資料快取 2306A 部分，還有關於向量單元 2310 及向量暫存器 2314 的更多細節。詳言之，向量單元 2310 為 16 寬的向量處理單元（VPU）（見 16 寬 ALU 2328），其執行整數、單精準浮點、及雙精準浮點指令。VPU 支援以拌和單元 2320 拌和暫存器輸入、數值轉換單元 2322A-B 的數值轉換、及以複製單元 2324 複製記憶體輸入。寫入遮罩暫存器 2326 允許預測所得之向量寫入。

可以各種方式拌和暫存器資料，例如，以支援矩陣複製。可跨 VPU 道複製來自記憶體的資料。這在圖形及非圖形平行資料處理兩者中為常見的操作，其顯著增加快取效率。

環狀網路為雙向，以允許比如 CPU 核心、L2 快取、及其他邏輯區塊的代理器在晶片內互相通訊。每一個環狀資料路徑為每一方向 512 位元寬。

示範無序處理器架構－第24圖

第24圖為繪示一示範無序處理器架構之區塊圖。詳言之，第24圖繪示已經修改以合併向量親和指令格式及其執行之眾所皆知的示範無序架構。在第24圖中，箭頭表示兩或更多個單元之間的耦合，且箭頭方向指示那些單元間的資料流之方向。第24圖包括前端單元2405，耦合到執行引擎單元2410及記憶體單元2415；執行引擎單元2410進一步耦合到記憶體單元2415。

前端單元2405包括耦合到二階（L2）分支預測單元2422之一階（L1）分支預測單元2420。L1及L2分支預測單元2420及2422耦合到L1指令快取單元2424。L1指令快取單元2424耦合到指令轉譯旁看緩衝器（TLB）2426，其進一步耦合到指令提取及預先解碼單元2428。指令提取及預先解碼單元2428耦合到指令佇列單元2430，其進一步耦合到解碼單元2432。解碼單元2432包含一個複雜解碼器單元2434及三個簡單解碼器單元2436、2438、及2440。解碼單元2432包括一個微碼ROM單元2442。解碼單元2432可如先前在解碼階段段落中所述般操作。L1指令快取單元2424進一步耦合到在記憶體單元2415中之L2快取單元2448。指令TLB單元2426進一步耦合到在記憶體單元2415中之第二階TLB單元2446。解碼單元2432、微碼ROM單元2442、及迴路流檢測器單元2444各耦合到執行引擎單元2410中之重命名/分配器單元2456。

執行引擎單元2410包括耦合到引退單元2474及統一排

程器單元2458的重命名/分配器單元2456。引退單元2474進一步耦合到執行單元2460並包括重排序緩衝器單元2478。統一排程器單元2458進一步耦合到實體暫存器檔單元2476，其耦合到執行單元2460。實體暫存器檔單元2476包含向量暫存器單元2477A、寫入遮罩暫存器單元2477B、及純量暫存器單元2477C；這些暫存器單元可提供向量暫存器510、寫入遮罩暫存器515、及通用暫存器525；且實體暫存器檔單元2476可包括未顯示之額外的暫存器檔（例如，別名於MMX分包整數扁平暫存器檔550上之純量浮點堆疊暫存器檔545）。執行單元2460包括三個混合純量及向量單元2462、2464、及2472；載入單元2466；儲存位址單元2468；儲存資料單元2470。載入單元2466、儲存位址單元2468、及儲存資料單元2470各進一步耦合到記憶體單元2415中之資料TLB單元2452。

記憶體單元2415包括耦合到資料TLB單元2452之第二階TLB單元2446。資料TLB單元2452耦合到L1資料快取單元2454。L1資料快取單元2454進一步耦合到L2快取單元2448。在一些實施例中，L2快取單元2448進一步耦合至記憶體單元2415內及/或外之L3及更高階快取單元2450。

舉例來說，該示範無序架構可如下般實現處理器管線2200：1）指令提取及預先解碼單元2428履行提取及長度解碼階段1910及2012；2）解碼單元2432履行解碼階段1920；3）重命名/分配器單元2456履行分配階段2122及重命名階段2124；4）統一排程器單元2458履行排程階段

2126；5）實體暫存器檔單元2476、重排序緩衝器單元2478、及記憶體單元2415履行暫存器讀取/記憶體讀取階段1930；執行單元2460履行執行/資料變換階段2245；6）記憶體單元2415及重排序緩衝器單元2478履行寫回/記憶體寫入階段1960；7）引退單元2474履行ROM讀取2162階段；8）各種單元可牽涉於異常處置階段2164中；及9）引退單元2474及實體暫存器檔單元2476履行確定階段2166。

示範單核心及多核心處理器－第29圖

第29圖為根據本發明之實施例的具有整合記憶體控制器及圖形之單核心處理器及多核心處理器2900之區塊圖。第29圖中之實線方框繪示具有單核心2902A之處理器2900、系統代理器單元2910單元、一組一或更多匯流排控制器單元2916，而虛線方框的可選添加繪示具有多核心2902A-N之替代處理器2900、在系統代理器單元2910單元中之一組一或更多整合記憶體控制器單元2914、及整合圖形邏輯2908。

記憶體階級包括在核心內之一或更多階的快取、一組一或更多共享快取單元2906、及耦合至該組整合記憶體控制器單元2914的外部記憶體（未圖示）。該組共享快取單元2906可包括一或更多中階快取，比如二階（L2）、三階（L3）、四階（L4）、或其他階的快取、最後一階快取（LLC）、及/或上述的組合。雖在一實施例中基於環形之互連單元2912互連整合圖形邏輯2908、該組共享快取單元

2906、及系統代理器單元2910，替代實施例可使用任何數量的眾所皆知技術來互連這種單元。

在本發明之一實施例中，核心2902A-N之一或更多者能進行多線程。系統代理器單元2910包括協調並操作核心2902A-N之那些組件。系統代理器單元2910可包括例如電力控制單元（PCU）及顯示單元。PCU可為或包括調節核心2902A-N及整合圖形邏輯2908之電力狀態所需之邏輯及組件。顯示單元用來驅動一或更多外部連接的顯示器。

核心2902A-N以架構及/或指令集而言可為同質或異質。例如，核心2902A-N之一些可為有序（例如，像第23A及23B圖中所示那個），同時其他為無序（像第24圖中所示那個）。舉另一例而言，核心2902A-N之兩或更多者能夠執行相同的指令集，而其他能夠僅執行那個指令集或不同指令集之一子集。核心的至少一者能夠執行本文中所述之向量親和指令格式。

處理器可為通用處理器，比如Core™i3、i5、i7、2 Duo及Quad、Xeon™、或Itanium處理器，可從美國加州聖塔克拉拉的英特爾公司購得。或者，處理器可來自另外的公司。處理器可為特殊目的處理器，比如，例如，網路或通訊處理器、壓縮引擎、圖形處理器、協處理器、嵌入式處理器等等。處理器可實現在一或更多晶片上。處理器2900可使用任何若干程序技術，比如，例如，BiCMOS、CMOS、或NMOS而為一或更多基板之一部分及/或可實現在一或更多基板上。

示範電腦系統及處理器－第25-28圖

第25-27圖為適合包括處理器2900的示範系統，而第28圖為示範系統晶片（SoC），其可包括核心2902之一或更多者。在膝上型電腦、桌上型電腦、手持PC、個人數位助理、工程站、伺服器、網路裝置、網路集線器、交換器、嵌入式處理器、數位信號處理器（DSP）、圖形裝置、視頻遊戲裝置、機上盒、微控制器、手機、可攜式媒體播放器、手持裝置、及各種其他電子裝置的記憶中之其他系統設計及組態為可行。一般而言，能夠併入本文所揭露之處理器及/或其他執行邏輯之大量各式各樣的系統或電子裝置一般都適合。

茲參照第25圖，顯示根據本發明之實施例的系統2500的區塊圖。系統2500可包括一或更多處理器2510及2515，其耦合到圖形記憶體控制器集線器（GMCH）2520。額外處理器2515之可選本質在第25圖中以間斷線標示。

各處理器2510及2515可為處理器2900的某種版本。然而，應注意到在處理器2510及2515中不大可能存在整合圖形邏輯及整合記憶體控制單元。

第25圖繪示GMCH 2520可耦合到記憶體2540，其可例如為動態隨機存取記憶體（DRAM）。DRAM可針對至少一些實施例與非依電性快取關聯。

GMCH 2520可為晶片組或晶片組之一部分。GMCH 2520可與處理器2510及2515通訊並控制處理器2510及2515

與記憶體 2540 之間的互動。GMCH 2520 亦可充當處理器 2510 及 2515 及系統 2500 的其他元件間之加速匯流排介面。針對至少一實施例，GMCH 2520 經由比如前端匯流排（FSB）2595 之多點匯流排與處理器 2510 及 2515 通訊。

此外，GMCH 2520 耦合到顯示器 2545（比如平板顯示器）。GMCH 2520 可包括整合圖形加速器。GMCH 2520 進一步耦合到輸入/輸出（I/O）控制器集線器（ICH）2550，其可用來耦合各種周邊裝置到系統 2500。在第 25 圖之實施例中顯示例如外部圖形裝置 2560，其可為連同另一周邊裝置 2570 耦合到 ICH 2550 之離散圖形裝置。

或者，額外或不同處理器亦可存在於系統 2500 中。例如，額外處理器 2515 可包括和處理器 2510 相同之額外處理器、與處理器 2510 異質或不對稱的額外處理器、加速器（比如，例如，圖形加速器或數位信號處理（DSP）單元）、現場可編程閘陣列、或任何其他處理器。以不同優點衡量上來說實體資源 2510 及 2515 之間可有各種差異，包括架構、微架構、熱、耗電量特性等等。這些差異可在處理元件 2510 及 2515 之間有效體現自己為不對稱及異質。針對至少一實施例，各種處理元件 2510 及 2515 可存在於相同晶粒封裝中。

茲參照第 26 圖，顯示根據本發明之一實施例的第二系統 2600 之區塊圖。如第 26 圖中所示，多處理器系統 2600 為點對點互連系統，並包括經由點對點互連 2650 耦合之第一處理器 2670 及第二處理器 2680。如第 26 圖中所示，處理器

2670及2680的各者可為處理器2900的某種版本。

或者，處理器2670及2680之一或更多可為非處理器之元件，比如加速器或現場可編程閘陣列。

雖僅顯示兩個處理器2670及2680，應了解的是本發明之範疇不限於此。在其他實施例中，一或更多額外處理元件可存在於給定處理器中。

處理器2670可進一步包括整合記憶體控制器集線器（IMC）2672及點對點（P-P）介面2676及2678。類似地，第二處理器2680可包括IMC 2682及P-P介面2686及2688。處理器2670及2680可經由點對點（PtP）介面2650使用PtP介面電路2678及2688來交換資料。如第26圖中所示，IMC 2672及2682耦合處理器到個別記憶體，即記憶體2632和記憶體2634，其可為本地附接到個別處理器的主記憶體之部分。

處理器2670及2680可各經由個別的P-P介面2652及2654使用點對點介面2676、2694、2686、及2698與晶片組2690交換資料。晶片組2690也可經由高性能圖形介面2639與高性能圖形電路2638交換資料。

可將共享快取（未圖示）包括在這兩個處理器外的處理器中，但經由P-P互連與處理器連接，使得若一處理器被置於低電力模式，則處理器之一或兩者的本地快取資訊可儲存在共享快取中。

晶片組2690可經由介面2696耦合至第一匯流排2616。在一實施例中，第一匯流排2616可為周邊組件互連

(PCI) 匯流排，或比如PCI Express匯流排或另外第三代I/O互連匯流排的匯流排，雖本發明之範疇不限於此。

如第26圖中所示，各種I/O裝置2614可耦合到第一匯流排2616，連同耦合第一匯流排2616至第二匯流排2620之匯流排橋接器2618。在一實施例中，第二匯流排2620可為低腳數(LPC)匯流排。各種裝置可耦合到第二匯流排2620包括，例如，在一實施例中，鍵盤/滑鼠2622、通訊裝置2626、及資料儲存單元2628，比如碟驅動機或其他大量儲存裝置，其可包括碼2630。此外，音頻I/O 2624可耦合到第二匯流排2620。注意到其他架構為可行。例如，取代第26圖的點對點架構，系統可實現多點匯流排或其他這種架構。

茲參照第27圖，顯示根據本發明之一實施例的第三系統2700之區塊圖。第26及27圖中之類似元件有類似參考符號；且第26圖之某些態樣在第27圖中省去以避免混淆第27圖之其他態樣。

第27圖繪示處理元件2670及2680可包括分別整合記憶體及I/O控制邏輯(「CL」)2672及2682。針對至少一實施例，CL 2672及2682可包括記憶體控制器集線器邏輯(IMC)，比如上述連同第29及26圖所述者。另外，CL 2672及2682亦可包括I/O控制邏輯。第27圖繪示不光是記憶體2632及2634耦合到CL 2672及2682，但I/O裝置2714也耦合到控制邏輯2672及2682。舊有I/O裝置2715耦合到晶片組2690。

茲參照第28圖，顯示根據本發明之一實施例的SoC 2800之區塊圖。在第29圖中類似的元件有類似參考符號。並且，虛線方框在更先進SoC上為可選特徵。在第28圖中，互連單元2802耦合到：應用處理器2810，其包括一組一或更多核心2902A-N及共享快取單元2906；系統代理器單元2910；匯流排控制器單元2916；整合記憶體控制器單元2914；一組或一或更多媒體處理器2820，其可包括整合圖形邏輯2908、提供靜止及/或視頻相機功能的影像處理器2824、提供硬體音頻加速之音頻處理器2826、及提供視頻編碼/解碼加速的視頻處理器2828；靜態隨機存取記憶體（SRAM）單元2830；直接記憶體存取（DMA）單元2832；及耦合一或更多外部顯示器之顯示單元2840。

本文中揭露的機制之實施例可以硬體、軟體、韌體、或這種實現方式的組合實現。本發明之實施例可實現為電腦程式或程式碼，其在包含至少一處理器、儲存系統（包括依電性及非依電性記憶體及/或儲存元件）、至少一輸入裝置、及至少一輸出裝置的可編程系統上執行。

程式碼，比如第26圖中所示之碼2630，可應用於輸入資料以履行本文所述之功能並產生輸出資訊。輸出資訊可以已知方式應用於一或更多輸出裝置。為了此說明書之目的，處理系統包括具有處理器（比如，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）之任何系統。

程式碼可以高階程序或物件導向編程語言實現來與處

理系統通訊。若希望的話，可以組合或機器語言實現程式碼。事實上，本文所述之機制範圍上不限於任何特定編程語言。在任何情況中，語言可為編譯或解譯語言。

可藉由儲存在機器可讀取媒體上之代表指令實現至少一實施例的一或更多態樣，該些代表指令代表處理器內的各種邏輯，當由機器讀取時會令機器編造邏輯以履行本文中所述之技術。可將這種表示，稱為「IP核心」儲存在有形機器可讀取媒體上，並供應至各種客戶或製造廠以載入到實際製作邏輯或處理器的製造機器之中。

這種機器可讀取媒體可包括而不限於，由機器或裝置製造或形成的物品之非暫態有形配置，包括比如硬碟、任何其他類型的碟（包括軟碟、光碟（光碟唯讀記憶體（CD-ROM）、可抹寫光碟（CD-RW）、及光磁碟））之儲存媒體、比如唯讀記憶體（ROM）、隨機存取記憶體（RAM）（比如動態隨機存取記憶體（DRAM）、靜態隨機存取記憶體（SRAM））、可抹除可編程唯讀記憶體（EPROM）、快閃記憶體、電性可抹除唯讀記憶體（EEPROM））之半導體裝置、磁或光卡、或適合儲存電子指令之任何其他類型的媒體。

依此，本發明之實施例亦可包括非暫態有形機器可讀取媒體，含有向量親和指令格式或含有設計資料，比如硬體描述語言（HDL），其界定本文所述之結構、電路、設備、處理器、及/或系統特徵。這種實施例亦可稱為程式產品。

在一些情況中，指令轉換器可用來將指令從來源指令集轉換至目的地指令集。例如，指令轉換器可轉譯（例如，使用靜態二元轉譯、包括動態編譯之動態二元轉譯）、變形、仿真、或否則轉換指令到將由核心處理的一或更多其他指令。可以軟體、硬體、韌體、或上述之組合實現指令轉換器。指令轉換器可在處理器上、處理器外、或部分在處理器上且部分在處理器外。

第30圖為根據本發明之實施例的對比使用軟體指令轉換器來將來源指令集中的二元指令轉換成目的地指令集中之二元指令的區塊圖。在所示實施例中，指令轉換器為軟體指令轉換器，雖替代地指令轉換器可以軟體、韌體、硬體、或上述的各種組合實現。第30圖顯示可使用x86編譯器3004編譯在高階語言3002中之程式以產生x86二元碼3006，其可由處理器以至少一個x86指令集核心3016原生執行（假設被編譯的指令之一些係在向量親和指令格式中）。具有至少一個x86指令集核心3016之處理器代表可履行與具有至少一個x86指令集核心的Intel處理器實質上相同功能之任何處理器，藉由相容地執行或者處理（1）Intel×86指令集核心之指令集的實質部分或（2）預計運作於具有至少一×86指令集核心之Intel處理器上的應用程式或其他軟體之物件碼版本，以實現與具有至少一個x86指令集核心的Intel處理器實質上相同的結果。x86編譯器3004代表可被操作以產生x86二元碼3006（例如物件碼）的編譯器，該碼可在有或無額外鏈結處理下以至少一個

x86指令集核心3016在處理器上加以執行。類似地，第30圖顯示在高階語言3002中之程式可使用一替代指令集編譯器3008編譯來產生替代指令集二元碼3010，其可由無至少一個x86指令集核心3014的處理器原生執行（例如，具有執行美國加州Sunnyvale的MIPS Technologies之MIPS指令集及/或執行美國加州Sunnyvale的ARM Holdings之ARM指令之核心的處理器）。指令轉換器3012用來轉換x86二元碼3006成為可由無x86指令集核心3014的處理器原生執行的碼。此經轉換的碼不大可能與替代指令集二元碼3010相同，因為能夠進行此之指令轉換器很難製造；然而，經轉換的碼將完成一般的操作並且係由來自替代指令集之指令構成。因此，指令轉換器3012代表軟體、韌體、硬體、或上述的組合，其透過仿真、模擬、或任何其他程序，允許不具有x86指令集處理器或核心之處理器或其他電子裝置執行x86二元碼3006。

在本文揭露的向量親和指令格式中之指令的某些運算可藉由硬體組件加以履行，並可體現在機器可執行指令中，其用來導致或至少造成，以該些指令編程之電路或其他硬體組件履行運算。電路可包括通用或特殊目的處理器或邏輯電路，僅舉數個例子。可藉由硬體與軟體之組合來可選地履行運算。執行邏輯及/或處理器可包括特定或具體電路或其他邏輯，其回應於機器指令或衍生自機器指令的一或更多控制信號而儲存指令特定結果運算元。例如，可在第25-28圖之系統的一或更多者中執行本文揭露的指

令之實施例，且在向量親和指令格式中的指令之實施例可儲存在將在系統中執行之程式碼。另外，這些圖的處理元件可利用本文詳述之詳細管線及/或架構之一（例如，有序及無序架構）。例如，有序架構的解碼單元可解碼指令、傳遞經解碼指令到向量或純量單元等等。

上述說明意圖繪示本發明之較佳實施例。從上述討論，很明顯地，尤其係在成長快速且未來進步不易預見的這種技術領域中，本發明之配置及細節可由熟悉此技藝人士加以修改而不悖離所附之申請專利範圍及其等效者的範疇內之本發明的原理。例如，可結合或進一步分離方法之一或更多操作。

替代實施例

雖已敘述可原生執行向量親和指令格式的實施例，本發明之替代實施例可透過運行在執行不同指令集的處理器（例如，執行美國加州 Sunnyvale 的 MIPS Technologies 之 MIPS 指令集的處理器、執行美國加州 Sunnyvale 的 ARM Holdings 之 ARM 指令之核心的處理器）上之仿真層來執行向量親和指令格式。並且，雖圖中的流程圖顯示由本發明之某些實施例所履行之操作的特定順序，應了解的是這種順序為示範性（例如，替代實施例可以不同順序、結合某些操作、重疊某些操作等等來履行操作）。

在上述說明中，為了便於解釋，已提出各種特定細節來提供本發明之實施例的詳盡理解。然而，對熟悉此技藝

人士而言很明顯地可在無這些特定細節下實行一或更多其他實施例。所述之特定實施例並非提供用來限制本發明而用來繪示本發明之實施例。本發明之範疇並非由以上提供的特定範例而係僅由下列申請專利範圍所判定。

【符號說明】

100A-100J：指令

110A-110J：指令

200：泛型向量親和指令格式

205：無記憶體存取

210：無記憶體存取全捨入控制類型運算

212：無記憶體存取寫入遮罩控制半捨入控制類型運算

215：無記憶體存取資料變換類型運算

217：無記憶體存取寫入遮罩控制 `vsize` 類型運算

220：記憶體存取

225：記憶體存取時間性

227：記憶體存取寫入遮罩控制

230：記憶體存取非時間性

240：格式欄位

242：基礎運算欄位

244：暫存器索引欄位

246：修飾符欄位

246A：無記憶體存取

246B：記憶體存取

- 250：擴充運算欄位
- 252：阿爾發欄位
- 252A：RS欄位
- 252A.1：捨入
- 252A.2：資料變換
- 252B：驅逐提示欄位
- 252B.1：時間性
- 252B.2：非時間性
- 252C：寫入遮罩控制（Z）欄位
- 254：貝他欄位
- 254A：捨入控制類型欄位
- 254B：資料變換欄位
- 254C：資料操縱欄位
- 256：抑制所有浮點異常欄位
- 257A：RL欄位
- 257A.1：捨入
- 257A.2：向量長度（VSIZE）
- 257B：廣播欄位
- 258：捨入運算控制欄位
- 259A：捨入運算欄位
- 259B：向量長度欄位
- 260：標度欄位
- 262A：位移欄位
- 262B：位移因子欄位

- 264：資料元件寬度欄位
- 268：類別欄位
- 268A：類別A欄位
- 268B：類別B欄位
- 270：寫入遮罩欄位
- 272：立即欄位
- 274：全運算碼欄位
- 300：特定向量親和指令格式
- 302：EVEX前綴
- 305：REX欄位
- 310：REX'欄位
- 315：運算碼映圖欄位
- 320：VVVV欄位
- 325：前綴編碼欄位
- 330：真實運算碼欄位
- 340：MOD R/M欄位
- 342：MOD欄位
- 344：MODR/M.reg欄位
- 346：MODR/M.r/m欄位
- 354：xxx欄位
- 356：bbb欄位
- 500：暫存器架構
- 510：向量暫存器檔
- 515：寫入遮罩暫存器

520：多媒體延伸控制狀態暫存器

525：通用暫存器

530：延伸旗標暫存器

535：浮點控制字暫存器

540：浮點狀態字暫存器

545：純量浮點堆疊暫存器檔

550：MMX分組整數扁暫存器檔

555：段暫存器

565：RIP暫存器

900：來源運算元

910：目的地運算元

1000：來源運算元

1010：目的地運算元

1020：來源運算元

1030：目的地運算元

1605：來源運算元

1610：來源/目的地運算元

1615：遮罩暫存器K1的內容

1620：目的地運算元

1625：運算

1630：固線遮罩

1635：目的地運算元

1640：暫存器

1900：處理器管線

1910：提取階段
 1920：解碼階段
 1930：暫存器讀取/記憶體讀取階段
 1940：資料變換階段
 1950：執行階段
 1960：寫回/記憶體寫入階段
 2000：處理器管線
 2012：長度解碼階段
 2100：處理器管線
 2122：分配階段
 2124：重命名階段
 2126：排程階段
 2162：重排序緩衝器讀取階段
 2164：異常處置階段
 2166：確定階段
 2200：處理器管線
 2245：執行/資料變換階段
 2300：指令解碼器
 2302：晶粒上互連網路
 2304：二階快取
 2306：一階快取
 2306A：一階資料快取
 2308：純量單元
 2310：向量單元

2312：純量暫存器
 2314：向量暫存器
 2320：拌和單元
 2322A-B：數值轉換單元
 2324：複製單元
 2326：寫入遮罩暫存器
 2328：16寬ALU
 2405：前端單元
 2410：執行引擎單元
 2415：記憶體單元
 2420：一階分支預測單元
 2422：二階分支預測單元
 2424：一階指令快取單元
 2426：指令轉譯旁看緩衝器
 2428：指令提取及預先解碼單元
 2430：指令佇列單元
 2432：解碼單元
 2434：複雜解碼器單元
 2436：簡單解碼器單元
 2438：簡單解碼器單元
 2440：簡單解碼器單元
 2442：微碼ROM單元
 2444：迴路流檢測器單元
 2446：第二階轉譯旁看緩衝器單元

- 2448：二階快取單元
- 2450：L3及更高階快取單元
- 2452：資料TLB單元
- 2454：L1資料快取單元
- 2456：重命名/分配器單元
- 2458：統一排程器單元
- 2460：執行單元
- 2462：混合純量及向量單元
- 2464：混合純量及向量單元
- 2466：載入單元
- 2468：儲存位址單元
- 2470：儲存資料單元
- 2472：混合純量及向量單元
- 2474：引退單元
- 2476：實體暫存器檔單元
- 2477A：向量暫存器單元
- 2477B：寫入遮罩暫存器單元
- 2477C：純量暫存器單元
- 2478：重排序緩衝器單元
- 2500：系統
- 2510：處理器
- 2515：處理器
- 2520：圖形記憶體控制器集線器
- 2540：記憶體

- 2545：顯示器
- 2550：輸入/輸出控制器集線器
- 2560：外部圖形裝置
- 2570：周邊裝置
- 2595：前端匯流排
- 2600：系統
- 2614：I/O裝置
- 2616：第一匯流排
- 2618：匯流排橋接器
- 2620：第二匯流排
- 2622：鍵盤/滑鼠
- 2624：音頻I/O
- 2626：通訊裝置
- 2628：資料儲存單元
- 2630：碼
- 2638：高性能圖形電路
- 2639：高性能圖形介面
- 2642：記憶體
- 2644：記憶體
- 2650：點對點互連
- 2670：第一處理器
- 2672：整合記憶體控制器集線器
- 2676：點對點介面
- 2678：點對點介面

2680：第二處理器
 2682：整合記憶體控制器集線器
 2686：點對點介面
 2688：點對點介面
 2690：晶片組
 2694：點對點介面
 2696：介面
 2698：點對點介面
 2700：系統
 2714：I/O裝置
 2715：舊有I/O裝置
 2800：系統晶片
 2802：互連單元
 2810：應用處理器
 2820：媒體處理器
 2824：影像處理器
 2826：音頻處理器
 2828：視頻處理器
 2830：靜態隨機存取記憶體單元
 2832：直接記憶體存取單元
 2840：顯示單元
 2900：處理器
 2902A：核心
 2902N：核心

- 2906：共享快取單元
- 2908：整合圖形邏輯
- 2910：系統代理器
- 2912：互連單元
- 2914：整合記憶體控制器單元
- 2916：匯流排控制器單元
- 3002：高階語言
- 3004：x86編譯器
- 3006：x86二元碼
- 3008：替代指令集編譯器
- 3010：替代指令集二元碼
- 3012：指令轉換器
- 3014：x86指令集核心
- 3016：x86指令集核心

申請專利範圍

1. 一種設備，包含：

處理器，可執行指令格式中之指令，其中該指令格式具有複數欄位，包括資料元件寬度（W）欄位、及寫入遮罩欄位，

其中該指令格式支援不同向量運算的指明，其中該些向量運算的每一者產生包括在不同資料元件位置之複數資料元件的目的地向量運算元，

其中該指令格式透過在該資料元件寬度欄位中的不同值來支援不同資料元件寬度的指明，

其中該指令格式透過在該寫入遮罩欄位中的不同值來支援該指明的向量運算所導致的該些資料元件的哪個將或將不反映在該目的地向量運算元之相應的資料元件位置中，

其中可能在該寫入遮罩欄位中的該些不同值之一保留來指示該指明的向量運算之所有結果將反映在該目的地向量運算元之相應的資料元件位置中，以及

其中可能在該寫入遮罩欄位中的該些不同值之其他區別儲存可組態寫入遮罩之不同的寫入遮罩暫存器，以及

其中該資料元件寬度欄位及該寫入遮罩欄位可各僅儲存一個值於該指令格式中之指令的各出現時。

2. 如申請專利範圍第 1 項之設備，其中該資料元件寬度包括 32 位元及 64 位元資料元件寬度。

3.如申請專利範圍第 1 或 2 項之設備，其中該些寫入遮罩暫存器為至少 64 位元。

4.如申請專利範圍第 1 或 2 項之設備，其中由該些不同向量運算所產生之該些目的地向量運算元為 512 位元。

5.如申請專利範圍第 1 或 2 項之設備，其中該寫入遮罩之單一位元係用於該目的地向量運算元之每一該些資料元件位置。

6.如申請專利範圍第 1 或 2 項之設備，其中該寫入遮罩暫存器之下 16 位元係用於具有 32 位元資料元件之 512 位元向量，而該寫入遮罩暫存器之下 8 位元係用於具有 32 位元資料元件之 512 位元向量。

7.如申請專利範圍第 1 或 2 項之設備，其中該些不同向量運算之至少某些係於兩來源運算元之上運算且該目的地向量運算元不會覆寫該兩來源運算元之任一者。

8.如申請專利範圍第 1 或 2 項之設備，其中該些不同向量運算之至少某些係於兩來源運算元之上運算且該目的地向量運算元係覆寫該兩來源運算元之一。

9.如申請專利範圍第 1 或 2 項之設備，其中係保留其將不反映該指明的向量運算的該目的地向量運算元之資料元件位置中的該些資料元件。

10.如申請專利範圍第 1 項之設備，其中該指令格式亦包括寫入遮罩控制欄位，其中該指令格式透過

在該資料元件寬度欄位中的不同值來支援是否履行合併寫入遮罩運算或歸零寫入遮罩運算。

11.如申請專利範圍第 10 項之設備，其中該些不同資料元件寬度包括 8 位元、16 位元、32 位元及 64 位元資料元件寬度。

12.如申請專利範圍第 10 項之設備，其中該指令格式亦包括真實運算碼欄位，其中該指令格式透過在該真實運算碼欄位中的不同值來支援該資料元件寬度欄位的內容是否針對該些出現的每一者在 64 位元與 32 位元資料元件大小之間做選擇或者在 16 位元與 8 位元資料元件大小之間做選擇。

13.如申請專利範圍第 1 項之設備，其中該出現的該資料元件寬度區別哪些資料元件位置相應於該些可組態寫入遮罩的哪些位元。

14.一種設備，包含：

處理器，可執行指令格式中之指令，其中該指令格式包括複數欄位，包括資料元件寬度（W）欄位、向量長度欄位、及寫入遮罩欄位，

其中該指令格式支援複數不同向量運算的指明，其中該些複數向量運算的每一者要求一運算被獨立地履行於至少一來源向量運算元之複數不同資料元件位置的每一者上來產生至少一個目的地向量運算元，

其中該指令格式透過在該資料元件寬度欄位中的不同值來支援不同資料元件寬度的指明，

其中該指令格式透過在該向量長度欄位中的不同值來支援複數不同向量長度的指明，

其中該指令格式透過在該寫入遮罩欄位中的不同值來支援不同寫入遮罩的指明，

其中該些不同值之僅一者可被放置在該資料元件寬度欄位及該寫入遮罩欄位之每一者中，於該指令格式中之指令的各出現時，以及

其中該出現的該資料元件寬度及該向量長度區別哪些資料元件位置相應於該出現之寫入遮罩的哪些位元，且其中將使用的該寫入遮罩以每資料元件位置為基礎指明該出現的向量運算之結果是否將或將不反映在該目的地向量運算元之相應的資料元件位置中。

15.如申請專利範圍第 14 項之設備，其中該些複數不同向量長度包括 128、256、及 512 位元。

16.如申請專利範圍第 14 或 15 項之設備，其中該些不同資料元件寬度包括 8 位元及 16 位元資料元件寬度。

17.如申請專利範圍第 14 或 15 項之設備，其中該些不同資料元件寬度包括 32 位元及 64 位元資料元件寬度。

18.如申請專利範圍第 14 或 15 項之設備，其中該指令格式支援歸零寫入遮罩運算。

19.如申請專利範圍第 14 或 15 項之設備，其中可放置在該寫入遮罩欄位中的該些不同值之一保留來

指示該出現的向量運算之所有結果將反映在該目的地向量運算元之相應的資料元件位置中。

20.如申請專利範圍第 14 或 15 項之設備，其中，其中可放置在該寫入遮罩欄位中的該些不同值之其他區別儲存可組態寫入遮罩之不同的寫入遮罩暫存器。

21.如申請專利範圍第 14 或 15 項之設備，其中該指令格式支援於兩來源運算元之上的運算且該目的地向量運算元不會覆寫該兩來源運算元之任一者。

22.如申請專利範圍第 14 或 15 項之設備，其中該指令格式支援於兩來源運算元之上的運算且該目的地向量運算元係覆寫該兩來源運算元之一。

23.一種設備，包含：

處理器，可執行指令格式中之指令，其中該指令格式透過在欄位中的特定值來支援具有位址產生之特定類型的記憶體存取之指明，其中該指令格式包括另一欄位，其內容係指明當具有位址產生之該特定類型的該記憶體存取被指明時的複數可能有正負符號的值之一，

其中當執行一於指令格式中並透過該欄位中之該特定值以指明具有位址產生之該特定類型的該記憶體存取之指令時，該處理器組態成

基於該指令格式中之複數其他欄位的該些內容以判定將使用複數記憶體存取大小之哪一個，以及

基於該判定的記憶體存取大小及該指明的有正負符號的值之相乘以產生位址。

24.如申請專利範圍第 23 項之設備，其中該指令格式亦透過該欄位中之其他值來支援具有無標度位移之記憶體存取及不具位移之記憶體存取的指明。

25.如申請專利範圍第 23 或 24 項之設備，其中該位址之該產生亦基於該指令格式之欄位中所指明的標度、索引、及基礎，當具有位址產生之該特定類型的該記憶體存取被指明時。

26.如申請專利範圍第 25 項之設備，其中該標度是從兩位元標度欄位拉出，該索引是從一位元 X 欄位與三位元 XXX 欄位之組合拉出，以及該基礎是從一位元 B 欄位與三位元 BBB 欄位之組合拉出。

27.如申請專利範圍第 23 或 24 項之設備，其中用來區別於該些複數記憶體存取大小之間的該些複數其他欄位選擇是基於擴充運算欄位內之類別欄位的內容。

28.如申請專利範圍第 27 項之設備，其中當該類別欄位的內容指明第一類別時，該用來區別於該些複數記憶體存取大小之間的欄位包括基礎運算欄位、資料操縱欄位、及資料元件寬度欄位的兩者或更多。

29.如申請專利範圍第 27 項之設備，其中，當該類別欄位的內容指明第二類別時，該用來區別於該些複數記憶體存取大小之間的欄位包括基礎運算欄位、

廣播欄位、向量長度欄位、及資料元件寬度欄位的兩者或更多。

30.如申請專利範圍第 23 或 24 項之設備，其中該另一欄位是單一位元組。

31.如申請專利範圍第 30 項之設備，其中該另一欄位的內容被解譯為介於 -128 與 127 之間的有正負符號的值。

32.如申請專利範圍第 23 或 24 項之設備，其中該些複數記憶體存取大小相應於該記憶體存取中之位元組數。

33.如申請專利範圍第 23 或 24 項之設備，其中該些複數記憶體存取大小為 1 位元組、2 位元組、4 位元組、8 位元組、16 位元組、32 位元組、及 64 位元組。

34. 一種設備，包含：

處理器，可執行指令格式中之指令，其中該指令集包括，其中該指令格式具有第一欄位、第二欄位、及第三欄位，其中該處理器組態成

在從不基於該第一欄位的內容的那些區別指明記憶體存取之該指令格式中的指令；

針對透過該第一欄位的內容指明記憶體存取的該些指令的每一者，基於該第三欄位的內容區別是否廣播及使用複數向量長度之哪一個；及

針對透過該第一欄位的內容不指明記憶體存

取的該些出現的每一者，基於在該第三欄位的內容，區別是否以捨入類型運算或以向量長度類型運算擴充；

針對透過該第一欄位的內容不指明記憶體存取並透過該第三欄位的內容指明該捨入類型運算的該些指令的每一者，基於該第三欄位的內容區別將施加複數捨入運算的哪一者；及

針對該些指令的每一者，基於該第二欄位的內容區別是否履行合併寫入遮罩運算或歸零寫入遮罩運算。

35.如申請專利範圍第 34 項之設備，其中該些複數捨入運算包括捨入至最接近者、無條件捨去、無條件進入、及朝零捨入。

36.如申請專利範圍第 34 或 35 項之設備，其中該指令格式進一步包括資料元件寬度欄位，其中該指令格式透過在該資料元件寬度欄位中的不同值來支援不同資料元件寬度的指明。

37.如申請專利範圍第 36 項之設備，其中該些不同資料元件寬度包括 32 位元和 64 位元。

38.如申請專利範圍第 36 項之設備，其中該指令格式亦透過第四欄位中的不同值來支援 8 位元及 16 位元資料元件寬度的指明。

39.如申請專利範圍第 36 項之設備，其中該指令格式進一步包括第四欄位，其中該第四欄位的內容區

別該資料元件寬度欄位的內容是否在 64 位元與 32 位元資料元件大小之間做選擇或者在 16 位元與 8 位元資料元件大小之間做選擇。

40.如申請專利範圍第 34 或 35 項之設備，其中該些複數向量長度包括 128 位元、256 位元、及 512 位元。

41.如申請專利範圍第 34 或 35 項之設備，其中該廣播運算包括 1 元件粒度廣播。

42.如申請專利範圍第 34 或 35 項之設備，其中該些指令之至少某些係於兩來源運算元之上運算且產生不會覆寫該兩來源運算元之任一者的目的地運算元。

43.如申請專利範圍第 34 或 35 項之設備，其中該些指令之至少某些係於兩來源運算元之上運算且產生覆寫該兩來源運算元之一的目的地運算元。

44.如申請專利範圍第 34 或 35 項之設備，其中該指令格式支援其指明一具有標度位移之記憶體存取運算的該第一欄位的內容，於此情況下該指令格式包括便於從該處拉出位移因子、標度、索引、及基礎之欄位，其中基於將記憶體存取的該大小乘以該位移因子來產生該標度位移，且其中該標度、索引、基礎、及標度位移被使用以產生該記憶體存取之位址。

45.如申請專利範圍第 44 項之設備，其中該位址是由 $2^{\text{標度}} + \text{該索引} + \text{該基礎} + \text{該標度位移}$ 所產生。

46.如申請專利範圍第 44 項之設備，其中該位移因子為介於 -128 與 127 之間的有正負符號的值。

47.如申請專利範圍第 44 項之設備，其中該些複數記憶體存取之支援大小為 1 位元組、2 位元組、4 位元組、8 位元組、16 位元組、32 位元組、及 64 位元組。

48. 一種設備，包含：

處理器，可執行指令格式中之指令，其中該指令格式具有當該指令格式中之指令不指明記憶體存取運算時的欄位，

其中該欄位的內容之第一部分區別是否將該欄位之剩餘者解譯成捨入運算欄位或者取代地解譯成向量長度欄位；

其中該捨入運算欄位的內容區別將施加複數捨入運算的哪一者；及

其中該向量長度欄位的內容區別將使用複數向量長度的哪一者。

49.如申請專利範圍第 48 項之設備，其中該些複數捨入運算包括捨入至最接近者、無條件捨去、無條件進入、及朝零捨入。

50.如申請專利範圍第 48 或 49 項之設備，其中該些複數向量長度包括 128 位元、256 位元、及 512 位元。

51.如申請專利範圍第 48 或 49 項之設備，其中

當該欄位之剩餘者被解譯成捨入運算欄位時，抑制基礎運算欄位所有浮點異常。

52.如申請專利範圍第 48 或 49 項之設備，其中該指令格式進一步包括資料元件寬度欄位，其中該指令格式透過在該資料元件寬度欄位中的不同值來支援不同資料元件寬度的指明。

53.如申請專利範圍第 52 項之設備，其中該些不同資料元件寬度包括 32 位元和 64 位元。

54.如申請專利範圍第 52 項之設備，其中該指令格式亦透過在另一欄位中的不同值來支援 8 位元及 16 位元資料元件寬度的指明。

55.如申請專利範圍第 52 項之設備，其中該指令格式進一步包括另一欄位，其中該另一欄位的內容區別該資料元件寬度欄位的內容是否在 64 位元與 32 位元資料元件大小之間做選擇或者在 16 位元與 8 位元資料元件大小之間做選擇。

第1B圖

在向量親和格式中之第一指令 110A
在向量親和格式中之第二指令 110B
在第二格式中之第一指令 110C
在第三格式中之第一指令 110D
在向量親和格式中之第三指令 110E
在第二格式中之第二指令 110F
在第二格式中之第三指令 110G
在第三格式中之第二指令 110H

■
■
■

第J指令 110J

第1A圖

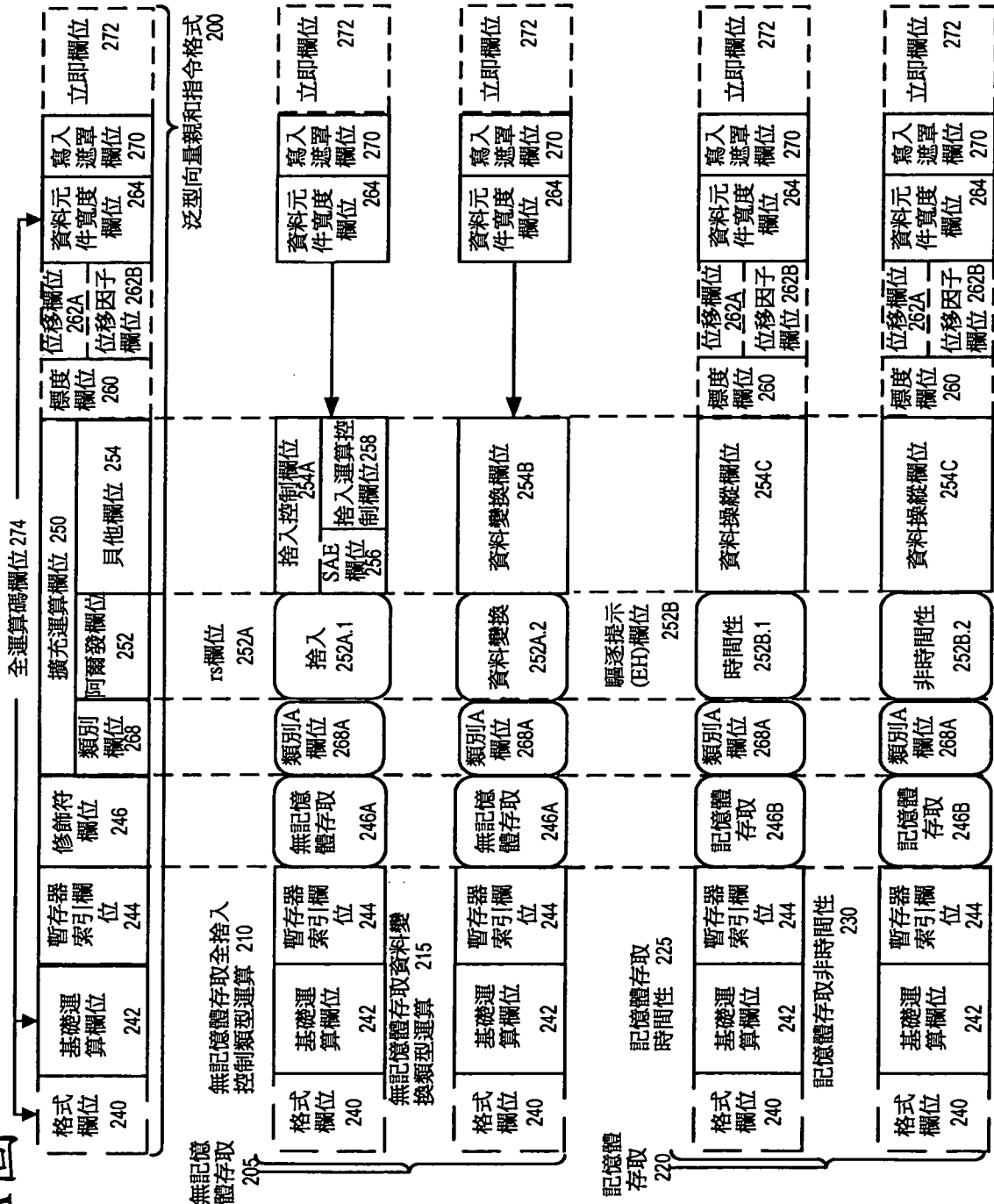
在向量親和格式中之第一指令 100A
在向量親和格式中之第二指令 100B
在向量親和格式中之第三指令 100C
在向量親和格式中之第四指令 100D

■
■
■

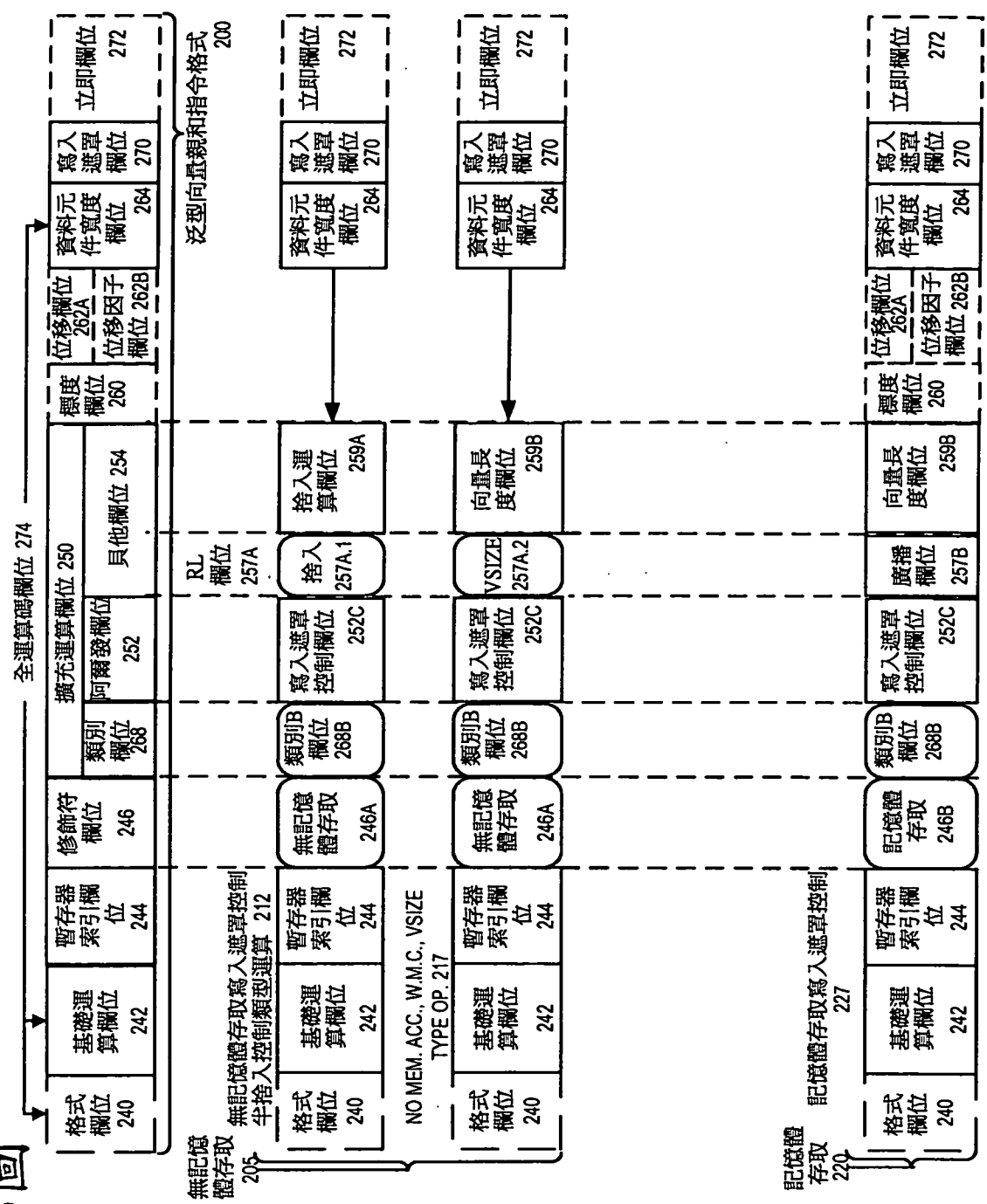
在向量親和格式中之第J指令 100J

圖 式

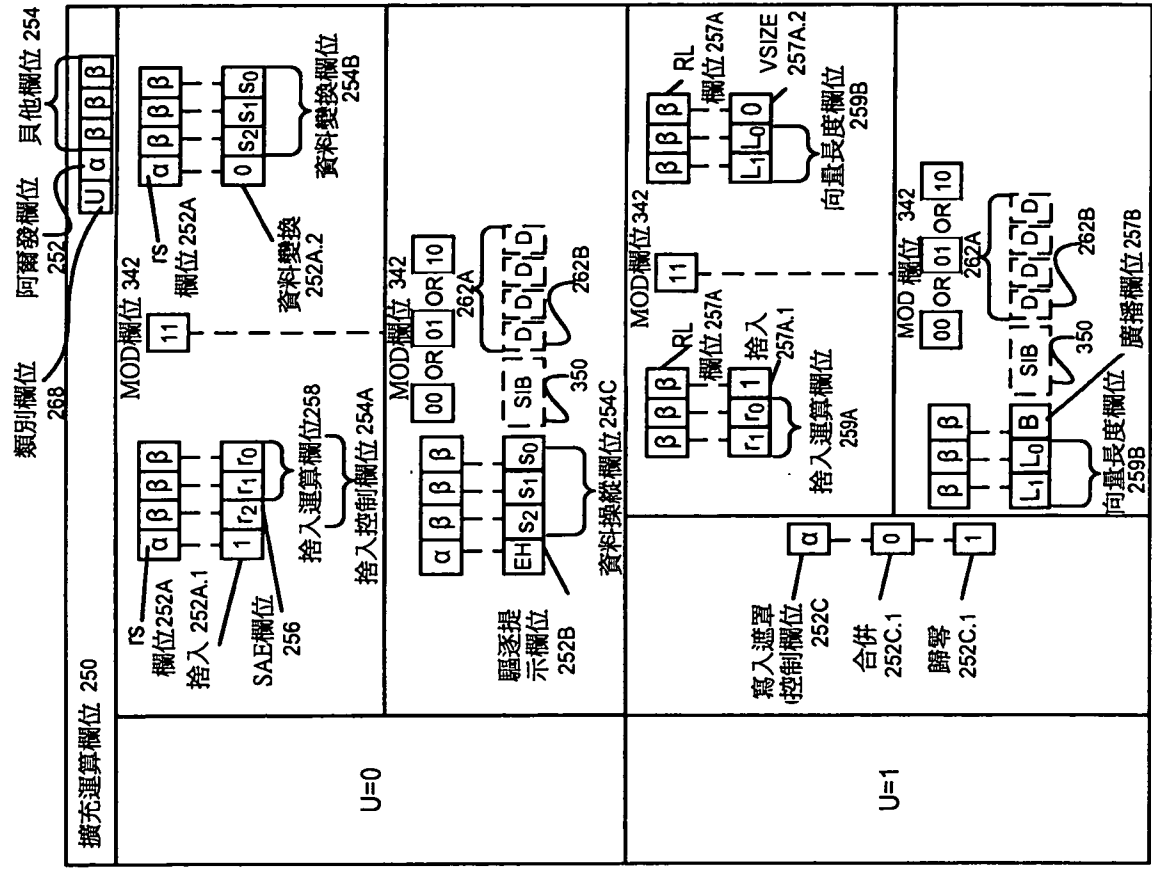
第2A圖



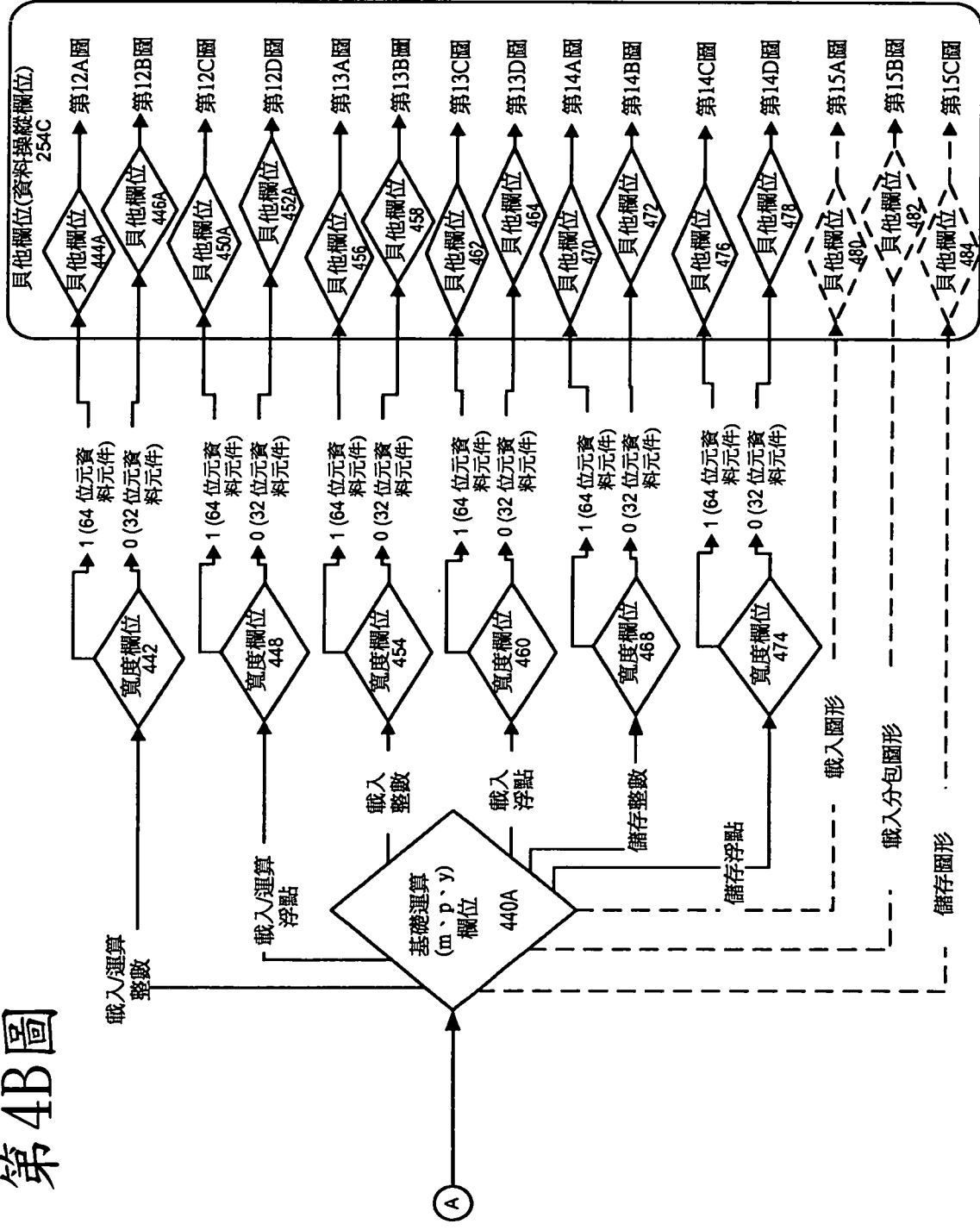
第2B圖



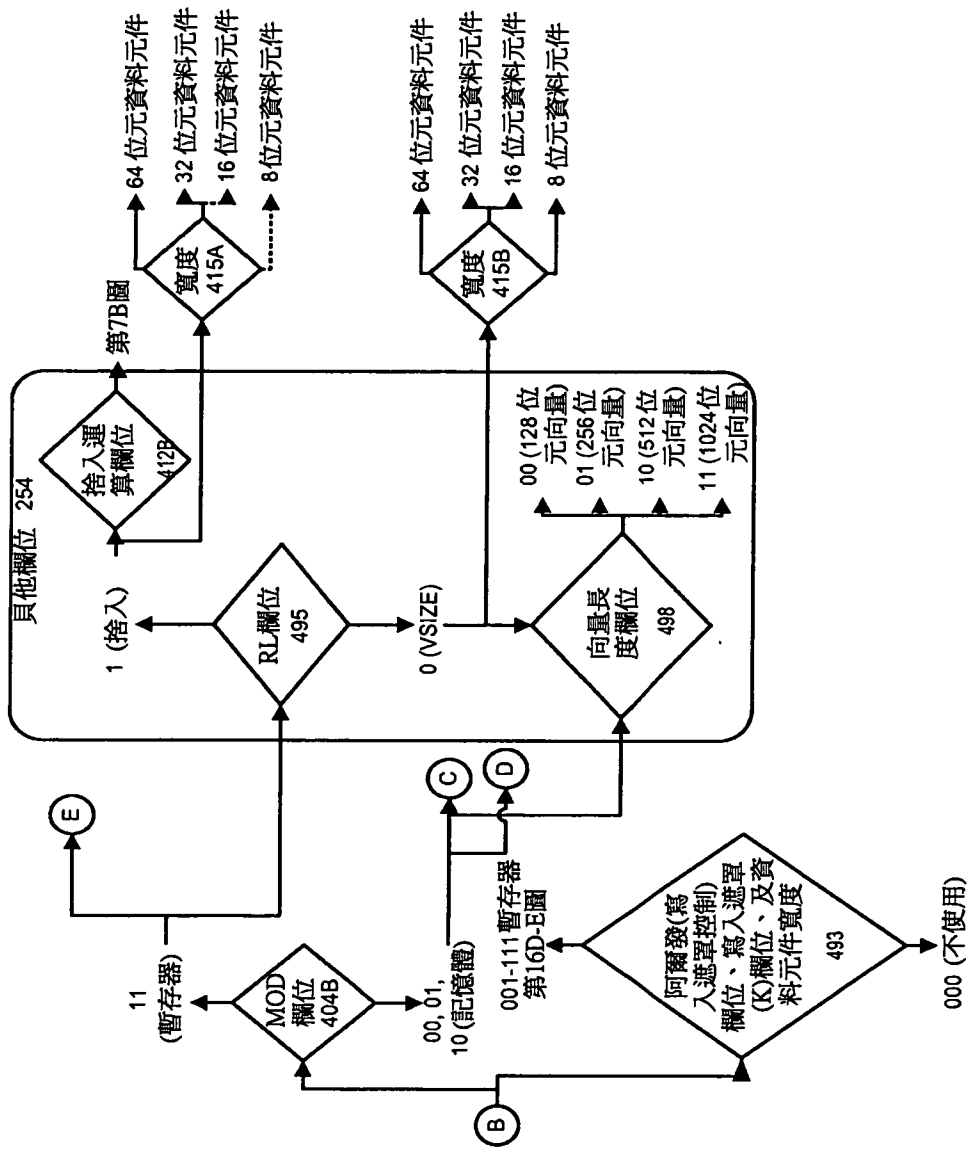
第3D圖



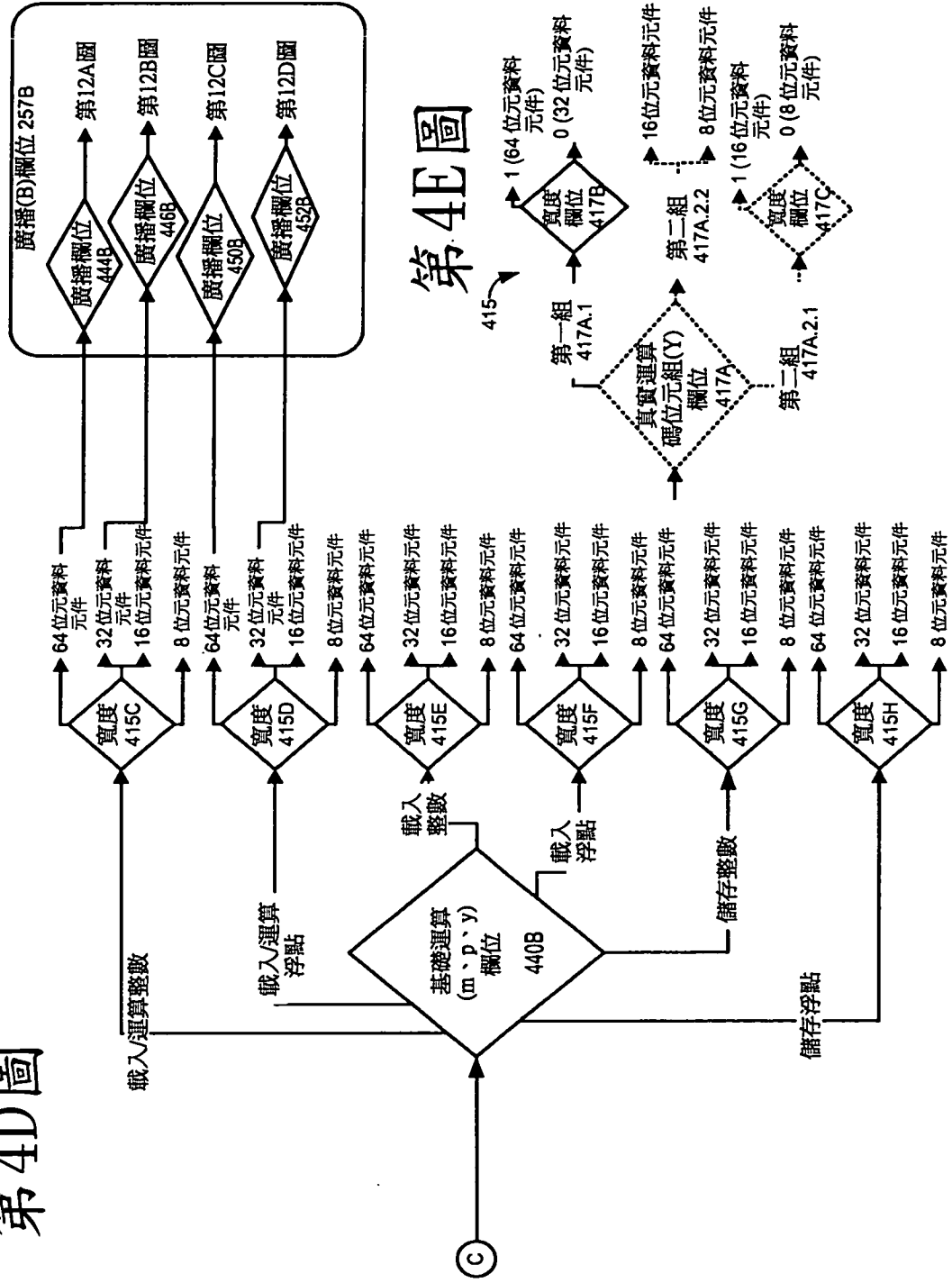
第4B圖



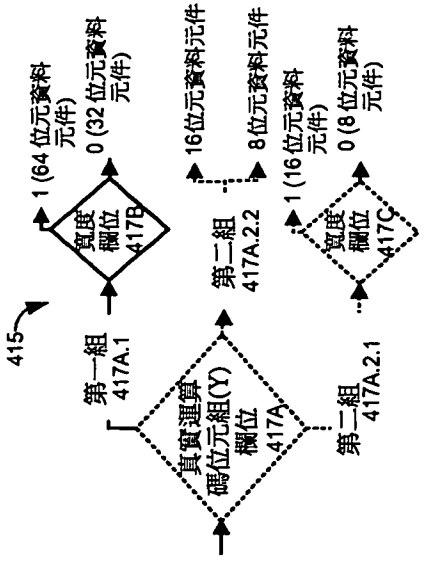
第4C圖



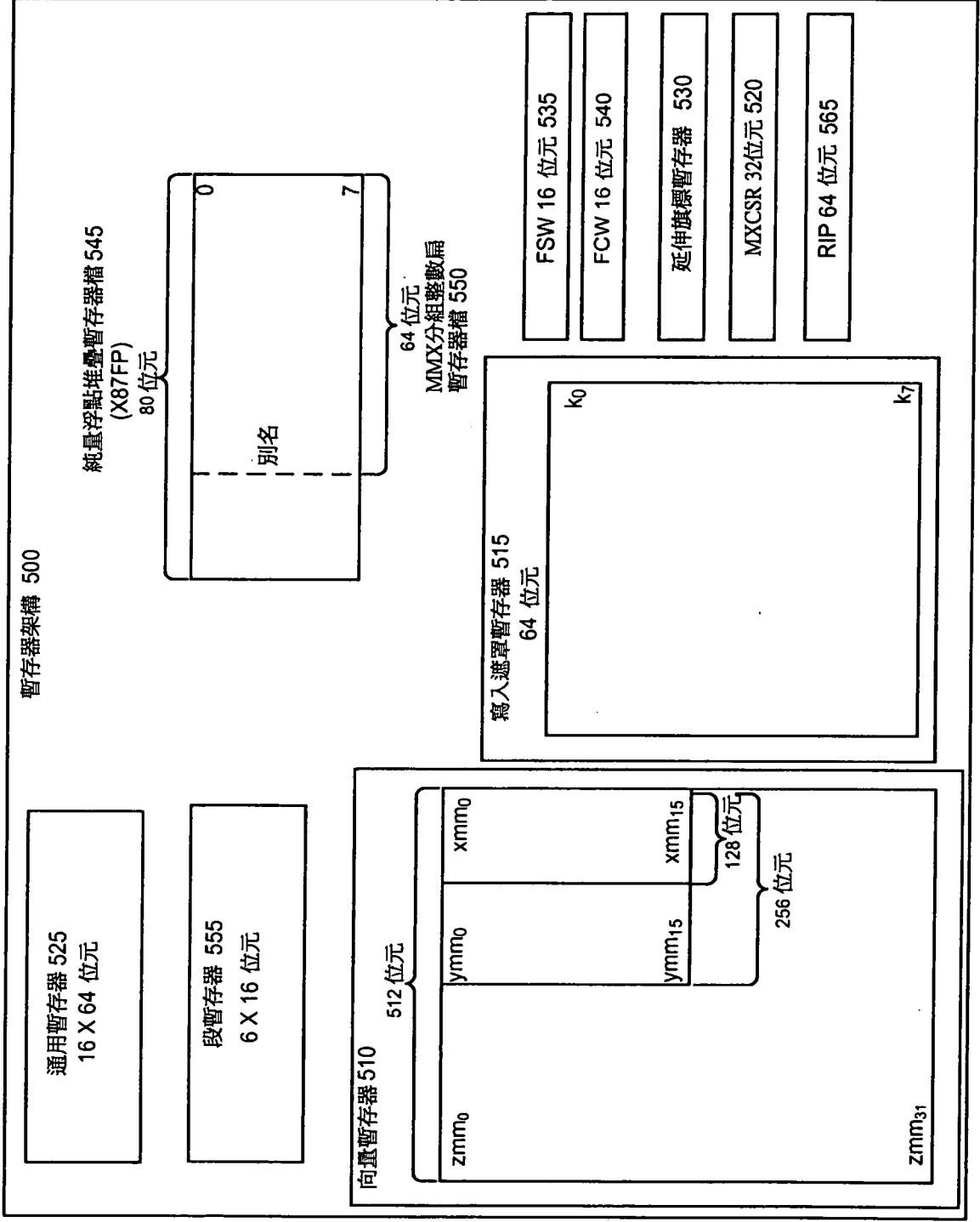
第4D圖



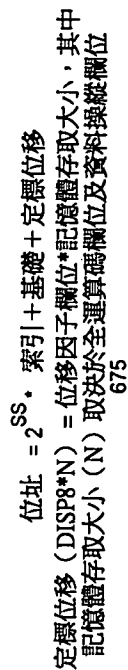
第4E圖



第5圖



第6B圖



第6C圖

位元組	DISP8欄位(1位元組)-128至127	DISP32欄位(4位元組) -2^{31} 至 $2^{31}-1$	示範DISP8*N欄位 [†] (1位元組)-128至127		
			其中 N=1	其中 N=2	其中 N=64
-128	•	•	•	•	•
:					
0	•	•	•	•	•
1	•	•	•		
2	•	•	•	•	
3	•	•	•		
4	•	•	•	•	
5	•	•	•		
6	•	•	•	•	
7	•	•	•		
:					
64	•	•	•	•	•
:					
127	•	•	•		
:					
254 (127*2)		•		•	
:					
8128 (127*64)		•			•
:					
$2^{31}-1$		•			

[†] DISP8*N用來指定標位移的概念，並因此在不同時間DISP8*N係指驅動計算之值的來源（位元因子欄位及/或N）及/或結果（定標位移）

捨入控制欄位
254A

第7A圖

其他欄位	捨入模式	用法
000	朝最近捨入(偶數) 向下捨入(負無限) 向上捨入(正無限) 朝零捨入	{RN}
001		{RD}
010		{RU}
011		{RZ}
100	朝最近捨入(偶數) 向下捨入(負無限) 向上捨入(正無限) 朝零捨入	{RN-SAE}
101		{RD-SAE}
110		{RU-SAE}
111		{RZ-SAE}

不抑制異常

抑制異常

SAE欄位 256 '捨入運算欄位 258

第7B圖

捨入運算欄位 259A	捨入模式	用法
00	朝最近捨入(偶數) 向下捨入(負無限) 向上捨入(正無限) 朝零捨入	{RN-SAE}
01		{RD-SAE}
10		{RU-SAE}
11		{RZ-SAE}

抑制異常

資料變換欄位
254B

418 64- 位元暫存器 SwizzUpConv

第8A圖

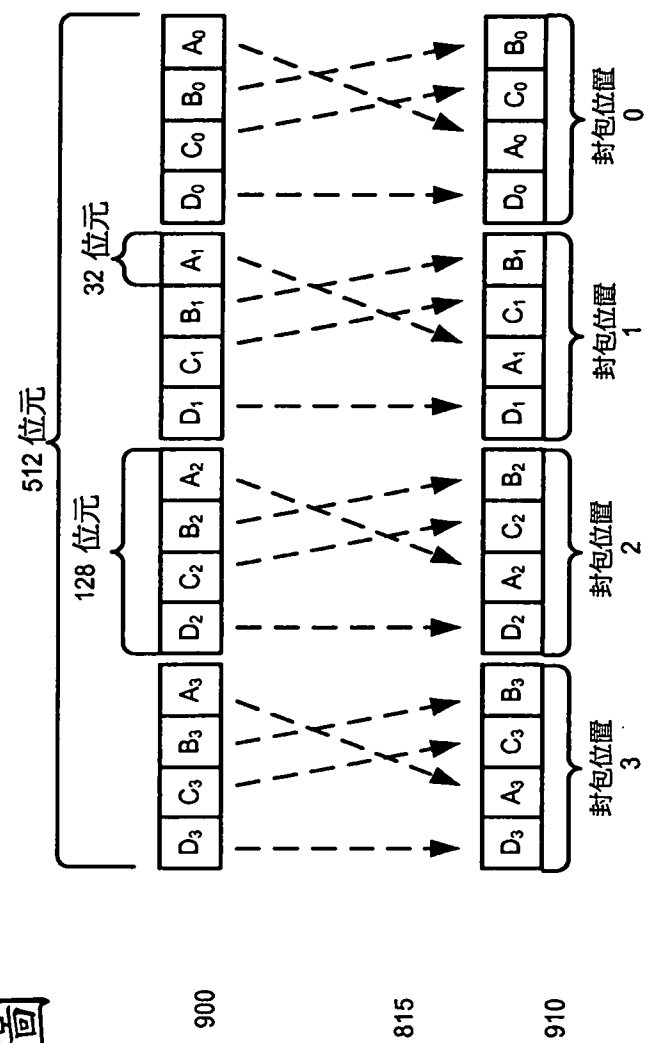
$S_2S_1S_0$	功能：4 x 64- 位元	用法
000	無拌和	zmm0 or zmm0 {dcba}
001	交換（內部）對	zmm0 {cdab}
010	兩兩交換	zmm0 {badc}
011	交叉乘積拌和	zmm0 {dacb}
100	跨四元件封包廣播a元件	zmm0 {aaaa}
101	跨四元件封包廣播b元件	zmm0 {bbbb}
110	跨四元件封包廣播c元件	zmm0 {cccc}
111	跨四元件封包廣播d元件	zmm0 {dddd}

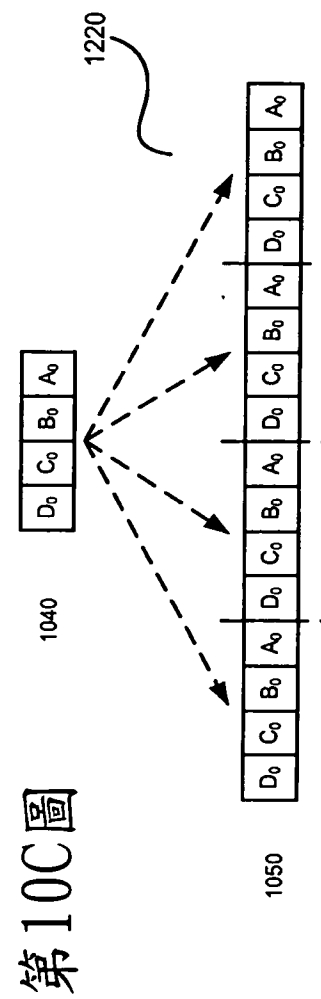
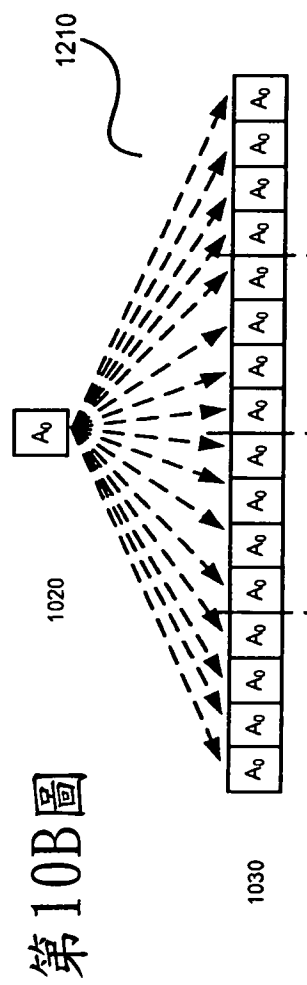
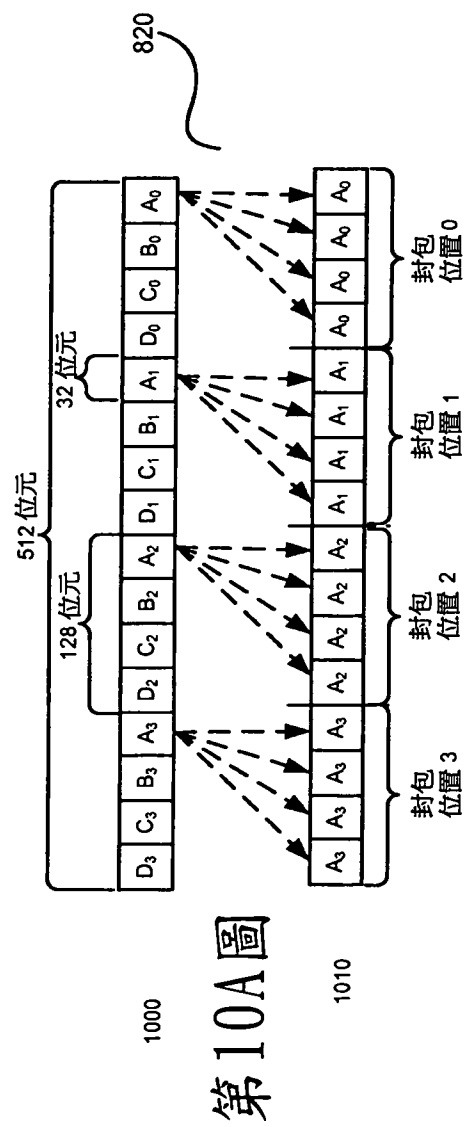
420 32- 位元暫存器 SwizzUpConv

第8B圖

$S_2S_1S_0$	功能：4 x 32- 位元	用法
000	無拌和	zmm0 or zmm0 {dcba}
001	交換（內部）對	zmm0 {cdab}
010	兩兩交換	zmm0 {badc}
011	交叉乘積拌和	zmm0 {dacb}
100	跨四元件封包廣播a元件	zmm0 {aaaa}
101	跨四元件封包廣播b元件	zmm0 {bbbb}
110	跨四元件封包廣播c元件	zmm0 {cccc}
111	跨四元件封包廣播d元件	zmm0 {dddd}

第9圖





第11A圖

運算碼映圖 mmmm	隱合的領先運算碼位元組	立即
0000 0001 0010 0011 0100 – 1111	保留 0000 1111 (0x0F) 0000 1111 0011 1000 (0x0F38) 0000 1111 0011 1010 (0x0F3A) 保留	立即 一些指令 無指令 全部指令

第11B圖

前綴編碼欄位	前綴意義
00	無前綴
01	66 前綴/運算碼延伸
10	F3 前綴/運算碼延伸
11	F2 前綴/運算碼延伸

第12A圖

444 64- 位元整數載入-運算 SwizzUpConv		
$S_2S_1S_0$	功能：	用法
000	無轉換	[rax] {8到8} 或 [rax]
001	廣播1元件 (x8)	[rax] {1到8}
010	廣播4元件 (x2)	[rax] {4到8}
011	保留 (必須 #UD)	N/A
100	保留 (必須 #UD)	N/A
101	保留 (必須 #UD)	N/A
110	保留 (必須 #UD)	N/A
111	保留 (必須 #UD)	N/A

257B

254C

第12B圖

446 32- 位元整數載入-運算SwizzUpConv.		
$S_2S_1S_0$	功能：	用法
000	無轉換	[rax] {16到16} 或 [rax]
001	廣播1元件 (x16)	[rax] {1到16} 1210
010	廣播4元件 (x4)	[rax] {4到16} 1220
011	保留 (必須 #UD)	N/A
100	uint8 到 uint32	[rax] {uint8}
101	sint8 到 sint32	[rax] {sint8}
110	uint16 到 uint32	[rax] {uint16}
111	sint16 到 sint32	[rax] {sint16}

257B

第12C圖

450 64- 位元浮點載入-運算 SwizzUpConv		
$S_2S_1S_0$	功能：	用法
000	無轉換	[rax] {8到8} 或 [rax]
001	廣播1元件 (x8)	[rax] {1到8}
010	廣播4元件 (x2)	[rax] {4到8}
011	保留 (必須 #UD)	N/A
100	保留 (必須 #UD)	N/A
101	保留 (必須 #UD)	N/A
110	保留 (必須 #UD)	N/A
111	保留 (必須 #UD)	N/A

257B

254C

第12D圖

452 32- 位元浮點載入-運算 SwizzUpConv		
$S_2S_1S_0$	功能：	用法
000	無轉換	[rax] {16到16} 或 [rax]
001	廣播1元件 (x16)	[rax] {1到16}
010	廣播4元件 (x4)	[rax] {4到16}
011	float16 到 float32	[rax] {float16}
100	uint8 到 float32	[rax] {uint8}
101	unorm8 到 float32	[rax] {unorm8}
110	uint16 到 float32	[rax] {uint16}
111	sint16 到 float32	[rax] {sint16}

257B

456

UpConv₁₆₄ (INT64)

第13A圖

$S_2S_1S_0$	功能：	用法
000	無轉換	[rax] {8到8} 或 [rax]
001	保留 (必須 #UD)	N/A
010	保留 (必須 #UD)	N/A
011	保留 (必須 #UD)	N/A
100	保留 (必須 #UD)	N/A
101	保留 (必須 #UD)	N/A
110	保留 (必須 #UD)	N/A
111	保留 (必須 #UD)	N/A

254C

458

UpConv₁₃₂ (INT32)

第13B圖

$S_2S_1S_0$	功能：	用法
000	無轉換	[rax] {16到16} 或 [rax]
001	保留 (必須 #UD)	N/A
010	保留 (必須 #UD)	N/A
011	保留 (必須 #UD)	N/A
100	uint8 到 uint32	[rax] {uint8}
101	sint8 到 sint32	[rax] {sint8}
110	uint16 到 uint32	[rax] {uint16}
111	sint16 到 sint32	[rax] {sint16}

第13C圖

462
UpConv_{f64} (FP64)

$S_2S_1S_0$	功能：	用法
000	無轉換	$[rax]$ {8到8} 或 $[rax]$
001	保留 (必須 #UD)	N/A
010	保留 (必須 #UD)	N/A
011	保留 (必須 #UD)	N/A
100	保留 (必須 #UD)	N/A
101	保留 (必須 #UD)	N/A
110	保留 (必須 #UD)	N/A
111	保留 (必須 #UD)	N/A

254C

第13D圖

464
UpConv_{f32} (FP32)

$S_2S_1S_0$	功能：	用法
000	無轉換	$[rax]$ {16到16} 或 $[rax]$
001	保留 (必須 #UD)	N/A
010	保留 (必須 #UD)	N/A
011	float16 到 float32	$[rax]$ {float16}
100	uint8 到 float32	$[rax]$ {uint8}
101	sint8 到 float32	$[rax]$ {sint8}
110	uint16 到 float32	$[rax]$ {uint16}
111	sint16 到 float32	$[rax]$ {sint16}

第14A圖

470
DownConv₆₄ (INT64)

$S_2S_1S_0$	功能：	用法
000	無轉換	zmm1
001	保留 (必須 #UD)	N/A
010	保留 (必須 #UD)	N/A
011	保留 (必須 #UD)	N/A
100	保留 (必須 #UD)	N/A
101	保留 (必須 #UD)	N/A
110	保留 (必須 #UD)	N/A
111	保留 (必須 #UD)	N/A

254C

第14B圖

472
DownConv₃₂ (INT32)

$S_2S_1S_0$	功能：	用法
000	無轉換	zmm1
001	保留 (必須 #UD)	N/A
010	保留 (必須 #UD)	N/A
011	保留 (必須 #UD)	N/A
100	uint32 到 uint8	zmm1 {uint8}
101	sint32 到 sint8	zmm1 {sint8}
110	uint32 到 uint16	zmm1 {uint16}
111	sint32 到 sint16	zmm1 {sint16}

第14C圖

476
DownConv_{f04} (FP04)

$S_2S_1S_0$	功能：	用法
000	無轉換	zmm1
001	保留 (必須 #UD)	N/A
010	保留 (必須 #UD)	N/A
011	保留 (必須 #UD)	N/A
100	保留 (必須 #UD)	N/A
101	保留 (必須 #UD)	N/A
110	保留 (必須 #UD)	N/A
111	保留 (必須 #UD)	N/A

254C

第14D圖

478
DownConv_{f32} (FP32)

$S_2S_1S_0$	功能：	用法
000	無轉換	zmm1
001	保留 (必須 #UD)	N/A
010	保留 (必須 #UD)	N/A
011	float32 到 float16	zmm1 {float16}
100	float32 到 uint8	zmm1 {uint8}
101	float32 到 sint8	zmm1 {sint8}
110	float32 到 uint16	zmm1 {uint16}
111	float32 到 sint16	zmm1 {sint16}

第15A圖

480
UpConv_{q32} (GR, FP32)

$S_2 S_1 S_0$	功能：	用法
000	無轉換	[rax] {16到16} ¹ 或 [rax]
001	保留 (必須 #UD)	N/A
010	srgb8 到 float32	[rax] {srgb8}
011	reserved (must #UD)	N/A
100	unorm8 到 float32	[rax] {unorm8}
101	snorm8 到 float32	[rax] {snorm8}
110	unorm16 到 float32	[rax] {unorm16}
111	snorm16 到 float32	[rax] {snorm16}

254C

第15B圖

482
UpConv_{fp32} (P.GR, FP32)

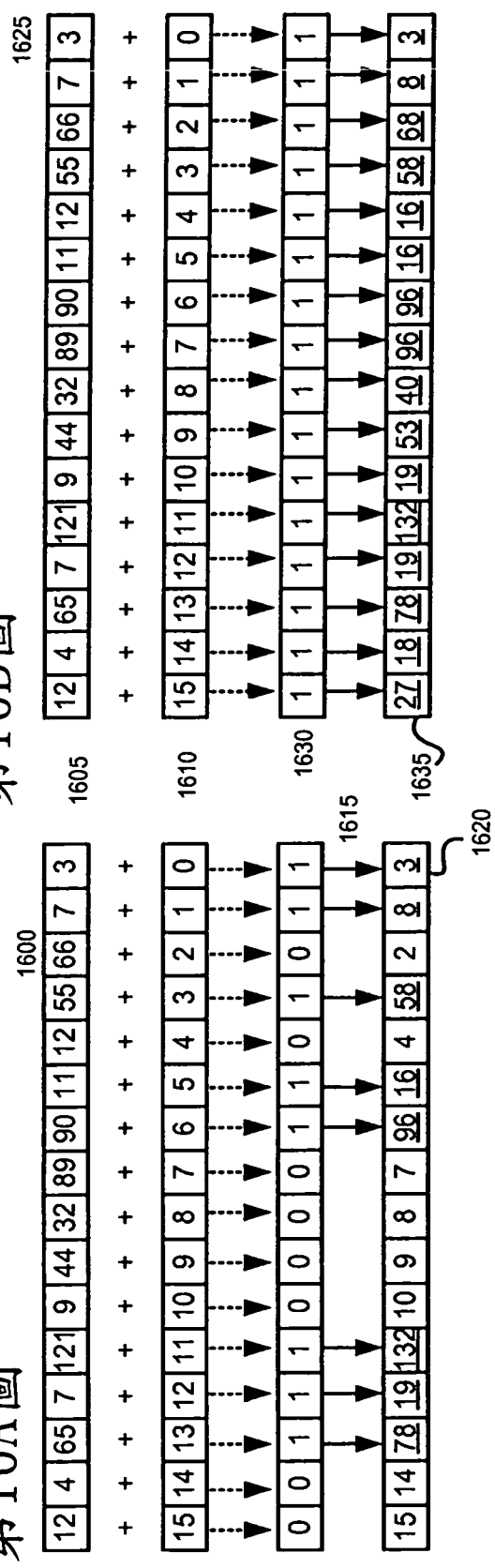
$S_2 S_1 S_0$	功能：	用法
000	無轉換	[rax] {16到16} ¹ 或 [rax]
001	float10C:11B:11A 槽 A	[rax] {float11A}
010	float10C:11B:11A 槽 B	[rax] {float11C}
011	float10C:11B:11A 槽 C	[rax] {float10C}
100	unorm2D:10C:10B:10A 槽 A	[rax] {unorm10A}
101	unorm2D:10C:10B:10A 槽 B	[rax] {unorm10B}
110	unorm2D:10C:10B:10A 槽 C	[rax] {unorm10C}
111	unorm2D:10C:10B:10A 槽 D	[rax] {unorm2D}

第15C圖

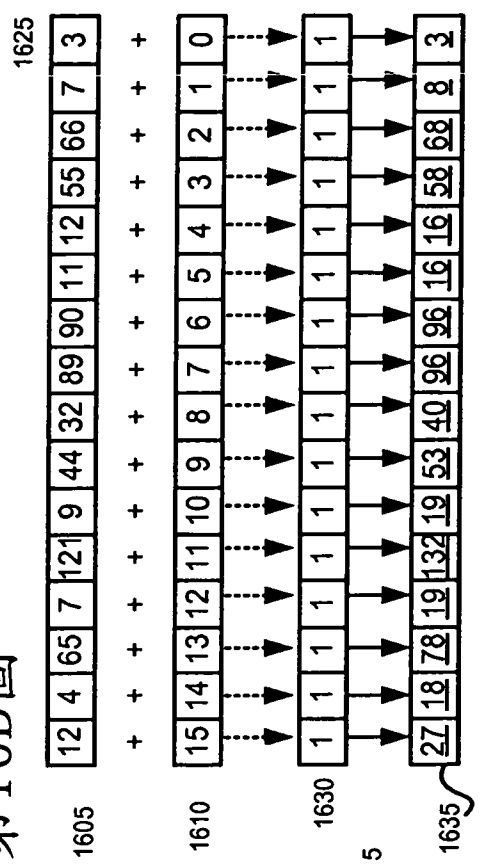
484
DownConv_{q32} (GR, FP32)

$S_2 S_1 S_0$	功能：	用法
000	無轉換	zmm1
001	保留 (必須 #UD)	N/A
010	保留 (必須 #UD)	N/A
011	float32 到具有朝零捨入之 float16	zmm1 {float16rz}
100	float32 到 unorm8	zmm1 {unorm8}
101	float32 到 snorm8	zmm1 {snorm8}
110	float32 到 unorm16	zmm1 {unorm16}
111	float32 到 snorm16	zmm1 {snorm16}

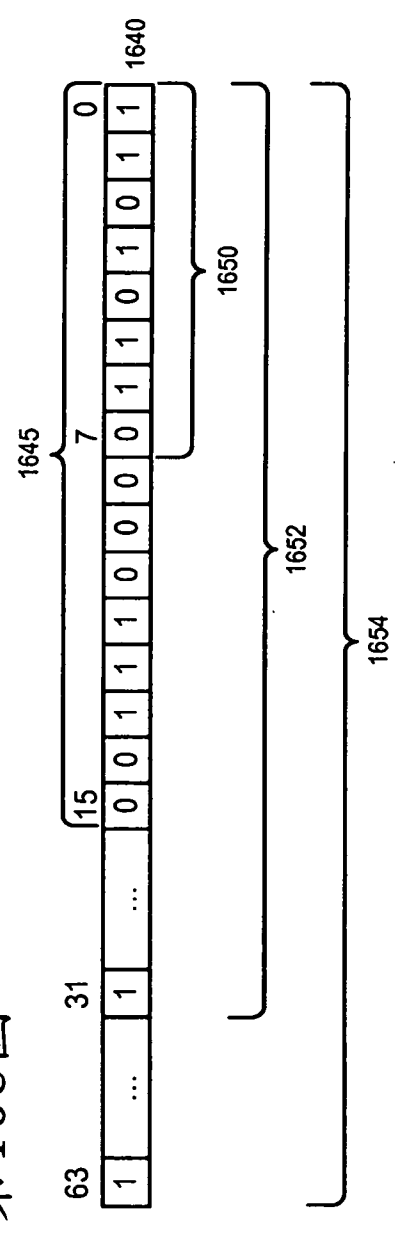
第16A圖



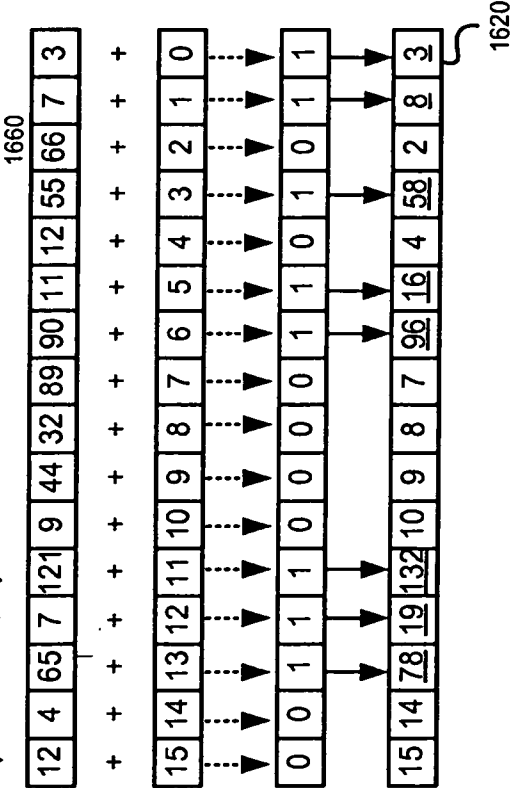
第16B圖



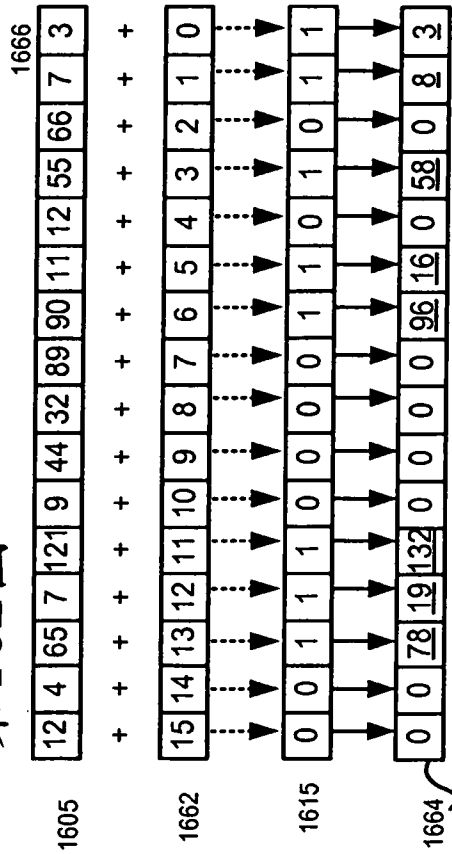
第16C圖



第16D圖

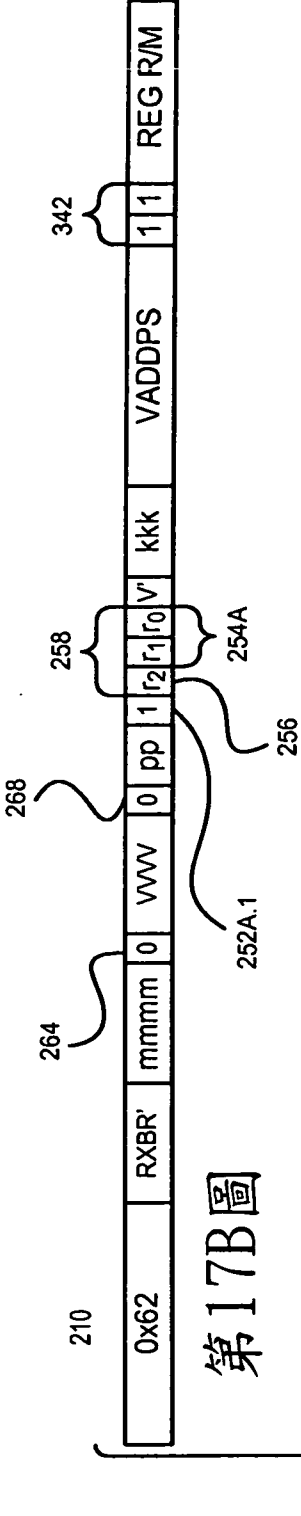


第16E圖



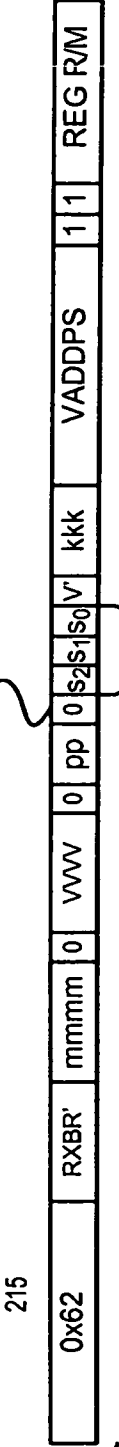


第17A圖

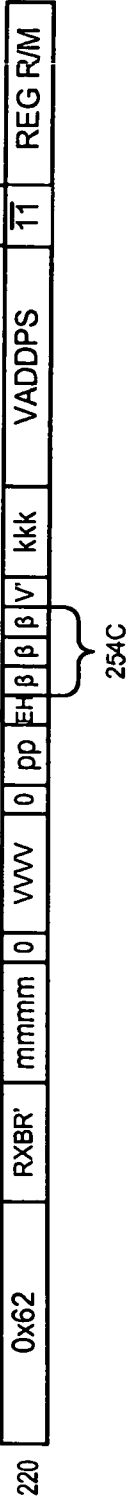


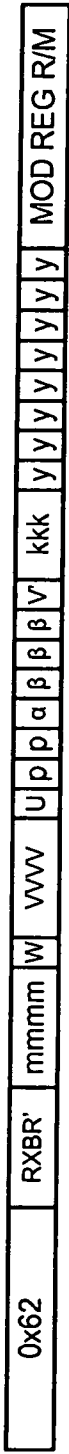
第17B圖

第17C圖

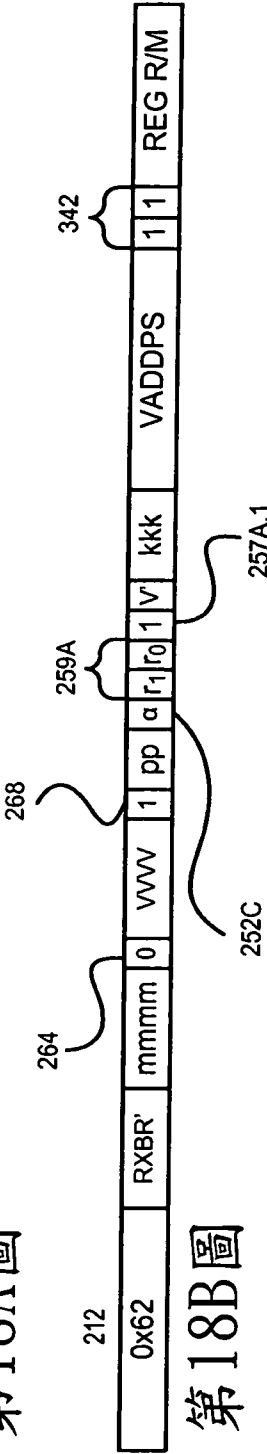


第17D圖

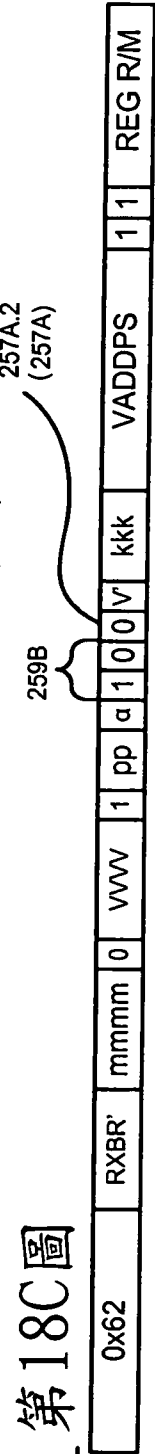




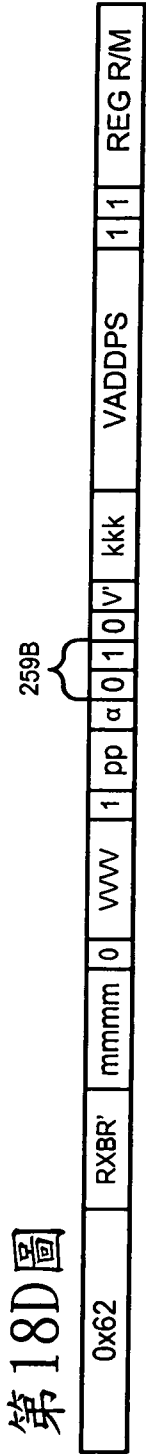
第18A圖



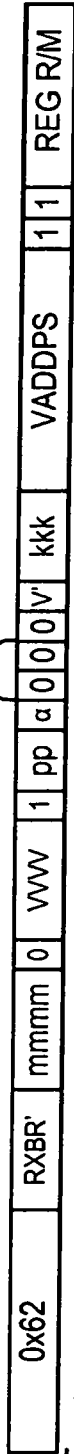
第18B圖



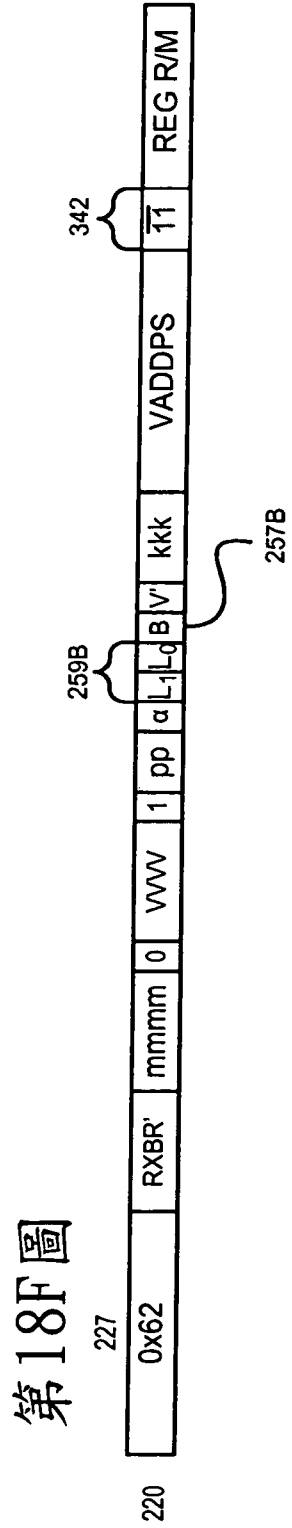
第18C圖



第18D圖

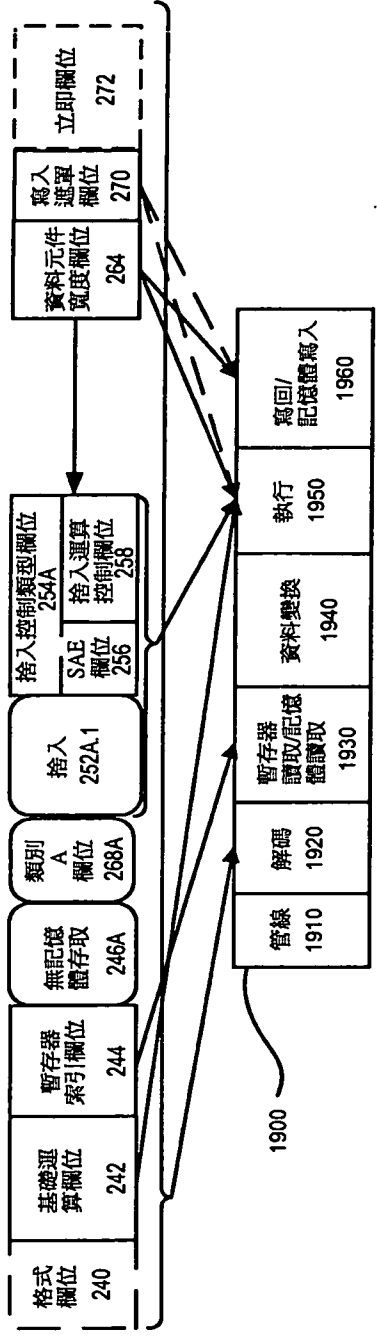


第18E圖

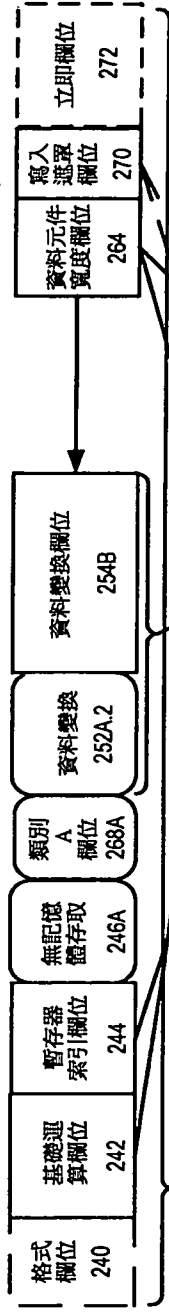


第18F圖

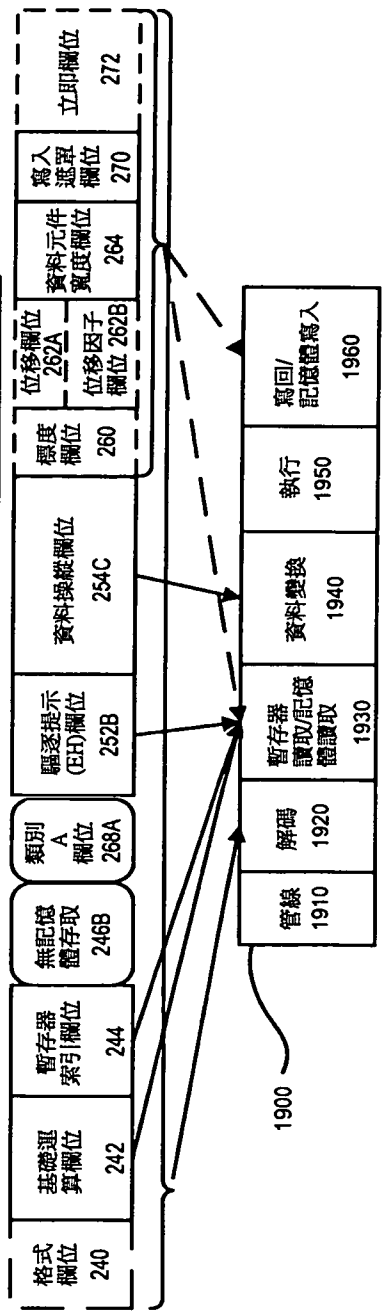
第19A圖 210



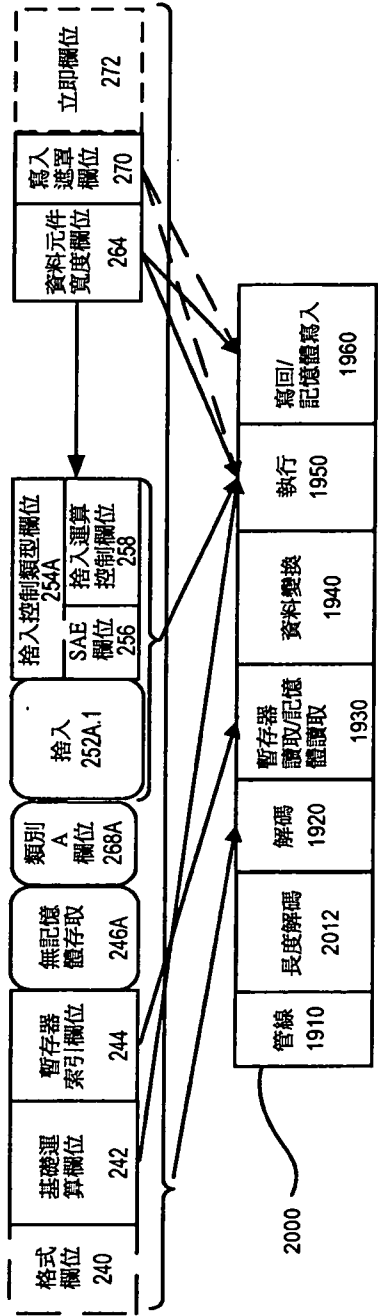
第19B圖 215



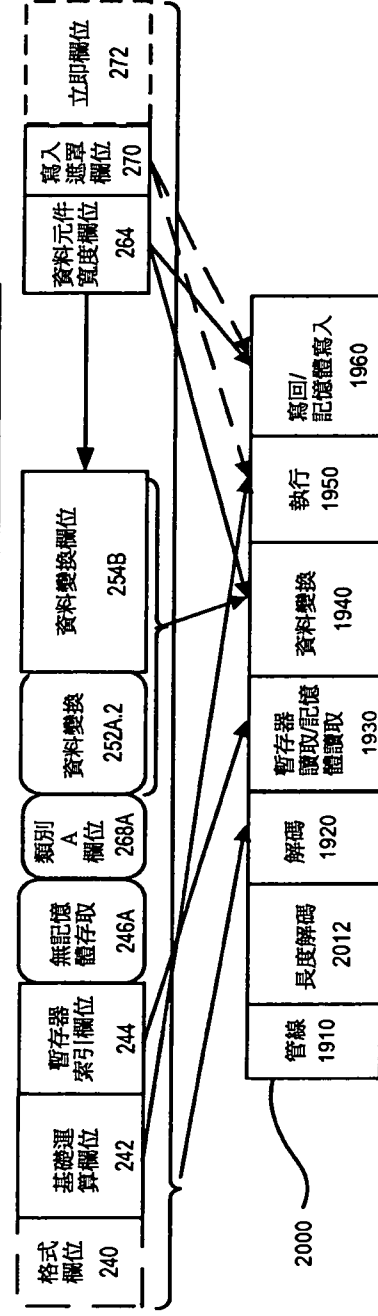
第19C圖 220



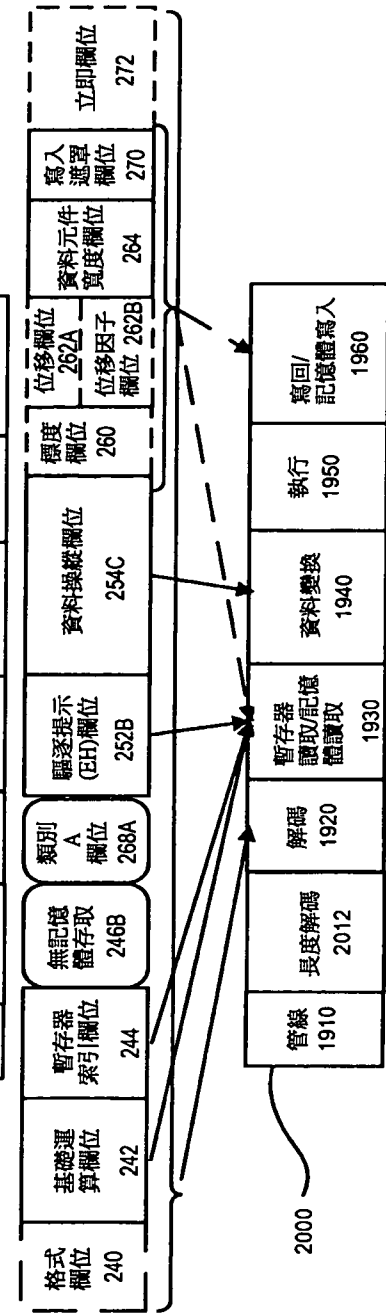
第20A圖 210



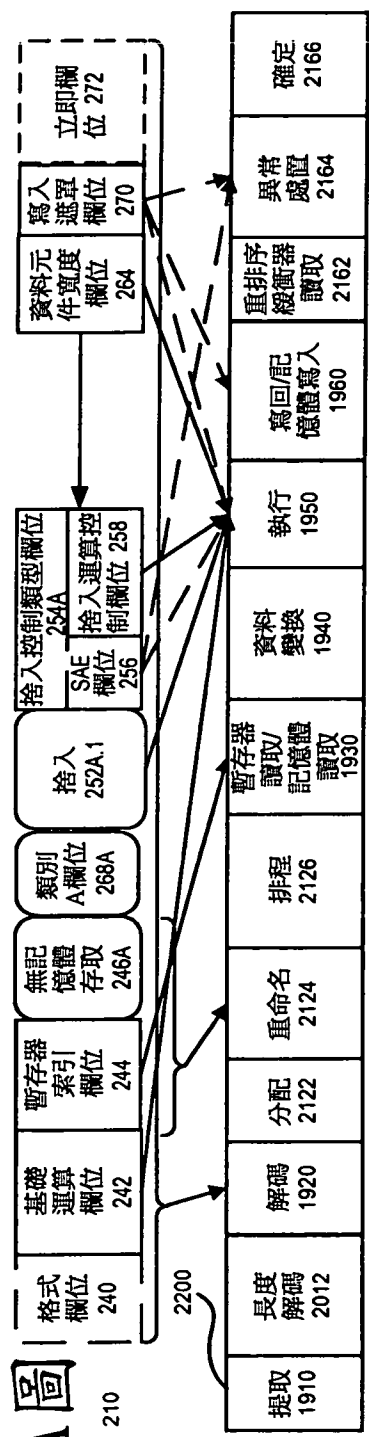
第20B圖 215



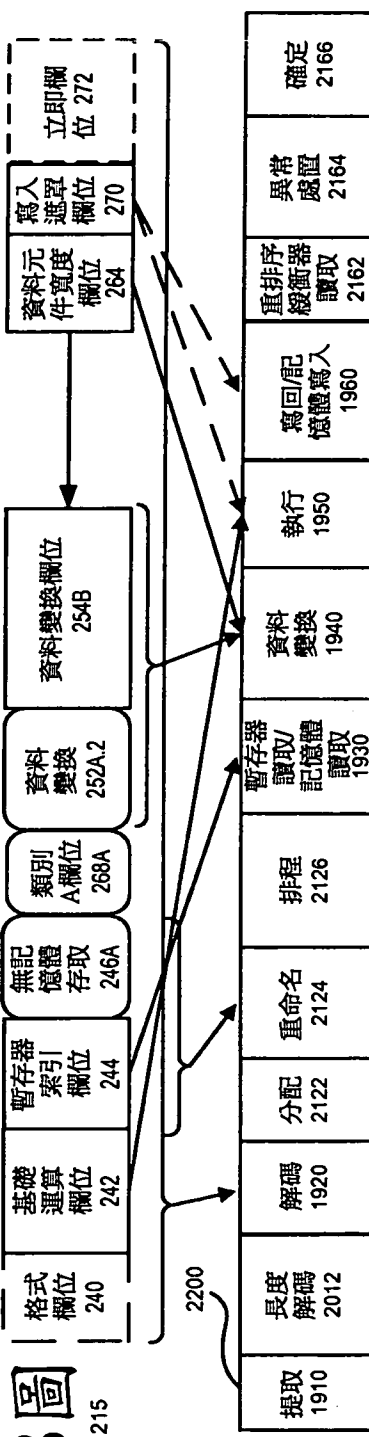
第20C圖 220



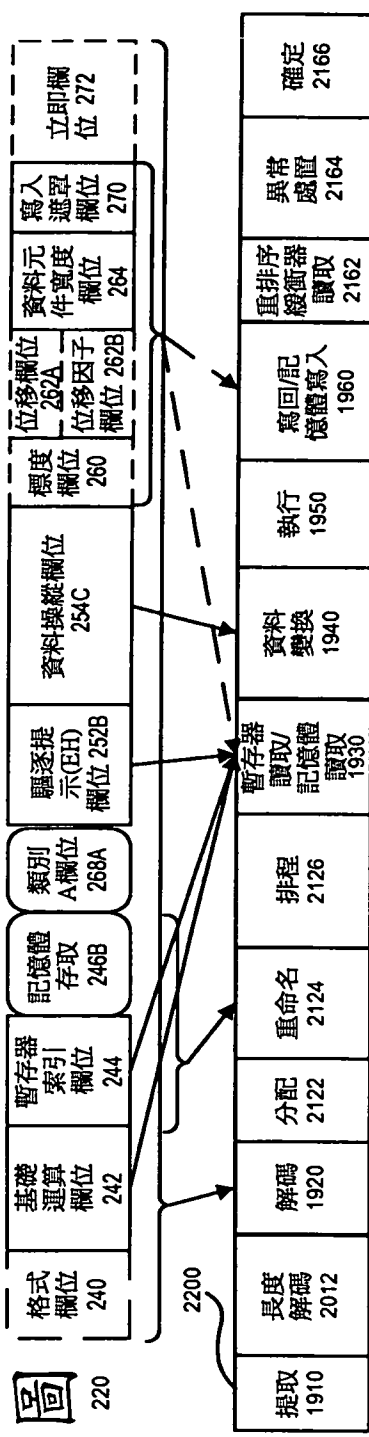
第21A圖



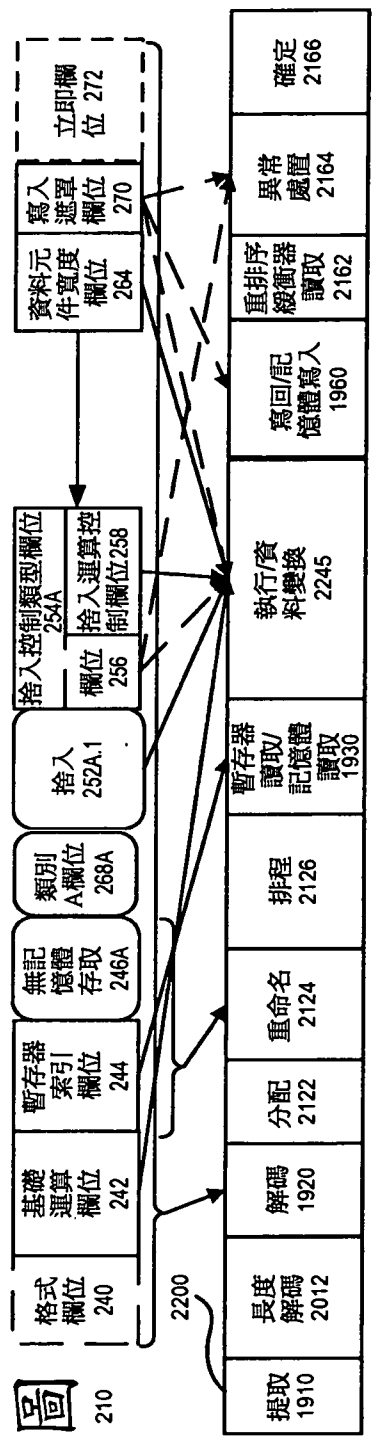
第21B圖



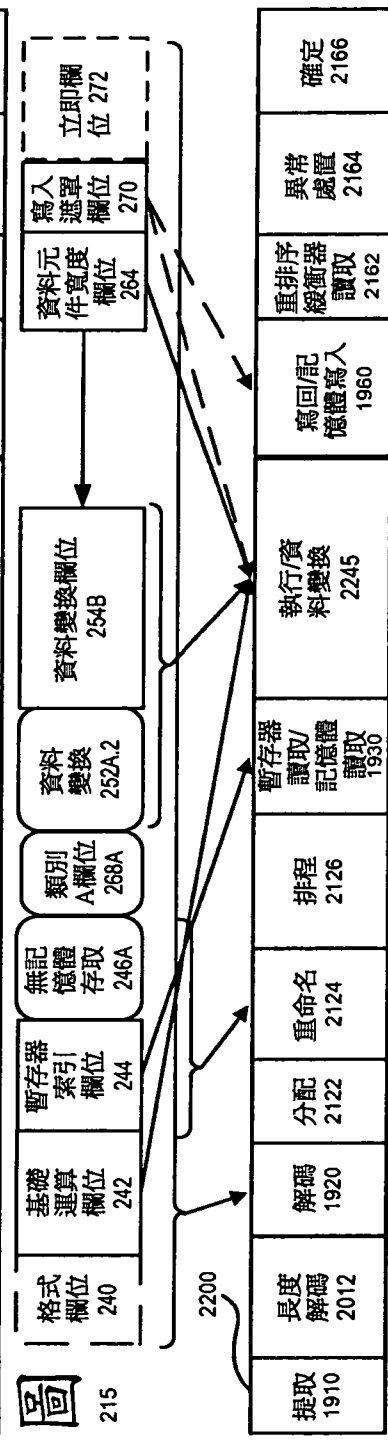
第21C圖



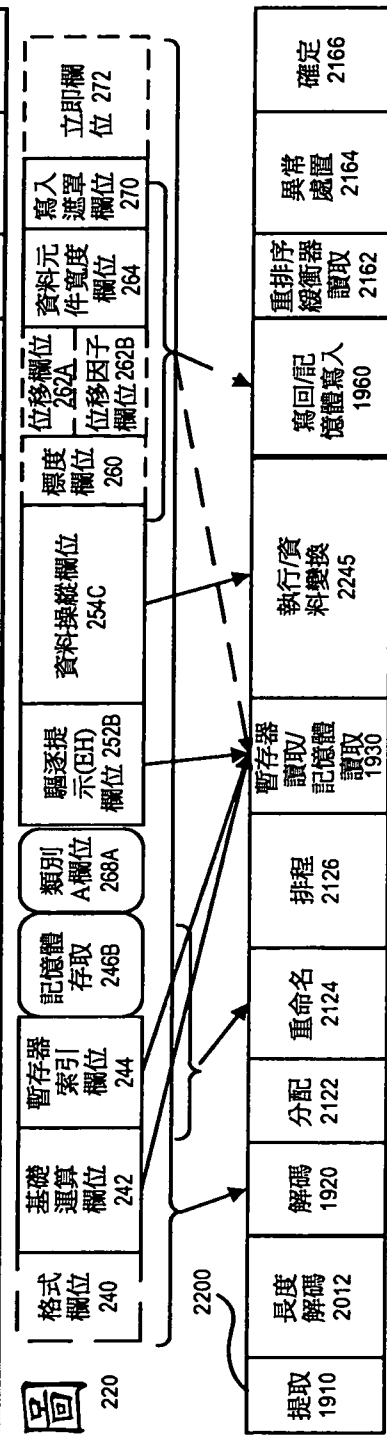
第22A圖



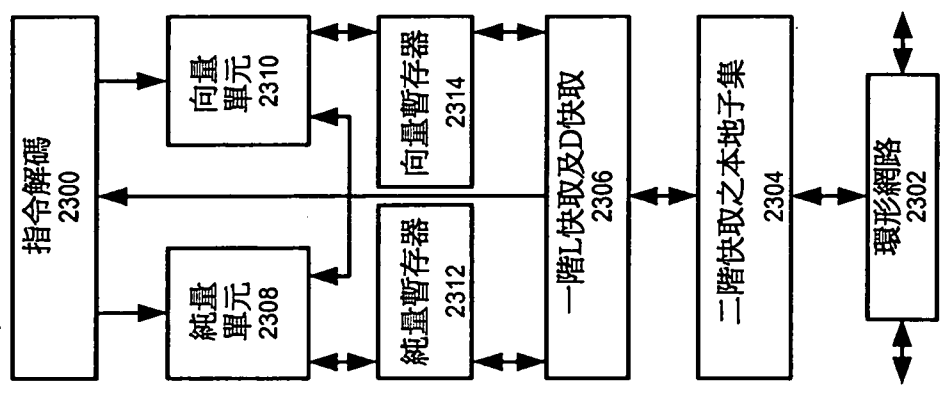
第22B圖



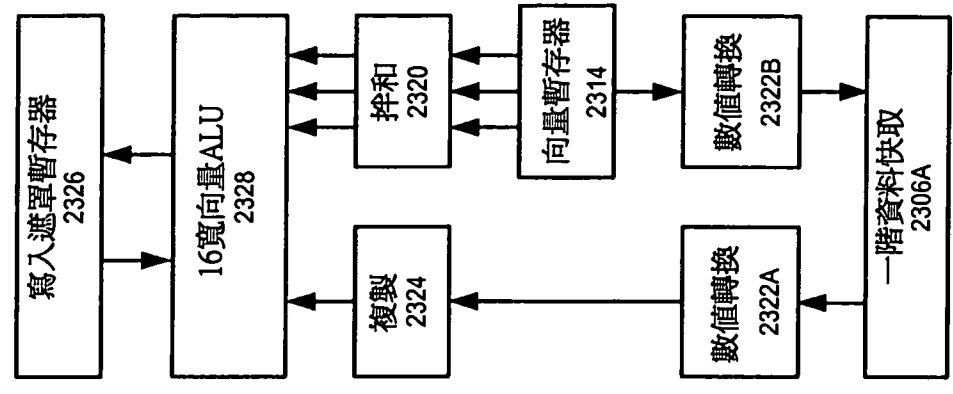
第22C圖

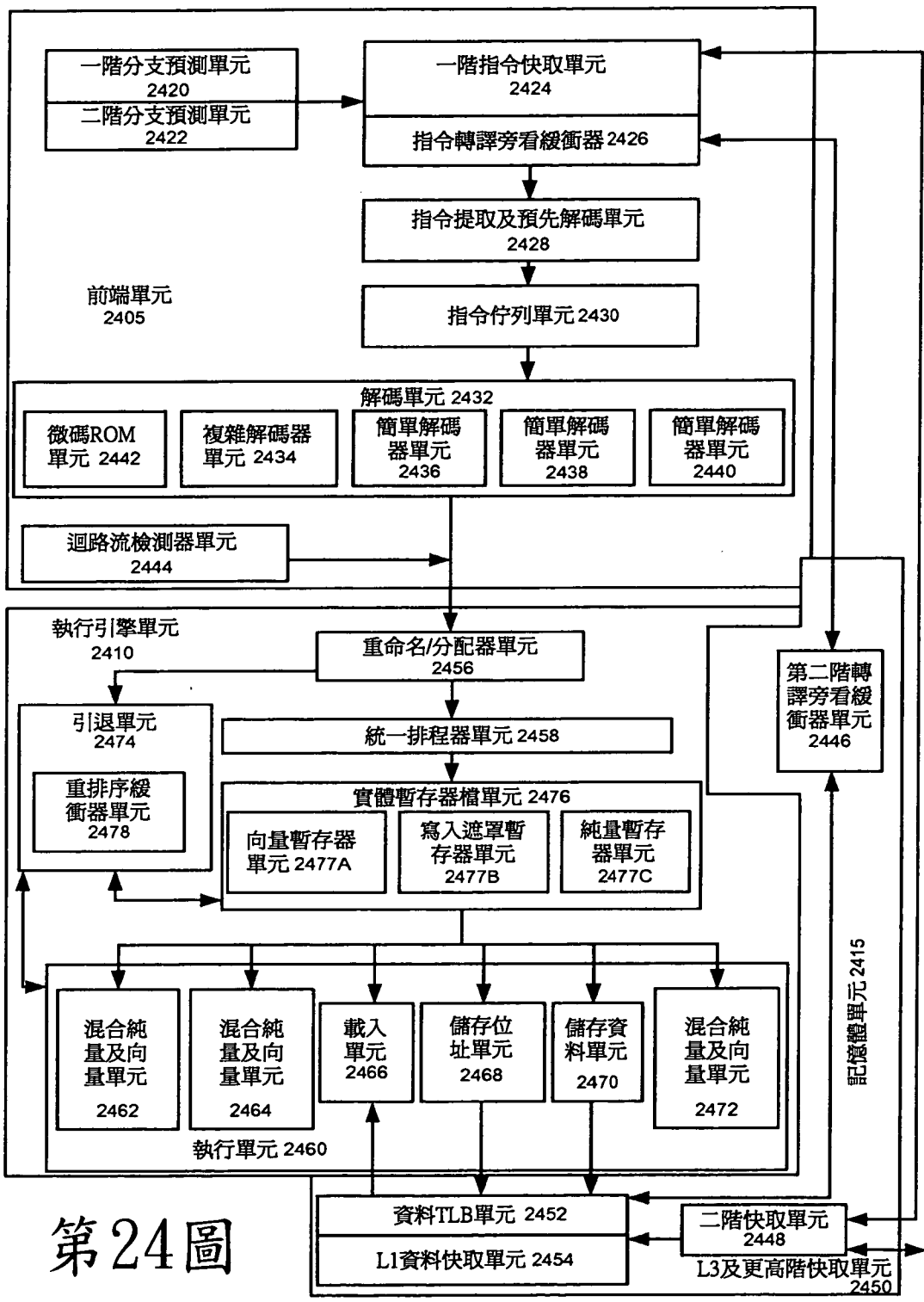


第23A圖

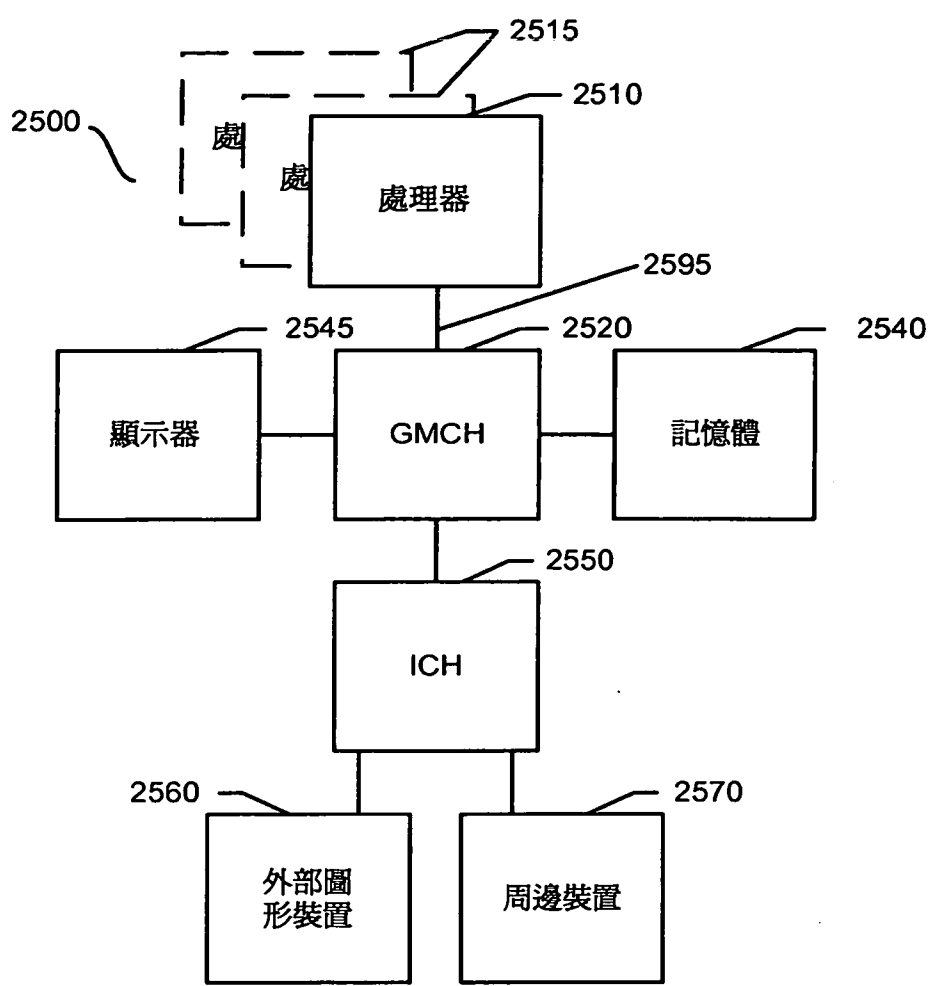


第23B圖

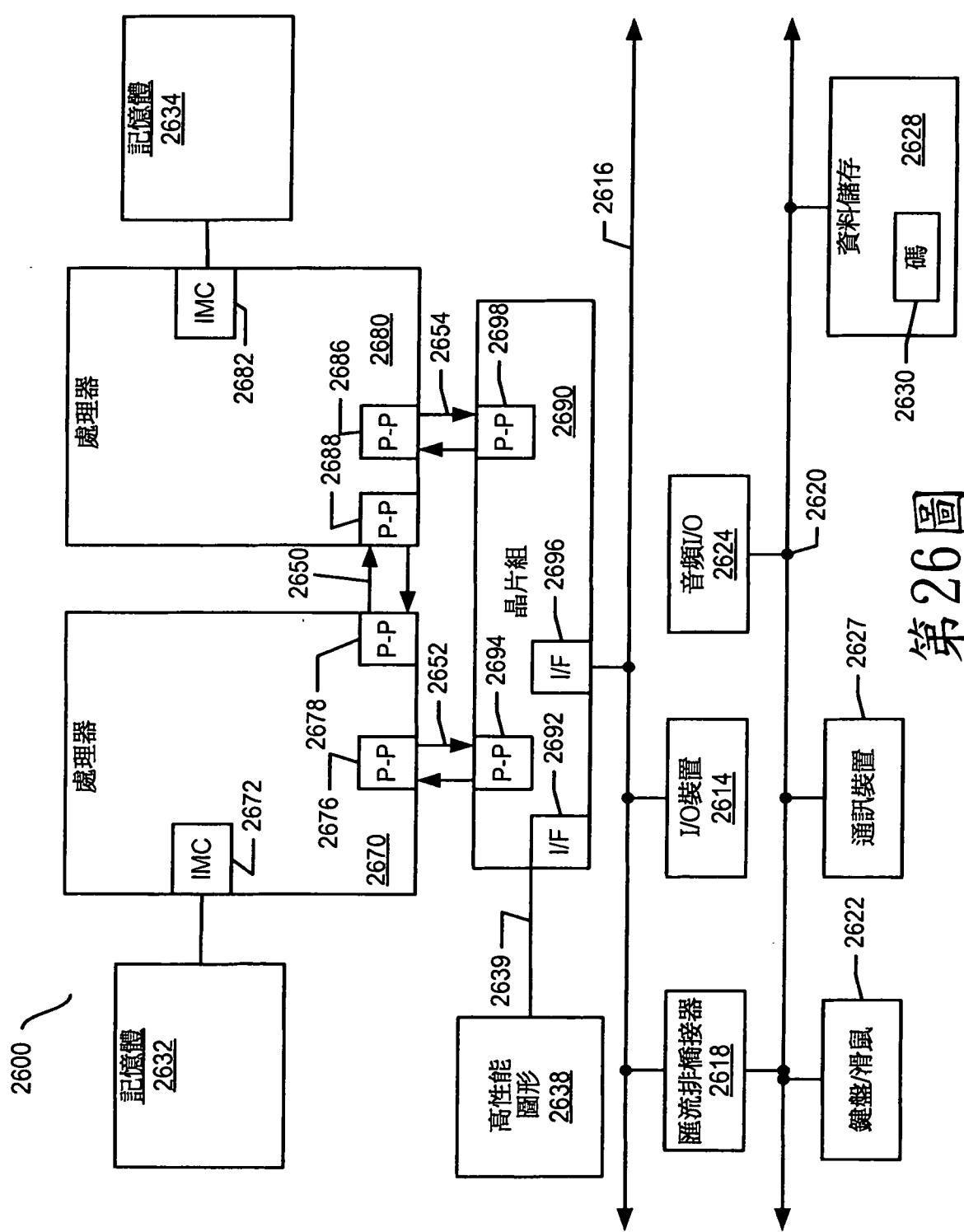




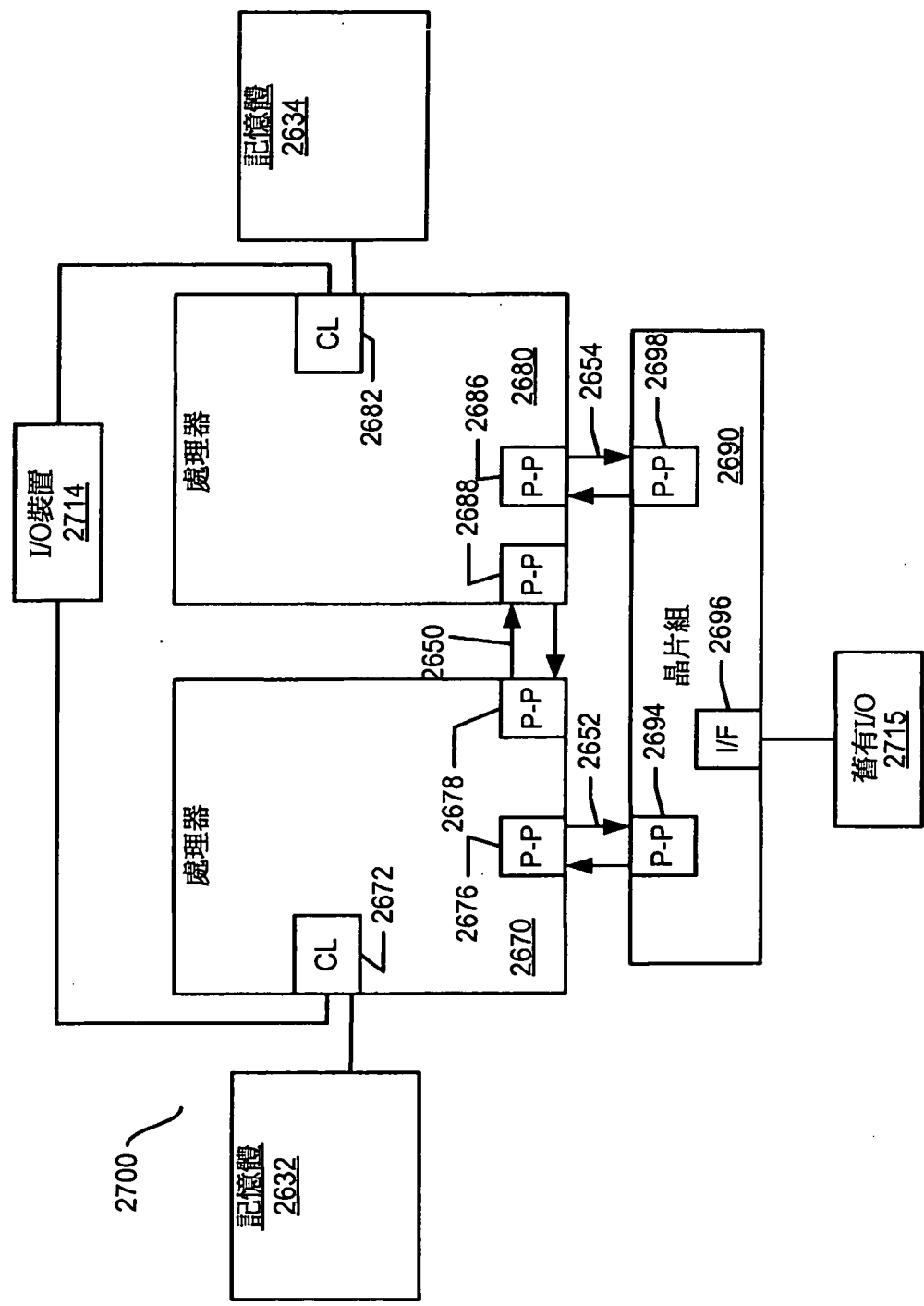
第24圖



第25圖



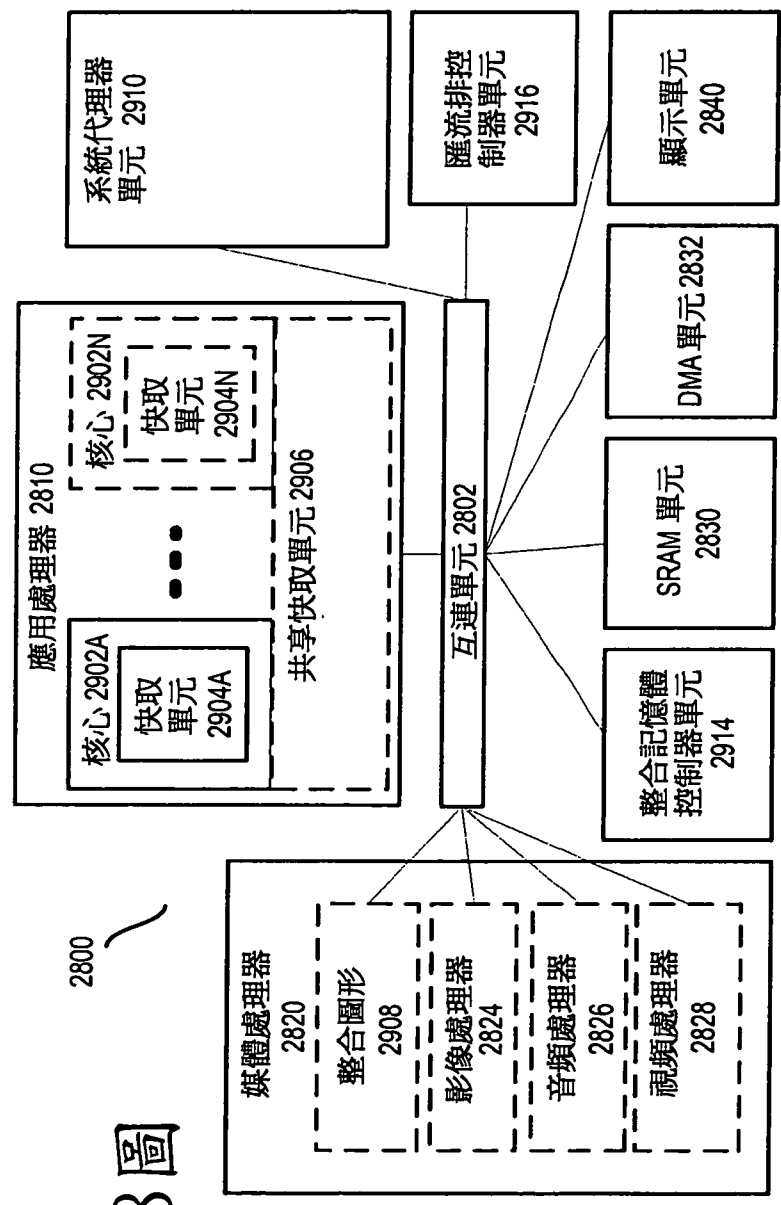
第26圖

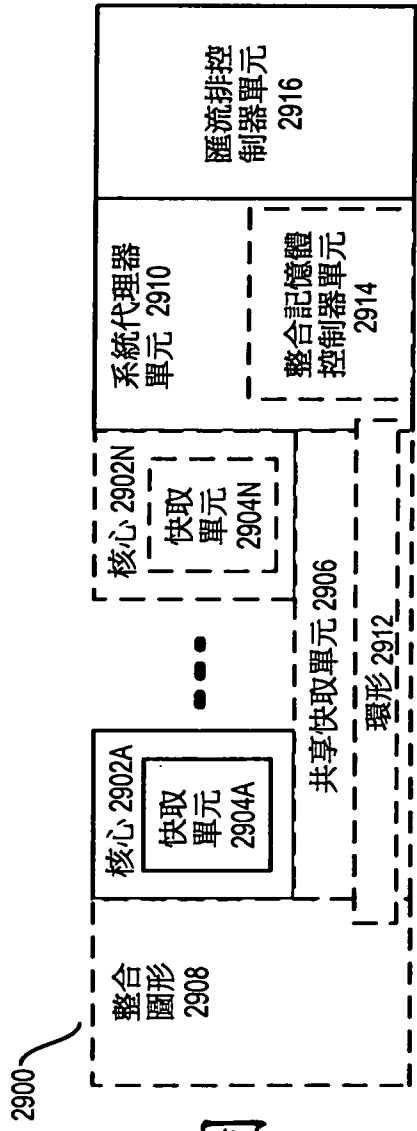


第27圖

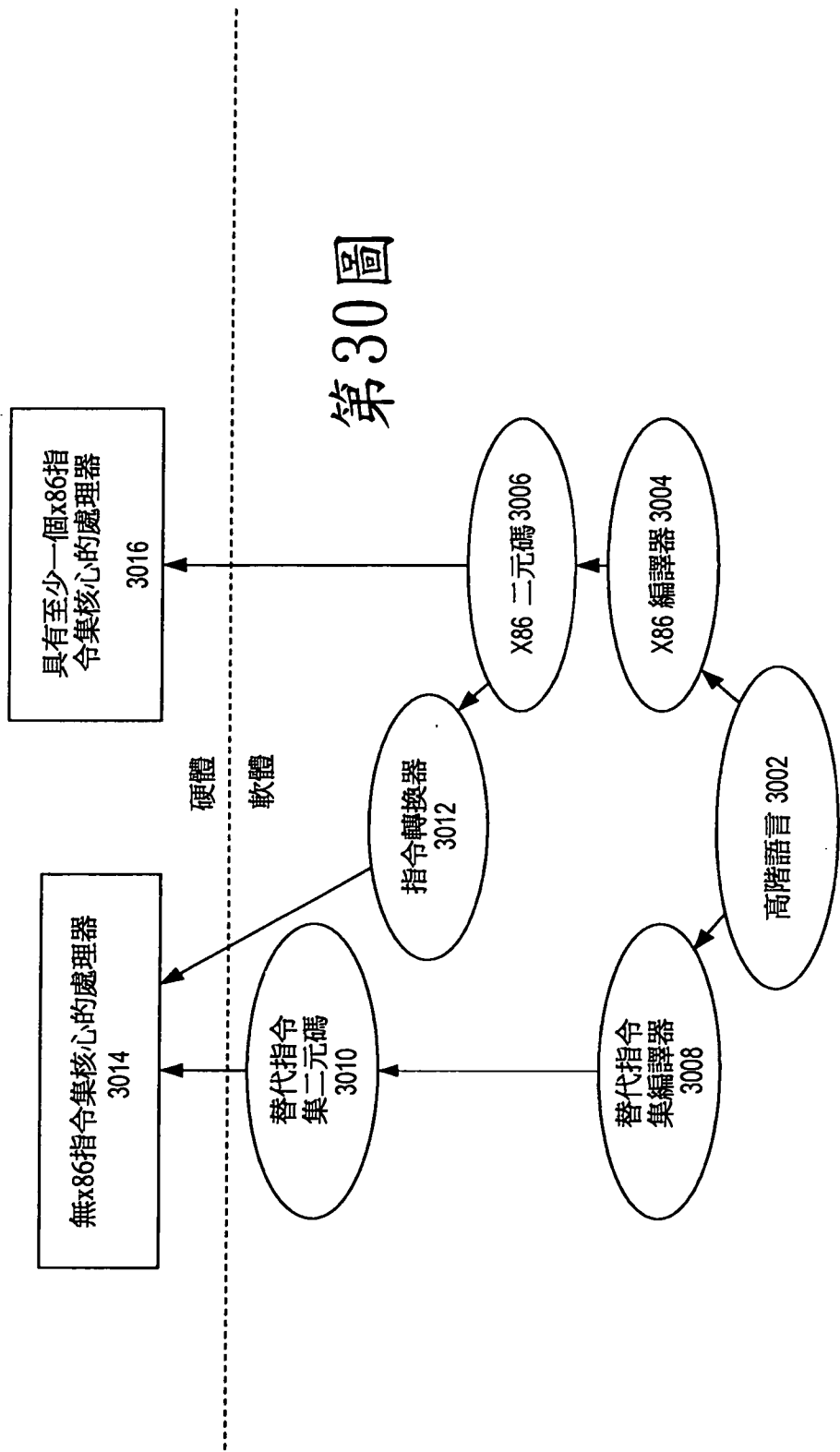
2800

第28圖





第29圖



第30圖