



(21)申請案號：105139273 (22)申請日：中華民國 105 (2016) 年 11 月 29 日
 (51)Int. Cl. : G06F40/12 (2020.01) G06F9/30 (2018.01)
 (30)優先權：2015/12/29 世界智慧財產權組織 PCT/RU2015/000950
 (71)申請人：美商英特爾股份有限公司(美國) INTEL CORPORATION (US)
 美國
 (72)發明人：羅本 義夫傑尼 ROUBAN, YEVGENY Y. (RU)；索科洛夫 丹尼爾 SOKOLOV,
 DANIIL Y. (RU)
 (74)代理人：林志剛
 (56)參考文獻：
 TW 201506782A
 "z/Architecture Principles of Operation", IBM, 2012 年 9 月, "<http://publibfp.dhe.ibm.com/epubs/pdf/dz9zr009.pdf>".
 審查人員：林彥廷
 申請專利範圍項數：24 項 圖式數：18 共 96 頁

(54)名稱

用於轉換編碼格式的硬體處理器、方法及非暫態機器可讀取媒體

(57)摘要

描述有關轉換編碼格式之方法及設備。於一實施例中，一種硬體處理器包括解碼電路，用以解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元；及執行電路，用以執行該指令以：將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式、以該第二編碼格式將該些元件儲存於該目的地向量運算元中、以該第二編碼格式將該些元件之總長度儲存於該狀態運算元中、及當來自該來源向量運算元之該些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中。

Methods and apparatuses relating to converting encoding formats are described. In one embodiment, a hardware processor includes a decode circuit to decode an instruction comprising a state operand, a source vector operand, a destination vector operand, and a control operand, and an execution circuit to execute the instruction to convert elements from the source vector operand in a first encoding format to a second encoding format, store the elements in the second encoding format in the destination vector operand, store a total length of the elements in the second encoding format in the state operand, and set a stream completion indication in the control operand when the elements from the source vector operand are a last elements in a data stream.

指定代表圖：

700



解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元

702



執行該指令，用以：將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式、以該第二編碼格式將這些元件儲存於該目的地向量運算元中、以該第二編碼格式將這些元件之總長度儲存於該狀態運算元中、及當來自該來源向量運算元之這些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中

704

圖 7

發明摘要

【發明名稱】(中文/英文)

用於轉換編碼格式的硬體處理器、方法及非暫態機器可讀取媒體

HARDWARE PROCESSORS, METHODS AND NON-TRANSITORY MACHINE READABLE MEDIUM FOR CONVERTING ENCODING FORMATS

【中文】

描述有關轉換編碼格式之方法及設備。於一實施例中，一種硬體處理器包括解碼電路，用以解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元；及執行電路，用以執行該指令以：將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式、以該第二編碼格式將該些元件儲存於該目的地向量運算元中、以該第二編碼格式將該些元件之總長度儲存於該狀態運算元中、及當來自該來源向量運算元之該些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中。

【英文】

Methods and apparatuses relating to converting encoding formats are described. In one embodiment, a hardware processor includes a decode circuit to decode an instruction comprising a state operand, a source vector operand, a destination vector operand, and a control operand, and an execution circuit to execute the instruction to convert elements from the source vector operand in a first encoding format to a second encoding format, store the elements in the second encoding format in the destination vector operand, store a total length of the elements in the second encoding format in the state operand, and set a stream completion indication in the control operand when the elements from the source vector operand are a last elements in a data stream.

【代表圖】

【本案指定代表圖】：第(7)圖。

【本代表圖之符號簡單說明】：無

【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)

用於轉換編碼格式的硬體處理器、方法及非暫態機器
可讀取媒體

HARDWARE PROCESSORS, METHODS AND NON-TRANSITORY
MACHINE READABLE MEDIUM FOR CONVERTING ENCODING
FORMATS

【技術領域】

本發明一般係有關電子電路，而更明確地，本發明之
實施例係有關用於轉換編碼格式的硬體處理器。

【先前技術】

處理器（或處理器之集合）係執行來自指令集（例
如，指令集架構（ISA））之指令。指令集為關於編程之
電腦架構的部分，且通常包括本機資料類型、指令、暫存
器架構、定址模式、記憶體架構、中斷和例外處置、及外
部輸入和輸出（I/O）。應注意：術語「指令」於此可指
稱巨集指令，例如，其為提供給處理器以供執行之指令；
或者微指令，例如，其得自處理器之解碼器解碼巨集指令
的結果之指令。

【圖式簡單說明】

本發明係藉由後附圖形之圖中的範例（而非限制）來
闡明，其中相似的參考符號係指示類似的元件，且其中：

圖 1 闡明依據本發明之實施例的硬體處理器。

圖 2 闡明硬體處理器，用以依據本發明之實施例來解碼並執行編碼格式轉換指令。

圖 3 闡明依據本發明之實施例的編碼格式轉換指令之格式。

圖 4 闡明依據本發明之實施例的編碼格式轉換操作之虛擬碼。

圖 5 闡明依據本發明之實施例的兩個編碼格式轉換指令之執行。

圖 6 闡明依據本發明之實施例的編碼格式轉換操作之流程圖。

圖 7 闡明依據本發明之實施例的流程圖。

圖 8A 為闡明一般性向量友善指令格式及其類別 A 指令模板的方塊圖，依據本發明之實施例。

圖 8B 為闡明一般性向量友善指令格式及其類別 B 指令模板的方塊圖，依據本發明之實施例。

圖 9A 為闡明圖 8A 及 8B 中之一般性向量友善指令格式的欄位之方塊圖，依據本發明之實施例。

圖 9B 為闡明其組成全運算碼欄位的圖 9A 中之特定向量友善指令格式的欄位之方塊圖，依據本發明之一實施例。

圖 9C 為闡明其組成暫存器指標欄位的圖 9A 中之特定向量友善指令格式的欄位之方塊圖，依據本發明之一實施例。

圖 9D 為闡明其組成擴增操作欄位 850 的圖 9A 中之

特定向量友善指令格式的欄位之方塊圖，依據本發明之一實施例。

圖 10 為一暫存器架構之方塊圖，依據本發明之一實施例。

圖 11A 為闡明範例依序管線及範例暫存器重新命名、失序發送/執行管線兩者之方塊圖，依據本發明之實施例。

圖 11B 為一方塊圖，其闡明將包括於依據本發明之實施例的處理器中之依序架構核心之範例實施例及範例暫存器重新命名、失序發送/執行架構核心兩者。

圖 12A 為單處理器核心之方塊圖，連同與晶粒上互連網路之其連接、以及第 2 階 (L2) 快取之其本地子集，依據本發明之實施例。

圖 12B 為圖 12A 中之處理器核心的部分之延伸視圖，依據本發明之實施例。

圖 13 為一種處理器之方塊圖，該處理器可具有多於一個核心、可具有集成記憶體控制器、且可具有集成圖形，依據本發明之實施例。

圖 14 為一系統之方塊圖，依據本發明之一實施例。

圖 15 為依據本發明之實施例的更特定範例系統之方塊圖。

圖 16 為依據本發明之實施例的第二更特定範例系統之方塊圖。

圖 17 顯示依據本發明之實施例的系統單晶片 (SoC)

的方塊圖。

圖 18 為一種對照軟體指令轉換器之使用的方塊圖，該轉換器係用以將來源指令集中之二元指令轉換至目標指令集中之二元指令，依據本發明之實施例。

【發明內容及實施方式】

於以下描述中，提出了數個特定細節。然而，應理解：本發明之實施例可被實行而無這些特定細節。於其他例子中，眾所周知的電路、結構及技術未被詳細地顯示以免模糊了對本說明書之瞭解。

說明書中對於「一個實施例」、「一實施例」、「一範例實施例」等等之參照係指示所述之實施例可包括特定的特徵、結構、或特性，但每一實施例可能不一定包括該特定的特徵、結構、或特性。此外，此等用詞不一定指稱相同的實施例。再者，當特定的特徵、結構、或特性配合實施例而描述時，係認為其落入熟悉此項技術人士之知識範圍內，以致能配合其他實施例（無論是否明確地描述）之此等特徵、結構、或特性。

一種（例如，硬體）處理器（例如，具有一或更多核心）可執行一或更多指令（例如，指令之執行緒）以操作於資料上（例如）來履行算術、邏輯、或其他功能。例如，軟體可請求一操作而硬體處理器（例如，其一核心或多數核心）可回應於該請求而履行該操作。資料可為第一格式而操作可將該第一格式轉換為第二格式。例如，資料

集（例如，資料之串流）可為複數字符，例如，包括字母、數字數位、標點符號、及空白。字符一起可形成語言之字元、而字元一起（例如，具有空白及標點符號）可形成句子，其可形成段落，等等。於計算時，字符（例如，文字）可被形成為字串。

於計算時，各字符可被編碼為（例如，一、二、三、或四位元組）碼值（例如，位元型態）。當作一範例，統一碼（Unicode）為一種計算工業標準，用於以許多世界書寫系統（例如，語言）所表達的字符（例如，文字）之一致性編碼、表示、及處置。編碼格式之一範例為統一碼變換格式 8（UTF-8），其係使用一至四位元組碼值（8 位元、16 位元、24 位元、或 32 位元）以表示各字符（例如，碼點）。編碼格式之另一範例為統一碼變換格式 16（UTF-16），其係使用一或二個十六位元碼值（16 位元或 32 位元）以表示各字符（例如，碼點）。編碼格式之另一範例為統一碼變換格式 32（UTF-32），其係使用一個三十二位元碼值（32 位元）以表示各字符（例如，碼點）。

應用程式（例如，那些運作於伺服器上者）可花費顯著的時間來轉換字符（例如，文字資料）從一編碼格式至另一編碼格式。於一實施例中，以通用組合指令從一編碼格式轉換至另一編碼格式可導致花費在碼值之各位元組上的複數（例如，五或更多）處理器（例如，中央處理單元（CPU））循環。如此可能導致用以處理字符（例如，碼

值)所需的顯著的時間及能量,例如,當剖析可延伸式標示語言(XML)資料時。例如,一編碼格式(例如,UTF-8)可被使用為網頁之編碼格式(例如,超文字標示語言(HTML)及XML)及其他編碼格式(例如,UTF-16)可被使用為用於處理網頁之編程語言(例如,Java、PHP、Net、Python等等)中之字符資料的編碼格式。因此介於兩不同編碼格式之間的轉換可經常被使用(例如)於伺服器及客戶上。

文中之某些實施例係有關用以從一編碼格式轉換至第二(不同)編碼格式之硬體處理器及方法。文中之某些實施例係有關用以(例如,解碼及)執行單指令來將字符碼值(例如,於資料串中)之多數位元組(例如,16位元組、32位元組、或64位元組)從一編碼格式轉換至第二(不同)編碼格式(例如,同時地)的硬體處理器及方法。文中之某些實施例係有關用以(例如,解碼及)執行單編碼格式轉換指令來將向量(例如,字符碼值之16位元組、32位元組、或64位元組)之多數元件(例如,8、16、24、或32位元碼值)從第一編碼格式轉換至第二編碼格式(例如,同時地)的硬體處理器及方法。

現在回到圖形,圖1闡明依據本發明之實施例的硬體處理器100。所述的硬體處理器100包括硬體解碼電路102,用以解碼指令,例如,一用以將碼值(例如,針對字符)從一編碼格式轉換至第二(不同)編碼格式之指令。碼值(例如,值之資料串)可被儲存於(例如,存取

自) 資料儲存裝置 110 (例如, 來源儲存 114) 中及/或於一或更多 (例如, 晶粒上) 暫存器 106 (例如, 來源暫存器 109) 中, 例如, 在轉換之前。所述的硬體執行電路 104 係用以執行已解碼指令, 例如, 用以將碼值 (例如, 針對字符) 從一編碼格式轉換至第二 (不同) 編碼格式。於一實施例中, 資料之串流為資料 (例如, 碼值) 之多數 (例如, 快取) 線。對於資料之指針 (例如, 用以儲存或載入) 可被利用。於某些實施例中, 碼值可被儲存以向量格式, 例如, 以致其各元件為一位元組。向量格式資料可被儲存於向量暫存器中及/或成為記憶體 (例如, 資料儲存裝置 110) 中之向量。於一實施例中, 向量中之各元件係儲存資料之一位元組、二位元組、三位元組、或四位元組。

硬體處理器 100 包括通至資料儲存裝置 110 之耦合 (例如, 連接)。資料儲存裝置 110 可為硬體處理器之本地的記憶體 (例如, 系統記憶體)。資料儲存裝置 110 (例如, 目的地儲存 112 和來源儲存 114 之一或兩者) 可為與硬體處理器分離之記憶體, 例如, 伺服器之記憶體。於一實施例中, 來源儲存 114 位於伺服器中及/或目的地儲存 112 是位於硬體處理器 100 客戶之本地。硬體處理器 100 可利用以下之一者、所有、或任何組合: 暫存器 106 及資料儲存裝置 110。

注意: 文中之圖形無法描繪所有的資料通訊連接。熟悉本技術人士將理解: 此係為了不混淆圖形中之某些細

節。注意：圖形中之雙頭箭號可不需要雙向通訊，例如，其可指示單向通訊（例如，至或自該組件或裝置）。通訊路徑之任一者或所有組合可被用於文中之某些實施例。

轉換操作可被履行於資料串（例如，輸入資料之串流）。資料之串流可被提供於較整個資料串更小的（例如，不同的）尺寸之區塊中，例如，其較小的區塊（例如，線）被提供直到該操作被履行於整個資料串上。狀態資料串可使其資料之各子集（例如，於其個別緩衝器中）依其完整且原始的順序而被呈送（例如，操作於其上）。例如，無狀態資料串之資料的各子集（例如，區塊）可被轉換為其本身的獨立工作。已轉換碼值可被儲存於資料儲存裝置 110（例如，目的地儲存 112）中及/或於一或更多（例如，晶粒上）暫存器 106（例如，目的地暫存器 108）中，例如，在轉換之前。

圖 2 闡明硬體處理器 200，用以依據本發明之實施例來解碼並執行編碼格式轉換指令 201。所描述的指令 201（例如，單指令）係由解碼電路 202 所解碼（例如，成為微指令及/或微操作），而已解碼指令係由執行單元 204 所執行。將由指令 201 所轉換之資料可被儲存於暫存器 206（例如，來源暫存器 209）及/或資料儲存裝置 210（例如，來源儲存 214 之區塊）中。從第一編碼格式轉換至第二編碼格式之資料可被載（例如，輸出）入暫存器 206（例如，目的地暫存器 208）及/或資料儲存裝置 210（例如，目的地儲存 212 之區塊）。

圖 3 闡明依據本發明之實施例的編碼格式轉換指令之格式。指令格式 300 可包括以下之一或更多者：運算碼欄位、狀態（例如，暫存器狀態）運算元欄位、目的地運算元欄位、來源運算元欄位、及控制運算元欄位。於一實施例中，第一運算碼係指示從第一編碼格式（例如，UTF-8）至第二編碼格式（例如，UTF-16）之轉換，而第二、不同的運算碼係指示從第二編碼格式（例如，UTF-16）至第二編碼格式（例如，UTF-8）之轉換。於一實施例中，單運算碼係指示介於第一編碼格式（例如，UTF-8）與第二編碼格式（例如，UTF-16）之間的轉換，而該指令之一欄位係指示來源編碼格式及目的地編碼格式。一欄位（例如，運算元）可為即刻值或非即刻值，例如，含有該值（或針對該值之指針）之暫存器或者含有該值（或針對該值之指針）之儲存元件的位址。於一實施例中，該指令（例如，運算碼或其他欄位）係指示該指令之處理器模式（例如，16 位元、32 位元、或 64 位元）。

狀態運算元可儲存以下之一或更多者（例如，或儲存含有以下資訊之暫存器的位址）：轉換之產生的結果長度及針對用以繼續資料串處理之後續（例如，屬於相同運算碼類型）指令的變遷狀態。變遷狀態可包括其由該指令所轉換的該碼值（例如，元件）為資料串中之最後碼值（例如，元件）的串流完成指示（例如，由處理器所設定），例如，以致其特定轉換工作被完成（例如，其可包括複數編碼格式轉換指令之執行）。於一實施例中，所被設定之

串流完成指示係指示其在此狀態停止轉換為有效的，例如，轉換操作尚未開始轉換其尚未被完成之字符。於一實施例中，串流完成指示為狀態運算元中之設定旗標（例如，位元），例如，假如資料串（例如，序列）之轉換為未完成則狀態運算元（例如，暫存器）位元（例如，[0:0]）設為 1，而假如資料串（例如，序列）之轉換為完成則設為 0。於一實施例中，狀態運算元，例如狀態暫存器（例如，狀態暫存器位元[7:1]），包括以目的地編碼格式之碼值的總長度，例如，寫入目的地運算元中之目的地編碼格式字元的數目（例如，輸出長度）。於一實施例中，狀態運算元，例如狀態暫存器（例如，狀態暫存器位元[47:8]），包括碼值（例如，元件），其並未結束於由該指令所轉換之碼值中，例如，未完成來源序列（例如，高達 5 位元組長）。狀態運算元可含有來源串尾部，其未被終止於最近處理的來源字元中。於一實施例中，狀態運算元（例如，欄位）係儲存上述項目之一或更多者於（例如，通用）暫存器中，例如，由控制運算元（例如，控制即刻運算元 `imm8`）之某些（例如，3:0）位元所指明的暫存器。

於一實施例中，來源運算元包括（例如，指明一包括以下之暫存器）碼值之複數位元組（例如，其中碼值可為複數位元組）。於一實施例中，目的地運算元包括（例如，指明一包括以下之暫存器）用以儲存已轉換碼值的存儲之複數位元組（例如，其中已轉換碼值可為複數位元

組)。於一實施例中，來源運算元及/或目的地運算元可為針對記憶體之指針，而指令之執行可依據有多少記憶體由轉換操作所消耗且輸出（個別地）來推進指針。

於一實施例中，控制運算元係控制編碼格式轉換指令如何操作。於一實施例中，控制運算元為即刻值（例如，8 位元即刻值）。控制運算元可包括（例如，針對以下之指示）以下之一或更多者：變遷狀態指明符（例如，控制位元 [3:0]），用以指明其儲存變遷狀態的可用（例如，通用 (64 位元)）暫存器（例如，`rax`, `rbx`, `rcx`, ..., `r15`）之一（例如，16）的位置，例如，用以履行後續指令（例如，相同的運算碼類型）；驗證指示符（例如，控制位元 [4:4]），例如，被設為 1 之位元值，用以造成（例如，完全）驗證被履行（例如，以致其無效編碼格式序列導致一旗標（例如，攜載旗標 (CF)）被設為 1）、及被設為 0 之位元值以造成無或最少驗證被履行（例如，和將由該指令所處理之碼值以及不被處理之毀損碼值，例如，其被視為遺失）；端讀取格式指示符，用以指示來源及/或目的地為小端讀取格式或者大端讀取格式，例如，假如該指令之結果將於小端讀取（LE）則控制位元（例如，控制位元 [5:5]）設為 0 而假如該指令之結果將於大端讀取（BE）則控制位元設為 1；及狀態指示符，例如，用以指示其來自狀態串之碼值的未完成序列將不被該指令所轉換（例如）直到碼值之序列完成。額外地或替代地，控制運算元可包括一位元或數個位元，用以造成處理器履行以下之一

或更多者：檢查來源和目的地限制、驗證（例如，檢查非法序列、設定最短編碼格式碼值、檢查代理碼點、設定編碼格式之一為修改的 UTF-8，等等）、容許或不容許未完成碼值或序列之讀取（例如，無狀態或狀態讀取，例如，使用狀態指示符），以不分裂輸出代理對（例如，以修改的 UTF-8）、錯誤處置（例如，復原及/或報告）、及產生字符偏移（例如，針對 ICU 全球化庫）。

指令格式 310、320、330、及 340 包括 8 位元即刻控制運算元。於其他實施例中，控制運算元可為任何大小。暫存器可為 32 位元、64 位元、128 位元、256 位元、512 位元等等暫存器。記憶體之區塊（例如，由編碼格式轉換指令所存取）可為任何大小，例如，64 位元、128 位元、256 位元、或 512 位元，等等。

所述的指令格式 310 具有 `u8u16` 運算碼，用以指示 UTF-8 編碼格式至 UTF-16 編碼格式轉換；暫存器或記憶體位址為目的地運算元，例如，用以儲存已轉換碼值；以及暫存器來源運算元。

所述的指令格式 320 具有 `u8u16` 運算碼，用以指示 UTF-8 編碼格式至 UTF-16 編碼格式轉換；暫存器或記憶體位址為來源運算元，例如，儲存將被轉換之碼值；以及暫存器目的地運算元。

所述的指令格式 330 具有 `u16u8` 運算碼，用以指示 UTF-16 編碼格式至 UTF-8 編碼格式轉換；暫存器或記憶體位址為目的地運算元，例如，用以儲存已轉換碼值；以

及暫存器來源運算元。

所述的指令格式 340 具有 `u16u8` 運算碼，用以指示 UTF-16 編碼格式至 UTF-8 編碼格式轉換；暫存器或記憶體位址為來源運算元，例如，儲存將被轉換之碼值；以及暫存器目的地運算元。

上述運算元及/或欄位之一或更多者可由處理器所設定，例如，在編碼格式轉換指令之執行以前。

於一實施例中，根據控制運算元（例如，控制旗標），則執行編碼格式轉換指令之硬體處理器係將碼值之第一編碼格式解碼為第二編碼格式（例如，從 UTF-8 與 UTF-16 之一至 UTF-8 與 UTF-16 之另一），從來源運算元至目的地運算元，並可接著改變後續編碼格式轉換指令之狀態。於一實施例中，該指令之執行造成一旗標（例如，攜載旗標 (CF)）被設定，假如發現驗證錯誤的話。於一實施例中，狀態暫存器（`rax`, `rbx`, `rcx`, ...）被指明於控制運算元（例如，控制旗標）中。與某些實施例中，編碼格式轉換指令之執行可造成以下之一或更多者：下個（例如，記憶體位址）目的地被計算（例如，自目的地運算元及狀態運算元）；檢查可用的來源空間是否夠大以從該處讀取某數目（例如，`N`）的位元組（例如，`N=16`、`32`、... 等等）；檢查可用目的地空間是否夠大以用最大結果長度（例如，`32`、`64`... 等等）來寫入那些 `N` 輸入位元組的已解碼輸出；及報告錯誤，例如，有錯誤細節或無錯誤細節（例如，以用來接著判定錯誤細節之軟體）。例如，一編

碼格式轉換指令為「`u8u16 ymm256 dest, m128 src, imm8` 控制（`r64` 狀態）」，而僅來源（`src`）之低半字元（128 位元）可被讀取，且於此轉換下，輸出可導致全字元（256 位元）之最大值而因此 256 位元暫存器可被用於目的地。於一實施例中，當所有來源位元組為 ASCII 碼值時，則有 UTF-8 之 16 位元組（16 位元組 = 128 位元）且被轉換至 16 個 UTF-16（128 位元 x 2 = 256 位元）。

圖 4 闡明依據本發明之實施例的編碼格式轉換操作之虛擬碼 400。虛擬碼 400 之線 05 包括編碼格式轉換指令。於一實施例中，虛擬碼 400 可被利用（例如，為內迴路）以轉換碼值之資料串。

圖 5 闡明依據本發明之實施例的兩個編碼格式轉換指令 510 及 520 之執行。區塊 500 包括記憶體之兩線，其包括複數元件於來源向量字元 00 及字元 16 中。那些熟悉此技藝人士可辨識：這些是針對該用詞之俄語字母，但不包括括號，（`Есть только две ...`）其係翻譯成英文為（`There are only two ...`）。注意：針對控制運算元，於此 010010 之位元值表示其 0 之第 5 位元位置（從右邊的位元位置 0 開始）值指示小端讀取、0 之第 4 位元位置值指示應履行驗證操作、及位元[3:0]中之 2 的值指示其狀態運算元被儲存於暫存器 `rcx` 中（其中 `r2` 映射至 `rcx`）。

圖 6 闡明依據本發明之實施例的編碼格式轉換操作之流程圖 600。所述的實施例具有開始 601，且接著其檢查是否有更多資料（例如，更多碼值或碼值的部分）留存於

資料串中以便被處理 602（例如，轉換）。假如沒有更多資料，則該流程可藉由變遷至無資料 603 而結束（例如，結束）。假如有更多資料待處理，則該資料可被載入至（例如，來源）暫存器 604，且接著編碼格式轉換指令可被執行 606 於來源暫存器中之該資料上（例如，碼值被載（或裝）入向量暫存器之元件）。目的地暫存器接著可被更新 608 以來自來源暫存器之已轉換資料。於一實施例中，狀態暫存器可被檢查其任何錯誤 610。假如無錯誤，則該流程可返回以檢查是否有更多資料應被處理 602（例如，相同資料串之更多資料）。假如有錯誤，則可執行一程式，例如，最後疊代之軟體實施方式以找出確實的錯誤 612。

圖 7 闡明依據本發明之實施例的流程圖 700。所述的流程 700 包括解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元 702；及執行該指令，用以：將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式、以該第二編碼格式將該些元件儲存於該目的地向量運算元中、以該第二編碼格式將該些元件之總長度儲存於該狀態運算元中、及當來自該來源向量運算元之該些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中 704。

於一實施例中，一種硬體處理器包括解碼電路，用以解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元；及執行電路，用以執行該

指令以：將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式、以該第二編碼格式將該些元件儲存於該目的地向量運算元中、以該第二編碼格式將該些元件之總長度儲存於該狀態運算元中、及當來自該來源向量運算元之該些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中。第一編碼格式可為 UTF-8 字符編碼與 UTF-16 字符編碼之一者，而第二編碼格式可為 UTF-8 字符編碼與 UTF-16 字符編碼之另一者。執行電路可執行該指令以：檢測來自該來源向量運算元之多元件序列的一或更多元件，其中該多元件序列不會結束於該來源向量運算元中，且不會將來自該來源向量運算元之多元件序列的一或更多元件從第一編碼格式轉換至第二編碼格式。執行電路可執行該指令以將來自該來源向量運算元之多元件序列的一或更多元件儲存於該狀態運算元中。執行電路可執行該指令以讀取該控制運算元之端讀取格式指示符以判定具有第二編碼格式之該些元件將何時（例如，如何）以大端讀取格式與小端讀取格式之一被儲存於該目的地向量運算元中。執行電路可執行該指令以讀取該控制運算元之驗證指示符來判定何時造成驗證操作被履行於該第一編碼格式及該第二編碼格式之該些元件上。執行電路可執行該指令以讀取該控制運算元之狀態指示符來判定何時將來自該來源向量運算元之該些元件的未完成序列從該第一編碼格式轉換至該第二編碼格式。該控制運算元可包括用以儲存該狀態運算元之暫存器的位址。

於另一實施例中，一種方法包括解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元；及執行該指令，用以：將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式、以該第二編碼格式將該些元件儲存於該目的地向量運算元中、以該第二編碼格式將該些元件之總長度儲存於該狀態運算元中、及當來自該來源向量運算元之該些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中。該第一編碼格式可為 UTF-8 字符編碼與 UTF-16 字符編碼之一者，而該第二編碼格式可為 UTF-8 字符編碼與 UTF-16 字符編碼之另一者。該方法可包括：檢測來自該來源向量運算元之多元件序列的一或更多元件，其中該多元件序列不會結束於該來源向量運算元中，且不會將來自該來源向量運算元之該多元件序列的該些一或更多元件從該第一編碼格式轉換至該第二編碼格式。該方法可包括將來自該來源向量運算元之該多元件序列的該些一或更多元件儲存於該狀態運算元中。該方法可包括讀取該控制運算元之端讀取格式指示符以判定具有該第二編碼格式之該些元件將何時以大端讀取格式與小端讀取格式之一被儲存於該目的地向量運算元中。該方法可包括讀取該控制運算元之驗證指示符以判定何時造成驗證操作被履行於該第一編碼格式及該第二編碼格式之該些元件上。該方法可包括讀取該控制運算元之狀態指示符以判定何時將來自該來源向量運算元之該些元件的未完成序列從該第一編碼格式轉換至該第二編

碼格式。該控制運算元可包括用以儲存該狀態運算元之暫存器的位址。

於又另一實施例中，一種儲存碼之非暫態機器可讀取媒體，當由機器所執行時該碼係造成該機器履行一方法，包括：解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元；及執行該指令，用以：將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式、以該第二編碼格式將該些元件儲存於該目的地向量運算元中、以該第二編碼格式將該些元件之總長度儲存於該狀態運算元中、及當來自該來源向量運算元之該些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中。該第一編碼格式可為 UTF-8 字符編碼與 UTF-16 字符編碼之一者，而該第二編碼格式可為 UTF-8 字符編碼與 UTF-16 字符編碼之另一者。該方法可包括：檢測來自該來源向量運算元之多元件序列的一或更多元件，其中該多元件序列不會結束於該來源向量運算元中，且不會將來自該來源向量運算元之該多元件序列的該些一或更多元件從該第一編碼格式轉換至該第二編碼格式。該方法可包括將來自該來源向量運算元之該多元件序列的該些一或更多元件儲存於該狀態運算元中。該方法可包括讀取該控制運算元之端讀取格式指示符以判定具有該第二編碼格式之該些元件將何時以大端讀取格式與小端讀取格式之一被儲存於該目的地向量運算元中。該方法可包括讀取該控制運算元之驗證指示符以判定何時造成驗證操

作被履行於該第一編碼格式及該第二編碼格式之該些元件上。該方法可包括讀取該控制運算元之狀態指示符以判定何時將來自該來源向量運算元之該些元件的未完成序列從該第一編碼格式轉換至該第二編碼格式。該控制運算元可包括用以儲存該狀態運算元之暫存器的位址。

於另一實施例中，一種硬體處理器包括解碼機構，用以解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元；及執行機構，用以執行該指令來：將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式、以該第二編碼格式將該些元件儲存於該目的地向量運算元中、以該第二編碼格式將該些元件之總長度儲存於該狀態運算元中、及當來自該來源向量運算元之該些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中。

於又另一實施例中，一種設備包含一儲存碼之資料儲存裝置，當由硬體處理器所執行時該碼係造成該硬體處理器履行文中所揭露之任何方法。一種設備可為如詳細說明中所描述者。一種方法可為如詳細說明中所描述者。

指令集可包括一或更多指令格式。既定指令格式可界定各種欄位（例如，位元之數目、位元之位置）以指明（除了別的以外）待履行操作（例如，運算碼）以及將於其上履行操作之運算元及/或其他資料欄位（例如，遮罩）。一些指令格式係透過指令模板（或子格式）之定義而被進一步分解。例如，既定指令格式之指令模板可被定

義以具有指令格式之欄位的不同子集（所包括的欄位通常係以相同順序，但至少某些具有不同的位元位置，因為包括了較少的欄位）及/或被定義以具有不同地解讀之既定欄位。因此，ISA 之各指令係使用既定指令格式（以及，假如被定義的話，以該指令格式之指令模板的既定一者）而被表達，並包括用以指明操作及運算元之欄位。例如，範例 ADD 指令具有特定運算碼及一指令格式，其包括用以指明該運算碼之運算碼欄位及用以選擇運算元（來源 1/目的地及來源 2）之運算元欄位；而於一指令串中之此 ADD 指令的發生將具有特定內容於其選擇特定運算元之運算元欄位中。被稱為先進向量延伸（AVX）（AVX1 及 AVX2）並使用向量延伸（VEX）編碼技術之一組 SIMD 延伸已被釋出及/或出版（例如，參見 Intel® 64 及 IA-32 架構軟體開發商手冊，2015 年九月；及參見 Intel® 架構指令集延伸編程參考，2015 年八月）。

範例指令格式

文中所述之指令的實施例可被實施以不同的格式。此外，範例系統、架構、及管線被詳述於下。指令之實施例可被執行於此等系統、架構、及管線上，但不限定於那些細節。

一般性向量友善指令格式

向量友善指令格式是一種適於向量指令之指令格式

（例如，有向量操作特定的某些欄位）。雖然實施例係描述其中向量和純量操作兩者均透過向量友善指令格式而被支援，但替代實施例僅使用具有向量友善指令格式之向量操作。

圖 8A-8B 為闡明一般性向量友善指令格式及其指令模板的方塊圖，依據本發明之實施例。圖 8A 為闡明一般性向量友善指令格式及其類別 A 指令模板的方塊圖，依據本發明之實施例；而圖 8B 為闡明一般性向量友善指令格式及其類別 B 指令模板的方塊圖，依據本發明之實施例。明確地，針對一般性向量友善指令格式 800 係定義類別 A 及類別 B 指令模板，其兩者均包括無記憶體存取 805 指令模板及記憶體存取 820 指令模板。於向量友善指令格式之背景下術語「一般性」指的是不與任何特定指令集連結的指令格式。

雖然本發明之實施例將描述其中向量友善指令格式支援以下：具有 32 位元（4 位元組）或 64 位元（8 位元組）資料元件寬度（或大小）之 64 位元組向量運算元長度（或大小）（而因此，64 位元組向量係由 16 雙字元大小的元件、或替代地 8 四字元大小的元件所組成）；具有 16 位元（2 位元組）或 8 位元（1 位元組）資料元件寬度（或大小）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）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）資料元件寬度（或大小）之 16 位元組向量運算元長度（或大小）；但是替代實施例可支援具有更大、更小、或不同資料元件寬度（例如，128 位元（16 位元組）資料元件寬度）之更大、更小及/或不同的向量運算元大小（例如，256 位元組向量運算元）。

圖 8A 中之類別 A 指令模板包括：1) 於無記憶體存取 805 指令模板內，顯示有無記憶體存取、全捨入控制類型操作 810 指令模板及無記憶體存取、資料變換類型操作 815 指令模板；以及 2) 於記憶體存取 820 指令模板內，顯示有記憶體存取、暫時 825 指令模板及記憶體存取、非暫時 830 指令模板。圖 8B 中之類別 B 指令模板包括：1) 於無記憶體存取 805 指令模板內，顯示有無記憶體存取、寫入遮蔽控制、部分捨入控制類型操作 812 指令模板及無記憶體存取、寫入遮蔽控制、v 大小類型操作 817 指令模板；以及 2) 於記憶體存取 820 指令模板內，顯示有記憶體存取、寫入遮蔽控制 827 指令模板。

一般性向量友善指令格式 800 包括以下欄位，依圖 8A-8B 中所示之順序列出如下。

格式欄位 840 - 此欄位中之一特定值（指令格式識別符值）係獨特地識別向量友善指令格式、以及因此在指令串中之向量友善指令格式的指令之發生。如此一來，此欄位是選擇性的，因為針對一僅具有一般性向量友善指令格

式之指令集而言此欄位是不需要的。

· 基礎操作欄位 842 – 其內容係分辨不同的基礎操作。

暫存器指標欄位 844 – 其內容（直接地或透過位址產生）係指明來源及目的地運算元之位置，假設其係於暫存器中或記憶體中。這些包括足夠數目的位元以從 $P \times Q$ （例如， 32×512 ， 16×128 ， 32×1024 ， 64×1024 ）暫存器檔選擇 N 暫存器。雖然於一實施例中 N 可高達三個來源及一個目的地暫存器，但是替代實施例可支援更多或更少的來源及目的地暫存器（例如，可支援高達兩個來源，其中這些來源之一亦作用為目的地；可支援高達三個來源，其中這些來源之一亦作用為目的地；可支援高達兩個來源及一個目的地）。

修飾符欄位 846 – 其內容係從不指明記憶體存取之那些指令分辨出其指明記憶體存取之一般性向量指令格式的指令之發生，亦即，介於無記憶體存取 805 指令模板與記憶體存取 820 指令模板之間。記憶體存取操作係讀取及/或寫入至記憶體階層（於使用暫存器中之值以指明來源及/或目的地位址之某些情況下），而非記憶體存取操作則不會（例如，來源及目的地為暫存器）。雖然於一實施例中此欄位亦於三個不同方式之間選擇以履行記憶體位址計算，但是替代實施例可支援更多、更少、或不同方式以履行記憶體位址計算。

擴增操作欄位 850 – 其內容係分辨多種不同操作之哪一個將被履行，除了基礎操作之外。此欄位是背景特定

的。於本發明之一實施例中，此欄位被劃分為類別欄位 868、 α 欄位 852、及 β 欄位 854。擴增操作欄位 850 容許操作之共同群組將被履行以單指令而非 2、3、或 4 指令。

比例欄位 860 - 其內容容許指標欄位之內容的定標，以供記憶體位址產生（例如，以供其使用 $2^{\text{比例}} * \text{指標} + \text{基礎}$ 之位址產生）。

置換欄位 862A - 其內容被使用為記憶體位址產生之部分（例如，以供其使用 $2^{\text{比例}} * \text{指標} + \text{基礎} + \text{置換}$ 之位址產生）。

置換因數欄位 862B（注意：直接在置換因數欄位 862B 上方之置換欄位 862A 的並列指示一者或另一者被使用）- 其內容被使用為位址產生之部分；其指明將被記憶體存取之大小（N）所定標的置換因數 - 其中 N 為記憶體存取中之位元組數目（例如，以供其使用 $2^{\text{比例}} * \text{指標} + \text{基礎} + \text{定標置換}$ 之位址產生）。冗餘低階位元被忽略而因此，置換因數欄位之內容被乘以記憶體運算元總大小（N）來產生最終置換以供使用於計算有效位址。N 之值係在運作時間由處理器硬體所判定，根據全運算碼欄位 874（稍後描述於文中）及資料調處欄位 854C。置換欄位 862A 及置換因數欄位 862B 是選擇性的，因為其未被使用於無記憶體存取 805 指令模板及/或不同的實施例可實施該兩欄位之僅一者或者無任何。

資料元件寬度欄位 864 - 其內容係分辨數個資料元件

寬度之哪一個將被使用（於針對所有指令之某些實施例中；於針對僅某些指令之其他實施例中）。此欄位是選擇性的，在於其假如僅有一資料元件寬度被支援及/或資料元件寬度係使用運算碼之某形態而被支援則此欄位是不需要的。

寫入遮蔽欄位 870 - 其內容係根據每資料元件位置以控制其目的地向量運算元中之資料元件位置是否反映基礎操作及擴增操作之結果。類別 A 指令模板支援合併-寫入遮蔽，而類別 B 指令模板支援合併-及歸零-寫入遮蔽兩者。當合併時，向量遮蔽容許目的地中之任何組的元件被保護自任何操作之執行期間（由基礎操作及擴增操作所指明）的更新；於另一實施例中，保留其中相應遮蔽位元具有 0 之目的地的各元件之舊值。反之，當歸零時，向量遮蔽容許目的地中之任何組的元件被歸零於任何操作之執行期間（由基礎操作及擴增操作所指明）；於一實施例中，當相應遮蔽位元具有 0 值時則目的地之一元件被設為 0。此功能之子集是其控制被履行之操作的向量長度（亦即，被修飾之元件的範圍，從第一者至最後者）的能力；然而，其被修飾之元件不需要是連續的。因此，寫入遮蔽欄位 870 容許部分向量操作，包括載入、儲存、運算、邏輯，等等。雖然本發明之實施例係描述其中寫入遮蔽欄位 870 之內容選擇其含有待使用之寫入遮蔽的數個寫入遮蔽暫存器之一（而因此寫入遮蔽欄位 870 之內容間接地識別其遮蔽將被履行），但是替代實施例取代地或者額外地容

許寫入遮蔽欄位 870 之內容直接地指明其遮蔽將被履行。

即刻欄位 872 - 其內容容許即刻之指明。此欄位是選擇性的，由於此欄位存在於其不支援即刻之一般性向量友善格式的實施方式中且此欄位不存在於其不使用即刻之指令中。

類別欄位 868 - 其內容分辨於不同類別的指令之間。參考圖 8A-B，此欄位之內容選擇於類別 A 與類別 B 指令之間。於圖 8A-B 中，圓化角落的方形被用以指示一特定值存在於一欄位中（例如，針對類別欄位 868 之類別 A 868A 及類別 B 868B，個別地於圖 8A-B 中）。

類別 A 之指令模板

於類別 A 之非記憶體存取 805 指令模板的情況下， α 欄位 852 被解讀為 RS 欄位 852A，其內容係分辨不同擴增操作類型之哪一個將被履行（例如，捨入 852A.1 及資料變換 852A.2 被個別地指明給無記憶體存取、捨入類型操作 810 及無記憶體存取、資料變換類型操作 815 指令模板），而 β 欄位 854 係分辨該些指明類型的操作之哪個將被履行。於無記憶體存取 805 指令模板中，比例欄位 860、置換欄位 862A、及置換比例欄位 862B 不存在。

無記憶體存取指令模板 - 全捨入控制類型操作

於無記憶體存取全捨入控制類型操作 810 指令模板中， β 欄位 854 被解讀為捨入控制欄位 854A，其內容係提

供靜態捨入。雖然於本發明之所述實施例中，捨入控制欄位 854A 包括抑制所有浮點例外 (SAE) 欄位 856 及捨入操作控制欄位 858，但替代實施例可支援可將這兩個觀念均編碼入相同欄位或僅具有這些觀念/欄位之一者或另一者 (例如，可僅具有捨入操作控制欄位 858)。

SAE 欄位 856 - 其內容係分辨是否除能例外事件報告；當 SAE 欄位 856 之內容指示抑制被致能時，則一既定指令不報告任何種類的浮點例外旗標且不引發任何浮點例外處置器。

捨入操作控制欄位 858 - 其內容係分辨一群捨入操作之哪一個將被履行 (例如向上捨入、向下捨入、朝零捨入及捨入至最接近)。因此，捨入操作控制欄位 858 容許以每指令為基之捨入模式的改變。於本發明之一實施例中，其中處理器包括一用以指明捨入模式之控制暫存器，捨入操作控制欄位 850 之內容係撤銷該暫存器值。

無記憶體存取指令模板 - 資料變換類型操作

於無記憶體存取資料變換類型操作 815 指令模板中， β 欄位 854 被解讀為資料變換欄位 854B，其內容係分辨數個資料變換之哪一個將被履行 (例如，無資料變換、拌合、廣播)。

於類別 A 之記憶體存取 820 指令模板的情況下， α 欄位 852 被解讀為逐出暗示欄位 852B，其內容係分辨逐出暗示之哪一個將被使用 (於圖 8A 中，暫時 852B.1 及非暫

時 852B.2 被個別地指明給記憶體存取、暫時 825 指令模板及記憶體存取、非暫時 830 指令模板)，而 β 欄位 854 被解讀為資料調處欄位 854C，其內容係分辨數個資料調處操作（亦已知為基元）之哪一個將被履行（例如，無調處；廣播；來源之向上轉換；及目的地之向下轉換）。記憶體存取 820 指令模板包括比例欄位 860、及選擇性地置換欄位 862A 或置換比例欄位 862B。

向量記憶體指令係履行向量載入自及向量儲存至記憶體，具有轉換支援。至於一般向量指令，向量記憶體指令係以資料元件式方式轉移資料自/至記憶體，以其被實際地轉移之元件由其被選為寫入遮蔽的向量遮蔽之內容所主宰。

記憶體存取指令模板 - 暫時

暫時資料為可能會夠早地被再使用以受惠自快取的資料。然而，此為一暗示，且不同的處理器可以不同的方式來實施，包括完全地忽略該暗示。

記憶體存取指令模板 - 非暫時

非暫時資料為不太可能會夠早地被再使用以受惠自第一階快取中之快取且應被給予逐出之既定優先權的資料。然而，此為一暗示，且不同的處理器可以不同的方式來實施，包括完全地忽略該暗示。

類別 B 之指令模板

於類別 B 之指令模板的情況下， α 欄位 852 被解讀為寫入遮蔽控制 (Z) 欄位 852C，其內容係分辨由寫入遮蔽欄位 870 所控制的寫入遮蔽是否應為合併或歸零。

於類別 B 之非記憶體存取 805 指令模板的情況下， β 欄位 854 之部分被解讀為 RL 欄位 857A，其內容係分辨不同擴增操作類型之哪一個將被履行（例如，捨入 857A.1 及向量長度 (VSIZE) 857A.2 被個別地指明給無記憶體存取、寫入遮蔽控制、部分捨入控制類型操作 812 指令模板及無記憶體存取、寫入遮蔽控制、VSIZE 類型操作 817 指令模板），而剩餘的 β 欄位 854 係分辨該些指明類型的操作之哪個將被履行。於無記憶體存取 805 指令模板中，比例欄位 860、置換欄位 862A、及置換比例欄位 862B 不存在。

於無記憶體存取中，寫入遮蔽控制、部分捨入控制類型操作 810 指令模板、剩餘的 β 欄位 854 被解讀為捨入操作欄位 859A 且例外事件報告被除能（既定指令則不報告任何種類的浮點例外旗標且不引發任何浮點例外處理器）。

捨入操作控制欄位 859A - 正如捨入操作控制欄位 858，其內容係分辨一群捨入操作之哪一個將被履行（例如向上捨入、向下捨入、朝零捨入及捨入至最接近）。因此，捨入操作控制欄位 859A 容許以每指令為基之捨入模式的改變。於本發明之一實施例中，其中處理器包括一用

以指明捨入模式之控制暫存器，捨入操作控制欄位 850 之內容係撤銷該暫存器值。

於無記憶體存取、寫入遮蔽控制、VSIZE 類型操作 817 指令模板中，剩餘的 β 欄位 854 被解讀為向量長度欄位 859B，其內容係分辨數個資料向量長度之哪一個將被履行（例如，128、256、或 512 位元組）。

於類別 B 之記憶體存取 820 指令模板的情況下， β 欄位 854 之部分被解讀為廣播欄位 857B，其內容係分辨廣播類型資料調處操作是否將被履行，而剩餘的 β 欄位 854 被解讀為向量長度欄位 859B。記憶體存取 820 指令模板包括比例欄位 860、及選擇性地置換欄位 862A 或置換比例欄位 862B。

關於一般性向量友善指令格式 800，全運算碼欄位 874 被顯示為包括格式欄位 840、基礎操作欄位 842、及資料元件寬度欄位 864。雖然一實施例被顯示為其中全運算碼欄位 874 包括所有這些欄位，全運算碼欄位 874 包括少於所有這些欄位在不支援其所有的實施例中。全運算碼欄位 874 提供操作碼（運算碼）。

擴增操作欄位 850、資料元件寬度欄位 864、及寫入遮蔽欄位 870 容許這些特徵以每指令為基被指明以一般性向量友善指令格式。

寫入遮蔽欄位與資料元件寬度欄位之組合產生類型化的指令，在於其容許遮蔽根據不同資料元件寬度而被施加。

類別 A 及類別 B 中所發現之各種指令模板在不同情況下是有利的。於本發明之某些實施例中，不同處理器或一處理器中之不同核心可支援僅類別 A、僅類別 B、或兩類別。例如，用於通用計算之高性能通用失序核心可支援僅類別 B；主要用於圖形及/或科學（通量）計算之核心可支援僅類別 A；及用於兩者之核心可支援兩者（當然，一種具有來自兩類別之模板和指令的某混合但非來自兩類別之所有模板和指令的核心是落入本發明之範圍內）。同時，單一處理器可包括多核心，其所有均支援相同的類別或者其中不同的核心支援不同的類別。例如，於一具有分離的圖形和通用核心之處理器中，主要用於圖形及/或科學計算的圖形核心之一可支援僅類別 A；而通用核心之一或更多者可為高性能通用核心，其具有用於支援僅類別 B 之通用計算的失序執行和暫存器重新命名。不具有分離的圖形核心之另一處理器可包括支援類別 A 和類別 B 兩者之一或更多通用依序或失序核心。當然，來自一類別之特徵亦可被實施於另一類別中，在不同實施例中。以高階語言寫入之程式將被置入（例如，僅以時間編譯或靜態地編譯）多種不同的可執行形式，包括：1）僅具有由用於執行之處理器所支援的類別之指令的形式；或 2）具有其使用所有類別之指令的不同組合所寫入之替代常式並具有控制流碼的形式，該控制流碼係根據由目前正執行該碼之處理器所支援的指令以選擇用來執行之常式。

範例特定向量友善指令格式

圖 9 為闡明範例特定向量友善指令格式的方塊圖，依據本發明之實施例。圖 9 顯示特定向量友善指令格式 900，其之特定在於其指明欄位之位置、大小、解讀及順序，以及那些欄位之部分的值。特定向量友善指令格式 900 可被用以延伸 x86 指令集，而因此某些欄位係類似於或相同於現存 x86 指令集及其延伸（例如，AVX）中所使用的那些。此格式保持與下列各者一致：具有延伸之現存 x86 指令集的前綴編碼欄位、真實運算碼位元組欄位、MOD R/M 欄位、SIB 欄位、置換欄位、及即刻欄位。闡明來自圖 8 之欄位投映入來自圖 9 之欄位。

應理解：雖然本發明之實施例係參考為說明性目的之一般性向量友善指令格式 800 的背景下之特定向量友善指令格式 900 而描述，但除非其中有聲明否則本發明不限於特定向量友善指令格式 900。例如，一般性向量友善指令格式 800 係考量各個欄位之多種可能大小，而特定向量友善指令格式 900 被顯示為具有特定大小之欄位。舉特定例而言，雖然資料元件寬度欄位 864 被闡明為特定向量友善指令格式 900 之一位元欄位，但本發明未如此限制（亦即，一般性向量友善指令格式 800 係考量資料元件寬度欄位 864 之其他大小）。

一般性向量友善指令格式 800 包括以下欄位，依圖 9A 中所示之順序列出如下。

EVEX 前綴（位元組 0-3）902 被編碼以四位元組形

式。

格式欄位 840 (EVEX 位元組 0, 位元 [7:0]) - 第一位元組 (EVEX 位元組 0) 為格式欄位 840 且其含有 0x62 (用於分辨本發明之一實施例中的向量友善指令格式之獨特值)。

第二-第四位元組 (EVEX 位元組 1-3) 包括數個提供特定能力之位元欄位。

REX 欄位 905 (EVEX 位元組 1, 位元 [7-5]) - 係包括: 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 857BEX 位元組 1, 位元 [5] - B)。EVEX.R、EVEX.X、及 EVEX.B 位元欄位提供如相應 VEX 位元欄位之相同功能, 且係使用 1 互補形式而被編碼, 亦即, ZMM0 被編碼為 1111B, ZMM15 被編碼為 0000B。指令之其他欄位編碼該些暫存器指標之較低三位元如本技術中所已知者 (rrr、xxx、及 bbb), 以致 Rrrr、Xxxx、及 Bbbb 可藉由加入 EVEX.R、EVEX.X、及 EVEX.B 而被形成。

REX'欄位 810 - 此為 REX'欄位 810 之第一部分且為 EVEX.R'位元欄位 (EVEX 位元組 1, 位元 [4] - R'), 其被用以編碼延伸的 32 暫存器集之上 16 個或下 16 個。於本發明之一實施例中, 此位元 (連同如以下所指示之其他者) 被儲存以位元反轉格式來分辨 (於眾所周知的 x86 32-位元模式) 自 BOUND 指令, 其真實運算碼位元組為 62, 但於 MOD R/M 欄位 (描述於下) 中不接受 MOD 欄

位中之 11 的值；本發明之替代實施例不以反轉格式儲存此及如下其他指示的位元。1 之值被用以編碼下 16 暫存器。換言之，R'Rrrr 係藉由結合 EVEX.R'、EVEX.R、及來自其他欄位之其他 RRR 而被形成。

運算碼映圖欄位 915 (EVEX 位元組 1，位元[3:0] - mmmm) - 其內容係編碼一暗示的領先運算碼位元組 (0F、0F 38、或 0F 3)。

資料元件寬度欄位 864 (EVEX 位元組 2，位元[7] - W) 係由記號 EVEX.W 所表示。EVEX.W 被用以界定資料類型 (32 位元資料元件或 64 位元資料元件) 之粒度 (大小)。

EVEX.vvvv 920 (EVEX 位元組 2，位元[6:3] - vvvv) - EVEX.vvvv 之角色可包括以下：1) EVEX.vvvv 編碼其以反轉 (1 之補數) 形式所指明的第一來源暫存器運算元且針對具有 2 或更多來源運算元為有效的；2) EVEX.vvvv 針對某些向量位移編碼其以 1 之補數形式所指明的目的地暫存器運算元；或 3) EVEX.vvvv 未編碼任何運算元，該欄位被保留且應含有 1111b。因此，EVEX.vvvv 欄位 920 係編碼其以反轉 (1 之補數) 形式所儲存的第一來源暫存器指明符之 4 個低階位元。根據該指令，一額外的不同 EVEX 位元欄位被用以延伸指明符大小至 32 暫存器。

EVEX.U 868 類別欄位 (EVEX 位元組 2，位元[2]-U) - 假如 EVEX.U = 0，則其指示類別 A 或 EVEX.U0；

假如 $EVEX.U = 1$ ，則其指示類別 B 或 $EVEX.U1$ 。

前綴編碼欄位 925 (EVEX 位元組 2，位元 [1:0] - pp) 提供額外位元給基礎操作欄位。除了提供針對 EVEX 前綴格式之舊有 SSE 指令的支援，此亦具有壓縮 SIMD 前綴之優點 (不需要一位元組來表達 SIMD 前綴，EVEX 前綴僅需要 2 位元)。於一實施例中，為了支援其使用以舊有格式及以 EVEX 前綴格式兩者之 SIMD 前綴 (66H、F2H、F3H) 的舊有 SSE 指令，這些舊有 SIMD 前綴被編碼為 SIMD 前綴編碼欄位；且在運作時間被延伸入舊有 SIMD 前綴，在其被提供至解碼器的 PLA 以前 (以致 PLA 可執行這些舊有指令之舊有和 EVEX 格式兩者而無須修改)。雖然較少的指令可將 EVEX 前綴編碼欄位之內容直接地使用為運算碼延伸，但某些實施例係以類似方式延伸以符合一致性而容許不同的意義由這些舊有 SIMD 前綴來指明。替代實施例可重新設計 PLA 以支援 2 位元 SIMD 前綴編碼，而因此不需要延伸。

α 欄位 852 (EVEX 位元組 3，位元 [7] - EH；亦已知為 $EVEX.EH$ 、 $EVEX.rs$ 、 $EVEX.RL$ 、 $EVEX$.寫入遮蔽控制、及 $EVEX.N$ ；亦闡明以 α) - 如先前所描述，此欄位是背景特定的。

β 欄位 854 (EVEX 位元組 3，位元 [6:4]-SSS，亦已知為 $EVEX.s_{2-0}$ 、 $EVEX.r_{2-0}$ 、 $EVEX.rr1$ 、 $EVEX.LL0$ 、 $EVEX.LLB$ ；亦闡明以 $\beta\beta\beta$) - 如先前所描述，此欄位是背景特定的。

REX'欄位 810 - 此為 REX'欄位之剩餘部分且為 EVEX.V'位元欄位 (EVEX 位元組 3, 位元[3] - V')，其被用以編碼延伸的 32 暫存器集之上 16 個或下 16 個。此位元被儲存以位元反轉格式。1 之值被用以編碼下 16 暫存器。換言之，V'VVVV 係藉由結合 EVEX.V'、EVEX.vvvv 所形成。

寫入遮蔽欄位 870 (EVEX 位元組 3, 位元[2:0]-kkk) - 其內容係指明在如先前所述之寫入遮蔽暫存器中的暫存器之指數。於本發明之一實施例中，特定值 EVEX.kkk=000 具有一特殊行為，其係暗示無寫入遮蔽被用於特別指令 (此可被實施以多種方式，包括使用其固線至所有各者之寫入遮蔽或者其旁路遮蔽硬體之硬體)。

真實運算碼欄位 930 (位元組 4) 亦已知為運算碼位元組。運算碼之部分被指明於此欄位。

MOD R/M 欄位 940 (位元組 5) 包括 MOD 欄位 942、Reg 欄位 944、及 R/M 欄位 946。如先前所述 MOD 欄位 942 之內容係分辨於記憶體存取與非記憶體存取操作之間。Reg 欄位 944 之角色可被概述為兩情況：編碼目的地暫存器運算元或來源暫存器運算元、或者被視為運算碼延伸而不被用以編碼任何指令運算元。R/M 欄位 946 之角色可包括以下：編碼其參考記憶體位址之指令運算元；或者編碼目的地暫存器運算元或來源暫存器運算元。

比例、指標、基礎 (SIB) 位元組 (位元組 6) - 如先前所述，比例欄位 850 之內容被用於記憶體位址產生。

SIB.xxx 954 及 SIB.bbb 956 - 這些欄位之內容先前已被參考針對暫存器指標 Xxxx 及 Bbbb。

置換欄位 862A (位元組 7-10) - 當 MOD 欄位 942 含有 10 時，位元組 7-10 為置換欄位 862A，且其工作如舊有 32 位元置換 (disp32) 之相同方式且工作以位元組粒度。

置換因數欄位 862B (位元組 7) - 當 MOD 欄位 942 含有 01 時，位元組 7 為置換因數欄位 862B。此欄位之位置係相同於舊有 x86 指令集 8 位元置換 (disp8) 之位置，其工作以位元組粒度。因為 disp8 是符號延伸的，所以其可僅定址於 -128 與 127 位元組偏移之間；關於 64 位元組快取線，disp8 係使用其可被設為僅四個真實可用值 -128、-64、0 及 64 之 8 位元；因為較大範圍經常是需要的，所以 disp32 被使用；然而，disp32 需要 4 位元組。相對於 disp8 及 disp32，置換因數欄位 862B 為 disp8 之再解讀；當使用置換因數欄位 862B 時，實際置換係由置換因數欄位之內容乘以記憶體運算元存取之大小 (N) 所判定。置換之類型被稱為 $\text{disp8} * N$ 。此係減少平均指令長度 (用於置換之單一位元組但具有更大的範圍)。此壓縮置換是基於假設其有效置換為記憶體存取之粒度的數倍，而因此，位址偏移之冗餘低階位元無須被編碼。換言之，置換因數欄位 862B 取代舊有 x86 指令集 8 位元置換。因此，置換因數欄位 862B 被編碼以如 x86 指令集 8 位元置換之相同方式 (以致 ModRM/SIB 編碼規則並無改變)，

唯一例外是其 disp8 被超載至 $\text{disp8} * N$ 。換言之，編碼規則或編碼長度沒有改變，但僅於藉由硬體之置換值的解讀（其需由記憶體運算元之大小來定標置換以獲得位元組式的位址偏移）。即刻欄位 872 係操作如先前所述。

全運算碼欄位

圖 9B 為闡明其組成全運算碼欄位 874 之特定向量友善指令格式 900 的欄位之方塊圖，依據本發明之一實施例。明確地，全運算碼欄位 874 包括格式欄位 840、基礎操作欄位 842、及資料元件寬度（W）欄位 864。基礎操作欄位 842 包括前綴編碼欄位 925、運算碼映圖欄位 915、及真實運算碼欄位 930。

暫存器指標欄位

圖 9C 為闡明其組成暫存器指標欄位 844 之特定向量友善指令格式 900 的欄位之方塊圖，依據本發明之一實施例。明確地，暫存器指標欄位 844 包括 REX 欄位 905、REX' 欄位 910、MODR/M.reg 欄位 944、MODR/M.r/m 欄位 946、VVVV 欄位 920、xxx 欄位 954、及 bbb 欄位 956。

擴增操作欄位

圖 9D 為闡明其組成擴增操作欄位 850 之特定向量友善指令格式 900 的欄位之方塊圖，依據本發明之一實施

例。當類別 (U) 欄位 868 含有 0 時，則其表示 EVEX.U0 (類別 A 868A)；當其含有 1 時，則其表示 EVEX.U1 (類別 B 868B)。當 $U=0$ 且 MOD 欄位 942 含有 11 (表示無記憶體存取操作) 時，則 α 欄位 852 (EVEX 位元組 3，位元[7] - EH) 被解讀為 rs 欄位 852A。當 rs 欄位 852A 含有 1 (捨入 852A.1) 時，則 β 欄位 854 (EVEX 位元組 3，位元[6:4]- SSS) 被解讀為捨入控制欄位 854A。捨入控制欄位 854A 包括一位元 SAE 欄位 856 及二位元捨入操作欄位 858。當 rs 欄位 852A 含有 0 (資料變換 852A.2) 時，則 β 欄位 854 (EVEX 位元組 3，位元[6:4]- SSS) 被解讀為三位元資料變換欄位 854B。當 $U=0$ 且 MOD 欄位 942 含有 00、01、或 10 (表示記憶體存取操作) 時，則 α 欄位 852 (EVEX 位元組 3，位元[7] - EH) 被解讀為逐出暗示 (EH) 欄位 852B 且 β 欄位 854 (EVEX 位元組 3，位元[6:4] - SSS) 被解讀為三位元資料調處欄位 854C。

當 $U=1$ 時，則 α 欄位 852 (EVEX 位元組 3，位元[7] - EH) 被解讀為寫入遮蔽控制 (Z) 欄位 852C。當 $U=1$ 且 MOD 欄位 942 含有 11 (表示無記憶體存取操作) 時，則 β 欄位 854 之部分 (EVEX 位元組 3，位元[4]- S_0) 被解讀為 RL 欄位 857A；當其含有 1 (捨入 857A.1) 時，則 β 欄位 854 之剩餘部分 (EVEX 位元組 3，位元[6-5]- S_{2-1}) 被解讀為捨入操作欄位 859A；而當 RL 欄位 857A 含有 0 (VSIZE857.A2) 時，則 β 欄位 854 之剩餘部

分 (EVEX 位元組 3, 位元 [6-5]- S_{2-1}) 被解讀為向量長度欄位 859B (EVEX 位元組 3, 位元 [6-5]- L_{1-0})。當 $U=1$ 且 MOD 欄位 942 含有 00、01、或 10 (表示記憶體存取操作) 時, 則 β 欄位 854 (EVEX 位元組 3, 位元 [6:4]-SSS) 被解讀為向量長度欄位 859B (EVEX 位元組 3, 位元 [6-5]- L_{1-0}) 及廣播欄位 857B (EVEX 位元組 3, 位元 [4]-B)。

範例暫存器架構

圖 10 為一暫存器架構 1000 之方塊圖, 依據本發明之一實施例。於所示之實施例中, 有 32 個向量暫存器 1010, 其為 512 位元寬; 這些暫存器被稱為 zmm_0 至 zmm_{31} 。較低的 16 個 zmm 暫存器之較低階 256 位元被重疊於暫存器 ymm_0-16 上。較低的 16 個 zmm 暫存器之較低階 128 位元 (ymm 暫存器之較低階 128 位元) 被重疊於暫存器 xmm_0-15 上。特定向量友善指令格式 900 係操作於這些重疊的暫存器檔上, 如以下表中所闡明。

可調整向量長度	類別	操作	暫存器
不包括向量長度欄位 859B 之指令模板	A(圖 8A; $U=0$)	810, 815, 825, 830	zmm 暫存器(向量長度為 64 位元組)
	B(圖 8B; $U=1$)	812	zmm 暫存器(向量長度為 64 位元組)
包括向量長度欄位 859B 之指令模板	B(圖 8B; $U=1$)	817, 827	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)根據向量長度欄位 859B

換言之，向量長度欄位 859B 於最大長度與一或更多其他較短長度之間選擇，其中每一此較短長度為前一長度之長度的一半；而無向量長度欄位 859B 之指令模板係操作於最大長度上。此外，於一實施例中，特定向量友善指令格式 900 之類別 B 指令模板係操作於緊縮或純量單/雙精確度浮點資料及緊縮或純量整數資料上。純量操作為履行於 `zmm/ymm/xmm` 暫存器中之最低階資料元件上的操作；較高階資料元件位置係根據實施例而被保留如其在該指令前之相同者或者被歸零。

寫入遮蔽暫存器 1015 - 於所示之實施例中，有 8 個寫入遮蔽暫存器 (`k0` 至 `k7`)，大小各為 64 位元。於替代實施例中，寫入遮蔽暫存器 1015 之大小為 16 位元。如先前所述，於本發明之一實施例中，向量遮蔽暫存器 `k0` 無法被使用為寫入遮蔽；當其通常將指示 `k0` 之編碼被用於寫入遮蔽時，其係選擇 `0xFFFF` 之固線寫入遮蔽，有效地除能該指令之寫入遮蔽。

通用暫存器 1025 - 於所示之實施例中，有十六個 64 位元通用暫存器，其係連同現存的 `x86` 定址模式來用以定址記憶體運算元。這些暫存器被參照以 `RAX`、`RBX`、`RCX`、`RDX`、`RBP`、`RSI`、`RDI`、`RSP`、及 `R8` 至 `R15`。

純量浮點堆疊暫存器檔 (`x87` 堆疊) 1045，`MMX` 緊縮整數平坦暫存器檔 1050 係別名於其上 - 於所示之實施例中，`x87` 堆疊為用以使用 `x87` 指令集延伸而在 32/64/80 位元浮點資料上履行純量浮點操作之八元件堆疊；而

MMX 暫存器被用以履行操作在 64 位元緊縮整數資料上、及用以保持運算元以供介於 MMX 與 XMM 暫存器間所履行的某些操作。

本發明之替代實施例可使用較寬或較窄的暫存器。此外，本發明之替代實施例可使用更多、更少、或不同的暫存器檔及暫存器。

範例核心架構，處理器，及電腦架構

處理器核心可被實施以不同方式、用於不同目的、以及於不同處理器中。例如，此類核心之實施方式可包括：

- 1) 用於通用計算之通用依序核心；
- 2) 用於通用計算之高性能通用失序核心；
- 3) 主要用於圖形及/或科學（通量）計算之特殊用途核心。

不同處理器之實施方式可包括：

- 1) CPU，其包括用於通用計算之一或更多通用依序核心及/或用於通用計算之一或更多通用失序核心；及
- 2) 核心處理器，其包括主要用於圖形及/或科學（通量）之一或更多特殊用途核心。

此等不同處理器導致不同的電腦系統架構，其可包括：

- 1) 在來自該 CPU 之分離晶片上的共處理器；
- 2) 在與 CPU 相同的封裝中之分離晶粒上的共處理器；
- 3) 在與 CPU 相同的晶粒上的共處理器（於該情況下，此一處理器有時被稱為特殊用途邏輯，諸如集成圖形及/或科學（通量）邏輯、或稱為特殊用途核心）；及
- 4) 在一可包括於相同晶粒上之所述 CPU（有時稱為應用程式核心或應用程式處理器）、上述共處理器、及額外功能的

晶片上之系統。範例核心架構被描述於下，接續著範例處理器及電腦架構之描述。

範例核心架構

依序或失序核心方塊圖

圖 11A 為闡明範例依序管線及範例暫存器重新命名、失序發送/執行管線兩者之方塊圖，依據本發明之實施例。圖 11B 為一方塊圖，其闡明將包括於依據本發明之實施例的處理器中之依序架構核心之範例實施例及範例暫存器重新命名、失序發送/執行架構核心兩者。圖 11A-B 中之實線方盒係闡明依序管線及依序核心，而虛線方盒之選擇性加入係闡明暫存器重新命名、失序發送/執行管線及核心。假設其依序形態為失序形態之子集，將描述失序形態。

於圖 11A 中，處理器管線 1100 包括提取級 1102、長度解碼級 1104、解碼級 1106、配置級 1108、重新命名級 1110、排程（亦已知為分派或發送）級 1112、暫存器讀取/記憶體讀取級 1114、執行級 1116、寫入回/記憶體寫入級 1118、例外處置級 1122、及確定級 1124。

圖 11B 顯示處理器核心 1190，其包括一耦合至執行引擎單元 1150 之前端單元 1130，且兩者均耦合至記憶體單元 1170。核心 1190 可為減少指令集計算（RISC）核心、複雜指令集計算（CISC）核心、極長指令字元（VLIW）核心、或者併合或替代核心類型。當作又另一

種選擇，核心 1190 可為特殊用途核心，諸如（例如）網路或通訊核心、壓縮引擎、共處理器核心、通用計算圖形處理單元（GPGPU）核心、圖形核心，等等。

前端單元 1130 包括一支預測單元 1132，其係耦合至指令快取單元 1134，其係耦合至指令變換後備緩衝（TLB）1136，其係耦合至指令提取單元 1138，其係耦合至解碼單元 1140。解碼單元 1140（或解碼器或解碼器單元）可解碼指令（例如，巨集指令）；並可將以下產生為輸出：一或更多微操作、微碼進入點、微指令、其他指令、或其他控制信號，其被解碼自（或者反應）、或被衍生自原始指令。解碼單元 1140 可使用各種不同的機制來實施。適當機制之範例包括（但不限定於）查找表、硬體實施方式、可編程邏輯陣列（PLA）、微碼唯讀記憶體（ROM），等等。於一實施例中，核心 1190 包括微碼 ROM 或者儲存用於某些巨指令之微碼的其他媒體（例如，於解碼單元 1140 中或者於前端單元 1130 內）。解碼單元 1140 被耦合至執行引擎單元 1150 中之重新命名/配置器單元 1152。

執行引擎單元 1150 包括重新命名/配置器單元 1152，其係耦合至撤回單元 1154 及一組一或更多排程器單元 1156。排程器單元 1156 代表任何數目的不同排程器，包括保留站、中央指令窗，等等。排程器單元 1156 被耦合至實體暫存器檔單元 1158。實體暫存器檔單元 1158 之各者代表一或更多實體暫存器檔，其不同者係儲存一或更多

不同的資料類型，諸如純量整數、純量浮點、緊縮整數、緊縮浮點、向量整數、向量浮點、狀態（例如，其為下一待執行指令之位址的指令指標），等等。於一實施例中，實體暫存器檔單元 1158 包含向量暫存器單元、寫入遮蔽暫存器單元、及純量暫存器單元。這些暫存器單元可提供架構向量暫存器、向量遮蔽暫存器、及通用暫存器。實體暫存器檔單元 1158 係由撤回單元 1154 所重疊以闡明其中暫存器重新命名及失序執行可被實施之各種方式（例如，使用記錄器緩衝器和撤回暫存器檔；使用未來檔、歷史緩衝器、和撤回暫存器檔；使用暫存器映圖和暫存器池，等等）。撤回單元 1154 及實體暫存器檔單元 1158 被耦合至執行叢集 1160。執行叢集 1160 包括一組一或更多執行單元 1162 及一組一或更多記憶體存取單元 1164。執行單元 1162 可履行各種操作（例如，偏移、相加、相減、相乘）以及於各種類型的資料上（例如，純量浮點、緊縮整數、緊縮浮點、向量整數、向量浮點）。雖然某些實施例可包括數個專屬於特定功能或功能集之執行單元，但其他實施例可包括僅一個執行單元或者全部履行所有功能之多數執行單元。排程器單元 1156、實體暫存器檔單元 1158、及執行叢集 1160 被顯示為可能複數的，因為某些實施例係針對某些類型的資料/操作產生分離的管線（例如，純量整數管線、純量浮點/緊縮整數/緊縮浮點/向量整數/向量浮點管線、及/或記憶體存取管線，其各具有本身的排程器單元、實體暫存器檔單元、及/或執行叢集 - 且

於分離記憶體存取管線之情況下，某些實施例被實施於其中僅有此管線之執行叢集具有記憶體存取單元 1164)。亦應理解：當使用分離管線時，這些管線之一或更多者可為失序發送/執行而其他者為依序。

該組記憶體存取單元 1164 被耦合至記憶體單元 1170，其包括資料 TLB 單元 1172，其耦合至資料快取單元 1174，其耦合至第二階 (L2) 快取單元 1176。於一範例實施例中，記憶體存取單元 1164 可包括載入單元、儲存位址單元、及儲存資料單元，其各者係耦合至記憶體單元 1170 中之資料 TLB 單元 1172。指令快取單元 1134 被進一步耦合至記憶體單元 1170 中之第二階 (L2) 快取單元 1176。L2 快取單元 1176 被耦合至一或更多其他階的快取且最終至主記憶體。

舉例而言，範例暫存器重新命名、失序發送/執行核心架構可實施管線 1100 如下：1) 指令提取 1138 履行提取和長度解碼級 1102 和 1104；2) 解碼單元 1140 履行解碼級 1106；3) 重新命名/配置器單元 1152 履行配置級 1108 和重新命名級 1110；4) 排程器單元 1156 履行排程級 1112；5) 實體暫存器檔單元 1158 和記憶體單元 1170 履行暫存器讀取/記憶體讀取級 1114；執行叢集 1160 履行執行級 1116；6) 記憶體單元 1170 和實體暫存器檔單元 1158 履行寫入回/記憶體寫入級 1118；7) 各個單元可參與例外處置級 1122；及 8) 撤回單元 1154 和實體暫存器檔單元 1158 履行確定級 1124。

核心 1190 可支援一或更多指令集（例如，x86 指令集，具有其已被加入以較新版本之某些延伸）；MIPS Technologies of Sunnyvale, CA 之 MIPS 指令集；ARM Holdings of Sunnyvale, CA 之 ARM 指令集（具有諸如 NEON 之選擇性額外延伸），包括文中所述之指令。於一實施例中，核心 1190 包括支援緊縮資料指令集延伸（例如，AVX1、AVX2）之邏輯，藉此容許由許多多媒體應用程式所使用的操作使用緊縮資料來履行。

應理解：核心可支援多線程（執行二或更多平行組的操作或線緒），並可以多種方式執行，包括時間切割多線程、同時多線程（其中單一實體核心提供邏輯核心給其實體核心正同時地多線程之每一線緒）、或者其組合（例如，時間切割提取和解碼以及之後的同時多線程，諸如 Intel® Hyperthreading 科技）。

雖然暫存器重新命名被描述於失序執行之背景，但應理解其暫存器重新命名可被使用於依序架構。雖然處理器之所述的實施例亦包括分離的指令和資料快取單元 1134/1174 以及共用 L2 快取單元 1176，但替代實施例可具有針對指令和資料兩者之單一內部快取，諸如（例如）第一階（L1）內部快取、或多階內部快取。於某些實施例中，該系統可包括內部快取與外部快取之組合，該外部快取是位於核心及/或處理器之外部。替代地，所有快取可於核心及/或處理器之外部。

特定範例依序核心架構

圖 12A-B 闡明更特定的範例依序核心架構之方塊圖，該核心將為晶片中之數個邏輯區塊之一（包括相同類型及/或不同類型之其他核心）。邏輯區塊係透過高頻寬互連網路（例如，環狀網路）來通訊，利用某些固定功能邏輯、記憶體 I/O 介面、及其他必要 I/O 邏輯，根據其應用而定。

圖 12A 為單處理器核心之方塊圖，連同與晶粒上互連網路 1202 之其連接、以及第二階（L2）快取 1204 之其本地子集，依據本發明之實施例。於一實施例中，指令解碼單元 1200 支援具有緊縮資料指令集延伸之 x86 指令集。L1 快取 1206 容許針對快取記憶體之低潛時存取入純量及向量單元。雖然於一實施例中（為了簡化設計），純量單元 1208 及向量單元 1210 使用分離的暫存器組（個別地，純量暫存器 1212 及向量暫存器 1214），且於其間轉移的資料被寫入至記憶體並接著從第一階（L1）快取 1206 被讀取回；但本發明之替代實施例可使用不同的方式（例如，使用單一暫存器組或者包括一通訊路徑，其容許資料被轉移於兩暫存器檔之間而不被寫入及讀取回）。

L2 快取 1204 之本地子集為其被劃分為分離本地子集（每一處理器核心有一個）之總體 L2 快取的部分。各處理器核心具有一直接存取路徑通至 L2 快取 1204 之其本身的本地子集。由處理器核心所讀取的資料被儲存於其 L2 快取子集 1204 中且可被快速地存取，平行於存取其本身

本地 L2 快取子集之其他處理器核心。由處理器核心所寫入之資料被儲存於其本身的 L2 快取子集 1204 中且被清除自其他子集，假如需要的話。環狀網路確保共用資料之一致性。環狀網路為雙向的，以容許諸如處理器核心、L2 快取及其他邏輯區塊等代理於晶片內部彼此通訊。各環狀資料路徑於每方向為 1012 位元寬。

圖 12B 為圖 12A 中之處理器核心的部分之延伸視圖，依據本發明之實施例。圖 12B 包括 L1 快取 1204 之 L1 資料快取 1206A 部分、以及有關向量單元 1210 和向量暫存器 1214 之更多細節。明確地，向量單元 1210 為 16 寬的向量處理單元 (VPU) (參見 16 寬的 ALU 1228)，其係執行整數、單精確度浮點、及雙精確度浮點指令之一或更多者。VPU 支援以拌合單元 1220 拌合暫存器輸入、以數字轉換單元 1222A-B 之數字轉換、及於記憶體輸入上以複製單元 1224 之複製。寫入遮蔽暫存器 1226 容許斷定結果向量寫入。

圖 13 為一種處理器 1300 之方塊圖，該處理器可具有多於一個核心、可具有集成記憶體控制器、且可具有集成圖形，依據本發明之實施例。圖 13 中之實線方塊闡明處理器 1300，其具有單核心 1302A、系統代理 1310、一組一或更多匯流排控制器單元 1316；而虛線方塊之選擇性加入闡明一替代處理器 1300，其具有多核心 1302A-N、系統代理單元 1310 中之一組一或更多集成記憶體控制器單元 1314、及特殊用途邏輯 1308。

因此，處理器 1300 之不同實施方式可包括：1) CPU，具有其為集成圖形及/或科學（通量）邏輯（其可包括一或更多核心）之特殊用途邏輯 1308、及其為一或更多通用核心（例如，通用依序核心、通用失序核心、兩者之組合）之核心 1302A-N；2) 共處理器，具有其為主要用於圖形及/或科學（通量）之大量特殊用途核心的核心 1302A-N；及 3) 共處理器，具有其為大量通用依序核心的核心 1302A-N。因此，處理器 1300 可為通用處理器、共處理器或特殊用途處理器，諸如（例如）網路或通訊處理器、壓縮引擎、圖形處理器、GPGPU（通用圖形處理單元）、高通量多數集成核心（MIC）共處理器（包括 30 或更多核心）、嵌入式處理器，等等。該處理器可被實施於一或更多晶片上。處理器 1300 可為一或更多基底之部分及/或可被實施於其上，使用數個製程技術之任一者，諸如（例如）BiCMOS、CMOS、或 NMOS。

記憶體階層包括該些核心內之一或更多階快取、一組或者一或更多共用快取單元 1306、及耦合至該組集成記憶體控制器單元 1314 之額外記憶體（未顯示）。該組共用快取單元 1306 可包括一或更多中階快取，諸如第二階（L2）、第三階（L3）、第四階（L4）、或其他階快取、最後階快取（LLC）、及/或其組合。雖然於一實施例中環狀為基的互連單元 1312 將以下裝置互連：集成圖形邏輯 1308、該組共用快取單元 1306、及系統代理單元 1310/集成記憶體控制器單元 1314，但替代實施例可使用

任何數目之眾所周知的技術以互連此等單元。於一實施例中，一致性被維持於一或更多快取單元 1306 與核心 1302A-N 之間。

於某些實施例中，一或更多核心 1302A-N 能夠進行多線程。系統代理 1310 包括協調並操作核心 1302A-N 之那些組件。系統代理單元 1310 可包括（例如）電力控制單元（PCU）及顯示單元。PCU 可為或者包括用以調節核心 1302A-N 及集成圖形邏輯 1308 之電力狀態所需的邏輯和組件。顯示單元係用以驅動一或更多外部連接的顯示。

核心 1302A-N 可針對架構指令集為同質的或異質的；亦即，二或更多核心 1302A-N 可執行相同的指令集，而其他者可執行該指令集或不同指令集之僅一子集。

範例電腦架構

圖 14-17 為範例電腦架構之方塊圖。用於膝上型電腦、桌上型電腦、手持式 PC、個人數位助理、工程工作站、伺服器、網路裝置、網路集線器、開關、嵌入式處理器、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、行動電話、可攜式媒體播放器、手持式裝置、及各種其他電子裝置之技術中已知的其他系統設計和組態亦為適當的。通常，能夠結合處理器及/或其他執行邏輯（如文中所揭露者）之多種系統或電子裝置為一般性適當的。

現在參考圖 14，其顯示依據本發明之一實施例的系

統 1400 之方塊圖。系統 1400 可包括一或更多處理器 1410、1415，其被耦合至控制器集線器 1420。於一實施例中，控制器集線器 1420 包括圖形記憶體控制器集線器（GMCH）1490 及輸入/輸出集線器（IOH）1450（其可於分離的晶片上）；GMCH 1490 包括記憶體及圖形控制器（耦合至記憶體 1440 及共處理器 1445）；IOH 1450 為通至 GMCH 1490 之耦合輸入/輸出（I/O）裝置 1460。另一方面，記憶體與圖形控制器之一或兩者被集成於處理器內（如文中所述者），記憶體 1440 及共處理器 1445 被直接地耦合至處理器 1410、及具有 IOH 1450 之單一晶片中的控制器集線器 1420。記憶體 1440 可包括編碼格式轉換模組 1440A，例如，用以儲存碼，當被執行時該碼係造成處理器履行本發明之任何方法。

額外處理器 1415 之選擇性本質於圖 14 中被標示以斷線。各處理器 1410、1415 可包括文中所述的處理核心之一或更多者並可為處理器 1300 之某版本。

記憶體 1440 可為（例如）動態隨機存取記憶體（DRAM）、相位改變記憶體（PCM）、或兩者之組合。針對至少一實施例，控制器集線器 1420 經由諸如前側匯流排（FSB）等多點分支匯流排、諸如 QuickPath 互連（QPI）等點對點介面、或類似連接 1495 而與處理器 1410、1415 通訊。

於一實施例中，共處理器 1445 為特殊用途處理器，諸如（例如）高通量 MIC 處理器、網路或通訊處理器、

壓縮引擎、圖形處理器、GPGPU、嵌入式處理器，等等。於一實施例中，控制器集線器 1420 可包括集成圖形加速器。

於實體資源 1410、1415 間可有多樣差異，針對價值矩陣之譜，包括架構、微架構、熱、功率耗損特性，等等。

於一實施例中，處理器 1410 執行其控制一般類型之資料處理操作的指令。指令內所嵌入者可為共處理器指令。處理器 1410 辨識這些共處理器指令為其應由裝附之共處理器 1445 所執行的類型。因此，處理器 1410 將共處理器匯流排或其他互連上之這些共處理器指令（或代表共處理器指令之控制信號）發送至共處理器 1445。共處理器 1445 接受並執行該些接收的共處理器指令。

現在參考圖 15，其顯示依據本發明之實施例的第一更特定範例系統 1500 之方塊圖。如圖 15 中所示，多處理器系統 1500 為點對點互連系統，並包括經由點對點互連 1550 而耦合之第一處理器 1570 及第二處理器 1580。處理器 1570 及 1580 之每一者可為處理器 1300 之某版本。於本發明之一實施例中，處理器 1570 及 1580 個別為處理器 1410 及 1415，而共處理器 1538 為共處理器 1445。於另一實施例中，處理器 1570 及 1580 個別為處理器 1410 及共處理器 1445。

處理器 1570 及 1580 被顯示為個別地包括集成記憶體控制器（IMC）單元 1572 及 1582。處理器 1570 亦包括其

匯流排控制器單元點對點 (P-P) 介面 1576 及 1578 之部分；類似地，第二處理器 1580 包括 P-P 介面 1586 及 1588。處理器 1570、1580 可使用 P-P 介面電路 1578、1588 而經由點對點 (P-P) 介面 1550 來交換資訊。如圖 15 中所示，IMC 1572 及 1582 將處理器耦合至個別記憶體，亦即記憶體 1532 及記憶體 1534，其可為本地地裝附至個別處理器之主記憶體的部分。

處理器 1570、1580 可各經由個別的 P-P 介面 1552、1554 而與晶片組 1590 交換資訊，使用點對點介面電路 1576、1594、1586、1598。晶片組 1590 可經由高性能介面 1539 而選擇性地與共處理器 1538 交換資訊。於一實施例中，共處理器 1538 為特殊用途處理器，諸如（例如）高通量 MIC 處理器、網路或通訊處理器、壓縮引擎、圖形處理器、GPGPU、嵌入式處理器，等等。

共用快取（未顯示）可被包括於任一處理器中或者於兩處理器外部，而經由 P-P 互連與處理器連接，以致處理器之任一者或兩者的本地快取資訊可被儲存於共用快取中，假如處理器被置於低功率模式時。

晶片組 1590 可經由一介面 1596 而被耦合至第一匯流排 1516。於一實施例中，第一匯流排 1516 可為周邊組件互連 (PCI) 匯流排、或者諸如 PCI 快速匯流排或其他第三代 I/O 互連匯流排等匯流排，雖然本發明之範圍未如此限制。

如圖 15 中所示，各種 I/O 裝置 1514 可被耦合至第一

匯流排 1516，連同匯流排橋 1518，其係將第一匯流排 1516 耦合至第二匯流排 1520。於一實施例中，一或更多額外處理器 1515（諸如共處理器、高通量 MIC 處理器、GPGPU 加速器（諸如，例如，圖形加速器或數位信號處理（DSP）單元）、場可編程閘極陣列、或任何其他處理器）被耦合至第一匯流排 1516。於一實施例中，第二匯流排 1520 可為低管腳數（LPC）匯流排。各個裝置可被耦合至第二匯流排 1520，其包括（例如）鍵盤及/或滑鼠 1522、通訊裝置 1527、及儲存單元 1528，諸如磁碟機或其他大量儲存裝置（其可包括指令/碼及資料 1530），於一實施例中。此外，音頻 I/O 1524 可被耦合至第二匯流排 1520。注意：其他架構是可能的。例如，取代圖 15 之點對點架構，系統可實施多點分支匯流排或其他此類架構。

現在參考圖 16，其顯示依據本發明之實施例的第二更特定範例系統 1600 之方塊圖。圖 15 與 16 中之類似元件具有類似的參考數字，且圖 15 之某些形態已從圖 16 省略以免混淆圖 16 之其他形態。

圖 16 闡明其處理器 1570、1580 可包括集成記憶體及 I/O 控制邏輯（「CL」）1572 和 1582，個別地。因此，CL 1572、1582 包括集成記憶體控制器單元並包括 I/O 控制邏輯。圖 16 闡明其不僅記憶體 1532、1534 被耦合至 CL 1572、1582，同時其 I/O 裝置 1614 亦被耦合至控制邏輯 1572、1582。舊有 I/O 裝置 1615 被耦合至晶片組 1590。

現在參考圖 17，其顯示依據本發明之一實施例的 SoC 1700 之方塊圖。圖 13 中之類似元件具有類似的參考數字。同時，虛線方塊為更多先進 SoC 上之選擇性特徵。於圖 17 中，互連單元 1702 被耦合至：應用程式處理器 1710，其包括一組一或更多核心 202A-N 及共用快取單元 1306；系統代理單元 1310；匯流排控制器單元 1316；集成記憶體控制器單元 1314；一組或者一或更多共處理器 1720，其可包括集成圖形邏輯、影像處理器、音頻處理器、及視頻處理器；靜態隨機存取記憶體（SRAM）單元 1730；直接記憶體存取（DMA）單元 1732；及顯示單元 1740，用以耦合至一或更多外部顯示。於一實施例中，共處理器 1720 包括特殊用途處理器，諸如（例如）網路或通訊處理器、壓縮引擎、GPGPU、高通量 MIC 處理器、嵌入式處理器，等等。

文中所揭露之（例如，機制的）實施例可被實施以硬體、軟體、韌體、或此等實施方式之組合。本發明之實施例可被實施為電腦程式或程式碼，其被執行於可編程系統上，該可編程系統包含至少一處理器、儲存系統（包括揮發性和非揮發性記憶體及/或儲存元件）、至少一輸入裝置、及至少一輸出裝置。

程式碼（諸如圖 15 中所示之碼 1530）可被應用於輸入指令以履行文中所述之功能並產生輸出資訊。輸出資訊可被應用於一或更多輸出裝置，以已知的方式。為了本申請案之目的，處理系統包括任何系統，其具有處理器，諸

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。

程式碼可被實施以高階程序或目標導向的編程語言來與處理系統通訊。程式碼亦可被實施以組合或機器語言，假如想要的話。事實上，文中所述之機制在範圍上不限於任何特定編程語言。於任何情況下，該語言可為編譯或解讀語言。

至少一實施例之一或更多形態可由其儲存在機器可讀取媒體上之代表性指令所實施，該機器可讀取媒體代表處理器內之各個邏輯，當由機器讀取時造成該機器製造邏輯以履行文中所述之技術。此等表示（已知為「IP 核心」）可被儲存在有形的、機器可讀取媒體上，且被供應至各個消費者或製造設施以載入其實際上製造該邏輯或處理器之製造機器。

此類機器可讀取儲存媒體可包括（無限制）由機器或裝置所製造或形成之物件的非暫態、有形配置，包括：儲存媒體，諸如硬碟、包括軟碟、光碟、微型碟唯讀記憶體（CD-ROM）、微型碟可再寫入（CD-RW）、及磁光碟等任何其他類型的碟片；半導體裝置，諸如唯讀記憶體（ROM）、諸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）等隨機存取記憶體（RAM）、快閃記憶體、電可抹除可編程唯讀記憶體（EEPROM）、相位改變記憶體（PCM）、磁或光學卡、或者適於儲存電子指令之任何

其他類型的媒體。

因此，本發明之實施例亦可包括含有指令或含有諸如硬體描述語言（HDL）等設計資料之非暫態、有形的機器可讀取媒體，該硬體描述語言（HDL）係定義文中所述之結構、電路、設備、處理器及/或系統特徵。此類實施例亦可被稱為程式產品。

仿真（包括二元翻譯、碼變形，等等）

於某些情況下，指令轉換器可被用以將來自來源指令集之指令轉換至目標指令集。例如，指令轉換器可將指令翻譯（例如，使用靜態二元翻譯、動態二元翻譯，包括動態編譯）、變形、仿真、或者轉換至一或更多其他指令以供由核心所處理。指令轉換器可被實施以軟體、硬體、韌體、或其組合。指令轉換器可位於處理器上、處理器外、或者部分於處理器上而部分於處理器外。

圖 18 為一種對照軟體指令轉換器之使用的方塊圖，該轉換器係用以將來源指令集中之二元指令轉換至目標指令集中之二元指令，依據本發明之實施例。於所述之實施例中，指令轉換器為一種軟體指令轉換器，雖然替代地該指令轉換器亦可被實施以軟體、韌體、硬體、或其各種組合。圖 18 顯示一種高階語言 1802 之程式可使用 x86 編譯器 1804 而被編譯以產生 x86 二元碼 1806，其可由具有至少一 x86 指令集核心之處理器 1816 來本機地執行。具有至少一 x86 指令集核心之處理器 1816 代表任何處理器，

其可藉由可相容地執行或者處理以下事項來履行實質上如一種具有至少一 x86 指令集核心之 Intel 處理器的相同功能：（1）Intel x86 指令集核心之指令集的實質部分或者（2）針對運作於具有至少一 x86 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，以獲得如具有至少一 x86 指令集核心之 Intel 處理器的相同結果。x86 編譯器 1804 代表一種編譯器，其可操作以產生 x86 二元碼 1806（例如，物件碼），其可（具有或沒有額外鏈結處理）被執行於具有至少一 x86 指令集核心之處理器 1816 上。類似地，圖 18 顯示高階語言 1802 之程式可使用替代的指令集編譯器 1808 而被編譯以產生替代的指令集二元碼 1810，其可由沒有至少一 x86 指令集核心之處理器 1814 來本地地執行（例如，具有其執行 MIPS Technologies of Sunnyvale, CA 之 MIPS 指令集及/或其執行 ARM Holdings of Sunnyvale, CA 之 ARM 指令集的核心之處理器）。指令轉換器 1812 被用以將 x86 二元碼 1806 轉換為其可由沒有一 x86 指令集核心之處理器 1814 來本地地執行的碼。已轉換碼不太可能相同於替代的指令集二元碼 1810，因為能夠執行此功能之指令很難製造；然而，已轉換碼將完成一般性操作並由來自替代指令集之指令所組成。因此，指令轉換器 1812 代表軟體、韌體、硬體、或其組合，其（透過仿真、模擬或任何其他程序）容許處理器或其他不具有 x86 指令集處理器或核心的電子裝置來執行 x86 二元碼 1806。

【符號說明】

- 100：硬體處理器
- 102：硬體解碼電路
- 104：硬體執行電路
- 106：暫存器
- 108：目的地暫存器
- 109：來源暫存器
- 110：資料儲存裝置
- 112：目的地儲存
- 114：來源儲存
- 200：硬體處理器
- 201：編碼格式轉換指令
- 202：解碼電路
- 204：執行單元
- 206：暫存器
- 208：目的地暫存器
- 209：來源暫存器
- 210：儲存裝置
- 212：目的地儲存
- 214：來源儲存
- 300、310、320、330、340：指令格式
- 400：虛擬碼
- 800：一般性向量友善指令格式

- 805：無記憶體存取
- 810：無記憶體存取、全捨入控制類型操作
- 812：無記憶體存取、寫入遮蔽控制、部分捨入控制類型操作
- 815：無記憶體存取、資料變換類型操作
- 817：無記憶體存取、寫入遮蔽控制、v 大小類型操作
- 820：記憶體存取
- 827：記憶體存取、寫入遮蔽控制
- 830：記憶體存取、非暫時
- 840：格式欄位
- 842：基礎操作欄位
- 844：暫存器指標欄位
- 846：修飾符欄位
- 850：擴增操作欄位
- 852： α 欄位
- 852A：RS 欄位
- 852A.1：捨入
- 852A.2：資料變換
- 852B：逐出暗示欄位
- 852B.1：暫時
- 852B.2：非暫時
- 854： β 欄位
- 854A：捨入控制欄位
- 854B：資料變換欄位

- 854C : 資料調處欄位
- 856 : SAE 欄位
- 857A : RL 欄位
- 857A.1 : 捨入
- 857A.2 : 向量長度 (VSIZE)
- 857B : 廣播欄位
- 858 : 捨入操作控制欄位
- 859A : 捨入操作欄位
- 859B : 向量長度欄位
- 860 : 比例欄位
- 862A : 置換欄位
- 862B : 置換因數欄位
- 864 : 資料元件寬度欄位
- 868 : 類別欄位
- 868A : 類別 A
- 868B : 類別 B
- 870 : 寫入遮蔽欄位
- 872 : 即刻欄位
- 874 : 全運算碼欄位
- 900 : 特定向量友善指令格式
- 902 : EVEX 前綴
- 905 : REX 欄位
- 910 : REX'欄位
- 915 : 運算碼映圖欄位

- 920 : VVVV 欄位
- 925 : 前綴編碼欄位
- 930 : 真實運算碼欄位
- 940 : Mod R/M 位元組
- 942 : MOD 欄位
- 944 : Reg 欄位
- 946 : R/M 欄位
- 954 : SIB.xxx
- 956 : SIB.bbb
- 1000 : 暫存器架構
- 1010 : 向量暫存器
- 1015 : 寫入遮蔽暫存器
- 1025 : 通用暫存器
- 1045 : 純量浮點堆疊暫存器檔
- 1050 : MMX 緊縮整數平坦暫存器檔
- 1100 : 處理器管線
- 1102 : 提取級
- 1104 : 長度解碼級
- 1106 : 解碼級
- 1108 : 配置級
- 1110 : 重新命名級
- 1112 : 排程級
- 1114 : 暫存器讀取/記憶體讀取級
- 1116 : 執行級

- 1118 : 寫入回/記憶體寫入級
- 1122 : 例外處置級
- 1124 : 確定級
- 1130 : 前端單元
- 1132 : 分支預測單元
- 1134 : 指令快取單元
- 1136 : 指令變換後備緩衝 (TLB)
- 1138 : 指令提取單元
- 1140 : 解碼單元
- 1150 : 執行引擎單元
- 1152 : 重新命名/配置器單元
- 1154 : 撤回單元
- 1156 : 排程器單元
- 1158 : 實體暫存器檔單元
- 1160 : 執行叢集
- 1162 : 執行單元
- 1164 : 記憶體存取單元
- 1170 : 記憶體單元
- 1172 : 資料 TLB 單元
- 1174 : 資料快取單元
- 1176 : 第二階 (L2) 快取單元
- 1190 : 處理器核心
- 1200 : 指令解碼單元
- 1202 : 晶粒上互連網路

- 1204：第二階（L2）快取
- 1206：L1 快取
- 1206A：L1 資料快取
- 1208：純量單元
- 1210：向量單元
- 1212：純量暫存器
- 1214：向量暫存器
- 1220：拌合單元
- 1222A-B：數字轉換單元
- 1224：複製單元
- 1226：寫入遮蔽暫存器
- 1228：16 寬的 ALU
- 1300：處理器
- 1302A-N：核心
- 1306：共用快取單元
- 1308：特殊用途邏輯
- 1310：系統代理
- 1312：環狀為基的互連單元
- 1314：集成記憶體控制器單元
- 1316：匯流排控制器單元
- 1400：系統
- 1410、1415：處理器
- 1420：控制器集線器
- 1440：記憶體

- 1440A：編碼格式轉換模組
- 1445：共處理器
- 1450：輸入/輸出集線器（IOH）
- 1460：輸入/輸出（I/O）裝置
- 1490：圖形記憶體控制器集線器（GMCH）
- 1495：連接
- 1500：多處理器系統
- 1514：I/O 裝置
- 1515：額外處理器
- 1516：第一匯流排
- 1518：匯流排橋
- 1520：第二匯流排
- 1522：鍵盤及/或滑鼠
- 1524：音頻 I/O
- 1527：通訊裝置
- 1528：儲存單元
- 1530：指令/碼及資料
- 1532：記憶體
- 1534：記憶體
- 1538：共處理器
- 1539：高性能介面
- 1550：點對點互連
- 1552、1554：P-P 介面
- 1570：第一處理器

- 1572、1582：集成記憶體控制器（IMC）單元
- 1576、1578：點對點（P-P）介面
- 1580：第二處理器
- 1586、1588：P-P 介面
- 1590：晶片組
- 1594、1598：點對點介面電路
- 1596：介面
- 1600：系統
- 1614：I/O 裝置
- 1615：舊有 I/O 裝置
- 1700：SoC
- 1702：互連單元
- 1710：應用程式處理器
- 1720：共處理器
- 1730：靜態隨機存取記憶體（SRAM）單元
- 1732：直接記憶體存取（DMA）單元
- 1740：顯示單元
- 1802：高階語言
- 1804：x86 編譯器
- 1806：x86 二元碼
- 1808：指令集編譯器
- 1810：指令集二元碼
- 1812：指令轉換器
- 1814：沒有至少一 x86 指令集核心之處理器

1816：具有至少一 x86 指令集核心之處理器

申請專利範圍

1. 一種用於轉換編碼格式的硬體處理器，包含：
解碼電路，用以解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元；及
執行電路，用以執行該指令以：
將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式，
以該第二編碼格式將該些元件儲存於該目的地向量運算元中，
以該第二編碼格式將該些元件之總長度儲存於該狀態運算元中，及
當來自該來源向量運算元之該些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中，
其中該控制運算元可包括：變遷狀態指明符，用以指明其儲存變遷狀態的可用暫存器之一的位置；驗證指示符，用以指示是否履行驗證；及端讀取格式指示符，用以指示來源及/或目的地為小端讀取格式或者大端讀取格式。
2. 如申請專利範圍第 1 項之硬體處理器，其中該第一編碼格式為 UTF-8 字符編碼與 UTF-16 字符編碼之一者，而該第二編碼格式為 UTF-8 字符編碼與 UTF-16 字符編碼之另一者。
3. 如申請專利範圍第 1 項之硬體處理器，其中該執行電路係用以執行該指令以：

檢測來自該來源向量運算元之多元件序列之一或更多元件，其中該多元件序列不會結束於該來源向量運算元中，且

不會將來自該來源向量運算元之該多元件序列的該些一或更多元件從該第一編碼格式轉換至該第二編碼格式。

4. 如申請專利範圍第 3 項之硬體處理器，其中該執行電路係用以執行該指令以將來自該來源向量運算元之該多元件序列的該些一或更多元件儲存於該狀態運算元中。

5. 如申請專利範圍第 1 項之硬體處理器，其中該執行電路係用以執行該指令以讀取該控制運算元之端讀取格式指示符來判定具有該第二編碼格式之該些元件將何時以大端讀取格式與小端讀取格式之一被儲存於該目的地向量運算元中。

6. 如申請專利範圍第 1 項之硬體處理器，其中該執行電路係用以執行該指令以讀取該控制運算元之驗證指示符來判定何時造成驗證操作被履行於該第一編碼格式及該第二編碼格式之該些元件上。

7. 如申請專利範圍第 1 項之硬體處理器，其中該執行電路係用以執行該指令以讀取該控制運算元之狀態指示符來判定何時將來自該來源向量運算元之該些元件的未完成序列從該第一編碼格式轉換至該第二編碼格式。

8. 如申請專利範圍第 1 項之硬體處理器，其中該控制運算元包含用以儲存該狀態運算元之暫存器的位址。

9. 一種用於轉換編碼格式的方法，包含：

解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元；及

執行該指令以：

將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式，

以該第二編碼格式將該些元件儲存於該目的地向量運算元中，

以該第二編碼格式將該些元件之總長度儲存於該狀態運算元中，及

當來自該來源向量運算元之該些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中，

其中該控制運算元可包括：變遷狀態指明符，用以指明其儲存變遷狀態的可用暫存器之一的位置；驗證指示符，用以指示是否履行驗證；及端讀取格式指示符，用以指示來源及/或目的地為小端讀取格式或者大端讀取格式。

10. 如申請專利範圍第 9 項之方法，其中該第一編碼格式為 UTF-8 字符編碼與 UTF-16 字符編碼之一者，而該第二編碼格式為 UTF-8 字符編碼與 UTF-16 字符編碼之另一者。

11. 如申請專利範圍第 9 項之方法，進一步包含：

檢測來自該來源向量運算元之多元件序列的一或更多元件，其中該多元件序列不結束於該來源向量運算元中，及

不將來自該來源向量運算元之該多元件序列的該些一或更多元件從該第一編碼格式轉換至該第二編碼格式。

12. 如申請專利範圍第 11 項之方法，進一步包含將來自該來源向量運算元之該多元件序列的該些一或更多元件儲存於該狀態運算元中。

13. 如申請專利範圍第 9 項之方法，進一步包含讀取該控制運算元之端讀取格式指示符以判定具有該第二編碼格式之該些元件將何時以大端讀取格式與小端讀取格式之一被儲存於該目的地向量運算元中。

14. 如申請專利範圍第 9 項之方法，進一步包含讀取該控制運算元之驗證指示符以判定何時造成驗證操作被履行於該第一編碼格式及該第二編碼格式之該些元件上。

15. 如申請專利範圍第 9 項之方法，進一步包含讀取該控制運算元之狀態指示符以判定何時將來自該來源向量運算元之該些元件的未完成序列從該第一編碼格式轉換至該第二編碼格式。

16. 如申請專利範圍第 9 項之方法，其中該控制運算元包含用以儲存該狀態運算元之暫存器的位址。

17. 一種儲存碼之非暫態機器可讀取媒體，當由機器所執行時，該非暫態機器可讀取媒體係造成該機器履行一方法，包含：

解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元；及

執行該指令以：

將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式，

以該第二編碼格式將該些元件儲存於該目的地向量運算元中，

以該第二編碼格式將該些元件之總長度儲存於該狀態運算元中，及

當來自該來源向量運算元之該些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中，

其中該控制運算元可包括：變遷狀態指明符，用以指明其儲存變遷狀態的可用暫存器之一的位置；驗證指示符，用以指示是否履行驗證；及端讀取格式指示符，用以指示來源及/或目的地為小端讀取格式或者大端讀取格式。

18. 如申請專利範圍第 17 項之非暫態機器可讀取媒體，其中該第一編碼格式為 UTF-8 字符編碼與 UTF-16 字符編碼之一者，而該第二編碼格式為 UTF-8 字符編碼與 UTF-16 字符編碼之另一者。

19. 如申請專利範圍第 17 項之非暫態機器可讀取媒體，其中該方法包含：

檢測來自該來源向量運算元之多元件序列的一或更多元件，其中該多元件序列不結束於該來源向量運算元中，及

不將來自該來源向量運算元之該多元件序列的該些一或更多元件從該第一編碼格式轉換至該第二編碼格式。

20. 如申請專利範圍第 19 項之非暫態機器可讀取媒體，其中該方法包含：

將來自該來源向量運算元之該多元件序列的該些一或更多元件儲存於該狀態運算元中。

21. 如申請專利範圍第 17 項之非暫態機器可讀取媒體，其中該方法包含：

讀取該控制運算元之端讀取格式指示符以判定具有該第二編碼格式之該些元件將何時以大端讀取格式與小端讀取格式之一被儲存於該目的地向量運算元中。

22. 如申請專利範圍第 17 項之非暫態機器可讀取媒體，其中該方法包含：

讀取該控制運算元之驗證指示符以判定何時造成驗證操作被履行於該第一編碼格式及該第二編碼格式之該些元件上。

23. 如申請專利範圍第 17 項之非暫態機器可讀取媒體，其中該方法包含：

讀取該控制運算元之狀態指示符以判定何時將來自該來源向量運算元之該些元件的未完成序列從該第一編碼格式轉換至該第二編碼格式。

24. 如申請專利範圍第 17 項之非暫態機器可讀取媒體，其中該控制運算元包含用以儲存該狀態運算元之暫存器的位址。

圖式

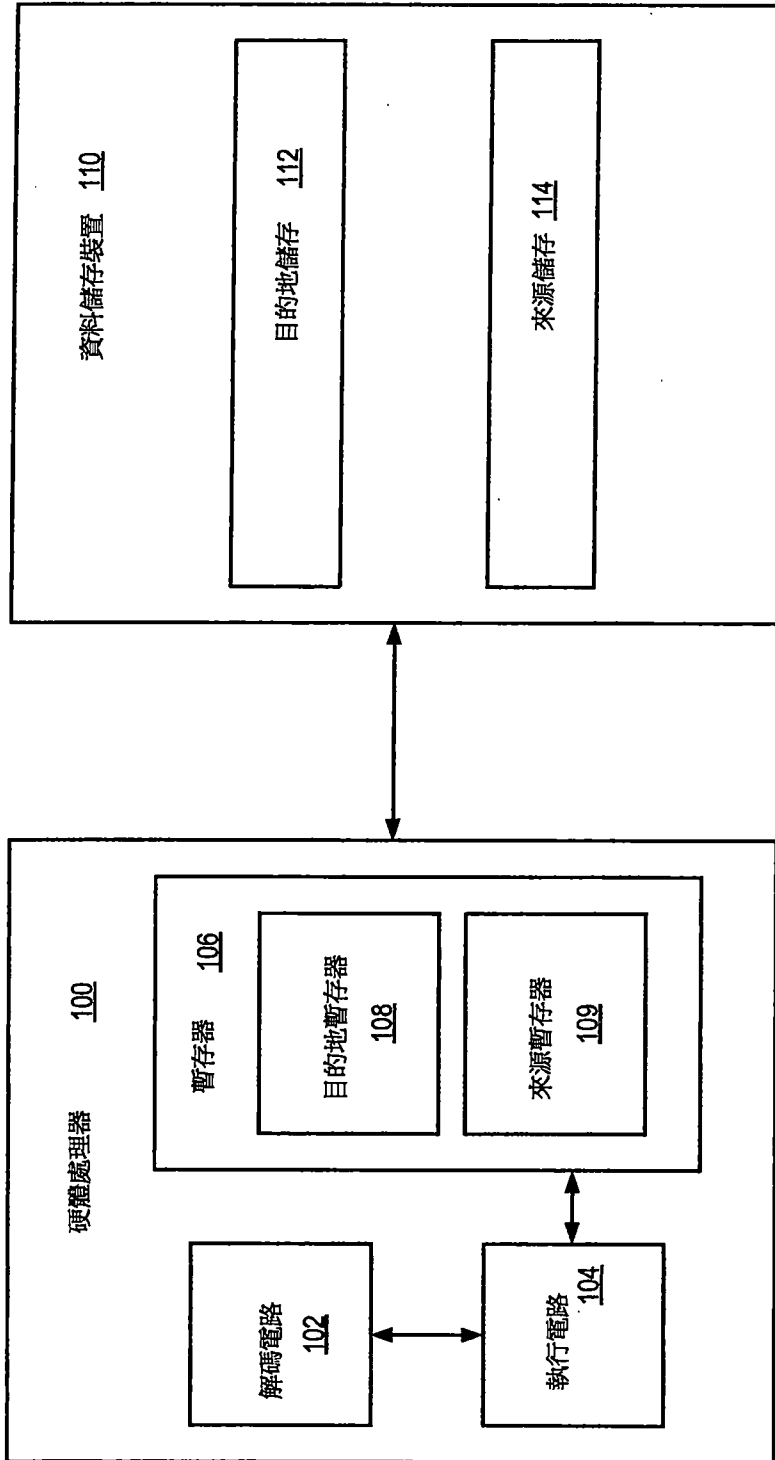


圖 1

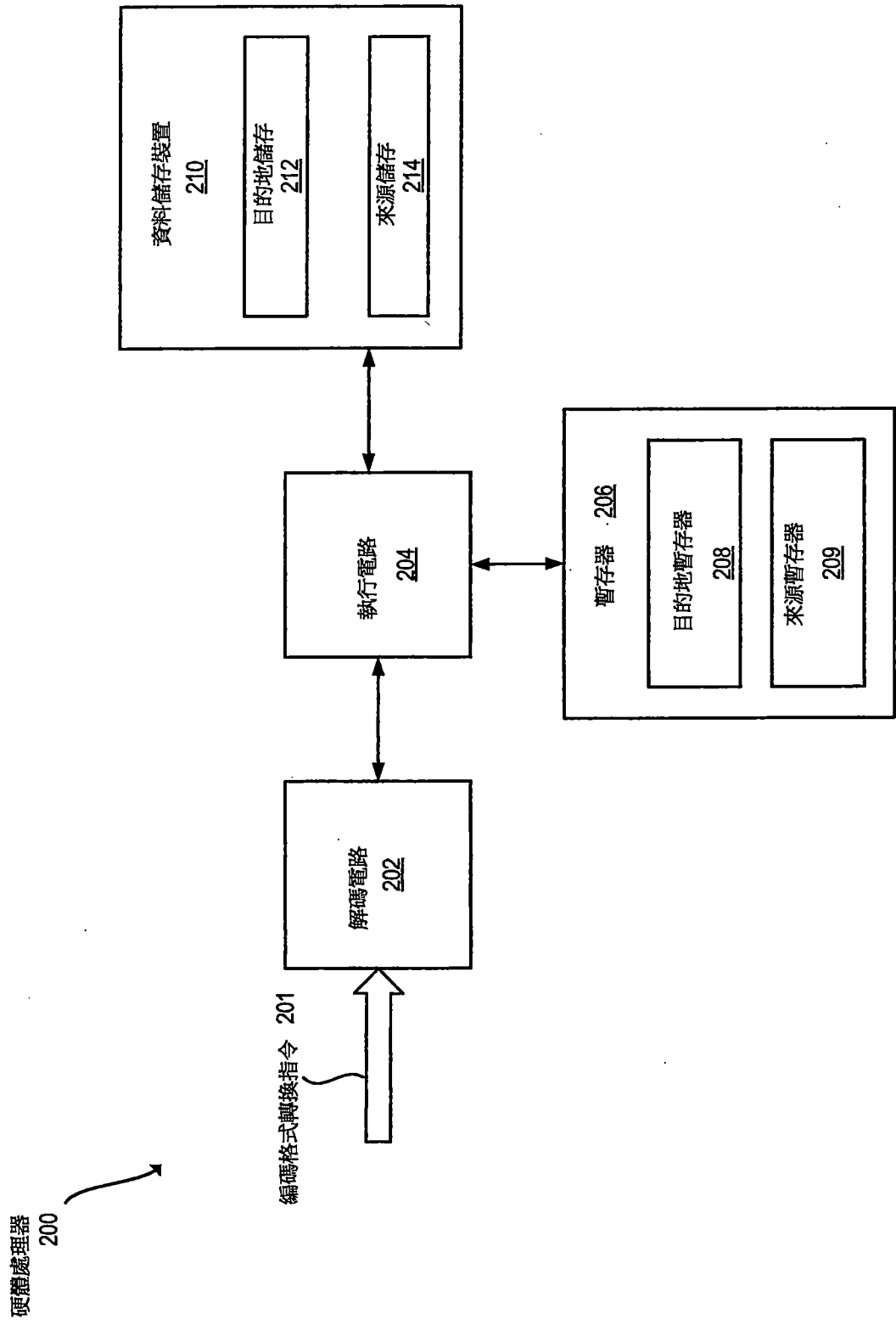


圖 2

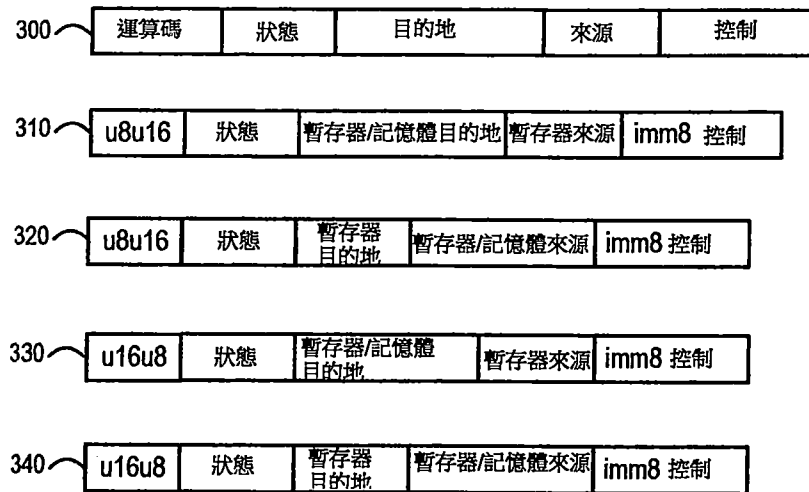


圖 3

編碼格式轉換虛擬碼		400:
00:	LLOOP	
01:	add rsi, 16	
02:	cmp rsi, srcEnd	
03:	jb LTAIL; // go and process the tail	
04:	movdqu xmm0, [rsi-16]	
05:	u8u16 [rdi], xmm0, 0xCTRL(rax)	
06:	jc LERROR; // go and find the error	
07:	// advance the destination pointer	
08:	mov rbx, rax	
09:	and rbx, STATE_STEP_MASK; // step mask = 0x3E	
10:	add rdi, rbx	
11:	jmp LLOOP	

圖 4

500 u8src-word00: D0 95 D1 81 D1 82 D1 8C 20 D1 82 D0 BE D0 BB D1 ; incomplete
 u8src-word16: 8C D0 BA D0 BE 20 D0 B4 D0 B2 D0 B5 20 2E 2E 2E ; complete

510 u8u16 ymm8, u8src-word00, 0x12
 ; control imm8 = 0x12 = 0b010010 = {low-endian + validate + r2=rcx}
 ; Right after the instruction there is the following:
 ; CF = 0 - no error
 ; ymm8 = 0415 0441 0442 044C 0020 0442 043E 043B
 ; rcx = 0xD111 = (0xD1 << 8) + (8 << 1) + 1 = {incompleteBytes(D1) + u16length(8)
 ; + incompleteFlag(1)}

520 u8u16 ymm9, u8src-word16, 0x12
 ; control imm8 = 0x12 = 0b010010 = {low-endian + validate + r2=rcx}
 ; Right after the instruction there is the following:
 ; CF = 0 - no error
 ; ymm9 = 044C 043A 043E 0020 0434 0432 0435 0020 002E 002E 002E
 ; rcx = 0x0016 = (0x00 << 8) + (11 << 1) + 0 = {incompleteBytes(0) + u16length(11)
 ; + incompleteFlag(0)}

圖 5

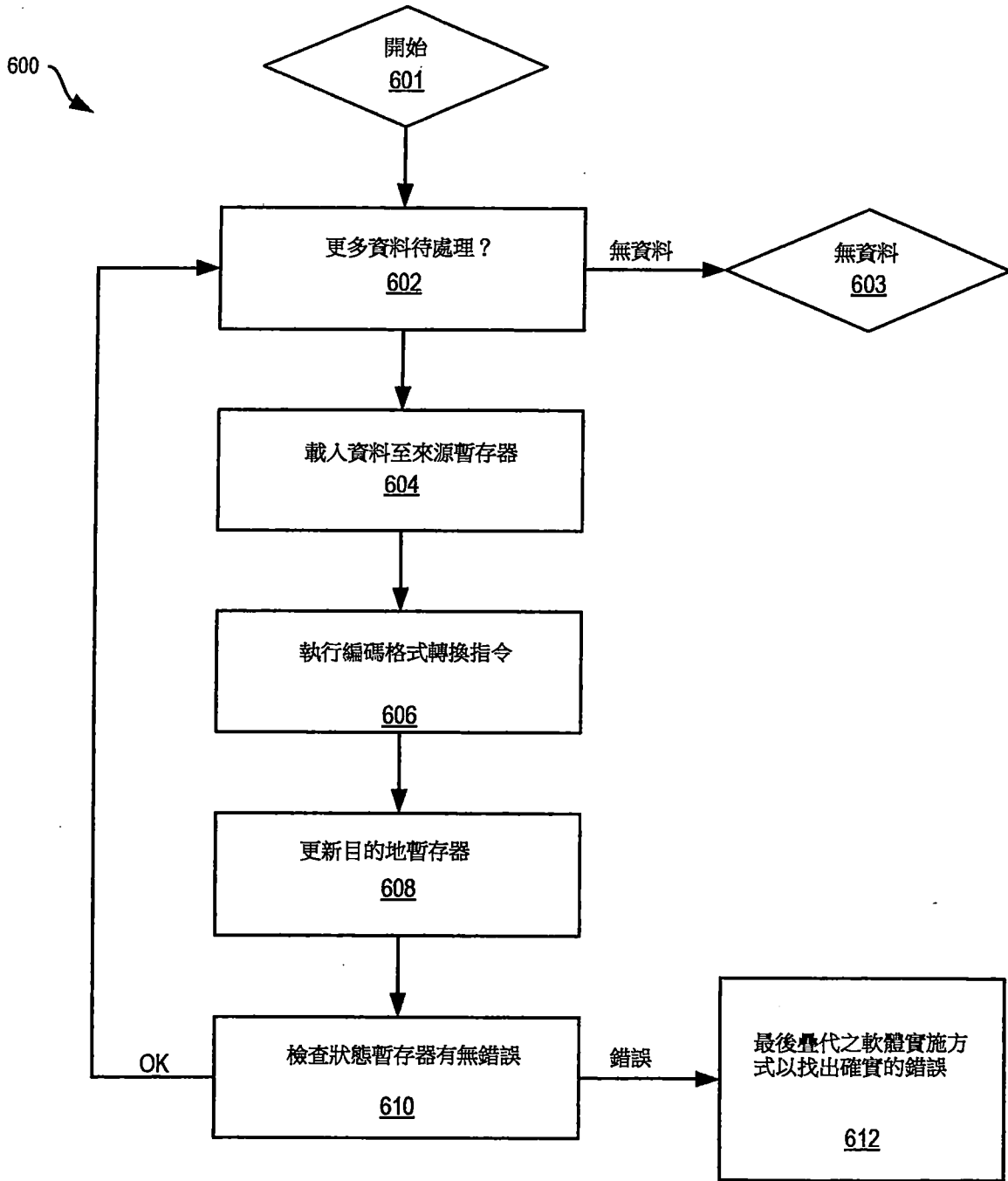


圖 6

700



解碼一指令，其包含狀態運算元、來源向量運算元、目的地向量運算元、和控制運算元 702



執行該指令，用以：將來自該來源向量運算元之元件從第一編碼格式轉換至第二編碼格式、以該第二編碼格式將該些元件儲存於該目的地向量運算元中、以該第二編碼格式將該些元件之總長度儲存於該狀態運算元中、及當來自該來源向量運算元之該些元件為資料串中之最後元件時，設定串流完成指示於該控制運算元中 704

圖 7

圖 8A

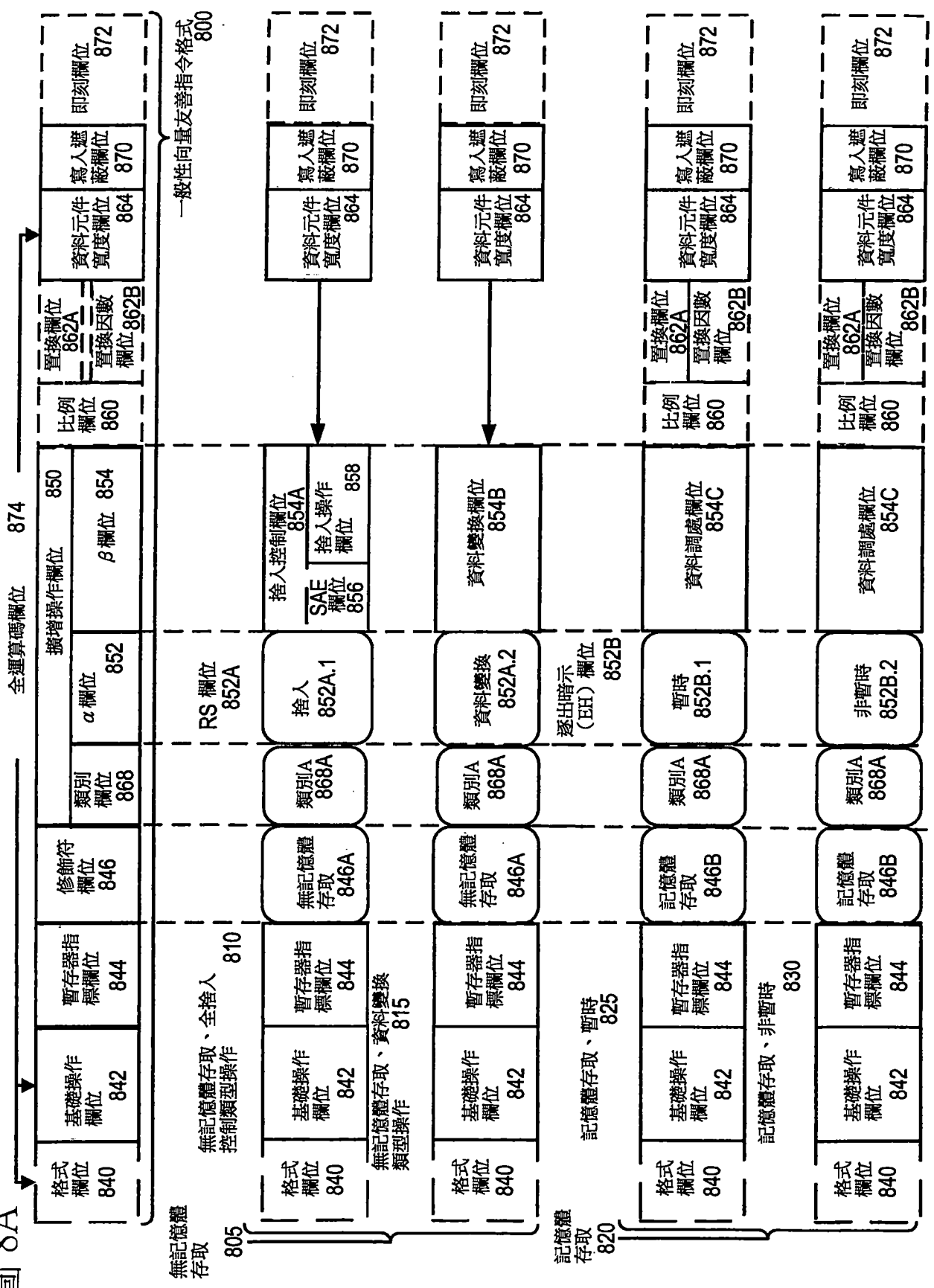


圖 8B

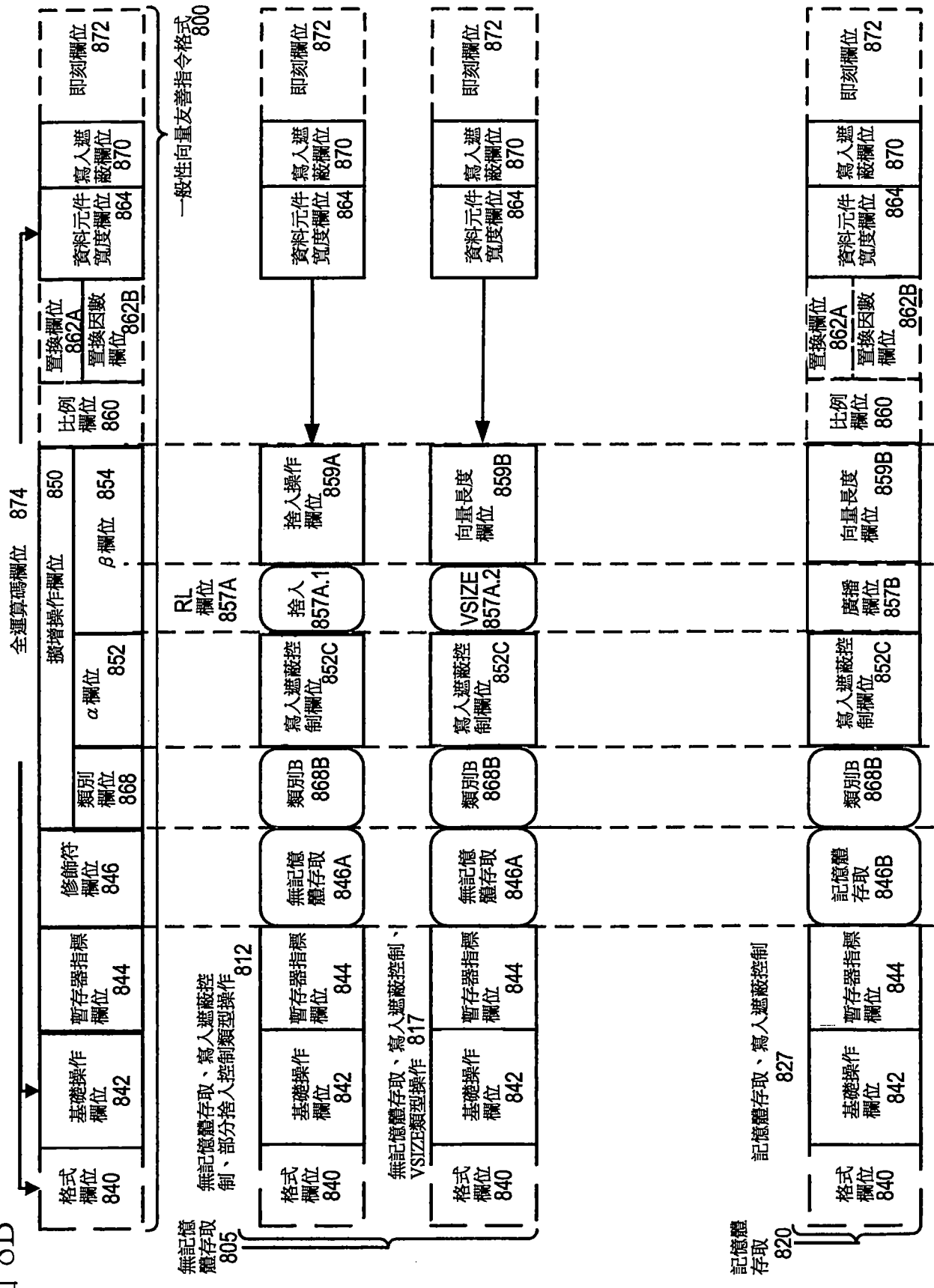
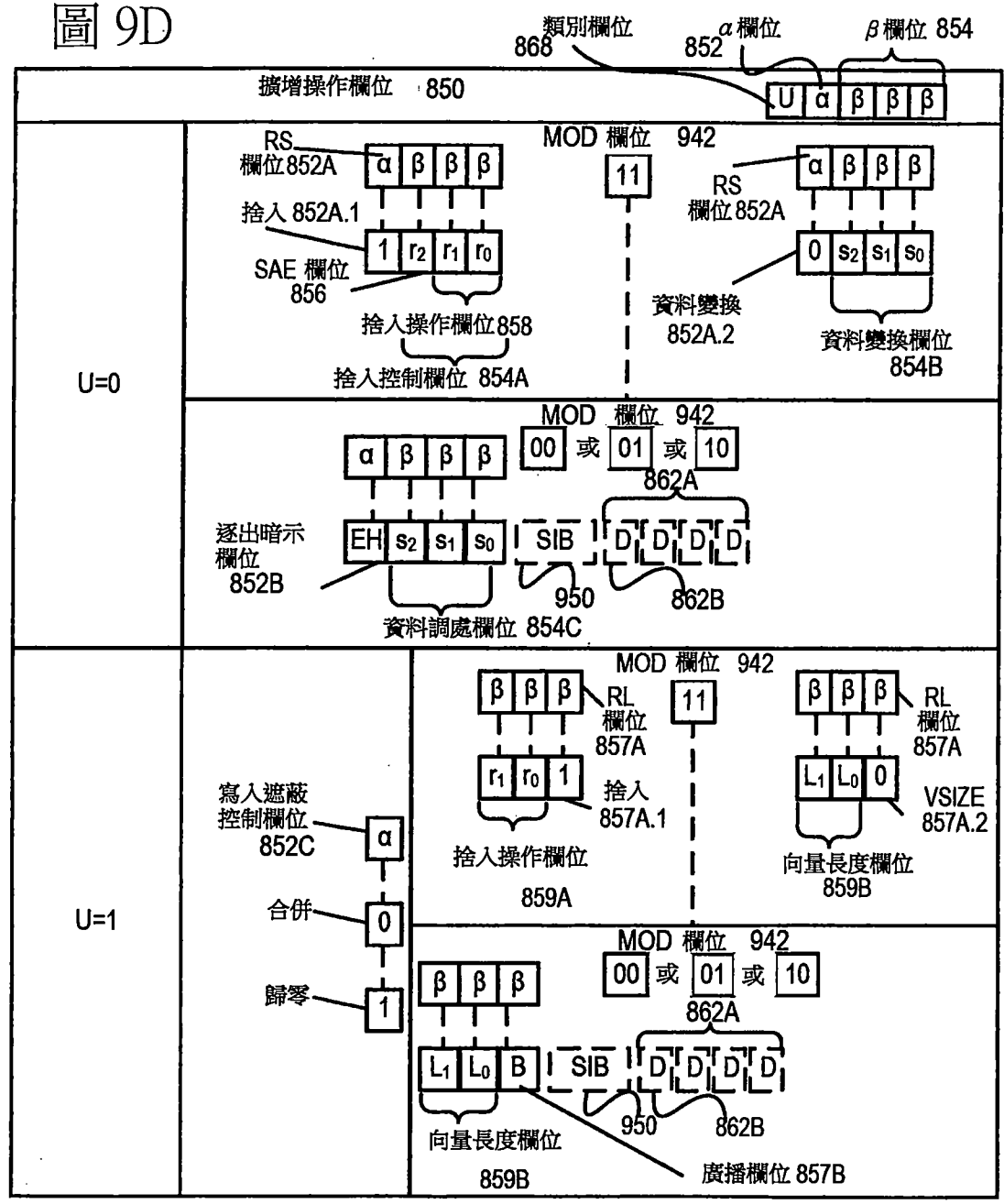


圖 9D



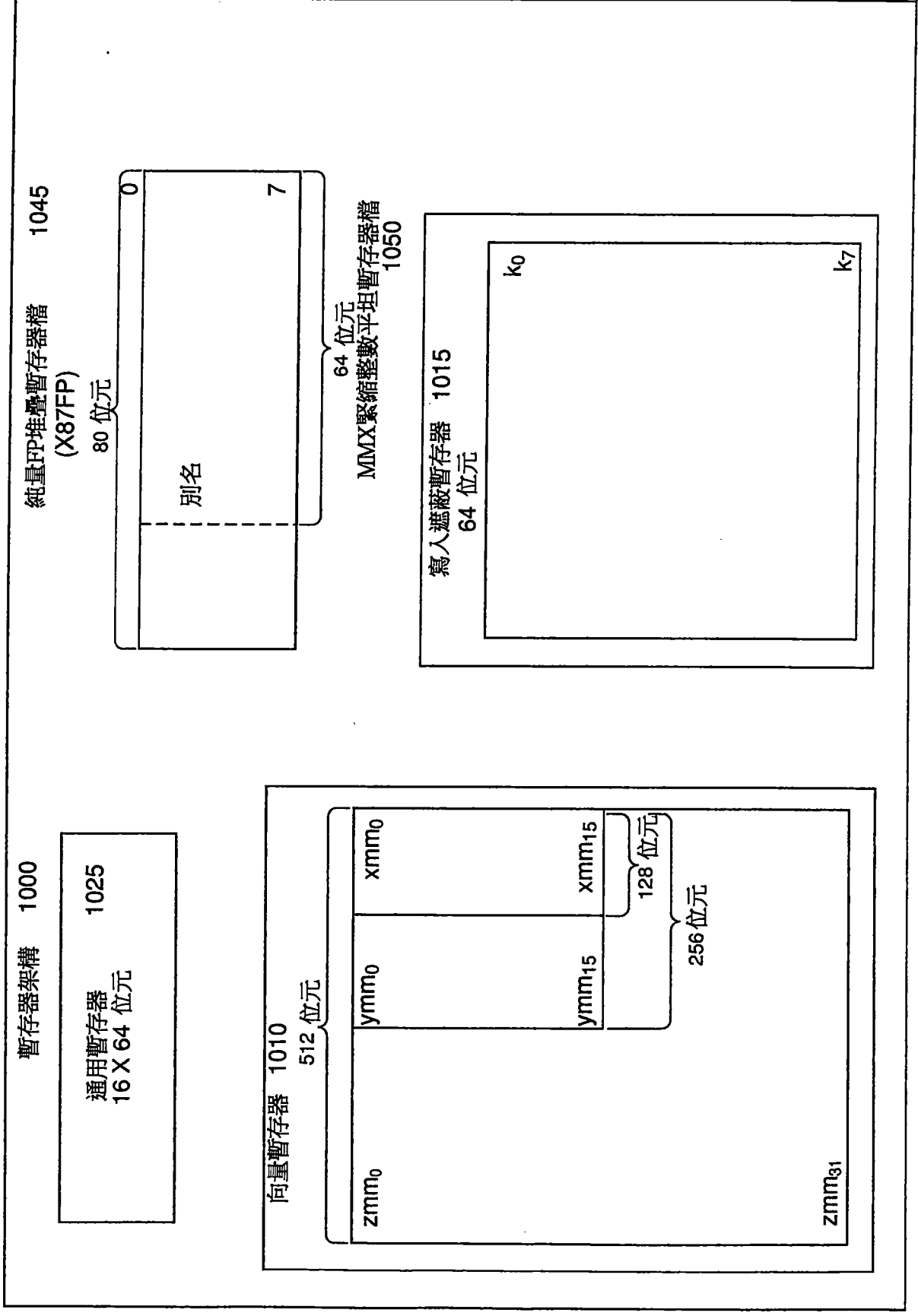
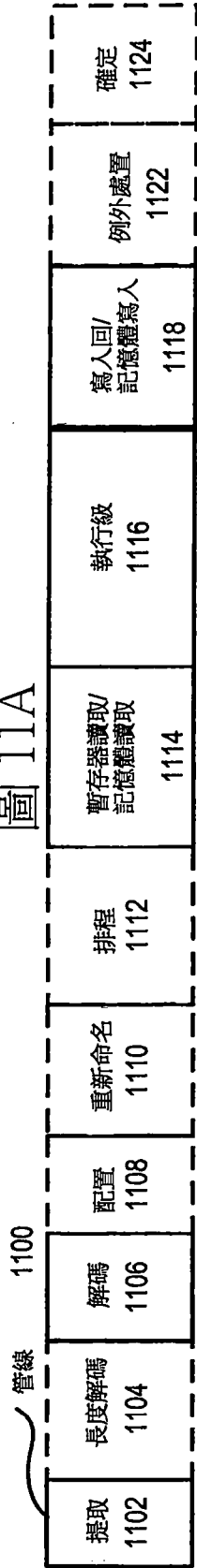


圖 10

圖 11A



核心 1190

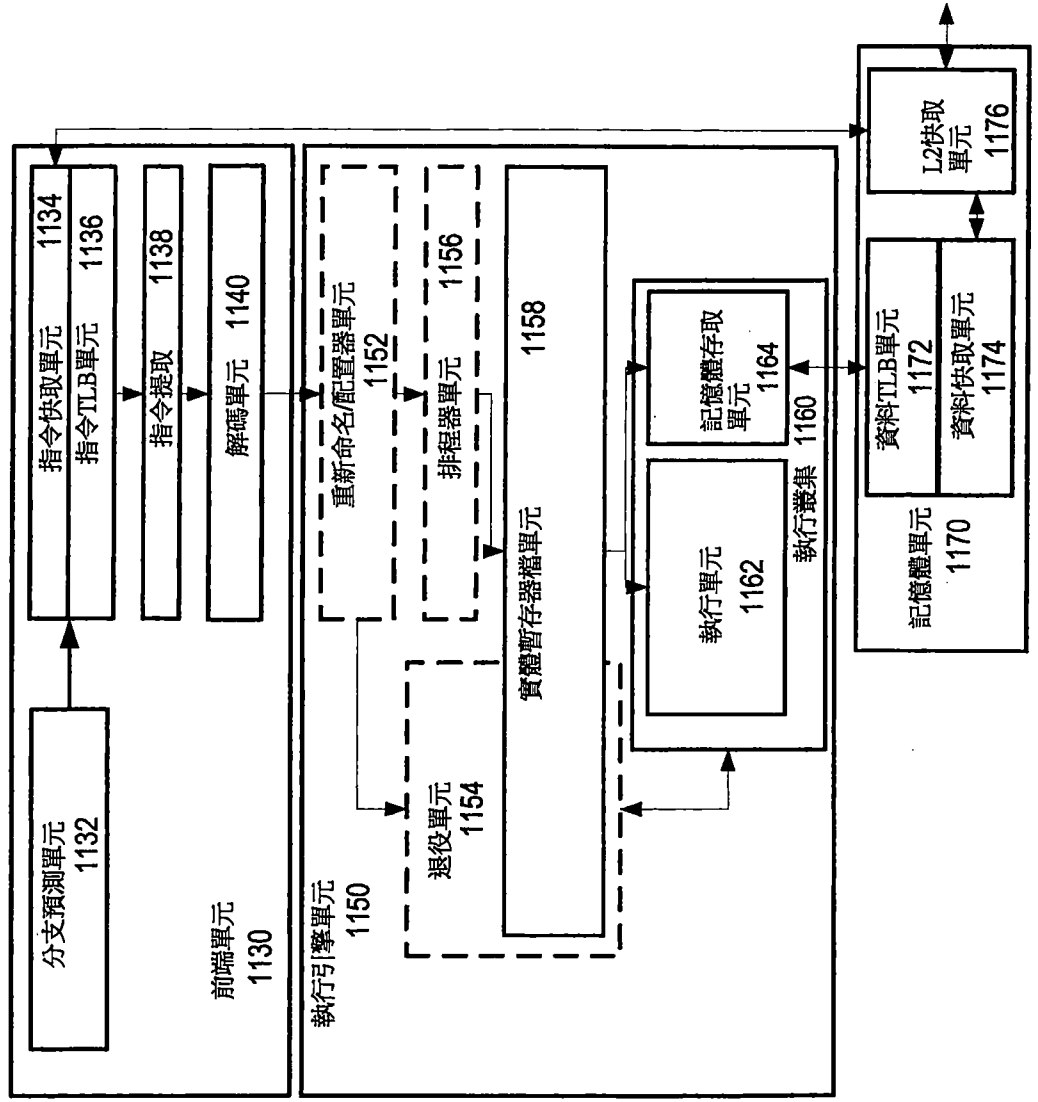


圖 11B

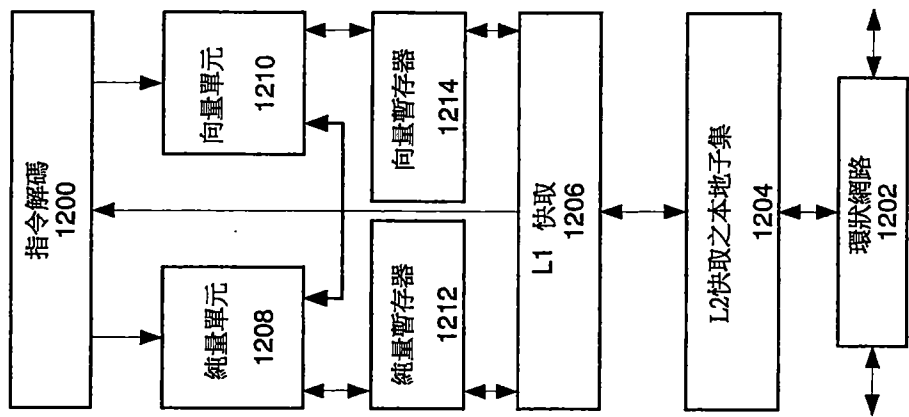


圖 12A

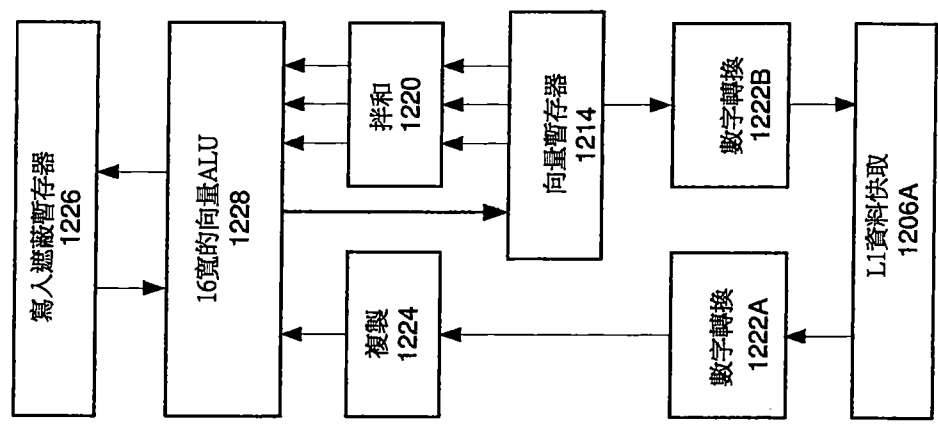


圖 12B

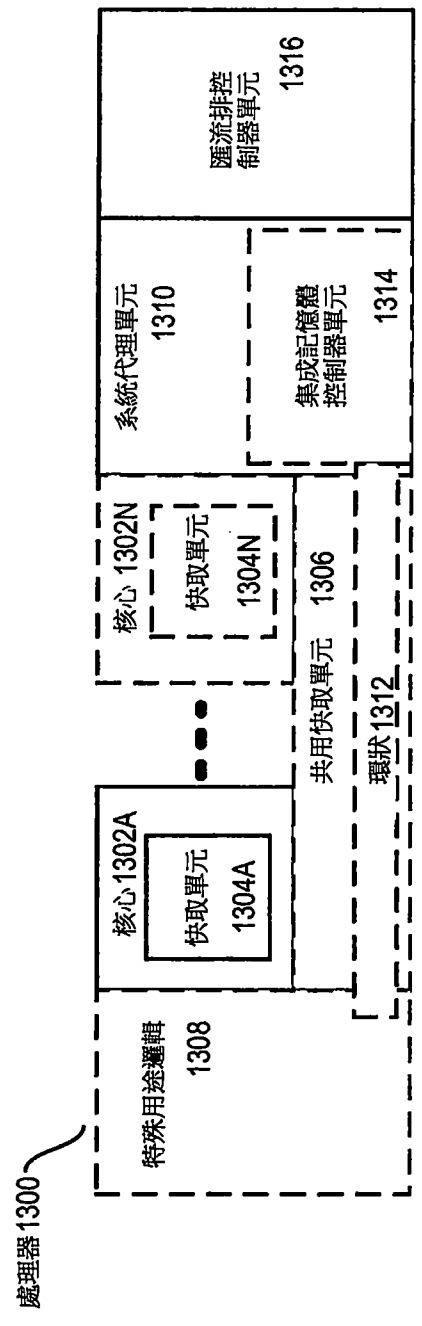


圖 13

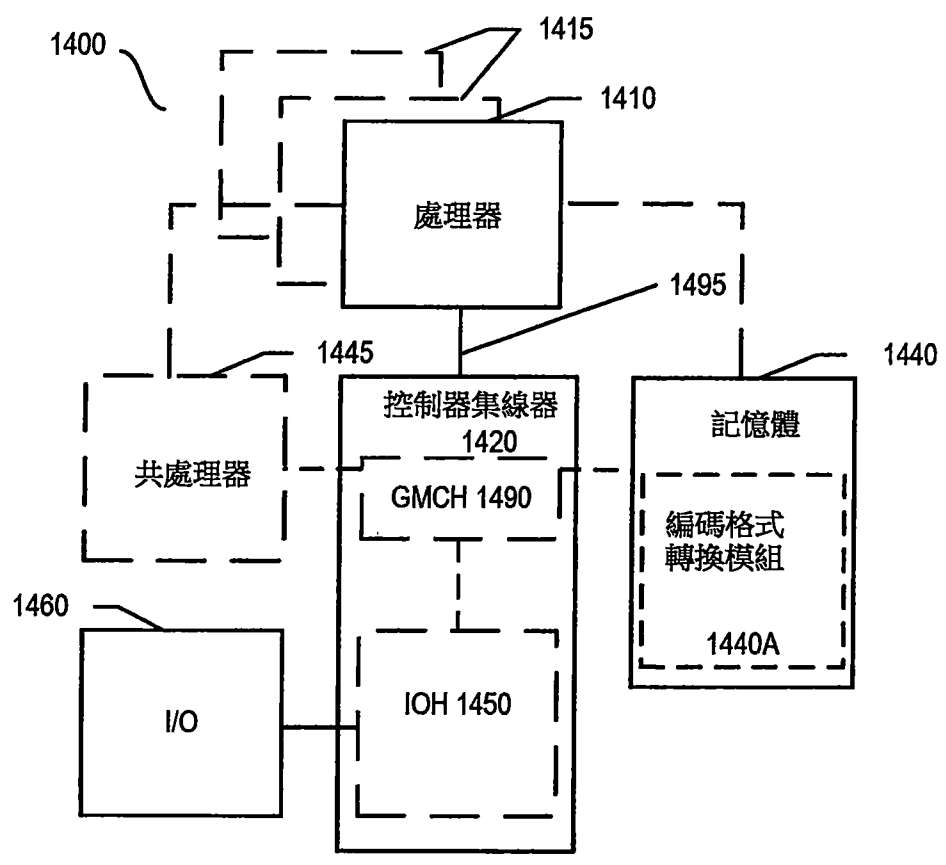


圖 14

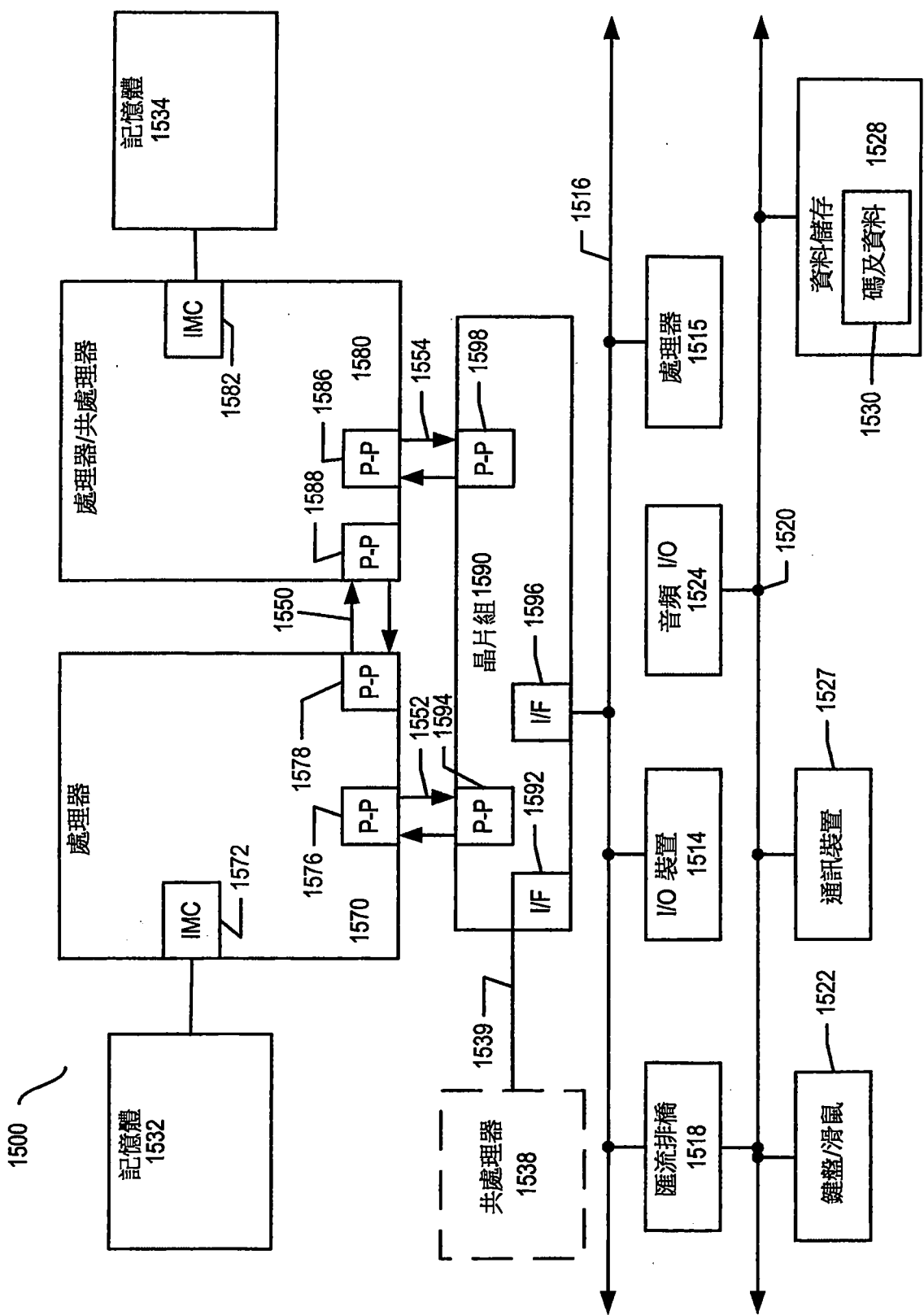


圖 15

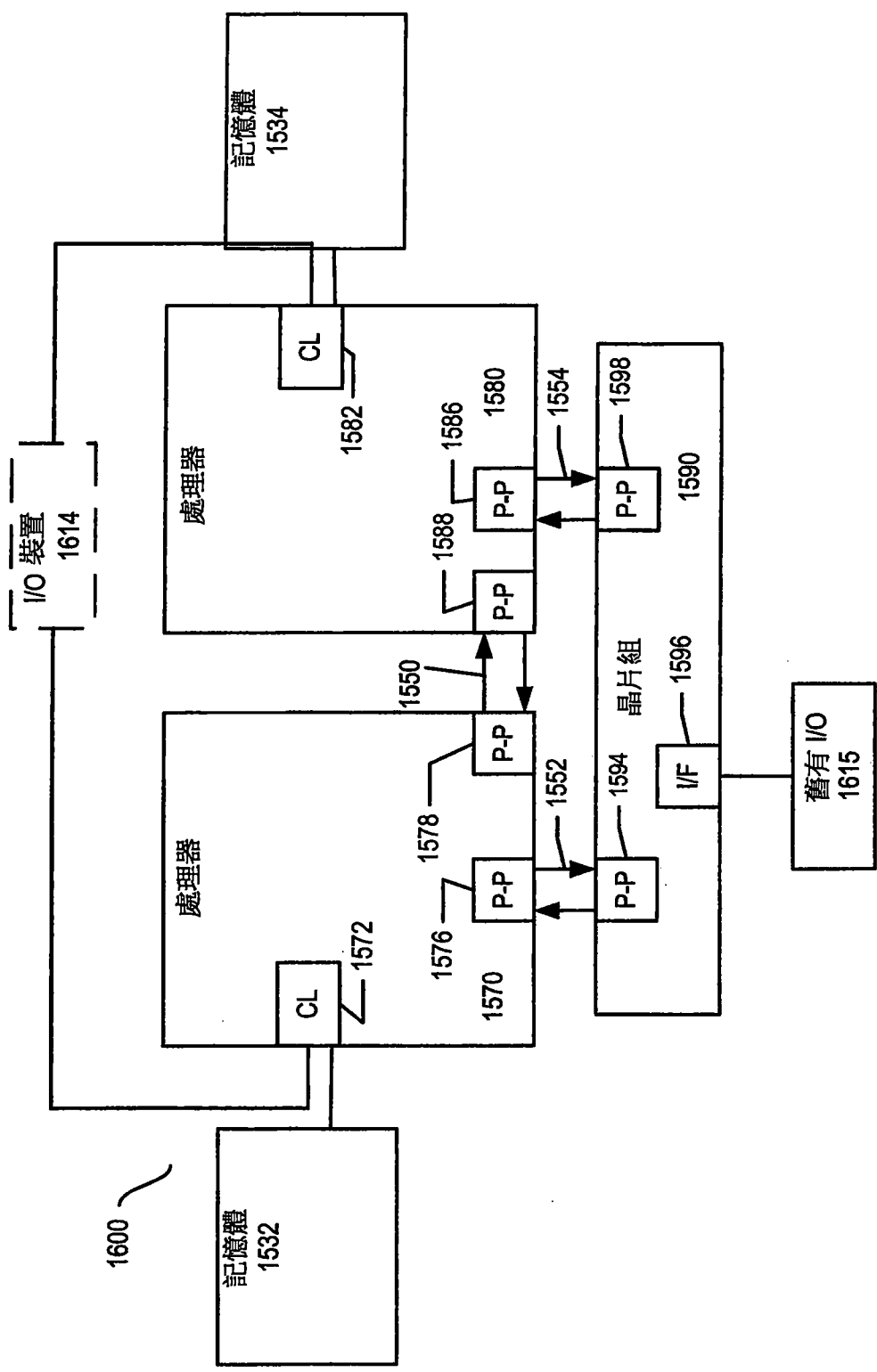


圖 16

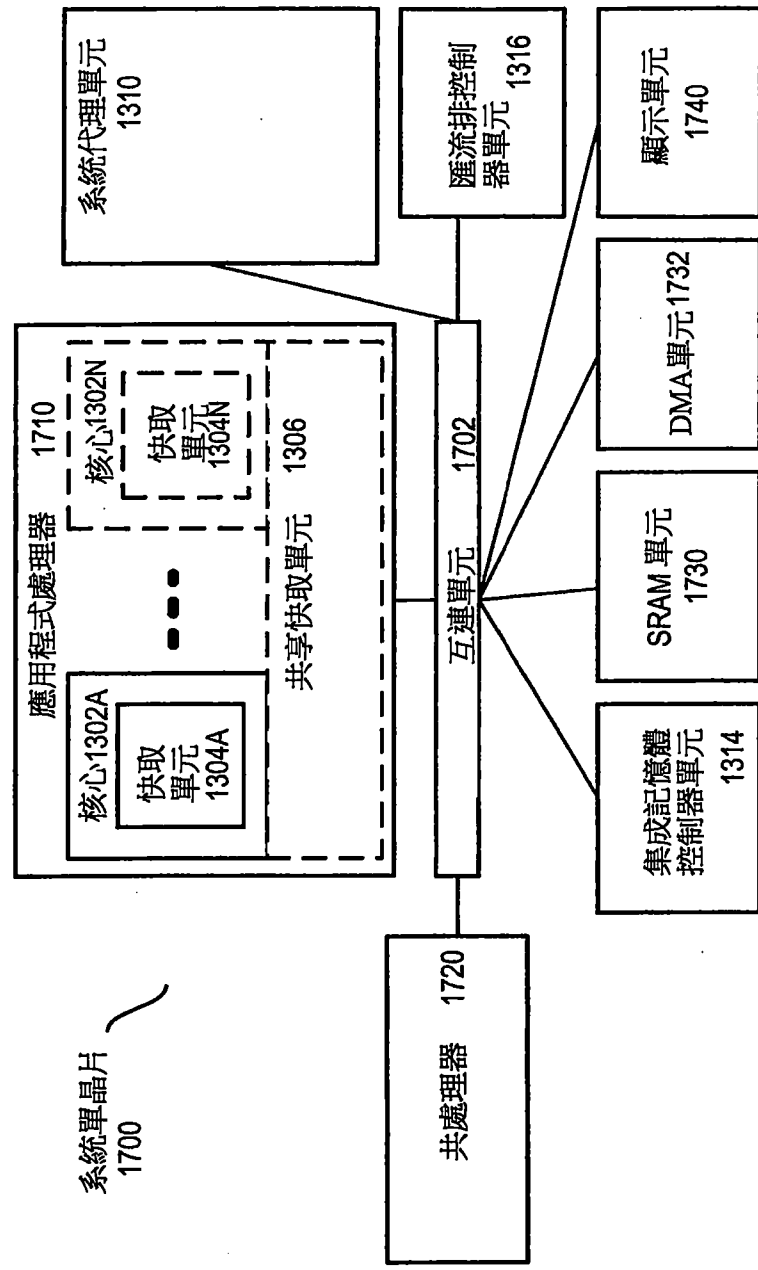


圖 17

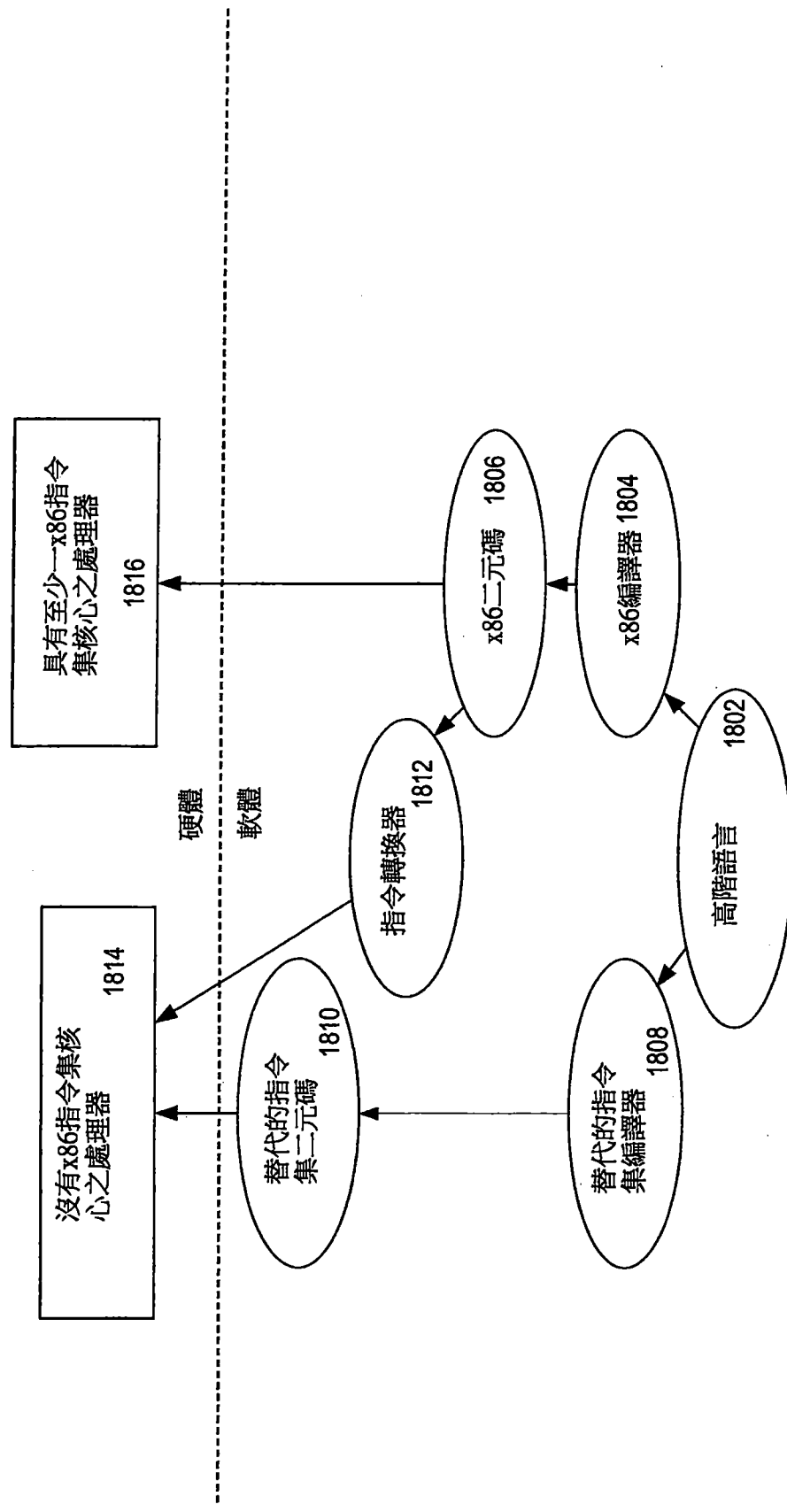


圖 18