



(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 I781142 B

(45)公告日：中華民國 111 (2022) 年 10 月 21 日

(21)申請案號：107105950

(22)申請日：中華民國 107 (2018) 年 02 月 22 日

(51)Int. Cl. : G06T1/20 (2006.01)

G06T1/40 (2006.01)

G06N20/00 (2019.01)

(30)優先權：2017/04/28 美國

15/581,167

(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：亞布 亞奇雪克 APPU, ABHISHEK (IN)；亞西鮑佛 班 ASHBAUGH, BEN (US)；金德煥 KIM, DUKHWAN (KR)；歐德亞麥德維爾 艾蒙斯特阿法 OULD-AHMED-VALL, ELMOUSTAPHA (MR)；唐 屏 TANG, PING (US)；陳曉明 CHEN, XIAOMING (CN)；拉克斯曼 巴拉斯 LAKSHMANAN, BARATH (IN)；尼爾斯 凱文 NEALIS, KEVIN (US)；史崔克蘭 麥可 STRICKLAND, MICHAEL (US)；寇克 歐塔 KOKER, ALTUG (US)；麥佛森 麥克 MACPHERSON, MIKE (US)；雷 喬伊迪普 RAY, JOYDEEP (US)；姚安邦 YAO, ANBANG (CN)；馬立偉 MA, LIWEI (CN)；赫德 琳達 HURD, LINDA (US)；班索艾 莎拉 BAGHSORKHI, SARA (IR)；威斯特 約翰 WEAST, JOHN C. (US)

(74)代理人：林志剛

(56)參考文獻：

TW 200937341A

TW 201346724A

US 2013/0212139A1

US 2017/0090924A1

審查人員：彭智輝

申請專利範圍項數：18 項 圖式數：36 共 188 頁

(54)名稱

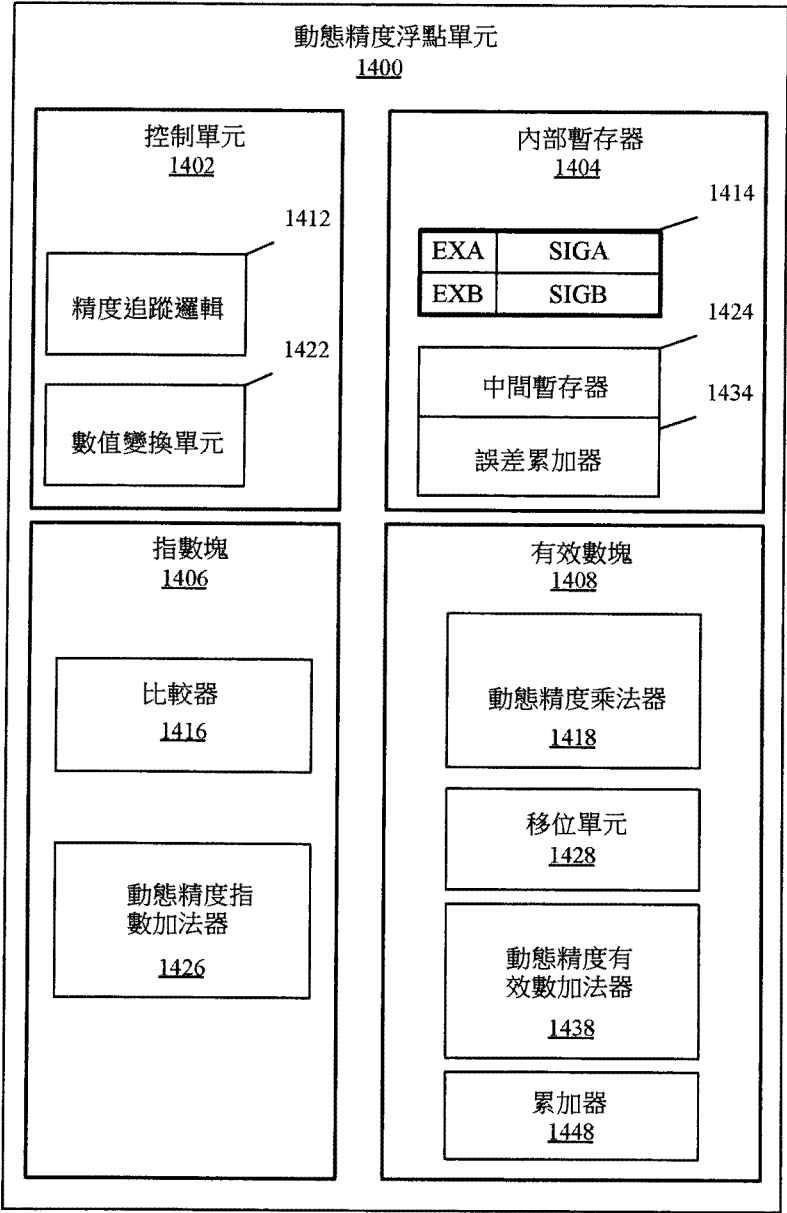
用於低精度機器學習操作之計算最佳化的通用圖形處理單元及資料處理系統

(57)摘要

一個實施例提供一通用圖形處理單元，包含動態精度浮點單元，其包括控制單元，具有精度追蹤硬體邏輯以追蹤相對於目標精度之用於計算資料的精度的可用位元數，其中該動態精度浮點單元包括計算邏輯，用來以多個精度輸出資料。

One embodiment provides a general-purpose graphics processing unit comprising a dynamic precision floating-point unit including a control unit having precision tracking hardware logic to track an available number of bits of precision for computed data relative to a target precision, wherein the dynamic precision floating-point unit includes computational logic to output data at multiple precisions.

指定代表圖：



- 符號簡單說明：
- 1400:動態精度浮點單元
 - 1402:控制單元
 - 1404:內部暫存器
 - 1406:指數塊
 - 1408:有效數塊
 - 1412:精度追蹤邏輯
 - 1414:運算元暫存器
 - 1416:比較器
 - 1418:動態精度乘法器
 - 1422:數值變換單元
 - 1424:中間暫存器
 - 1426:動態精度指數加法器
 - 1428:移位單元
 - 1434:誤差累加器
 - 1438:動態精度有效數加法器
 - 1448:累加暫存器

圖 14



【發明摘要】

【中文發明名稱】

用於低精度機器學習操作之計算最佳化的通用圖形處理單元及資料處理系統

【英文發明名稱】

GENERAL-PURPOSE GRAPHICS PROCESSING UNIT AND DATA PROCESSING SYSTEM FOR COMPUTE OPTIMIZATIONS FOR LOW PRECISION MACHINE LEARNING OPERATIONS

【中文】

一個實施例提供一通用圖形處理單元，包含動態精度浮點單元，其包括控制單元，具有精度追蹤硬體邏輯以追蹤相對於目標精度之用於計算資料的精度的可用位元數，其中該動態精度浮點單元包括計算邏輯，用來以多個精度輸出資料。

【英文】

One embodiment provides a general-purpose graphics processing unit comprising a dynamic precision floating-point unit including a control unit having precision tracking hardware logic to track an available number of bits of precision for computed data relative to a target precision, wherein the dynamic precision floating-point unit includes computational logic to output data at multiple precisions.

第 107105950 號

民國 111 年 4 月 11 日修正

【指定代表圖】第(14)圖。

【代表圖之符號簡單說明】

1400：動態精度浮點單元

1402：控制單元

1404：內部暫存器

1406：指數塊

1408：有效數塊

1412：精度追蹤邏輯

1414：運算元暫存器

1416：比較器

1418：動態精度乘法器

1422：數值變換單元

1424：中間暫存器

1426：動態精度指數加法器

1428：移位單元

1434：誤差累加器

1438：動態精度有效數加法器

1448：累加暫存器

【特徵化學式】無

【發明說明書】

【中文發明名稱】

用於低精度機器學習操作之計算最佳化的通用圖形處理單元及資料處理系統

【英文發明名稱】

GENERAL-PURPOSE GRAPHICS PROCESSING UNIT AND DATA PROCESSING SYSTEM FOR COMPUTE OPTIMIZATIONS FOR LOW PRECISION MACHINE LEARNING OPERATIONS

【技術領域】

【0001】實施例一般關於資料處理，更具體地關於透過通用圖形處理單元的資料處理。

【先前技術】

【0002】當前的平行圖形資料處理包括已開發用以對圖形資料執行特定操作的系統和方法，諸如，舉例來說，線性內插、曲面細分(tessellation)、光柵化(rasterization)、紋理映射、深度測試等。傳統上，圖形處理器使用固定功能計算單元來處理圖形資料；然而，近來，部分的圖形處理器已成為可編程的，使得這些處理器能支援更廣泛的操作來處理頂點和片段(fragment)資料。

【0003】為了進一步提高性能，圖形處理器通常實現諸如管線操作的處理技術，其試圖在整個圖形管線的不同部分平行處理盡可能多的圖形資料。具有單指令多線程(SIMT)結構的平行圖形處理器旨在將圖形管線中的平行處

理量最大化。在SIMT結構中，平行線程組試圖盡可能經常地同步執行程式指令，以提高處理效率。用於SIMT結構之軟體和硬體的總體概要可於Shane Cook、CUDA Programming (CUDA程式設計)、第3章、第37-51頁(2013)及/或Nicholas Wilt、CUDA Handbook (CUDA手冊)、A Comprehensive Guide to GPU Programming (GPU程式設計綜合指南)，第2.6.2至3.1.2節(2013年6月)中找到。

【圖式簡單說明】

【0004】為了能夠詳細理解本發明的特徵，可透過參考實施例來對本發明進行更具體的描述，在附圖中示出其中一些實施例。然而，應注意的是，附圖僅示出典型的實施例，因此不應被視為限制所有實施例的範圍。

【0005】圖1是示出被配置成實現本文所述之實施例的一或多個態樣的計算機系統的方塊圖；

【0006】圖2A-2D示出依據實施例之平行處理器組件；

【0007】圖3A-3B是依據實施例之圖形多處理器的方塊圖；

【0008】圖4A-4F示出其中複數個GPU通訊地耦接至複數個多核心處理器的示例性結構；

【0009】圖5示出依據實施例之圖形處理管線；

【0010】圖6示出依據實施例之機器學習軟體堆疊；

【0011】圖7示出依據實施例之高度平行通用圖形處

理單元；

【0012】圖8示出依據實施例之多GPU計算系統；

【0013】圖9A-9B示出示例性深度神經網路的層；

【0014】圖10示出示例性遞迴神經網路；

【0015】圖11示出深度神經網路的訓練和部署；

【0016】圖12是示出分散式學習的方塊圖；

【0017】圖13示出適用於使用訓練好的模型來執行推論的示例性推論系統單晶片(SOC)；

【0018】圖14示出依據實施例之動態精度浮點單元的組件；

【0019】圖15提供依據實施例之關於動態精度浮點單元的額外細節；

【0020】圖16示出依據實施例之用於動態精度處理系統的線程分配；

【0021】圖17示出依據實施例之以低於要求的精度執行數值運算的邏輯；

【0022】圖18示出依據實施例之用於SIMD單元的迴圈向量化；

【0023】圖19示出依據實施例之線程處理系統；

【0024】圖20示出依據實施例之分配線程用於計算的邏輯；

【0025】圖21示出可使用由本文所述之實施例所提供的計算邏輯來處理的深度神經網路；

【0026】圖22是依據實施例之邏輯的流程圖，該邏輯

用以防止在機器學習執行低精度運算時的錯誤或顯著的精度損失；

【0027】圖23是依據實施例之處理系統的方塊圖；

【0028】圖24是具有一或多個處理器核心、整合式記憶體控制器、及整合式圖形處理器之處理器的實施例的方塊圖；

【0029】圖25是圖形處理器的方塊圖，該圖形處理器可以是分離式圖形處理單元、或者可以是與複數個處理核心整合的圖形處理器；

【0030】圖26是依據一些實施例之圖形處理器的圖形處理引擎的方塊圖；

【0031】圖27是由另外的實施例提供的圖形處理器的方塊圖；

【0032】圖28示出線程執行邏輯，其包括在一些實施例中採用的處理元件的陣列；

【0033】圖29是示出依據一些實施例之圖形處理器指令格式的方塊圖；

【0034】圖30是依據另一實施例之圖形處理器的方塊圖；

【0035】圖31A-31B示出依據一些實施例之圖形處理器命令格式和命令序列；

【0036】圖32示出依據一些實施例之用於資料處理系統的示例性圖形軟體結構；

【0037】圖33是示出依據實施例之IP核心開發系統的

方塊圖；

【0038】圖34是示出依據實施例之示例性系統單晶片積體電路的方塊圖；

【0039】圖35是示出依據實施例之另外的圖形處理器的方塊圖；以及

【0040】圖36是示出依據實施例之系統單晶片積體電路之另外的示例性圖形處理器的方塊圖。

【發明內容】及【實施方式】

【0041】在一些實施例中，圖形處理單元(GPU)通訊地耦接至主機(host)/處理器核心，以加速圖形運算、機器學習運算、型樣分析(pattern analysis)運算、及各種通用GPU(GPGPU)功能。GPU可以透過匯流排或另一互連(例如，諸如PCIe或NVLink的高速互連)通訊地耦接至主機處理器/核心。在其他實施例中，GPU可以與核心整合在相同的封裝或晶片上，並且透過內部處理器匯流排/互連(即，封裝或晶片的內部)通訊地耦接至核心。不管GPU連接的方式如何，處理器核心都可以工作描述符中所包含的命令/指令序列的形式，將工作分配給GPU。然後GPU使用專用電路/邏輯來有效地處理這些命令/指令。

【0042】在下面的描述中，闡述了許多具體細節以提供更透徹的理解。然而，對於本領域之技術人員而言將顯而易見的是，可在沒有這些具體細節中的一或多個的情況下實施本文所述的實施例。在其他實例中，未描述公知的

特徵以避免模糊本實施例的細節。

系統概要

【0043】圖1是示出被配置成實現本文所述之實施例的一或多個態樣的計算系統100的方塊圖。計算系統100包括處理子系統101，其具有一或多個處理器102及系統記憶體104，透過可包括記憶體集線器105的互連路徑進行通訊。記憶體集線器105可以是晶片組組件內的獨立組件，或者可以整合在一或多個處理器102內。記憶體集線器105透過通訊鏈路106與I/O子系統111耦接。I/O子系統111包括I/O集線器107，其可以使計算系統100能夠接收來自一或多個輸入裝置108的輸入。另外，I/O集線器107可以使可被包括在一或多個處理器102中的顯示控制器向一或多個顯示裝置110A提供輸出。在一個實施例中，與I/O集線器107耦接的一或多個顯示裝置110A可包括本地、內部或嵌入式顯示裝置。

【0044】在一個實施例中，處理子系統101包括透過匯流排或其他通訊鏈路113耦接至記憶體集線器105的一或多個平行處理器112。通訊鏈路113可以是許多基於標準的通訊鏈路技術或協定的其中一者，諸如但不限於PCI Express，或者可以是供應商特定的通訊介面或通訊結構(fabric)。在一個實施例中，一或多個平行處理器112形成計算上集中的平行或向量處理系統，其包括大量處理核心及/或處理叢集(cluster)，諸如許多整合式核心(MIC)處理

器。在一個實施例中，一或多個平行處理器 112 形成一圖形處理子系統，其可輸出像素給透過 I/O 集線器 107 耦接的一或多個顯示裝置 110A 之其中一者。一或多個平行處理器 112 還可包括顯示控制器及顯示介面(未示出)，使得能夠直接連接到一或多個顯示裝置 110B。

【0045】在 I/O 子系統 111 內，系統儲存單元 114 可連接到 I/O 集線器 107，以為計算系統 100 提供儲存機制。I/O 開關 116 可被用來提供介面機制，以致能 I/O 集線器 107 和其他組件之間的連接，例如，可被整合進平台的網路配接器 118 及/或無線網路配接器 119、及可透過一或多個附加(add-in)裝置 120 而添加的各種其他裝置。網路配接器 118 可以是乙太網路(Ethernet)配接器或另一有線的網路配接器。無線網路配接器 119 可包括 Wi-Fi、藍牙、近場通訊(NFC)、或包括一或多個無線電(wireless radio)之其他網路裝置的其中一或多者。

【0046】計算系統 100 可包括未明確示出的其他組件，包括 USB 或其他埠連接、光學儲存驅動器、視訊擷取裝置等等，也可連接到 I/O 集線器 107。圖 1 中互連各種組件的通訊路徑可使用任何合適的協定來實現，諸如基於 PCI(週邊組件互連)的協定(例如，PCI-Express)、或者任何其他匯流排或點對點通訊介面及/或協定，諸如 NV-Link 高速互連、或本領域已知的互連協定。

【0047】在一個實施例中，一或多個平行處理器 112 結合針對圖形及視訊處理而最佳化的電路，包括例如視訊

輸出電路，並且構成圖形處理單元(GPU)。在另一實施例中，一或多個平行處理器112結合針對通用處理而最佳化的電路，同時保留底層的計算結構，本文有更詳細的描述。在又另一實施例中，計算系統100的組件可與單一積體電路上的一或多個其他系統元件整合在一起。例如，可將一或多個平行處理器112、記憶體集線器105、處理器102、及I/O集線器107整合進系統單晶片(SoC)積體電路。替代地，可將計算系統100的組件整合進單一封裝中，以形成系統級封裝(SIP)配置。在一個實施例中，可將計算系統100的至少一部分的組件整合進多晶片模組(MCM)，其可以與其他多晶片模組互連成一模組化計算系統。

【0048】應當理解的是，本文所示的計算系統100是示例性的，並且變形和修改是可能的。可視需要修改連接拓樸，包括橋接器的數量和配置、處理器102的數量、以及平行處理器112的數量。例如，在一些實施例中，系統記憶體104不是透過橋接器而是直接連接到處理器102，而其他的裝置則是透過記憶體集線器105及處理器102與系統記憶體104通訊。在其他替代的拓樸中，平行處理器112連接到I/O集線器107或者直接連接到一或多個處理器102之其中一者，而不是連接到記憶體集線器105。在其他實施例中，可將I/O集線器107和記憶體集線器105整合進單一晶片中。一些實施例可包括透過多個插座(socket)附接的兩組或更多組的處理器102，其可以與平行處理器112的兩個或更多個實例耦接。

【0049】本文所示的特定組件的其中一些是可選的，並且可能不包括在計算系統100的所有實現中。舉例來說，可以支持許多插卡(add-in card)或週邊，或可以省去一些組件。此外，一些結構可能使用不同的術語來描述類似於圖1中所示的那些組件的組件。舉例來說，在一些結構中，記憶體集線器105可被稱為北橋，而I/O集線器107可被稱為南橋。

【0050】圖2A示出依據實施例之平行處理器200。平行處理器200的各種組件可使用一或多個積體電路裝置，諸如可編程處理器、特定應用積體電路(ASIC)、或現場可編程閘陣列(FPGA)來實現。依據實施例，所示出的平行處理器200是圖1中所示之一或多個平行處理器112的變形。

【0051】在一個實施例中，平行處理器200包括平行處理單元202。平行處理單元包括I/O單元204，其致能與包括平行處理單元202的其他實例的其他裝置進行通訊。可將I/O單元204直接連接到其他裝置。在一個實施例中，I/O單元204透過使用集線器或開關介面，諸如記憶體集線器105，與其他裝置連接。記憶體集線器105和I/O單元204之間的連接形成通訊鏈路113。在平行處理單元202內，I/O單元204與主機介面206及記憶體交叉(memory crossbar)216連接，其中主機介面206接收指向執行處理操作的命令，而記憶體交叉216接收指向執行記憶體操作的命令。

【0052】當主機介面206透過I/O單元204接收命令緩衝器時，主機介面206可以將用以執行那些命令的工作操

作指向前端 208。在一個實施例中，前端 208 與排程器 210 耦接，該排程器 210 被配置成將命令或其他工作項目分發到處理叢集陣列 212。在一個實施例中，排程器 210 確保處理叢集陣列 212 被適當地配置，並且在任務被分發到處理叢集陣列 212 的處理叢集之前處於有效狀態。在一個實施例中，排程器 210 透過在微控制器上執行的韌體邏輯來實現。微控制器實現的排程器 210 可被配置成以粗粒度和精細粒度執行複雜的排程和工作分發操作，其致能在處理陣列 212 上執行的線程的快速先佔 (preemption) 和上下文切換 (context switching)。在一個實施例中，主機軟體可透過多個圖形處理門鈴之其中一者來查驗在處理陣列 212 上進行排程的工作量。然後可以透過排程器微控制器內的排程器 210 邏輯，將工作量自動地分配在處理陣列 212 中。

【0053】處理叢集陣列 212 可包括多達“N”個處理叢集 (例如，叢集 214A、叢集 214B、至叢集 214N)。處理叢集陣列 212 的每個叢集 214A-214N 可以執行大量的並行線程。排程器 210 可使用各種排程及/或工作分配演算法，將工作分配給處理叢集陣列 212 的叢集 214A-214N，該些演算法可依據每種類型的程式或計算所產生的工作量而不同。排程可以由排程器 210 動態地處理，或者可在編譯被配置為由處理叢集陣列 212 執行的程式邏輯的期間，由編譯器邏輯部分地協助。在一個實施例中，可將處理叢集陣列 212 的不同叢集 214A-214N 分配用於處理不同類型的程式或者用於執行不同類型的計算。

【0054】處理叢集陣列212可被配置成執行各種類型的平行處理操作。在一個實施例中，處理叢集陣列212被配置成執行通用平行計算操作。例如，處理叢集陣列212可包括邏輯，用以執行處理任務，包括過濾視訊及/或音訊資料、執行包括物理操作的模型化操作，以及執行資料轉換。

【0055】在一個實施例中，處理叢集陣列212被配置成執行平行圖形處理操作。在平行處理器200被配置成執行圖形處理操作的實施例中，處理叢集陣列212可包括用以支援執行此種圖形處理操作的額外的邏輯，包括但不限於用以執行紋理操作的紋理取樣邏輯、以及曲面細分邏輯和其他頂點處理邏輯。此外，處理叢集陣列212可被配置成執行圖形處理相關的著色器程式，諸如，但不限於頂點著色器、曲面細分著色器、幾何著色器、及像素著色器。平行處理單元202可透過I/O單元204從系統記憶體傳送資料用於處理。在處理期間，被傳送的資料可在處理期間被儲存到晶片上記憶體(例如，平行處理器記憶體222)，然後被寫回到系統記憶體。

【0056】在一個實施例中，當平行處理單元202被用來執行圖形處理時，排程器210可被配置成將處理工作量分成大約相等大小的任務，以更佳地使圖形處理操作能夠分配至處理叢集陣列212的多個叢集214A-214N。在一些實施例中，處理叢集陣列212的部分可被配置成執行不同類型的處理。例如，第一部分可被配置成執行頂點著色和

拓樸產生，第二部分可被配置成執行曲面細分和幾何著色，以及第三部分可被配置成執行像素著色或其他螢幕空間操作，以產生用於顯示的呈現影像。由叢集214A-214N的其中一或多者所產生的中間資料可被儲存在緩衝器中，以允許中間資料在叢集214A-214N之間傳輸以用於進一步處理。

【0057】在操作期間，處理叢集陣列212可以透過排程器210接收待執行的處理任務，該排程器210從前端208接收定義處理任務的命令。針對圖形處理操作，處理任務可包括待處理的資料的索引，例如，表面(修補(patch))資料、基元資料、頂點資料、及/或像素資料，以及狀態參數和定義資料將如何被處理的命令(例如，將執行什麼程式)。排程器210可被配置成提取對應於任務的索引，或者可從前端208接收索引。前端208可被配置成確保在啟動由輸入命令緩衝器(例如，批次緩衝器(batch-buffer)、推入緩衝器(push buffer)等)指定的工作量之前，處理叢集陣列212被配置為有效狀態。

【0058】平行處理單元202的一或多個實例的各者可與平行處理器記憶體222耦接。可透過記憶體交叉216來存取平行處理器記憶體222，該記憶體交叉216可接收來自處理叢集陣列212以及I/O單元204的記憶體請求。記憶體交叉216可透過記憶體介面218存取平行處理器記憶體222。記憶體介面218可包括多個分割(partition)單元(例如，分割單元220A、分割單元220B、至分割單元220N)，每個分

割單元可以耦接至平行處理器記憶體222的一部分(例如，記憶體單元)。在一個實施方式中，分割單元220A-220N的數量被配置成等於記憶體單元的數量，使得第一分割單元220A具有對應的第一記憶體單元224A、第二分割單元220B具有對應的記憶體單元224B、而第N個分割單元220N具有對應的第N個記憶體單元224N。在其他實施例中，分割單元220A-220N的數量可能不等於記憶體裝置的數量。

【0059】在各種實施例中，記憶體單元224A-224N可包括各種不同類型的記憶體裝置，包括動態隨機存取記憶體(DRAM)或圖形隨機存取記憶體，諸如同步圖形隨機存取記憶體(SGRAM)，包括圖形雙倍資料速率(GDDR)記憶體。在一個實施例中，記憶體單元224A-224N亦可包括3D堆疊式記憶體，包括但不限於高頻寬記憶體(HBM)。本領域之技術人士將理解到，記憶體單元224A-224N的具體實現可以有所變化，並且可選自各種傳統的設計之一。諸如幀緩衝器或紋理圖等的渲染目標可被儲存在記憶體單元224A至224N，允許分割單元220A-220N平行地寫入每個渲染目標的部分以有效地使用平行處理器記憶體222的可用頻寬。在一些實施例中，為了有利於共同利用系統記憶體與本地快取記憶體的統一記憶體設計，可排除平行處理器記憶體222的本地實例。

【0060】在一個實施例中，處理叢集陣列212之叢集214A-214N中的任一者可處理將被寫入到平行處理器記憶體222內的任何記憶體單元224A-224N的資料。記憶體交

又 216 可被配置成將每個叢集 214A-214N 的輸出傳送到任何分割單元 220A-220N 或者傳送到另一叢集 214A-214N，其可對該輸出執行額外的處理操作。每個叢集 214A-214N 可透過記憶體交叉 216 與記憶體介面 218 通訊，以讀取自或寫入至各種外部記憶體裝置。在一個實施例中，記憶體交叉 216 具有至記憶體介面 218 的連接以與 I/O 單元 204 通訊，以及具有至平行處理器記憶體 222 之本地實例的連接，使得不同處理叢集 214A-214N 內的處理單元能夠與系統記憶體或不在平行處理單元 202 本地的其他記憶體通訊。在一個實施例中，記憶體交叉 216 可使用虛擬通道來分離叢集 214A-214N 和分割單元 220A-220N 之間的流量流 (traffic streams)。

【0061】雖然在平行處理器 200 內示出了平行處理單元 202 的單一實例，但可以包括平行處理單元 202 的任意數量的實例。例如，平行處理單元 202 的多個實例可被設置在單一插卡上，或者可互連多個插卡。平行處理單元 202 的不同實例可被配置成即使不同實例具有不同數量的處理核心、不同數量的本地平行處理器記憶體、及/或其他配置差異，仍可進行交互操作 (inter-operate)。例如並且在一個實施例中，平行處理單元 202 的一些實例可包括相對於其他實例更高精度的浮點單元。併入平行處理單元 202 之一或多個實例或者平行處理器 200 的系統可以各種配置和形狀因素來實現，包括但不限於桌上型、膝上型、或手持式個人電腦、伺服器、工作站、遊戲主控台、及/或嵌入

式系統。

【0062】圖 2B 是依據實施例之分割單元 220 的方塊圖。在一個實施例中，分割單元 220 是圖 2A 之分割單元 220A-220N 之其中一者的實例。如所示，分割單元 220 包括 L2 快取 221、幀緩衝器介面 225、及 ROP 226(光柵(raster)操作單元)。L2 快取 221 是讀/寫快取，其被配置成執行從記憶體交叉 216 和 ROP 226 接收到的加載及儲存操作。讀取失誤(miss)和緊急寫回請求是由 L2 快取 221 輸出到幀緩衝器介面 225 進行處理。更新也可透過幀緩衝器介面 225 被發送到幀緩衝器進行處理。在一個實施例中，幀緩衝器介面 225 透過介面與平行處理器記憶體中的記憶體單元之其中一者，諸如圖 2A 的記憶體單元 224A-224N(例如，在平行處理器記憶體 222 內)，連接。

【0063】在圖形應用中，ROP 226 是執行諸如模板、z 測試、混色等的光柵操作的處理單元。ROP 226 隨後輸出被儲存在圖形記憶體中的經處理的圖形資料。在一些實施例中，ROP 226 包括壓縮邏輯，用以壓縮被寫入到記憶體的深度或色彩資料，以及解壓縮從記憶體讀取的深度或色彩資料。壓縮邏輯可以是利用多種壓縮演算法中的一或多種的無失真壓縮邏輯。由 ROP 226 執行的壓縮類型可以依據待壓縮資料的統計特性而變化。例如，在一個實施例中，在每個圖磚的基礎上，對深度和色彩資料執行色差壓縮(delta color compression)。

【0064】在一些實施例中，ROP 226 被包括在每個處

理叢集(例如,圖2A的叢集214A-214N)內,而不是包括在分割單元220內。在這種實施例中,透過記憶體交叉216傳輸像素資料的讀取和寫入請求,而不是像素片段資料。可將經處理的圖形資料顯示在顯示裝置上,諸如圖1的一或多個顯示裝置110之其中一者,由處理器102路由用於進一步處理,或者由圖2A的平行處理器200內的處理實體的其中一者路由用於進一步處理。

【0065】圖2C是依據實施例之在平行處理單元內的處理叢集214的方塊圖。在一個實施例中,該處理叢集是圖2A的處理叢集214A-214N之其中一者的實例。處理叢集214可被配置成平行執行許多線程,其中術語「線程」是指在特定的一組輸入資料上執行的特定的程式的實例。在一些實施例中,單指令、多資料(SIMD)指令發佈(issue)技術被用來支援大量線程的平行執行而不需提供多個獨立的指令單元。在其他實施例中,單指令、多線程(SIMT)技術被用來支援大量大致上同步的線程的平行執行,其使用被配置成向每個處理叢集內的一組處理引擎發佈指令的共通指令(common instruction)單元。不同於所有處理引擎通常執行相同指令的SIMD執行體制(regime),SIMT執行允許不同的線程更容易地遵循給定線程程式中不同的執行路徑。本領域之技術人士將理解到,SIMD處理體制代表SIMT處理體制的一功能子集。

【0066】處理叢集214的操作可經由管線管理器232來控制,管線管理器232將處理任務分配給SIMT平行處理

器。管線管理器 232 從圖 2A 的排程器 210 接收指令，並且經由圖形多處理器 234 及 / 或紋理單元 236 來管理那些指令的執行。所示的圖形多處理器 234 是 SIMT 平行處理器的一示例性實例。然而，處理叢集 214 內可包括不同架構的各種類型的 SIMT 平行處理器。處理叢集 214 內可包括圖形多處理器 234 的一或多個實例。圖形多處理器 234 可處理資料，並且資料交叉 240 可被用來將經處理的資料分發給多個可能的目的地之一，包括其他的著色 (shader) 單元。管線管理器 232 可藉由指定要透過資料交叉 240 分發的經處理的資料的目的地來促進經處理的資料的分發。

【0067】處理叢集 214 內的每個圖形多處理器 234 可包括相同的一組功能執行邏輯 (例如，算術邏輯單元、加載-儲存單元等)。可以管線式方式來配置功能執行邏輯，在該管線式方式中，在完成先前的指令之前可以發佈新的指令。功能執行邏輯支援各種操作，包括整數及浮點運算、比較運算、布林 (Boolean) 運算、位元位移、以及各種代數函數計算。在一個實施例中，可利用相同的功能單元硬體來執行不同的運算，並且可存在功能單元的任意組合。

【0068】發送到處理叢集 214 的指令構成線程。在該組平行處理引擎上執行的一組線程是一線程組 (thread group)。線程組對不同的輸入資料執行相同的程式。線程組內的每個線程可被分配給圖形多處理器 234 內的不同處理引擎。線程組可包括比圖形多處理器 234 內的處理引擎數量更少的線程。當線程組包括比處理引擎數量更少的線

程時，在處理該線程組的週期期間，處理引擎之其中一或多者可以是閒置的。線程組還可包括比圖形多處理器 234 內的處理引擎數量更多的線程。當線程組包括比圖形多處理器 234 內的處理引擎數量更多的線程時，可在連續的時脈週期執行處理。在一個實施例中，多個線程組可以同時在圖形多處理器 234 上執行。

【0069】在一個實施例中，圖形多處理器 234 包括內部快取記憶體，用以執行加載和儲存操作。在一個實施例中，圖形多處理器 234 可放棄(forego)內部快取，並使用處理叢集 214 內的快取記憶體(例如，L1 快取 248)。每個圖形多處理器 234 還可以存取在所有處理叢集 214 之間共用並且可被用來在線程之間傳輸資料的分割單元(例如，圖 2A 的分割單元 220A-220N)內的 L2 快取。圖形多處理器 234 也可存取晶片外(off-chip)全域記憶體，其可包括本地平行處理器記憶體及/或系統記憶體中之一或多者。平行處理單元 202 外部的任何記憶體可被用作全域記憶體。處理叢集 214 包括圖形多處理器 234 之多個實例的實施例可共享共同的指令和資料，其可被儲存在 L1 快取 248 中。

【0070】每個處理叢集 214 可包括 MMU 245(記憶體管理單元)，其被配置成將虛擬位址映射到實體位址。在其他實施例中，MMU 245 的一或多個實例可常駐在圖 2A 的記憶體介面 218 內。MMU 245 包括用於將虛擬位址映射到圖磚之實體位址的一組分頁表項目(page table entries, PTEs)、以及可選地一快取線索引。MMU 245 可包括位址

轉譯後備緩衝區(translation lookaside buffer, TLB)或者可常駐在圖形多處理器 234 或 L1 快取或處理叢集 214 內的快取。處理實體位址來分發表面資料存取區域性(locality)，以允許在分割單元之間的高效請求交錯。快取線索引可被用來確定對快取線的請求是否命中或是未命中。

【0071】在圖形及計算應用中，可配置處理叢集 214 使得每個圖形多處理器 234 耦接到紋理單元 236 以執行紋理映射操作，例如，確定紋理樣本位置、讀取紋理資料、及過濾紋理資料。紋理資料是讀取自內部紋理 L1 快取(未示出)或者在一些實施例中是讀取自圖形多處理器 234 內的 L1 快取，並且依據需要提取自 L2 快取、本地平行處理器記憶體、或系統記憶體。每個圖形多處理器 234 將經處理的任務輸出到資料交叉 240，以將經處理的任務提供給另一處理叢集 214 用於進一步處理，或者透過記憶體交叉 216 將經處理的任務儲存在 L2 快取、本地平行處理器記憶體、或系統記憶體中。preROP 242(預光柵(pre-raster)操作單元)被配置成從圖形多處理器 234 接收資料、將資料引導至 ROP 單元，該 ROP 單元可與可與如本文所述之分割單元(例如，圖 2A 的分割單元 220A-220N)位在一起。preROP 242 單元可執行色彩混色的最佳化、組織像素色彩資料、以及執行位址轉譯。

【0072】應當理解的是，本文所示的核心結構是示例性的，並且變形和修改是可能的。處理叢集 214 內可以包括任意數量的處理單元，例如，圖形多處理器 234、紋理

單元 236、preROP 242 等等。此外，雖然僅示出一個處理叢集 214，但如本文所述之平行處理單元可包括處理叢集 214 的任意數量的實例。在一個實施例中，每個處理叢集 214 可被配置成使用單獨的且不同的處理單元、L1 快取等來獨立於其他的處理叢集 214 進行操作。

【0073】圖 2D 示出依據一個實施例之圖形多處理器 234。在此實施例中，圖形多處理器 234 與處理叢集 214 的管線管理器 232 耦接。圖形多處理器 234 具有執行管線，包括但不限於指令快取 252、指令單元 254、位址映射單元 256、暫存器檔案 258、一或多個通用圖形處理單元 (GPGPU) 核心 262、以及一或多個加載/儲存單元 266。GPGPU 核心 262 以及加載/儲存單元 266 透過記憶體及快取互連 268 與快取記憶體 272 以及共用記憶體 270 耦接。

【0074】在一個實施例中，指令快取 252 從管線管理器 232 接收要執行的指令流。該些指令被快取儲存在指令快取 252 中，並且由指令單元 254 分派執行。指令單元 254 可將指令分派為線程組 (例如，執行包 (warp))，線程組的每個線程被分配給 GPGPU 核心 262 內的不同執行單元。指令可藉由在統一位址空間內指定位址來存取本地、共用、或全域位址空間中的任一者。位址映射單元 256 可被用來將統一位址空間中的位址轉譯成可由加載/儲存單元 266 存取的不同記憶體位址。

【0075】暫存器檔案 258 提供一組暫存器給圖形多處理器 234 的功能單元。暫存器檔案 258 為連接至圖形多處理

器 234 的功能單元(例如，GPGPU 核心 262、加載/儲存單元 266)的資料路徑的運算元提供暫時的儲存。在一個實施例中，暫存器檔案 258 在每個功能單元之間被劃分，使得每個功能單元被分配有暫存器檔案 258 的專用部分。在一個實施例中，暫存器檔案 258 在由圖形多處理器 234 執行的不同執行包之間被劃分。

【0076】每個 GPGPU 核心 262 可包括被用來執行圖形多處理器 234 之指令的浮點單元(FPU)及/或整數算術邏輯單元(ALU)。依據實施例，GPGPU 核心 262 可以在結構上相似或者可以在結構上不同。例如且在一個實施例中，GPGPU 核心 262 的第一部分包括單精度 FPU 及整數 ALU，而 GPGPU 核心的第二部分包括雙精度 FPU。在一個實施例中，FPU 可實現用於浮點運算的 IEEE 754-2008 標準或者啟用可變精度浮點運算。圖形多處理器 234 可另外包括一或多個固定功能或特殊功能單元以執行特定功能，諸如複製矩形或像素混色操作。在一個實施例中，GPGPU 核心中的一或多個還可包括固定或特殊功能邏輯。

【0077】在一個實施例中，GPGPU 核心 262 包括能夠對多組資料執行單指令的 SIMD 邏輯。在一個實施例中，GPGPU 核心 262 可實體上執行 SIMD4、SIMD8、及 SIMD16 指令，並且邏輯上執行 SIMD1、SIMD2、及 SIMD32 指令。用於 GPGPU 核心的 SIMD 指令可由著色編譯器在編譯時產生，或者當執行針對單程式多資料(SPMD)或 SIMT 結構所編寫和編譯的程式時自動地產生。可透過單一的 SIMD 指

令執行被配置用於SIMT執行模型之程式的多個線程。例如並且在一個實施例中，可透過單一的SIMD8邏輯單元平行地執行八個SIMT線程，該八個SIMT線程執行相同或相似的操作。

【0078】記憶體及快取互連268是一互連網路，其將圖形多處理器234的每個功能單元連接到暫存器檔案258及連接到共用記憶體270。在一個實施例中，記憶體及快取互連268是一交叉互連(crossbar interconnect)，其允許加載/儲存單元266在共用記憶體270和暫存器檔案258之間實現加載和儲存操作。暫存器檔案258可以與GPGPU核心262相同的頻率操作，因此GPGPU核心262和暫存器檔案258之間的資料傳輸延遲非常低。共用記憶體270可被用來啟用在圖形多處理器234內的功能單元上執行的線程之間的通訊。快取記憶體272可被用作資料快取，例如，以快取儲存在功能單元和紋理單元236之間通訊的紋理資料。共用記憶體270還可被用作管理快取的程式。除了自動快取儲存已被儲存在快取記憶體272內的資料之外，在GPGPU核心262上執行的線程還可以程式設計方式將資料儲存在共用記憶體內。

【0079】圖3A-3B示出依據實施例之另外的圖形多處理器。所示圖形多處理器325、350是圖2C的圖形多處理器234的變形。所示圖形多處理器325、350可被配置為能夠同時執行大量執行線程的串流多處理器(streaming multiprocessor, SM)。

【0080】圖3A示出依據另外的實施例之圖形多處理器325。圖形多處理器325包括關於圖2D之圖形多處理器234的執行資源單元的多個額外的實例。例如，圖形多處理器325可包括指令單元332A-332B、暫存器檔案334A-334B、及紋理單元344A-344B的多個實例。圖形多處理器325還包括多組圖形或計算執行單元(例如，GPGPU核心336A-336B、GPGPU核心337A-337B、GPGPU核心338A-338B)及多組加載/儲存單元340A-340B。在一個實施例中，執行資源單元具有共通指令快取330、紋理及/或資料快取記憶體342、以及共用記憶體346。

【0081】多種組件可以透過互連結構(interconnect fabric)327通訊。在一個實施例中，互連結構327包括一或多個交叉開關(crossbar switch)以致能圖形多處理器325的各種組件之間的通訊。在一個實施例中，互連結構327是單獨的高速網路結構層，其上堆疊了圖形多處理器325的每個組件。圖形多處理器325的組件透過互連結構327與遠端組件通訊。例如，GPGPU核心336A-336B、337A-337B、及338A-338B之各者可透過互連結構327與共用記憶體346通訊。互連結構327可仲裁圖形多處理器325內的通訊以確保組件之間的公平頻寬分配。

【0082】圖3B示出依據另外的實施例之圖形多處理器350。圖形處理器包括多組執行資源356A-356D，其中各組執行資源包括多個指令單元、暫存器檔案、GPGPU核心、及加載儲存單元，如圖2D及圖3A中所示。執行資源

356A-356D可與紋理單元360A-360D合作用於紋理操作，同時共享指令快取354及共用記憶體362。在一個實施例中，執行資源356A-356D可共享指令快取354及共用記憶體362，以及紋理及/或資料快取記憶體358A-358B的多個實例。各種組件可透過類似圖3A之互連結構327的互連結構352通訊。

【0083】本領域之技術人士將理解到，在圖1、2A-2D及3A-3B中描述的結構是描述性的而非限制本實施例的範圍。因此，本文所描述的技術可以在任何適當配置的處理單元上實現，包括但不限於，一或多個行動應用處理器、包括多核中央處理單元(CPU)的一或多個桌上型或伺服器CPU、一或多個平行處理單元，諸如圖2A的平行處理單元202，以及一或多個圖形處理器或專用處理單元，而不背離本文所述之實施例的範圍。

【0084】在一些實施例中，如本文所述的平行處理器或GPGPU通訊地耦接至主機/處理器核心以加速圖形運算、機器學習運算、型樣分析運算、及各種通用GPU(GPGPU)功能。GPU可以透過匯流排或其他互連(例如，諸如PCIe或NVLink的高速互連)通訊地耦接至主機處理器/核心。在其他實施例中，GPU可以與核心整合在相同的封裝或晶片上，並且透過內部處理器匯流排/互連(即，封裝或晶片的內部)通訊地耦接至核心。不管GPU連接的方式如何，處理器核心都可以工作描述符中所包含的命令/指令序列的形式，將工作分配給GPU。然後GPU使用

專用電路/邏輯來有效地處理這些命令/指令。

GPU至主處理器(Host Processor)互連的技術

【0085】圖4A示出其中複數個GPU 410-413透過高速鏈路440A-440D(例如，匯流排、點對點互連等等)通訊地耦接至複數個多核心處理器405-406的示例性結構。在一個實施例中，高速鏈路440A-440D依據實施狀況支援4GB/s、30GB/s、80GB/s或更高的通訊通量。可使用各種互連協定，包括但不限於PCIe 4.0或5.0以及NVLink 2.0。然而，本發明的基本原理不限於任何特定的通訊協定或通量。

【0086】此外，在一個實施例中，GPU 410-413中的兩個或更多個透過高速鏈路442A-442B互連，其可使用與用於高速鏈路440A-440D之協定/鏈路相同或不同的協定/鏈路來實現。類似地，多核心處理器405-406中的兩個或更多個可透過高速鏈路443連接，高速鏈路443可以是以20GB/s、30GB/s、120GB/s或更高速率運行的對稱多處理器(SMP)匯流排。替代地，可使用相同的協定/鏈路(例如，透過常見的互連結構)來完成圖4A中所示的各種系統組件之間的所有通訊。然而，如上所述，本發明的基本原理不限於任何特定類型的互連技術。

【0087】在一個實施例中，每個多核心處理器405-406分別透過記憶體互連430A-430B通訊地耦接至處理器記憶體401-402，並且每個GPU 410-413分別透過GPU記憶體

互連 450A-450D 通訊地耦接至 GPU 記憶體 420-423。記憶體互連 430A-430B 及 450A-450D 可利用相同或不同的記憶體存取技術。透過舉例而非限制，處理器記憶體 401-402 及 GPU 記憶體 420-423 可以是揮發性記憶體，諸如動態隨機存取記憶體 (DRAM)(包括堆疊式 DRAM)、圖形 DDR SDRAM(GDDR)(例如，GDDR5、GDDR6)、或高頻寬記憶體 (HBM)，及/或可以是非揮發性記憶體，諸如 3D XPoint 或奈米隨機記憶體 (Nano-Ram)。在一個實施例中，記憶體的一些部分可以是揮發性記憶體，而另一部分可以是非揮發性記憶體(例如，使用二階記憶體 (2LM) 階層)。

【0088】如下所述，雖然各種處理器 405-406 及 GPU 410-413 可以分別實體地耦接至特定的記憶體 401-402、420-423，但也可實現統一的記憶體結構，其中相同的虛擬系統位址空間(也稱為「有效位址」空間)分佈在所有各種實體記憶體中。例如，處理器記憶體 401-402 之各者可包含 64GB 的系統記憶體位址空間，以及 GPU 記憶體 420-423 之各者可包含 32GB 的系統記憶體位址空間(在此範例中導致總共 256GB 的可定址記憶體)。

【0089】圖 4B 示出依據一個實施例之多核心處理器 407 及圖形加速模組 446 之間的互連的額外細節。圖形加速模組 446 可包括一或多個 GPU 晶片，其整合在透過高速鏈路 440 耦接到處理器 407 的線路卡上。替代地，圖形加速模組 446 可以與處理器 407 整合在相同的封裝或晶片上。

【0090】所示的處理器 407 包括複數個核心 460A-

460D，每個核心具有轉譯後備緩衝區 461A-461D 及一或多個快取 462A-462D。核心可包括用於執行指令和處理資料的各種其他組件，為了避免模糊本發明的基本原理而未示出(例如，指令提取單元、分支預測單元、解碼器、執行單元、重排序緩衝器等)。快取 462A-462D 可包含第 1 階(L1)及第 2 階(L2)快取。此外，一或多個共用快取 456 可被包括在快取階層中，並且由多組核心 460A-460D 共用。例如，處理器 407 的一個實施例包括 24 個核心，每個核心具有其自己的 L1 快取、12 個共用 L2 快取、以及 12 個共用 L3 快取。在此實施例中，L2 及 L3 快取的其中一者由兩個相鄰核心共用。處理器 407 及圖形加速器積體模組 446 與系統記憶體 441 連接，系統記憶體 441 可包括處理器記憶體 401-402。

【0091】透過在一致性匯流排 464 上的核心間通訊來保持儲存在各種快取 462A-462D、456 及系統記憶體 441 中的資料和指令的一致性。例如，每個快取可具有與其相關聯的快取一致性邏輯/電路，以回應偵測到的對特定快取線的讀取或寫入而在一致性匯流排 464 上進行通訊。在一個實施方式中，在一致性匯流排 464 上實現快取監視協定(snooping protocol)以監視快取存取。快取監視/一致性技術已為本領域之技術人士所完全理解，為了避免模糊本發明的基本原理，此處不再詳細描述。

【0092】在一個實施例中，代理(proxy)電路 425 將圖形加速模組 446 通訊地耦接至一致性匯流排 464，允許圖形

加速模組 446 作為核心的同級(peer)參與快取一致性協定。特別是，介面 435 提供透過高速鏈路 440 (例如，PCIe 匯流排、NVLink 等) 到代理電路 425 的連接，且介面 437 將圖形加速模組 446 連接到高速鏈路 440。

【0093】在一個實施方式中，加速器整合電路 436 代表圖形加速模組 446 的複數個圖形處理引擎 431、432、N 提供快取管理、記憶體存取、上下文管理、及中斷管理服務。圖形處理引擎 431、432、N 之各者可包括單獨的圖形處理單元(GPU)。替代地，圖形處理引擎 431、432、N 可包括在 GPU 內的不同類型的圖形處理引擎，諸如圖形執行單元、媒體處理引擎(例如，視訊編碼器/解碼器)、取樣器、及區塊影像轉移(blit)引擎。換句話說，圖形加速模組可以是具有複數個圖形處理引擎 431-432、N 的 GPU 或者圖形處理引擎 431-432、N 可以是整合在共通封裝、線路卡或晶片上的個別 GPU。

【0094】在一個實施例中，加速器整合電路 436 包括記憶體管理單元(MMU)439，用於執行各種記憶體管理功能，諸如用於存取系統記憶體 441 的虛擬至實體記憶體轉譯(也被稱為有效至真實記憶體轉譯)以及記憶體存取協定。MMU 439 亦可包括轉譯後備緩衝區(TLB)(未示出)用於快取虛擬/有效至實體/真實位址轉譯。在一個實施方式中，快取 438 儲存命令何資料以供圖形處理引擎 431-432、N 進行高效存取。在一個實施例中，儲存在快取 438 和圖形記憶體 433-434、M 中的資料與核心快取 462A-462D、456

和系統記憶體 411 保持一致。如上所述，這可透過代表快取 438 及記憶體 433-434、M 參與快取一致性機制的代理電路 425 來完成(例如，發送關於處理器快取 462A-462D、456 上的快取線的修改/存取的更新至快取 438，並且從快取 438 接收更新)。

【0095】一組暫存器 445 儲存由圖形處理引擎 431-432、N 執行的線程的上下文資料，並且上下文管理電路 448 管理該線程上下文。例如，上下文管理電路 448 可執行保存(save)和恢復(restore)操作，以在上下文切換期間保存和恢復各種線程的上下文(例如，其中保存第一線程以及儲存第二線程，從而該第二線程可被圖形處理引擎執行)。例如，在上下文切換時，上下文管理電路 448 可將當前的暫存器值儲存到(例如，由上下文指標所標識的)記憶體中的指定區域。然後當返回上下文時，其可恢復暫存器值。在一個實施例中，中斷管理電路 447 接收並且處理從系統裝置接收的中斷。

【0096】在一個實施方式中，由 MMU 439 將來自圖形處理引擎 431 的虛擬/有效位址轉譯成在系統記憶體 411 中的真實/實體位址。加速器整合電路 436 的一個實施例支援多個(例如，4、8、16 個)圖形加速器模組 446 及/或其他加速器裝置。圖形加速器模組 446 可專用於在處理器 407 上執行的單一應用，或者可在多個應用之間被共用。在一個實施例中，呈現虛擬化圖形執行環境，其中圖形處理引擎 431-432、N 的資源與多個應用程式或者虛擬機器(VM)共

用。資源可被細分為「片(slices)」，其依據與VM及/或應用程式相關聯的處理要求和優先權而被分配給不同的VM及/或應用程式。

【0097】因此，加速器整合電路充當圖形加速模組446至系統的橋樑，並提供位址轉譯和系統記憶體快取服務。此外，加速器整合電路436可為主處理器提供虛擬化設施，以管理圖形處理引擎的虛擬化、中斷、和記憶體管理。

【0098】因為圖形處理引擎431-432、N的硬體資源明確地被映射到由主處理器407看到的真實位址空間，所以任何主處理器可以使用有效位址值來直接定址這些資源。在一個實施例中，加速器整合電路436的一個功能是圖形處理引擎431-432、N的實體分離，使得它們作為獨立的單元出現在系統。

【0099】如所述，在所示實施例中，一或多個圖形記憶體433-434、M分別耦接至圖形處理引擎431-432、N之各者。圖形記憶體433-434、M儲存要由圖形處理引擎431-432、N之各者處理的指令及資料。圖形記憶體433-434、M可以是揮發性記憶體，諸如DRAM(包括堆疊式DRAM)、GDDR記憶體(例如，GDDR5、GDDR6)、或HBM，及/或可以是非揮發性記憶體，諸如3D XPoint或奈米隨機記憶體。

【0100】在一個實施例中，為了減少高速鏈路440上的資料流量，使用偏置(biasing)技術來確保儲存在圖形記

記憶體 433-434、M 中的資料是圖形處理引擎 431-432、N 將會最常使用、且較佳地不會被核心 460A-460D 使用(至少不頻繁)的資料。類似的，偏置機制試圖將核心(並且較佳地不是圖形處理引擎 431-432、N)所需的資料保持在核心的快取 462A-462D、456 及系統記憶體 411 內。

【0101】圖 4C 示出加速器整合電路 436 被整合在處理器 407 內的另一實施例。在此實施例中，圖形處理引擎 431-432、N 經由介面 437 及介面 435(其也可以利用任何形式的匯流排或介面協定)透過高速鏈路 440 直接通訊至加速器整合電路 436。加速器整合電路 436 可執行與關於圖 4B 所描述的操作相同的操作，但考慮到其與一致性匯流排 464 及快取 462A-462D、456 非常靠近，可能以較高的通量進行操作。

【0102】一個實施例支援不同的程式設計模型，包括專用程序(dedicated-process)程式設計模型(沒有圖形加速模組虛擬化)以及共用程式設計模型(有虛擬化)。後者可包括由加速器整合電路 436 控制的程式設計模型以及由圖形加速模組 446 控制的程式設計模型。

【0103】在專用程序模型的一個實施例中，圖形處理引擎 431-432、N 專用於單一作業系統下的單一應用程式或程序。單一應用程式可將其他的應用程式請求彙入圖形處理引擎 431-432、N，在 VM/分區內提供虛擬化。

【0104】在專用程序程式設計模型中，圖形處理引擎 431-432、N 可被多個 VM/應用程式分區共用。共用模型需

要系統超管理器來虛擬化圖形處理引擎431-432、N，以允許每個作業系統存取。針對沒有超管理器的單分區系統，由作業系統擁有圖形處理引擎431-432、N。在兩種情況中，作業系統都可虛擬化圖形處理引擎431-432、N來提供對每個程序或應用程式的存取。

【0105】針對共用程式設計模型，圖形加速模組446或個別的圖形處理引擎431-432、N使用處理處置(handle)來選擇處理元件。在一個實施例中，處理元件被儲存在系統記憶體411中，並且可使用本文所述之有效位址至真實位址轉譯技術來定址。處理處置可以是當主程序向圖形處理引擎431-432、N註冊其上下文時被提供給主程序之特定於實施方式的值(即，呼叫系統軟體將程序元件加入程序元件鏈接表)。處理處置的低16位元可以是程序元件鏈接表內的程序元件的偏移。

【0106】圖4D示出示例性加速器整合片(integration slice)490。如本文所使用的，「片(slice)」包含加速器整合電路436的處理資源的指定部分。系統記憶體411內的應用程式有效位址空間482儲存程序元件483。在一個實施例中，回應於來自在處理器407上執行的應用程式480的GPU調用(invocation)481而儲存程序元件483。程序元件483包含相對應的應用程式480的程序狀態。程序元件483中包含的工作描述符(WD)484可以是應用程式所請求的單一工作(job)或者可包含指向工作佇列的指標。在後者情況中，WD 484是指向在應用程式的位址空間482中的工作請求佇

列的指標。

【0107】圖形加速模組446及/或個別的圖形處理引擎431-432、N可由系統中的所有程序或程序的子集共用。本發明的實施例包括用於建立程序狀態以及將WD 484發送至圖形加速模組446以在虛擬環境中開始工作的基礎設施。

【0108】在一個實施方式中，專用程序程式設計模型是特定於實施方式的。在此模型中，單一程序擁有圖形加速模組446或個別的圖形處理引擎431。由於圖形加速模組446是屬於單一程序的，因此超管理器(hypervisor)為所屬分區(partition)初始化加速器整合電路436，以及作業系統在圖形加速模組446被分配時，為所屬程序初始化加速器整合電路436。

【0109】在操作中，加速器整合片490中的WD提取單元491提取下一個WD 484，其包括待由圖形加速模組446之圖形處理引擎中之一者完成的工作的指示。來自WD 484的資料可被儲存在暫存器445中，並且由MMU 439、中斷管理電路447及/或上下文管理電路448使用，如所示出的。舉例來說，MMU 439的一個實施例包括區段(segment)/頁面步行(page walk)電路，用於存取OS虛擬位址空間485內的區段/頁表486。中斷管理電路447可處理接收自圖形加速模組446的中斷事件492。當執行圖形操作時，由圖形處理引擎431-432、N產生的有效位址493透過MMU 439被轉譯成真實位址。

【0110】在一個實施例中，相同組的暫存器445係針對每個圖形處理引擎431-432、N及/或圖形加速模組446被複製，並且可由超管理器或作業系統初始化。這些複製的暫存器之各者可被包含在加速器整合片490中。表1示出可由超管理器初始化的示例性暫存器。

表 1 – 超管理器初始化的暫存器

1	片控制暫存器
2	真實位址(RA)經排程程序區域指標
3	權限遮罩置換暫存器
4	中斷向量表進入點偏移(Entry Offset)
5	中斷向量表進入點限制(Entry Limit)
6	狀態暫存器
7	邏輯分區ID
8	真實位址(RA)超管理器加速器利用記錄指標
9	儲存描述暫存器

【0111】表2示出可由作業系統初始化的示例性暫存器。

表 2 – 作業系統初始化的暫存器

1	程序及線程標識
2	有效位址(EA)上下文保存/恢復指標
3	虛擬位址(VA)加速器利用記錄指標
4	虛擬位址(VA)儲存分段表指標
5	權限遮罩
6	工作描述符

【0112】在一個實施例中，每個WD 484係特定於特定的圖形加速模組446及/或圖形處理引擎431-432、N。它包含圖形處理引擎431-432、N完成其工作所需的所有資訊，或者它可以是指向應用程式已經建立待完成之工作的命令佇列的記憶體位置的指標。

【0113】圖4E示出共用模型的一個實施例的額外細節。此實施例包括超管理器真實位址空間498，其中儲存有程序元件表499。透過超管理器496可存取超管理器真實位址空間498，超管理器496為作業系統495虛擬化圖形加速模組。

【0114】共用程式設計模型允許來自系統中的所有分割或分割的子集的所有程序或程序的子集使用圖形加速模組446。存在兩種程式設計模型，其中圖形加速模組446由多個程序和分割共用：時間切片(time-sliced)的共用以及圖形指向的共用。

【0115】在此模型中，系統超管理器496擁有圖形加速模組446，並且使其之功能為所有作業系統495可用。對於透過系統超管理器496來支援虛擬化的圖形加速模組446，圖形加速模組446可遵循以下要求：(1)應用程式的工作請求必須是自主的(即，不需要在工作之間維護狀態)，或者圖形加速模組446必須提供上下文保存及恢復機制。(2)應用程式的工作請求由圖形加速模組446保證以在指定的時間內完成，包括任何轉譯錯誤，或者圖形加速模組446提供搶占工作處理的能力。(3)當在指向的共用程式

設計模型中操作時，圖形加速模組 446 必須保證程序之間的公平性。

【0116】在一個實施例中，針對共用模型，應用程式 480 被要求以圖形加速模組 446 類型、工作描述符 (WD)、權限遮罩暫存器 (AMR) 值、及上下文保存/恢復區域指標 (CSRP) 來進行作業系統 495 系統呼叫。圖形加速模組 446 類型描述用於系統呼叫的目標加速函數。圖形加速模組 446 類型可以是系統特定的值。WD 係特別針對圖形加速模組 446 被格式化，並且可以是圖形加速模組 446 命令、指向使用者定義的結構的有效位址指標、指向命令佇列的有效位址指標、或者描述待由圖形加速模組 446 完成之工作的任何其他資料結構的形式。在一個實施例中，AMR 值是用於當前程序的 AMR 狀態。傳送給作業系統的該值類似於設定 AMR 的應用程式。若加速器整合電路 436 及圖形加速模組 446 實施方式不支援使用者權限遮罩置換暫存器 (User Authority Mask Override Register, UAMOR)，則作業系統可在超管理器呼叫中，在傳送 AMR 之前將當前的 UAMOR 值應用於 AMR 值。在將 AMR 放入程序元件 483 之前，超管理器 496 可以可選地應用當前的權限遮罩置換暫存器 (AMOR) 值。在一個實施例中，CSRP 是暫存器 445 之其中一者，其包含應用程式的位址空間 482 中的區域的有效位址，以供圖形加速模組 446 保存及恢復上下文狀態。若不需要在工作之間保存狀態或者當工作被搶占時，則此指標是可選的。上下文保存/恢復區域可以是固定的 (pinned) 系

統記憶體。

【0117】一旦接收到系統呼叫，作業系統495可驗證應用程式480已註冊並且被授權使用圖形加速模組446。接著，作業系統495使用表3中所示的資訊呼叫超管理器496。

表 3 – OS至超管理器呼叫參數

1	工作描述符(WD)
2	權限遮罩暫存器(AMR)值(可能被遮蔽)
3	有效位址(EA)上下文保存/恢復區域指標(CSRP)
4	程序ID(PID)及可選的線程ID(TID)
5	虛擬位址(VA)加速器利用記錄指標(AURP)
6	儲存分段表指標(SSTP)的虛擬位址
7	邏輯中斷服務號碼(LISN)

【0118】一旦接收到超管理器呼叫，超管理器496驗證作業系統495已註冊並且被授權使用圖形加速模組446。接著，超管理器496將程序元件483放入用於相應的圖形加速模組446類型的程序元件鏈接表中。程序元件可包括表4中所示的資訊。

表 4 – 程序元件資訊

1	工作描述符(WD)
2	權限遮罩暫存器(AMR)值(可能被遮蔽)
3	有效位址(EA)上下文保存/恢復區域指標(CSRP)
4	程序ID(PID)及可選的線程ID(TID)
5	虛擬位址(VA)加速器利用記錄指標(AURP)
6	儲存分段表指標(SSTP)的虛擬位址
7	邏輯中斷服務號碼(LISN)
8	中斷向量表，源自超管理器呼叫參數
9	狀態暫存器(SR)值
10	邏輯分區ID(LPID)
11	真實位址(RA)超管理器加速器利用記錄指標
12	儲存描述符暫存器(SDR)

【0119】在一個實施例中，超管理器初始化複數個加速器整合片490的暫存器445。

【0120】如圖4F中所示，本發明的一個實施例採用統一的記憶體，可透過用以存取實體處理器記憶體401-402及GPU記憶體420-423的共通虛擬記憶體位址空間來定址。在此實施方式中，GPU 410-413上執行的操作利用相同的虛擬/有效記憶體位址空間來存取處理器記憶體401-402，且反之亦然，藉此簡化可編程性。在一個實施例中，虛擬/有效記憶體位址空間的第一部分被分配給處理器記憶體401，第二部分被分配給第二處理器記憶體402，第三部分被分配給GPU記憶體420等等。整個虛擬/有效記

記憶體空間(有時稱為有效位址空間)藉此被分散在處理器記憶體 401-402 及 GPU 記憶體 420-423 之各者，允許任何處理器或 GPU 存取任何實體記憶體，其透過映射到該記憶體的虛擬位址。

【0121】在一個實施例中，MMU 439A-439E 之其中之一或多者內的偏置/一致性管理電路 494A-494E 確保主處理器(例如，405)和 GPU 410-413 之快取之間的快取一致性，並且實現指示應儲存某些類型之資料的實體記憶體的偏置技術。雖然圖 4F 中示出偏置/一致性管理電路 494A-494E 的多個實例，但偏置/一致性電路可在一或多個主處理器 405 的 MMU 內及/或在加速器整合電路 436 內實現。

【0122】一個實施例允許 GPU 附加的記憶體 420-423 被映射為系統記憶體的一部分，並且使用共用虛擬記憶體(SVM)技術來存取，但不會遭受到與完整系統快取一致性相關聯的典型的性能缺陷。GPU 附加的記憶體 420-423 作為系統記憶體被存取而不會造成繁重的快取一致性負擔的能力為 GPU 卸載提供了有利的操作環境。這種配置允許主處理器 405 軟體設置運算元並存取計算結果，而沒有傳統 I/O DMA 資料副本的負擔。這種傳統副本涉及驅動程式呼叫、中斷及記憶體映射 I/O (MMIO) 存取，相對於簡單的記憶體存取而言這些都是低效率的。同時，存取 GPU 附加的記憶體 420-423 而沒有快取一致性負擔的能力對於卸載計算的執行時間是非常重要的。例如，在具有相當大的串流寫入記憶體流量的情況下，快取一致性負擔可以顯著地降

低 GPU 410-413所看見的有效寫入頻寬。運算元設置的效率、結果存取的效率、以及 GPU計算的效率均在確定 GPU卸載的效益中發揮作用。

【0123】在一個實施方式中，由偏置追蹤器資料結構驅動 GPU偏置及主處理器偏置之間的選擇。例如，可使用偏置表，其可以是每 GPU附加的記憶體頁面包括 1或 2個位元的頁面粒度 (page-granular)結構 (即，以記憶體頁面的粒度控制)。偏置表可在一或多個 GPU附加的記憶體 420-423的遭竊 (stolen)的記憶體範圍中實現，GPU 410-413中具有或不具有偏置快取 (例如，用以快取頻繁/最近使用的偏置表的進入點)。替代地，可將整個偏置表保持在 GPU內。

【0124】在一個實施方式中，在實際存取 GPU記憶體之前，存取與至 GPU附加的記憶體 420-423的每個存取相關聯的偏置表進入點，從而導致以下操作。首先，將來自 GPU 410-413之在 GPU偏置中尋找其頁面的本地請求直接轉送到相對應的 GPU記憶體 420-423。將來自 GPU之在主機偏置中尋找其頁面的本地請求轉送到處理器 405 (例如，透過如上所述的高速鏈路)。在一個實施例中，來自處理器 405之在主處理器偏置中尋找請求頁面的請求如同正常記憶體讀取般完成該請求。替代地，可將指向 GPU偏置頁面的請求轉送到 GPU 410-413。接著，若 GPU當前未使用該頁面，則可將該頁面轉移到主處理器偏置。

【0125】頁面的偏置狀態可以透過基於軟體的機制、硬體輔助之基於軟體的機制來改變，或者針對有限的狀

況，可透過純基於硬體的機制來改變。

【0126】改變偏置狀態的一個機制採用 API 呼叫(例如，OpenCL)，其依次呼叫 GPU 的裝置驅動程式，該驅動程式依次向 GPU 傳送訊息(或將命令描述符排入佇列)，引導其改變偏置狀態，且針對某些轉換，在主機中執行快取排清(flushing)操作。快取排清操作對於從主處理器 405 偏置轉換到 GPU 偏置是必須的，但對於相反的轉換是不需要的。

【0127】在一個實施例中，藉由暫時性地呈現主處理器 405 不可快取的 GPU 偏置頁面來維持快取一致性。為了存取這些頁面，處理器 405 可請求來自 GPU 410 的存取，其取決於實現方式可以或者不可以立即授權存取。因此，為了減少處理器 405 和 GPU 410 之間的通訊，確保 GPU 偏置頁面是 GPU 所需要的頁面而不是主處理器 405 所需要的頁面是有利的，反之亦然。

圖形處理管線

【0128】圖 5 示出依據實施例之圖形處理管線 500。在一個實施例中，圖形處理器可實現所示的圖形處理管線 500。圖形處理器可被包括在本文所述的平行處理子系統內，諸如圖 2A 的平行處理器 200，其在一個實施例中是圖 1 之平行處理器 112 的變形。各種平行處理系統可透過如本文所述的平行處理單元(例如，圖 2A 的平行處理單元 202)的一或多個實例來實現圖形處理管線 500。舉例來說，著

色器單元(例如，圖2C的圖形多處理器234)可被配置成執行頂點處理單元504、曲面細分控制處理單元508、曲面細分評估處理單元512、幾何處理單元516、及片段/像素處理單元524之其中一或多者的功能。資料組合器502、基元組合器506、514、518、曲面細分單元510、光柵器(rasterizer)522、及光柵(raster)操作單元526的功能亦可由處理叢集(例如，圖2A的處理叢集214)內的其他處理引擎以及相應的分割單元(例如，圖2A的分割單元220A-220N)執行。圖形處理管線500也可使用針對一或多種功能的專用處理單元來實現。在一個實施例中，圖形處理管線500的一或多個部分可由通用處理器(例如，CPU)內的平行處理邏輯來執行。在一個實施例中，圖形處理管線500的一或多個部分可透過記憶體介面528存取晶載記憶體(例如，如在圖2A中的平行處理器記憶體222)，記憶體介面528可以是圖2A之記憶體介面218的實例。

【0129】在一個實施例中，資料組合器502是收集用於表面和基元之頂點資料的處理單元。接著資料組合器502將包括頂點屬性的頂點資料輸出至頂點處理單元504。頂點處理單元504是可編程的執行單元，其執行頂點著色器程式，將頂點著色器程式指定的頂點資料進行打光(lighting)及轉換。頂點處理單元504讀取儲存在快取、本地或系統記憶體中用於處理頂點資料的資料，並且可被編程為將頂點資料從基於物件的座標表示轉換為世界空間座標空間或正規化的裝置座標空間。

【0130】基元組合器506的第一實例從頂點處理單元504接收頂點屬性。基元組合器506根據需要讀取儲存的頂點屬性，並且建構圖形基元以供曲面細分控制處理單元508處理。圖形基元包括三角形、線段、點、修補等等，如由各種圖形處理應用程式編程介面(API)所支援的。

【0131】曲面細分控制處理單元508將輸入頂點視為幾何修補的控制點。將該些控制點從修補的輸入表示(例如，修補的基礎)轉換成適合於曲面細分評估處理單元512進行表面評估的表示。曲面細分控制處理單元508亦可計算用於幾何修補之邊緣的曲面細分因子。曲面細分因子適用於單一邊緣，並且量化與該邊緣相關聯的視點相依的細節程度。曲面細分單元510被配置成接收用於修補之邊緣的曲面細分因子，並且將該修補曲面細分成多個幾何基元，例如線、三角形或四邊形基元，其被傳輸至曲面細分評估處理單元512。曲面細分評估處理單元512對被細分的修補的參數化座標進行操作，以為每個頂點產生與幾何基元相關聯的表面表示及頂點屬性。

【0132】基元組合器514的第二實例從曲面細分評估處理單元512接收頂點屬性，根據需要讀取儲存的頂點屬性，並建構圖形基元以供幾何處理單元516處理。幾何處理單元516是可編程的執行單元，其執行幾何著色器程式以轉換如由幾何著色器程式指定的從基元組合器514接收的圖形基元。在一個實施例中，幾何處理單元516可被編程為將圖形基元細分為一或多個新的圖形基元，並計算用

以光柵化新圖形基元的參數。

【0133】在一些實施例中，幾何處理單元516可在幾何流中添加或刪除元素。幾何處理單元516將指定新圖形基元的參數及頂點輸出至基元組合器518。基元組合器518接收來自幾何處理單元516的參數及頂點，並且建構圖形基元以供視埠縮放、剔除及剪裁單元520處理。幾何處理單元516讀取儲存在平行處理器記憶體或系統記憶體中的資料，用於處理幾何資料。視埠縮放、剔除及剪裁單元520執行剪裁(clipping)、剔除(culling)、及視埠縮放(viewport scaling)，並將經處理的圖形基元輸出到光柵器522。

【0134】光柵器522可執行深度剔除以及其他基於深度的最佳化。光柵器522亦對新圖形基元執行掃描轉換以產生片段及將那些片段和相關聯的覆蓋資料輸出到片段/像素處理單元524。片段/像素處理單元524是可編程的執行單元，其被配置以執行片段著色器程式或像素著色器程式。片段/像素處理單元524轉換接收自光柵器522的片段或像素，如片段或像素著色器程式所指定的。例如，片段/像素處理單元524可被編程為執行包括但不限於紋理映射、著色、混色、紋理校正及透視校正的操作，以產生被輸出至光柵操作單元526的經著色的片段或像素。片段/像素處理單元524可以讀取儲存在平行處理器記憶體或系統記憶體中的資料，用於處理片段資料時使用。片段或像素著色器程式可被配置成依據為處理單元配置的取樣率，以

樣本、像素、圖磚或其他粒度進行著色。

【0135】光柵操作單元526是執行包括但不限於模板、z測試、混色等光柵操作的處理單元，並且輸出像素資料作為經處理的圖形資料，以被儲存於圖形記憶體中(例如，圖2A中的平行處理器記憶體222、及/或圖1中的系統記憶體104)、以被顯示在一或多個顯示裝置110上或者用於由一或多個處理器102之其中一者或平行處理器112進一步處理。在一些實施例中，光柵操作單元526被配置成壓縮寫入記憶體的z或色彩資料，並且解壓縮讀取自記憶體的z或色彩資料。

機器學習概述

【0136】機器學習演算法是可以基於一組資料學習的演算法。機器學習演算法的實施例可被設計成在資料集內將高階抽象模型化。例如，可使用影像識別演算法來確定給定的輸入屬於數個類別中的哪個類別；回歸演算法可以輸出給定輸入的數值；以及型樣識別演算法可被用來產生經翻譯的文本或者執行文本至語音及/或語音辨識。

【0137】機器學習演算法之一示例性類型是神經網路。有許多類型的神經網路；一簡單類型的神經網路是前饋式(feedforward)網路。前饋網路可被實現為非循環圖，其中節點分層排列。通常，前饋網路拓模包括由至少一個隱藏層分隔的輸入層和輸出層。隱藏層將輸入層接收到的輸入轉換成對在輸出層中產生輸出有用的表示。網路節點

透過邊(edge)完全連接到相鄰層中的節點，但在各層內的節點之間沒有邊。在前饋網路之輸入層的節點處接收到的資料透過激勵函數(activation function)被傳送(即，“前饋”)至輸出層的節點，激勵函數基於分別與將層連接的每個邊相關聯的係數(“權重”)來計算網路中每個接續層的節點的狀態。依據正在執行的演算法所表示的特定模型，神經網路演算法的輸出可以採取各種形式。

【0138】在可以使用機器學習演算法來將特定問題模型化之前，使用訓練資料集來訓練演算法。訓練神經網路涉及選擇網路拓樸，使用一組訓練資料表示由網路模型化的問題，以及調整權重直到網路模型對訓練資料集的所有實例以最小誤差執行。舉例來說，在針對神經網路的監督式學習(supervised learning)訓練過程期間，將回應於表示訓練資料集中的實例的輸入而由網路產生的輸出與該實例之被標記「正確」的輸出進行比較，計算表示輸出與被標記的輸出之差的誤差信號，以及隨著誤差信號反向傳遞通過網路的層，調整與該些連接相關聯的權重以最小化該誤差。當從訓練資料集的實例所產生的每個輸出的誤差被最小化時，網路被認為是「經過訓練的」。

【0139】用來訓練演算法的資料集的品質會顯著地影響機器學習演算法的準確度。訓練過程可能是計算密集的，並且在傳統通用處理器上可能需要大量時間。因此，使用平行處理硬體來訓練許多種類型的機器學習演算法。這對於最佳化神經網路的訓練是特別有用的，因為在調整

神經網路中的係數時所執行的計算自然地適合於平行實現。具體而言，許多機器學習演算法和軟體應用程式已經適應於利用通用圖形處理裝置內的平行處理硬體。

【0140】圖6是機器學習軟體堆疊600的廣義圖。機器學習應用602可被配置成使用訓練資料集訓練神經網路或者被配置成使用經過訓練的深度神經網路來實現機器智慧。機器學習應用602可包括用於神經網路的訓練和推論(inference)功能及/或在部署之前可被用來訓練神經網路的專用軟體。機器學習應用602可以實現任何類型的機器智慧，包括但不限於影像識別、映射和定位、自主導航、語音合成、醫學成像、或語言翻譯。

【0141】用於機器學習應用602的硬體加速可透過機器學習框架604啟用。機器學習框架604可以提供機器學習基元庫。機器學習基元是通常由機器學習演算法執行的基本操作。若沒有機器學習框架604，機器學習演算法的開發者將需要創建和最佳化與機器學習演算法相關聯的主要計算邏輯，然後在開發新的平行處理器時重新最佳化計算邏輯。相反的，機器學習應用可被配置成使用由機器學習框架604提供的基元來執行必要的計算。示例性基元包括張量卷積(tensor convolutions)、激勵函數、及池化(pooling)，這些是訓練卷積神經網路(CNN)時執行的計算操作。機器學習框架604還可提供基元來實現由許多機器學習演算法執行的基本線性代數子程式，例如矩陣及向量運算。

【0142】機器學習框架604可處理從機器學習應用602接收到的輸入資料，並且產生適當的輸入給計算框架606。計算框架606可以將提供給GPGPU驅動器608的底層指令抽象化，以使機器學習框架604能夠透過GPGPU硬體610來利用硬體加速，而不需要機器學習框架604深入了解GPGPU硬體610的架構。此外，計算框架606可以跨各種類型及世代的GPGPU硬體610啟用針對機器學習框架604的硬體加速。

GPGPU機器學習加速

【0143】圖7示出依據實施例之高度平行通用圖形處理單元700。在一個實施例中，通用處理單元(GPGPU)700可被配置為在處理與訓練深度神經網路相關聯的計算工作量類型時特別有效。此外，GPGPU 700可被直接鏈接到GPGPU的其他實例，以創建多GPU叢集來提高特別針對深度神經網路的訓練速度。

【0144】GPGPU 700包括主機介面702，以致能與主處理器的連接。在一個實施例中，主機介面702是PCI Express介面。然而，主機介面也可以是供應商特定的通訊介面或通訊結構。GPGPU 700從主處理器接收命令，並使用全域排程器704來將與那些命令相關聯的執行線程分發給一組計算叢集706A-706H。計算叢集706A-706H共用快取記憶體708。快取記憶體708可充當計算叢集706A-706H內的快取記憶體的高階快取。

【0145】 GPGPU 700 包括透過一組記憶體控制器 712A-712B 與計算叢集 706A-H 耦接的記憶體 714A-714B。在各種實施例中，記憶體 714A-714B 可包括各種類型的記憶體裝置，包括動態隨機存取記憶體 (DRAM) 或圖形隨機存取記憶體，諸如同步圖形隨機存取記憶體 (SGRAM)，包括圖形雙倍資料速率 (GDDR) 記憶體，或 3D 堆疊式記憶體，包括但不限於高頻寬記憶體 (HBM)。

【0146】 在一個實施例中，每個計算叢集 706A-706H 包括一組圖形多處理器，諸如圖 2C 的圖形多處理器 234。計算叢集的圖形多處理器包括多種類型的整數及浮點邏輯單元，其可以在包括適合機器學習計算的精度範圍執行計算。例如且在一個實施例中，計算叢集 706A-H 之各者中的浮點單元之至少一子集可被配置成執行 16 位元或 32 位元浮點運算，而浮點單元的不同子集可被配置成執行 64 位元浮點運算。

【0147】 GPGPU 700 的多個實例可被配置成作為計算叢集操作。計算叢集用於同步和資料交換的通訊機制在各實施例間不同。在一個實施例中，GPGPU 700 的多個實例透過主機介面 702 通訊。在一個實施例中，GPGPU 700 包括 I/O 集線器 709，其將 GPGPU 700 與致能至 GPGPU 之其他實例的直接連接的 GPU 鏈路 710 耦接。在一個實施例中，GPU 鏈路 710 耦接到致能 GPGPU 700 之多個實例間的通訊和同步的專用 GPU 至 GPU 橋。在一個實施例中，GPU 鏈路 710 與高速互聯耦接以向其他 GPGPU 或平行處理器發送和

接收資料。在一個實施例中，GPGPU 700的多個實例位於分離的資料處理系統，並且透過可經由主機介面702存取的網路裝置進行通訊。在一個實施例中，GPU鏈路710可被配置成除了主機介面702之外或者作為主機介面702的替代，致能至主處理器的連接。

【0148】雖然GPGPU 700之所示配置可被配置來訓練神經網路，但一個實施例提供了可在高性能或低功耗推論平台內被配置用於部署的GPGPU 700的替代配置。在推論配置中，相對於訓練配置，GPGPU 700包括更少的計算叢集706A-H。此外，與記憶體714A-714B相關聯的記憶體技術在推論和訓練配置之間可能不同。在一個實施例中，GPGPU 700的推論配置可支援推論特定的指令。例如，推論配置可提供支援一或多個8位元整數點乘積指令，這些指令通常在已部署的神經網路推論操作期間被使用。

【0149】圖8示出依據實施例之多GPU計算系統800。多GPU計算系統800可包括透過主機介面開關804耦接至多個GPGPU 806A-806D的處理器802。在一個實施例中，主機介面開關804是將處理器802耦接至PCI Express匯流排的PCI Express開關裝置，處理器802可透過該PCI Express匯流排與該組GPGPU 806A-D通訊。多個GPGPU 806A-806D之各者可以是圖7之GPGPU 700的實例。GPGPU 806A-D可透過一組高速點對點GPU至GPU鏈路816互連。高速GPU至GPU鏈路可透過專用GPU鏈路，諸如圖7中的GPU鏈路710，連接至GPGPU 806A-806D之各者。P2P GPU鏈路816

可使 GPGPU 806A-806D 之各者之間能夠直接通訊，而不需要透過處理器 802 所連接的主機介面匯流排進行通訊。透過被引導至 P2P GPU 鏈路的 GPU 至 GPU 流量，主機介面匯流排仍然可用於系統記憶體存取或者與多 GPU 計算系統 800 的其他實例通訊，例如，透過一或多個網路裝置。雖然在所示的實施例中 GPGPU 806A-806D 透過主機介面開關 804 連接至處理器 802，但在一個實施例中，處理器 802 包括直接支援 P2P GPU 鏈路 816 並且可直接連接至 GPGPU 806A-806D。

機器學習神經網路實現

【0150】本文所述實施例所提供的計算結構可被配置成執行特別適合於訓練及部署針對機器學習之神經網路的平行處理的類型。神經網路可以概括為具有圖形關係的函數網路。如本領域所熟知的，機器學習中使用了各種類型的神經網路實現。如前所述，神經網路的一個示例性類型為前饋網路。

【0151】神經網路的第二示例性類型為卷積神經網路 (CNN)。CNN 是一種專用的前饋神經網路，用於處理具有已知、格柵狀拓樸的資料，諸如影像資料。因此，CNN 通常被用於計算視覺及影像識別應用，但它們也可被用於其他類型的型樣識別，諸如語音及語言處理。CNN 輸入層中的節點被組織為一組「過濾器」(特徵檢測器靈感來自於視網膜中發現的接受域)，並且每組過濾器的輸出被傳遞

給網路的接續層中的節點。用於CNN的計算包括將卷積數學運算應用於每個過濾器以產生該過濾器的輸出。卷積是一種專門的數學運算，由兩個函數執行以產生第三個函數，其為兩個原始函數之一的修改版本。在卷積網路術語中，卷積的第一函數可被稱為輸入，而第二函數可被稱為卷積核(convolution kernel)。該輸出可被稱為特徵圖(feature map)。例如，卷積層的輸入可以是定義輸入影像的各種色彩分量的資料的多維矩陣。卷積核可以是參數的多維矩陣，其中透過神經網路的訓練過程來調整參數。

【0152】遞迴神經網路(RNN)是前饋神經網路的家族，其包括層之間的反饋連接。RNN透過跨神經網路的不同部分共用參數資料來致能順序資料的模型化。用於RNN的結構包括週期(cycle)。週期代表變數的當前值對於其自身在未來時間的值的影響，因為來自RNN的輸出資料的至少一部分被用作處理序列中後續輸入的反饋。由於可以構成語言資料的可變性，因此這個特徵使得RNN對於語言處理特別有用。

【0153】下面描述的圖式給出了示例性的前饋、CNN、及RNN網路，並且描述了分別訓練及部署那些類型的網路的一般過程。應理解的是，這些描述對於本文所述的任何特定實施例是示例性的且非限制性的，並且所示的概念通常可應用於一般深度神經網路及機器學習技術。

【0154】上述的示例性神經網路可被用來執行深度學習。深度學習是使用深度神經網路的機器學習。深度學習

中使用的深度神經網路是由多個隱藏層組成的人工神經網路，而不是僅包括單一隱藏層的淺層神經網路。更深的神經網路通常在計算上需要更多的訓練。然而，網路的額外隱藏層致能多步驟型樣識別，其相對於淺機器學習技術導致輸出錯誤減少。

【0155】深度學習中使用的深度神經網路通常包括耦接至後端網路之執行特徵識別的前端網路，後端網路代表可以執行操作(例如，物件分類、語音識別等)的數學模型，其基於提供給該模型的特徵表示。深度學習使得機器學習能夠被執行，而無需針對模型執行手工製作的特徵工程。相反的，深度神經網路可以依據輸入資料中的統計結構或相關性來學習特徵。可以將學習到的特徵提供給數學模型，該數學模型可將檢測到的特徵圖到輸出。網路所使用的數學模型通常專門針對要執行的特定任務，並且將使用不同的模型來執行不同的任務。

【0156】一旦神經網路被結構化，可將學習模型應用於網路以訓練網路執行特定的任務。學習模型描述如何調整模型內的權重以減少網路的輸出誤差。反向傳播誤差是用來訓練神經網路的常用方法。輸入向量被呈現給網路進行處理。使用損失函數將網路的輸出與期望的輸出進行比較，並且針對輸出層中的每個神經元計算誤差值。然後將誤差值向後傳播直到每個神經元具有相關聯的誤差值，該誤差值大致代表其對原始輸出的貢獻。然後網路可使用演算法，諸如隨機梯度下降(stochastic gradient descent)演算

法，從這些誤差中學習以更新神經網路的權重。

【0157】圖9A-9B示出示例性卷積(convolutional)神經網路。圖9A示出CNN內的各種層。如圖9A中所示，被用來模型化影像處理的示例性CNN可以接收描述輸入影像之紅色、綠色、及藍色(RGB)分量的輸入902。輸入902可以由多個卷積層(例如，卷積層904、卷積層906)處理。來自多個卷積層的輸出可以可選地由一組完全連接層908處理。完全連接層中的神經元與前一層中的所有激勵(activation)有完全的連接，如先前針對前饋網路所描述的。來自完全連接層908的輸出可被用來產生出自網路的輸出結果。可以使用矩陣乘法而不是卷積來計算完全連接層908內的激勵。不是所有的CNN實現都利用完全連接層908。例如，在一些實現中，卷積層906可針對CNN產生輸出。

【0158】卷積層是稀疏連接的，這不同於完全連接層908中可見的傳統神經網路配置。傳統神經網路層是完全連接的，使得每個輸出單元與每個輸入單元進行交互。然而，卷積層是稀疏連接的，因為區域(field)的卷積的輸出被輸入(而不是區域中每個節點的個別狀態值)到後續層的節點，如圖所示。與卷積層相關聯的核執行卷積運算，其輸出被發送到下一層。卷積層內執行的降維是使CNN能夠縮放以處理大型影像的一個態樣。

【0159】圖9B示出在CNN的卷積層內的示例性計算階段。至CNN之卷積層912的輸出可在卷積層914的三個階段

中進行處理。三個階段可包括卷積階段916、檢測器階段918、及池化階段920。接著卷積層914可以輸出資料給接續的卷積層。網路的最終卷積層可以產生輸出特徵圖資料或者提供輸入至完全連接層，例如，以針對該至CNN的輸入產生分類值。

【0160】在卷積階段916中平行執行數個卷積以產生一組線性激勵。卷積階段916可包括仿射變換，仿射變換是可以被指定為線性變換加上平移的任何變換。仿射變換包括旋轉、平移、縮放、及這些變換的組合。卷積階段計算連接到輸入中的特定區域的函數(例如，神經元)的輸出，其可以被確定為與神經元相關聯的局部區域。神經元計算神經元的權重與神經元所連接的局部輸入中的區域之間的點積。卷積階段916的輸出定義了由卷積層914的後續階段處理的一組線性激勵。

【0161】線性激勵可以由檢測器階段918進行處理。在檢測器階段918中，由非線性激勵函數處理每個線性激勵。非線性激勵函數增加整體網路的非線性特性，而不影響卷積層的感知區域(receptive fields)。可使用幾種類型的非線性激勵函數。一種特定類型是線性整流函數(rectified linear unit, ReLU)，其使用定義為 $f(x) = \max(0, x)$ 的激勵函數，使得激勵閾值為零。

【0162】池化階段920使用池化函數，其用鄰近輸出的摘要統計替換卷積層906的輸出。池化函數可被用來將平移不變性引入神經網路，使得對輸入的小的平移不會改

變已池化的輸出。對局部平移的不變性在輸入資料中存在特徵比特徵的精確位置更重要的情況下是非常有用的。在池化階段920期間可以使用各種類型的池化函數，包括最大池化、平均池化、及L2-norm池化。此外，一些CNN實現不包括池化階段。相反的，這種實現取而代之，並且額外的卷積階段具有相對於先前的卷積階段增加的步長(stride)。

【0163】卷積層914的輸出可接著由下一層922進行處理。下一層922可以是額外的卷積層或者是完全連接層908的其中一者。例如，圖9A的第一卷積層904可以輸出至第二卷積層906，而第二卷積層可以輸出至完全連接層908的第一層。

【0164】圖10示出示例性遞迴神經網路1000。在遞迴神經網路(RNN)中，網路的先前狀態會影響網路的當前狀態的輸出。可以使用各種函數以各種方式來構建RNN。RNN的使用通常圍繞著使用數學模式來依據先前的輸入順序預測未來。例如，RNN可被用來執行統計語言模型化以根據字的先前順序預測即將到來的字。所示出的RNN 1000可被描述為具有接收輸入向量的輸入層1002、實現循環函數的隱藏層1004、致能先前狀態之「記憶體」的反饋機制1005、以及輸出結果的輸出層1006。RNN 1000基於時間步驟運行。給定時間步驟的RNN的狀態係透過反饋機制1005基於先前的時間步驟而受到影響。對於給定的時間步驟，隱藏層1004的狀態是由當前時間步驟的先前狀態和

輸入定義。第一時間步驟的初始輸入(x_1)可以由隱藏層 1004 進行處理。第二輸入(x_2)可以由隱藏層 1004 使用在初始輸入(x_1)的處理期間被決定的狀態資訊來進行處理。一給定狀態可被計算為 $s_t = f(Ux_t + Ws_{t-1})$ ，其中 U 和 W 是參數矩陣。函數 f 通常是非線性函數，諸如雙曲正切函數 (Tanh) 或整流函數 $f(x) = \max(0, x)$ 的變形。然而，隱藏層 1004 中所使用的具體數學函數可以依據 RNN 1000 的具體實現細節而不同。

【0165】除了所述的基本 CNN 及 RNN 網路之外，也可啟用那些網路的變形。一個示例性 RNN 變形是長短期記憶體 (LSTM) RNN。LSTM RNN 能夠學習處理更長的語言序列所需的長期相依性。CNN 的變形是卷積深度信念網路，其具有類似於 CNN 的結構，並且以類似於深度信念網路的方式進行訓練。深度信念網路 (DBN) 是由多層隨機(隨機)變量組成的生成神經網路。可以使用貪心無監督式學習來逐層訓練 DBN。接著，透過決定針對神經網路的最佳初始權重組，DBN 之已學習到的權重可被用來提供預訓練神經網路。

【0166】圖 11 示出深度神經網路的訓練和部署。一旦給定的網路已針對任務被結構化，使用訓練資料集 1102 來訓練神經網路。已經發展出各種訓練框架 1104 來致能訓練過程的硬體加速。例如，圖 6 的機器學習框架 604 可被配置為訓練框架 604。訓練框架 604 可嵌入未經訓練的神經網路 1106，並且能夠使用本文所述的平行處理資源來訓練未經

訓練的神經網，以產生經訓練的神經網路 1108。

【0167】為了起始訓練過程，可以隨機選擇初始權重或者使用深度信念網路預訓練初始權重。接著可以監督式或無監督式來執行訓練週期。

【0168】監督式學習是將訓練執行作為中介運算的學習方法，諸如當訓練資料集 1102 包括輸入，該輸入與針對該輸入之期望的輸出配對，或者訓練資料集包括具有已知輸出的輸入，且神經網路的輸出是手動分級的。網路處理輸入並將結果輸出與一組預期的或期望的輸出比較。然後將誤差回傳通過系統。訓練框架 1104 可以調整控制未經訓練的神經網路 1106 的權重。訓練框架 1104 可以提供工具來監測未經訓練的神經網路 1106 朝向適合於依據已知的輸入資料來產生正確答案的模型收斂的程度。隨著網路的權重被調整以改進神經網路產生的輸出，訓練過程會重複地發生。訓練過程可以持續直到神經網路達到與經訓練的神經網路 1108 相關聯的統計上期望的精確度。然後經訓練的神經網路 1108 可以被部署來實現任何數量的機器學習運算。

【0169】無監督式學習是網路嘗試使用未經標示的資料來訓練其自身的學習方法。因此，對於無監督式學習，訓練資料集 1102 將包括無任何相關聯的輸出資料的輸入資料。未經訓練的神經網路 1106 可以學習在未經標識的輸入內的分組，並且可以確定個別的輸入如何與整個資料集相關。可以使用無監督式學習來產生自組織的映射，該自組織的映射是一種經訓練的神經網路 1108，能夠執行有益於

降低資料維度的運算。也可使用無監督式學習來執行異常檢測，其允許識別輸入資料集中偏離資料之正常模式的資料點。

【0170】也可採用監督式和無監督式訓練的變形。半監督式學習是在訓練資料集 1102 中包括相同分佈的已標示與未經標示之資料的混合的技術。增量學習是監督式學習的變形，其中輸入資料被持續地用以進一步訓練模型。增量學習使得經訓練的神經網路 1108 能夠適應新資料 1112 而不會忘記在初始訓練期間被灌輸到網路內的知識。

【0171】無論是監督式或無監督式，特別深度神經網路的訓練過程對於單一計算節點而言可能計算量過大。取代使用單一計算節點，可使用計算節點的分散式網路來加速訓練過程。

【0172】圖 12 是示出分散式學習的方塊圖。分散式學習是使用多個分散式計算節點來執行神經網路的監督式或無監督式訓練的訓練模型。分散式計算節點可以各包括一或多個主機處理器以及通用處理節點中的一或多個，諸如圖 7 中的高度平行通用圖形處理單元 700。如所示，分散式學習可以是執行的模型平行處理 (parallelism) 1202、資料平行處理 1204、或模型及資料平行處理的組合 1206。

【0173】在模型平行處理 1202 中，分散式系統中的不同計算節點可以針對單一網路的不同部分執行訓練計算。例如，神經網路的每一層可以由分散式系統的不同處理節點來訓練。模型平行處理的好處包括能夠擴展至特別大的

模型。分開與神經網路之不同層相關聯的計算使得能夠訓練非常大的神經網路，其中所有層的權重無法填入單一計算節點的記憶體。在某些情況下，模型平行處理可能特別有用於執行大型神經網路的無監督式訓練。

【0174】在資料平行處理1204中，分散式網路的不同節點具有模型的完整實例，並且每個節點接收資料的不同部分。然後合併來自不同節點的結果。雖然進行資料平行處理的不同方法是可能的，但資料平行訓練方法均需要合併結果以及在各節點之間同步模型參數的技術。組合資料的示例性方法包括參數平均以及基於更新的資料平行處理。參數平均訓練訓練資料之子集上的每個節點，並將全域參數(例如，權重、偏差)設定為來自每個節點的參數的平均值。參數平均使用維護參數資料的中央參數伺服器。基於更新的資料平行處理類似於參數平均，不同之處在於並非將參數從節點傳輸至參數伺服器，而是傳輸模型的更新。此外，可以分散的方式來執行基於更新的資料平行處理，其中在節點之間壓縮和傳送更新。

【0175】組合的模型及資料平行處理1206可被執行例如在分散式系統中，該分散式系統中每個計算節點包括多個GPU。每個節點可以具有模型的完整實例，每個節點內的單獨GPU被用來訓練模型的不同部分。

【0176】相對於在單一機器上訓練，分散式訓練增加了負擔。然而，本文所述的平行處理器及GPGPU可以各自實現各種技術以減少分散式訓練的負擔，包括致能高頻寬

GPU至GPU資料傳輸和加速的遠端資料同步的技術。

示例性機器學習應用

【0177】機器學習可被應用來各種技術問題，包括但不限於計算機視覺、自主駕駛和導航、語音識別、和語言處理。傳統上計算機視覺已是機器學習應用最活躍的研究領域之一。計算機視覺應用的範圍從諸如識別臉孔的重製人類視覺能力到創造視覺能力的新類別。例如，計算機視覺應用可被配置成識別來自視訊中可見物體引起的振動的聲波。平行處理器加速的機器學習使得計算機視覺應用能夠使用比先前可行的顯著較大的訓練資料集來進行訓練，並且使得推論系統能夠使用低功耗平行處理器來進行部署。

【0178】平行處理器加速的機器學習具有自動駕駛應用，包括車道及道路標誌識別、避障、導航、及駕駛控制。加速的機器學習技術可被用來依據資料集訓練駕駛模型，該資料集定義對特定訓練輸入的適當反應。本文所述的平行處理器可以實現用於自動駕駛解決方案之日益複雜的神經網路的快速訓練，並且能夠在適合整合到自動駕駛車輛中的行動平台中部署低功率推論處理器。

【0179】平行處理器加速的深度神經網路已能使機器學習方法實現自動語音識別(ASR)。ASR包括建立一函數，該函數考量給定的輸入聲音序列計算最可能語言序列。使用深度神經網路的加速的機器學習已能夠替代先前

用於 ASR 的隱藏馬可夫模型 (HMM) 和高斯混合模型 (GMM)。

【0180】平行處理器加速的機器學習也可被用來加速自然語言處理。自動學習程序可以利用統計推論演算法來產生對錯誤或不熟悉的輸入很強健的模型。示例性自然語言處理器應用包括人類語言之間的自動機器翻譯。

【0181】可將用於機器學習的平行處理平台分為訓練平台和部署平台。訓練平台通常是高度平行的，並且包括最佳化以加速多 GPU 單節點的訓練和多節點、多 GPU 的訓練。適於訓練的示例性平行處理器包括圖 7 的高度平行通用圖形處理單元 700 及圖 8 的多 GPU 計算系統 800。相反的，部署的機器學習平台通常包括低功率平行處理器，適合用於諸如相機、自動機器人、及自動車等產品中。

【0182】圖 13 示出適用於使用訓練好的模型來執行推論的示例性推論系統單晶片 (SOC) 1300。SOC 1300 可以整合處理組件，其包括媒體處理器 1302、視覺處理器 1304、GPGPU 1306 及多核心處理器 1308。SOC 1300 可以另外包括晶載記憶體 1305，其可致能可由處理組件之各者存取的共用晶載資料池。處理組件可以針對低功率運算被最佳化以使得能夠部署到各種的機器學習平台，包括自動車及自動機器人。例如，SOC 1300 的一個實施方式可被用作自動車之主控制系統的一部分。在將 SOC 1300 配置用於自動車中的情況下，SOC 被設計並配置為遵守部署管轄 (jurisdiction) 之相關功能安全標準。

【0183】在操作期間，媒體處理器 1302 和視覺處理器 1304 可一起工作以加速計算機視覺運算。媒體處理器 1302 可致能多個高解析度(例如，4K、8K)視訊流的低延遲解碼。經解碼的視訊流可被寫入晶載記憶體 1305 中的緩衝器。然後，視覺處理器 1304 可剖析經解碼的視訊，並且在準備使用經訓練的影像識別模型來處理幀的情況下對經解碼的視訊的幀執行初步的處理操作。例如，視覺處理器 1304 可以加速針對被用來對高解析度的視訊資料執行影像識別的 CNN 的卷積操作，而由 GPGPU 1306 執行後端模型計算。

【0184】多核心處理器 1308 可包括控制邏輯以輔助由媒體處理器 1302 和視覺處理器 1304 執行的資料傳輸和共用記憶體操作的定序和同步。多核心處理器 1308 還可以用作應用處理器以執行可以利用 GPGPU 1306 之推論計算能力的軟體應用。例如，導航及驅動邏輯的至少一部分可以在多核心處理器 1308 上執行的軟體中實現。此種軟體可以將計算工作負荷直接發佈給 GPGPU 1306，或者可以將計算工作負荷發佈給多核心處理器 1308，多核心處理器 1308 可以將那些操作的至少一部分卸載到 GPGPU 1306。

【0185】GPGPU 1306 可包括計算叢集，諸如高度平行通用圖形處理器單元 700 內的計算叢集 706A-706H 的低功率配置。GPGPU 1306 內的計算叢集可支援專門最佳化以在經訓練的神經網路上執行推論計算的指令。例如，GPGPU 1306 可支援用以執行諸如 8 位元及 4 位元整數向量

運算的低精度計算的指令。

用於機器學習的動態浮點單元精度降低

【0186】IEEE 754單精度二進制浮點格式指定32位元二進制表示，其具有1位元正負號(sign)、8位元指數、及24位元有效數(significand)，其中明確地儲存23個位元。IEEE 754半精度二進制浮點格式指定16位元二進制表示，其具有1位元正負號、5位元指數、及11位元有效數，其中明確地儲存10個位元。針對非零指數值，隱含有效數位元被定義為1，且當所有指數位元為零時，隱含有效數位元被定義為零。本領域中已知能夠以單精度和半精度執行算術運算的浮點單元。例如，現有的浮點單元可以執行32位元單精度浮點運算(FP32)或雙16位元半精度浮點運算(FP16)。

【0187】本文所述的實施例透過對指令和相關聯的邏輯提供支持來擴展此能力以致能可變精度運算。允許可變精度運算的浮點指令可以藉由以盡可能低的精度執行運算而動態地增加通量。在一個實施例中，提供一組指令和相關聯的邏輯，其中透過以盡可能最低的精度執行浮點運算而沒有顯著的資料丟失來增加通量。在一個實施例中，提供一組指令和相關聯的邏輯，其中浮點邏輯將對比以較高精度執行的結果來驗證較低精度的結果，以確定是否發生任何顯著的資料丟失。

【0188】圖14示出依據實施例之動態精度浮點單元

1400的組件。在一個實施例中，動態精度浮點單元1400包括控制單元1402、一組內部暫存器1404、指數塊1406、及有效數塊1408。除了本領域中已知的浮點控制邏輯之外，在一個實施例中，控制單元1402額外地包括精度追蹤邏輯1412和數值變換單元1422。

【0189】在一個實施例中，精度追蹤邏輯1412是硬體邏輯，被配置為追蹤計算資料相對於目標精度的可用精度位元數。精度追蹤邏輯1412可以追蹤指數塊1406和有效數塊1408內的精度暫存器，以追蹤精度度量，諸如儲存由指數塊1406和有效數塊1408產生的計算值所需的最少精度位元數。在一個實施例中，精度度量包括在一組計算中表示資料所需的數值精度的移動平均值(running average)。在一個實施例中，精度度量包括在一組給定資料內的最大所需精度。在一個實施例中，動態精度浮點單元1400支援讀取或重置精度追蹤邏輯1412所使用的暫存器資料的指令，以產生本文所述的精度度量。在一個實施例中，容納動態精度浮點單元的計算單元支援設置或重置精度追蹤邏輯1412所使用的暫存器檔案的指令。在一個實施例中，精度追蹤邏輯1412監控該組內部暫存器1404內的誤差累加器1434。誤差累加器可被用來追蹤一組浮點運算中的累積誤差(例如，捨入誤差)。在一個實施例中，動態精度浮點單元1400支援一組指令，其包括重置誤差累加器1434的指令和讀取誤差累加器1434的指令。在一個實施例中，可以回應於作為運算元而被提供給指令的位元或旗標而重置誤差

累加器。

【0190】在一個實施例中，數值變換單元1422可被用來在執行較低精度運算時對資料執行中間數值變換，以防止或減輕執行運算時溢位(overflow)或欠位(underflow)的可能性。例如，當接近給定資料類型的精度限制時，數值變換單元1422可以使用對數來執行乘法或除法運算，並且透過指數轉換結果值。圖22中提供關於精度追蹤邏輯1412和數值變換單元1422的更多細節。

【0191】內部暫存器1404包括一組運算元暫存器1414，其儲存用於動態精度浮點單元1400的輸入值。在一個實施例中，運算元暫存器1414包括兩個運算元(A, B)。對於浮點輸入資料，輸入資料值可被分為指數部分(EXA, EXB)和有效數部分(SIGA, SIGB)。在各種實施例中，運算元暫存器1414並不限於支援兩個浮點輸入。在一個實施例中，運算元暫存器1414包括三個輸入運算元，例如，用以支援融和乘加、乘減、乘積-累加、或相關運算。在一個實施例中，運算元暫存器1414也可以儲存整數值，如在一個實施例中，動態精度浮點單元支援32位元、16位元、及8位元整數運算。在一個實施例中，具體的資料類型和基線精度可透過控制單元1402的輸入進行配置。

【0192】在一個實施例中，使用指數塊1406和有效數塊1408以動態精度執行浮點運算。在一個實施例中，可以透過有效數塊1408執行整數運算。在一個實施例中，可以使用指數塊1406和有效數塊1408執行雙8位元整數運算。

【0193】在一個實施例中，指數塊1406包括比較器1416和動態精度指數加法器1426。比較器確定指數之間的差異並且確定兩個指數中較小的一個。在浮點加法期間，將具有較小數值的指數調整成匹配具有較大數值的指數。動態精度指數加法器1426可被用來為FP16或FP32值加上指數值。有效數塊1408包括動態精度乘法器1418、移位單元1428、動態精度有效數加法器1438、及累加暫存器1448。

【0194】在一個實施例中，可以針對運算來指定FP16或FP32資料類型。在指定FP16的情況下，動態精度浮點單元1400可以供電執行FP32運算非必要的閘元件，同時保持邏輯以追蹤精度損失或誤差（例如，透過誤差累加器1434）。例如且在一個實施例中，誤差累加器1434可被用來追蹤一段指令週期內的一些捨入運算。在一個實施例中，誤差累加器保持一組指令中的累加捨入誤差總值。動態精度浮點單元1400可以致能支援從軟體清除或讀取誤差累加器1434的指令。當指定FP32的情況下，動態精度浮點單元1400可以嘗試以FP16精度執行FP32運算，同時供電超出以FP16精度執行運算所需的閘控元件和組件。基於輸入或中間值，在要求動態精度浮點單元1400以FP32執行運算的情況下，動態精度浮點單元1400可以初始地嘗試以FP16執行運算，並且依據需要擴展精度直到FP32。在可以FP16精度執行FP32運算的情況下，每個運算的功率需量減少，允許同時啟用更大量的計算元件。例如，諸如電池供電配置或僅被動冷卻配置的給定配置的動態電容及/或功率預

算限制可能不允許同時啟用所有的浮點單元或 GPGPU 內的其他計算元件。透過致能動態較低精度計算來降低一組浮點單元的動態功率可以增加給定的功率包絡內的 GPGPU 的計算單元的整體通量，因為可基於每個週期處理更多數量的線程而不超過動態功率限制。

【0195】圖 15 提供依據實施例之關於圖 14 的動態精度浮點單元 1400 的額外細節。在一個實施例中，動態精度乘法器 1418 包括一組輸入緩衝器 1502 以儲存有效數資料。在一個實施例中，該組輸入緩衝器包括儲存用於乘法或除法運算的兩個輸入值的兩個緩衝器。對於融和運算（例如，乘加、乘減），運算的乘積可以透過加法器被加到第三輸入及/或被儲存到累加器暫存器中。

【0196】在一個實施例中，動態精度乘法器 1418 的一些配置包括 24 位元輸入的輸入緩衝器，其可以明確地儲存用於單精度浮點輸入的 24 個位元的有效數資料，或者用於半精度浮點值的 11 個位元的有效數資料。在一些配置中，輸入緩衝器 1502 也可以是 32 位元的緩衝器以實現 32 位元整數值的乘法。在一個實施例中，呈現輸入緩衝器 1502 的單一配置，其可在 32 位元及 24 位元之間進行選擇或配置。在一個實施例中，輸出緩衝器 1510 類似地可在 24 位元及 32 位元之間進行配置或選擇，以選擇性地致能針對 32 位元或 16 位元浮點數的全精度 32 位元整數或 24 位元及/或 11 位元有效數值的儲存。

【0197】在一個實施例中，動態精度乘法器 1418 包括

乘法器 1506 及溢位乘法器 1504。乘法器 1506 被配置成針對資料類型以半精度執行乘法或除法運算。例如，乘法器 1506 可以針對 FP16 浮點值的有效數執行 11 位元乘法運算及 / 或針對 16 位元整數運算執行 16 位元乘法運算。乘法器 1506 亦可針對 INT8 整數值執行 8 位元乘法運算。對於 32 位元浮點值或 32 位元整數值，乘法器 1506 可以以 11 位元 (例如，FP16 精度) 針對 24 位元有效數執行乘法運算。若需要，乘法器 1506 可以針對 24 位元 FP32 有效數以 16 位元的精度執行乘法。在一個實施例中，可以透過精度暫存器 1508 追蹤對一給定組的輸入的運算所需的及結果的精度。在一個實施例中，可以在精度暫存器 1508 內透過精度損失來表示所需的和結果的精度，若乘法器 1506 的輸出透過輸出緩衝器 1510 輸出會導致該精度損失。在這種實施例中，精度暫存器 1508 可以追蹤與使用較低精度資料類型相關聯的精度損失以及與以低於所需精度執行運算相關聯的精度損失。

【0198】在一個實施例中，與動態精度乘法器 1418 相關聯的控制邏輯 (例如，在圖 14 的控制單元 1402 內)，可以監控與以較低精度 (例如，FP16、INT16、INT8) 執行較高精度 (例如，FP32、INT32) 的運算相關聯的精度損失。若精度損失將是顯著的，則控制邏輯可以使溢位乘法器 1504 能夠執行額外位元的精度的運算。此外，若控制邏輯依據當前的輸入確定將發生溢位或欠位，則致能溢位乘法器 1504 並且使用溢位乘法器 1504 和乘法器 1506 來執行乘法運

算。

【0199】對於動態精度指數加法器 1426 和動態精度有效數加法器 1438，執行相似的控制操作。動態精度指數加法器 1426 包括一組 8 位元輸入緩衝器，其可以儲存 FP32(8 位元)及 FP16(5 位元)的指數資料。8 位元輸入緩衝器 1512 也可以儲存一組 INT-8 輸入。可以類似地配置用於動態精度指數加法器 1426 的輸出緩衝器 1520。動態精度有效數加法器 1438 包括一組輸入緩衝器 1522，其可選自一組 24 位元及 32 位元緩衝器的其中一者，或者可被動態地配置成儲存具有 24 位元或 32 位元的輸入資料。在一個實施例中，輸入緩衝器 1522 僅單純地是 32 位元緩衝器，其亦可以儲存 24 位元輸入資料。可類似地配置用於動態精度有效數加法器 1438 的輸出緩衝器 1530。動態精度指數加法器 1426 內的精度暫存器 1518 及動態精度有效數加法器 1438 內的精度暫存器 1528 可被配置成追蹤所執行的運算的精度損失。控制邏輯可以根據需要致能溢位加法器 1514 及 / 或溢位加法器 1524，以防止溢位或欠位條件或者防止精度損失超過閾值。

【0200】返回圖 14，在一個實施例中，動態精度浮點單元 1400 可以使用動態精度指數加法器 1426 和動態精度有效數加法器 1438 來執行雙 INT8 運算。例如，取代在整數運算期間禁用指數塊 1406，指數塊 1406 可被配置成對第一組 8 位元整數運算元執行操作，而有效數塊 1408 可被配置成對第二組 8 位元運算元執行操作。為了能夠支援雙 8 位元乘

法、雙融和乘加、雙融和乘減、及/或其他基於乘法的運算，在一個實施例中，指數塊 1406 可以包括額外的乘法器 1436。該乘法器可以是固定的 8 位元乘法器，以使用指數塊 1406 和有效數塊 1408 致能同時的雙 8 位元乘法運算。

【0201】圖 16 示出依據實施例之用於動態精度處理系統 1600 的線程分配。在一個實施例中，動態精度處理系統 1600 包括一組動態浮點單元 1608A-1608D。動態浮點單元 1608A-1608D 可以執行一組操作線程 1606A-1606D，其可以執行混合精度運算並且以可變的精度產生輸出資料。在一個實施例中，可在第一動態浮點單元 1608A 上由第一操作線程 1606A 執行第一操作(例如，加法、減法、乘法、除法等)，其中第一操作線程 1606A 接受作為輸入的兩個 16 位元浮點值 1602A-1602B，並且輸出一 16 位元浮點值 FP16。第一操作可被執行為雙操作，其中由 GPGPU 執行的單指令允許混和精度 FP16/FP32 雙操作。雙操作的第二操作可以由第二動態浮點單元 1608B 執行的第二操作線程 1606B 執行，第二動態浮點單元 1608B 可以產生第二輸出 1612，其為 32 位元浮點輸出。第二操作線程 1606B 配置第二動態浮點單元 1608B 接收兩個 32 位元浮點輸入值 1603A-1603B。在一個實施例中，可以 16 位元的精度執行對兩個 32 位元浮點操作的操作，如果可以在沒有欠位、溢位、或過度(excessive)的情況下執行操作，則以較低精度執行操作不會發生精度損失。

【0202】在一個實施例中，動態精度處理系統 1600 可

以執行具有16位元運算元1604A及32位元運算元1604B的單指令。可以在動態浮點單元1608C上執行操作線程1606C。動態浮點單元1608C將嘗試以16位元精度執行混合精度的16位元/32位元操作，除非將發生顯著的精度損失或誤差。在一個實施例中，動態精度處理系統1600也可被配置成執行整數操作。例如，可以透過操作線程1606D經由動態浮點單元1608D對一對8位元整數輸出1605A-1605B執行操作，以產生一8位元整數輸出1616。在一個實施例中，動態浮點單元1608D被配置成執行雙8位元整數操作，其中兩個8位元整數操作可在單循環中執行。

【0203】圖17示出依據實施例之以低於要求的精度執行數值運算的邏輯1700。在一個實施例中，邏輯1700透過整合在圖14的動態精度浮點單元1400內的硬體來實現。在一個實施例中，邏輯1700部分地透過在圖14的動態精度浮點單元1400內的控制單元1402來執行。

【0204】在一個實施例中，邏輯1700可以接收以第一精度執行數值運算的請求，如方塊1702所示。數值運算可以是浮點運算或整數運算。第一精度可以是，例如，32位元精度。在一個實施例中，數值運算可以是在具有混和精度的運算時執行的第一精度的運算。邏輯1700可接著使用與低於第一精度的第二精度相關聯的位元數執行數值運算，如方塊1704所示。例如且在一個實施例中，用來執行運算的位元數可以是與16位元運算相關聯的位元數，同時第一精度為32位元精度。邏輯1700可以在方塊1706以第二

精度產生中間結果。邏輯 1700可接著確定中間結果相對於第一精度的精度損失。可從暫存器讀取精度損失，該暫存器儲存在運算期間儲存的精度損失的指標。

【0205】邏輯 1700可以在方塊 1709確定精度損失是否小於閾值。在一個實施例中，與精度損失相關聯的閾值可以是軟體可配置的，雖然在一些實施例中使用硬體預設的閾值。在一個實施例中，還可透過在未使用的計算單元上平行執行全精度操作來確定精度損失的程度。然後可將降低的精度結果與全精度結果進行比較。若精度損失小於閾值，則邏輯 1700可以以第二精度輸出結果，如方塊 1712所示。若在方塊 1709精度損失不小於閾值，則邏輯 1700可以在方塊 1710計算結果的剩餘位元，並且以第一精度輸出結果，如方塊 1714所示。在一個實施例中，在方塊 1710計算結果的剩餘位元可以透過溢位邏輯單元，諸如圖 15中的溢位乘法器 1504、溢位加法器 1514、及/或溢位加法器 1524來執行。

用於 16位元浮點運算的操作的垂直堆疊

【0206】當以較低精度執行單指令多線程(SIMT)操作，在某些情況下，由於填充所有 SIMD 通道(lanes)需要的元件較多，可能難以保持底層的單指令多資料(SIMD)邏輯的充分利用。例如，被配置用於在 128位元輸出暫存器上運行 FP32的 SIMD邏輯單元可以對四組輸入資料執行單一操作。若該邏輯單元被配置成對相同的四組輸入資料執

行FP16操作，則由於操作的精度較低，操作的底層的通量可能會增加，但SIMD利用會減半。SIMD未充分利用的一個解決方案是對八組輸入資料執行操作。然而，在邏輯單元上執行的軟體可能不需要如底層硬體可提供的那麼多的平行性。

【0207】例如，可將對輸入陣列執行迭代操作的迴圈向量化，使得陣列的每個迭代作為單獨的SIMT線程平行被執行。可以在計算單元內的底層SIMD/向量邏輯上以單一操作執行單獨的SIMT線程。當執行透過編譯器迴圈向量化邏輯衍生的平行指令時，短於八次迭代的迴圈將不會填滿可用於執行針對那些運算所產生(spawn)的線程的全部八個SIMD通道，從而降低計算單元的整體利用。另外，若底層硬體具有N個SIMD通道，非N倍數的任何數量的向量化迭代將需要在一不完整的SIMD單元上執行剩餘迭代。此外，在執行向量化操作的主體之前，向量化可能需要單獨執行剝離(peel)迴圈。

【0208】本文所述的一些實施例可以透過將多個不相關的FP16操作堆疊到單一SIMD單元中以供執行來提高SIMD利用。在SIMD單元具有8個通道可用於執行的情況下，線程排程邏輯可以以N/2或N/4為單位分派線程，並且允許執行相同的或相容的操作的不相關的幾組線程共用單一SIMD單元。另外，一個實施例致能SIMD通道排程，其允許動態組裝的SIMT線程組與向量SIMD線程的混合。

【0209】圖18示出依據實施例之用於SIMD單元的迴

圈向量化(loop vectorization)。在一個實施例中，軟體邏輯可以包括透過在資料處理系統上執行的編譯器軟體自動向量化的迴圈。迴圈可以包括剝離迴圈1802、向量化的主迴圈1804、及剩餘迴圈1806。在一些配置中，迴圈向量化在對存取對齊的記憶體的资料執行時效率最高。例如，可以配置GPGPU使得向量記憶體存取可以在64位元組塊1801A-1801F中最高效率地被執行。在此種配置中，剝離迴圈1802包括從主迴圈剝離以使未對齊的記憶體存取能夠從主迴圈中區隔出來的迴圈迭代的子集。向量化的主迴圈1804包括迴圈的大部分迭代。可以平行執行向量化的主迴圈的每個迭代，並且針對每個元素的記憶體存取在特定的記憶體邊界上對齊。剩餘迴圈1806包括在向量化的主迴圈1804之後的一組迭代。剩餘迴圈1806中的迭代通常可能不像主迴圈那樣有效率地平行執行。

【0210】在一個實施例中，也可將剝離迴圈1802和剩餘迴圈1806向量化。在一個實施例中，剝離迴圈1802、主迴圈1804、及剩餘迴圈1806之各者可以在FP16 SIMD8單元上執行，其中可以平行執行相同操作的八個實例。可以使用執行遮罩1812、執行遮罩1814、及執行遮罩1816在SIMD硬體(例如，FP16 SIMD8單元1808A-1808C)上平行執行迴圈迭代，該等執行遮罩之各者為操作週期啟用及禁用SIMD通道。對於所示的剝離迴圈1802及剩餘迴圈1806，在執行遮罩1812及執行遮罩1816中選擇元素的子集。在向量化的主迴圈1804的執行遮罩1814中選擇所有的通道。

【0211】在一個實施例中，對於具有不活動的通道的 SIMD 單元，可被配置成在那些不活動的通道上執行其他操作。對於給定的週期，在排程邏輯為 SIMD 單元(例如，FP16 SIMD8 1808A、FP16 SIMD8 1808C)配置一組不活動的通道的情況下，取代在循環的期間閒置那些通道，排程器可以堆疊其他多元素 SIMD 線程或分配 SIMT 線程給其他閒置的 SIMD 通道。

【0212】圖 19 示出依據實施例之線程處理系統 1900。在一個實施例中，線程處理系統 1900 包括 SIMD 計算單元，諸如 SIMD8 浮點單元 1920，其包括多個動態浮點單元 1922A-1922H。取決於操作，SIMD8 浮點單元 1920 可以在單一循環中執行八個或更多個相同或相似的操作。例如且在一個實施例中，八個動態浮點單元 1922A-1922H 之各者可以以 FP16 精度執行單一操作。在一個實施例中，八個動態浮點單元 1922A-1922H 之各者可以在單一循環中執行兩對 INT8 操作。

【0213】在一些情況下，例如具有如圖 18 中所示的剝離或剩餘迴圈，在循環的期間不是所有的 SIMD 浮點單元的通道都將是活動的。為了提高利用，可以較小的粒度分配 SIMD 槽，以使得其他未使用的 SIMD 通道能夠被利用。例如，SIMD8 浮點單元 1920 通常會以八個操作粒度被分配線程或操作，其中少於八個操作會造成潛在的計算效率的損失。在一個實施例中，SIMD 通道可被單向量 SIMD 線程佔用，其包括選擇至少八個元素的執行遮罩或具有至少八

個元素的SIMT線程組。

【0214】為了提高SIMD利用，一個實施例將八個SIMD通道分成兩個SIMD4槽（例如，SIMD4槽1910、SIMD4槽1912）。可以各種方式填充SIMD4槽。在一個實施例中，將結合以覆蓋總共四個SIMD通道的兩個單獨的SIMD線程（SIMD線程1902、SIMD線程1904）分配給SIMD4槽（例如，SIMD4槽1910）。在一個實施例中，可以將SIMT線程組1906分配給SIMD4槽1912。SIMT線程組1906可以包括四個線程的倍數（例如，4、8、12、16等）的任何數量的線程。可以一次處理四個線程來處理SIMT線程組1906內的線程，處理SIMT線程組1906內所有線程所需的循環數取決於組內的線程數。

【0215】圖20示出依據實施例之分配線程用於計算的邏輯2000。在一個實施例中，透過圖19中的線程處理系統1900執行邏輯2000。在一個實施例中，邏輯2000可以在具有第一數量的通道的SIMD單元處接收第一組線程，如方塊2002所示。邏輯2000接著可以確定第一組線程是否填充SIMD單元之全部的SIMD通道，如方塊2003所示。若第一組線程包括足夠的SIMT線程或者第一組線程的線程包括足夠的SIMD向量元素來填充全部的SIMD通道，則邏輯2000可以將第一組線程分配給SIMD單元，如方塊2004所示。

【0216】若第一組線程未填充全部的SIMD通道，如方塊2003處所確定的，則邏輯2000可以在方塊2006處將第

一組線程分配給少於第一數量的通道的第二數量的通道。可以透過將 SIMD 線程分配給 SIMD 單元並遮蔽掉不活動的通道來執行分配。也可以透過將一組 SIMT 線程分配給 SIMD 單元來執行分配。然後，邏輯可以堆疊一或多個額外組的線程來填充全部的 SIMD 通道，如方塊 2008 所示。額外組的線程可以指定佔用未被初始線程佔用的通道的活動 SIMD 通道。

用以致能用於低精度資料之標準化及變換的系統

【0217】當以低精度資料類型執行操作時，必須小心避免數值運算期間資料的溢位或欠位。這種責任通常落在正在開發低精度演算法的資料科學家身上。由於低精度算術的侷限性，許多神經網路已經適應於使用每個元素僅佔用一或兩個位元的二進制及/或三進制值。然而，需要可以使用防護邏輯來致能 N 位元低精度算術的整數及浮點運算邏輯單元在算術運算期間警告或試圖防止顯著的精度損失。在一個實施例中，本文所述之動態精度浮點單元包括當數值計算接近低精度計算的極限時發出警告的邏輯。

【0218】如圖 14 中所示，動態精度浮點單元 1400 可包括精度追蹤邏輯 1412 及數值變換單元 1422。在一個實施例中，精度追蹤邏輯 1412 追蹤計算資料相對於目標精度的剩餘精度的可用位元。若可以追蹤精度，中間資料的可用位元用以確定在一個實施例中相對於輸入資料或輸出資料以更高精度計算的中間資料值是否可以輸出精度儲存而沒有

顯著的精度損失或捨入誤差。例如且在一個實施例中，特定的低精度運算可以以更高的精度有效地執行，並且精度追蹤邏輯 1412 可以確定計算的結果是否會溢出給定的輸出精度。在一個實施例中，本文所述的邏輯單元可以輸出狀態資訊，其指示由於捨入誤差導致的精度損失的程度。在一個實施例中，邏輯單元可以對資料執行中間數值變換以防止顯著的資料損失。接著，邏輯單元可以輸出經變換的值。在一個實施例中，可以基於輸出提供的輸出及狀態資訊以編程方式導出全精度或接近全精度輸出值。

【0219】圖 21 示出可使用由本文所述之實施例所提供的計算邏輯來處理的深度神經網路 2100。深度神經網路 (DNN) 是包括多個神經網路層 2102A-2102N 的人工神經網路。每一層代表一組非線性計算操作，以與本文所述之機器學習神經網路一致的方式執行特徵抽取及變換。每個接續層使用來自前一層的輸出作為輸入。在卷積神經網路的情況下，融合乘加邏輯 (例如，FMA 邏輯 2104A、2104B) 可被用來計算特徵圖和過濾資料之間的點積以產生激勵圖 (activation map) 資料，其被提供作為接續層的輸入。

【0220】可使用與二進制、三進制、或 N 位元特徵圖相結合的二進制或三進制權重來實現低精度神經網路。一些神經網路仍然可受益於使用 N 位元特徵圖及 N 位元過濾器的計算的增加精度。在一些實施方式中，可以以低精度處理神經網路的 N 位元特徵和權重而不會顯著降低輸出誤差。然而，實現低精度 N 位元神經網路 (例如，FP16、

INT8)的資料科學家通常應該知道由於以低精度連續計算而可能出現的捨入誤差或越界資料。若FMA邏輯2104A-2104B中的精度追蹤邏輯(例如,圖14的精度追蹤邏輯1412)確定權重或特徵圖資料正接近資料類型的可用精度的限度,則狀態位元可由FMA邏輯2104A-2014B設置。狀態位元可以作為正在開發存在於神經網路層2102A-2102N內的神經網路模型的資料科學家的指標,指示該模型可能需要最佳化或可能需要更大的數值精度。

【0221】在一個實施例中,可以致能正規化及變換邏輯2106A-2106B以在將特徵圖資料提供給下一個神經網層用於輸入之前,對特徵圖資料執行權重正規化或數值變換。在正規化及變換邏輯2106A-2106B的應用在每個階段都是可選的,並且只有在處理即將到來的層的期間可能發生顯著的精度損失、溢位或欠位條件時才被執行。在一個實施例中,輸出自神經網路之一層的權重或特徵圖可以透過正規化及變換邏輯2106A-2106B的實例而被自動地正規化。

【0222】在一個實施例中,正規化及變換邏輯2106A-2106B可以使用圖14的數值變換單元1422來變換特徵圖資料或權重資料。輸出自神經層的特徵圖資料可以基於輸出自一組函數的一組資料。在此種實施例中,提供了一組特定的低精度指令,其能夠自動調整N位元神經網路資料以防止災難性的精度損失。可由正規化及變換邏輯2106A-2106B執行的示例性變換或正規化包括權重正規化至一範

圍的值，或一組永久的及可逆的特徵資料變換。在一個實施例中，可以執行權重正規化以將一組過濾器權重的動態範圍壓縮到預定範圍內。可以在例如 $[-1,1]$ 的範圍內將權重資料正規化，這可以保持權重值之間的相對差異，同時縮小權重值的整體數值。在一個實施例中，可透過資料集的平均值將神經網路權重或特徵圖資料正規化。

【0223】在一個實施例中，使用接近資料類型之範圍限制的資料的神經網路計算可以在資料被用於計算之前進行變換。例如，使用可能導致溢位的大的值的乘法運算可被執行為對數的加法而不是乘法運算。雖然這種變換可能導致一定程度的精度損失，但是將能夠在不溢出被分配來執行運算的位元數的情況下執行計算。例如，可以如方程式(1)中那樣呈現一系列的運算。

$$f = \frac{A \times B \times C}{D \times E} \quad (1)$$

【0224】若計算單元內的精度追蹤邏輯確定這樣的運算可能溢位或欠位，則運算可被轉換成方程式(2)。

$$f = 10^{(\log(A) + \log(B) + \log(C) - \log(D) - \log(E))} \quad (2)$$

【0225】方程式(2)可被執行來產生結果而不觸發資料類型的溢位。在一個實施例中，正規化及變換邏輯2106A-2016B可以將輸出值轉換成對數值進行儲存，並且在將該值使用於本文所述之機器學習計算之前透過取冪轉換值。

【0226】圖22是依據實施例之邏輯2200的流程圖，該

邏輯 2200用以防止在機器學習執行低精度運算時的錯誤或顯著的精度損失。在一個實施例中，可以透過如圖 14 中的動態精度浮點單元 1400 內的精度追蹤邏輯 1412 及數值變換單元 1422 來執行邏輯 2200。

【0227】在一個實施例中，邏輯 2200 可以依據與神經網路的層相關聯的過濾器 and 特徵圖資料來計算激勵圖，如方塊 2202 所示。然後，邏輯 2200 可以追蹤在計算用於神經網路的層的激勵圖期間所發生的精度損失。邏輯 2200 接著可以在方塊 2205 處確定精度損失是否接近閾值。若在方塊 2205 處精度損失沒有接近預設或配置的閾值，則邏輯 2200 可以繼續計算用於接續層的激勵圖（並且應用激勵函數），直到且除非在方塊 2205 處發生接近閾值的精度損失。當精度損失接近閾值，邏輯 2200 可以在方塊 2207 處確定是否啟用自動數值變換。若在方塊 2207 處，例如，透過被用來執行該組數值運算的指令啟用自動變換，則邏輯 2200 可以在方塊 2208 處轉換神經網路資料以減少由於精度損失引起的誤差。邏輯 2200 可以執行本文所述的任何數值變換，包括範圍內的資料正規化或透過平均值。不管在方塊 2207 處是否啟用自動變換，邏輯 2200 可以在方塊 2210 處輸出指示精度損失正接近閾值的狀態。可以將狀態輸出為狀態旗標，其輸出自計算單元作為執行運算的結果。程式設計師可以透過對執行程式執行演算法調整或者調整用於執行機器學習的神經網路模型來配置軟體邏輯以回應這種狀態。

附加的示例性圖形處理系統

【0228】上述實施例的細節可被併入在下述的圖型處理系統和裝置內。圖23至圖36的圖形處理系統和裝置示出可實現上述任何及所有技術的替代系統和圖形處理硬體。

附加的示例性圖形處理系統概要

【0229】圖23是依據實施例之處理系統2300的方塊圖。在各種實施例中，系統2300包括一或多個處理器2302及一或多個圖形處理器2308，並且可以是單一處理器桌上型系統、多處理器工作站系統、或者是具有大量處理器2302或處理器核心2307的伺服器系統。在一個實施例中，系統2300是被併入用於行動、手持式、或嵌入式裝置的系統單晶片(SoC)積體電路內的處理平台。

【0230】系統2300之一實施例可包括基於伺服器的遊戲平台、包括遊戲和媒體控制台、行動遊戲機、手持式遊戲機、或線上遊戲機等的遊戲機，或者併入於前述各者內。在一些實施例中，系統2300是行動電話、智慧型電話、平板計算裝置或行動上網裝置。資料處理系統2300還可包括可穿戴裝置、與可穿戴裝置耦接、或者被整合進可穿戴裝置內，該可穿戴裝置諸如智慧型手錶可穿戴裝置、智慧型眼鏡裝置、擴增實境裝置、或虛擬實境裝置。在一些實施例中，資料處理系統2300是電視或機上盒裝置，其具有一或多個處理器2302和由一或多個圖形處理器2308所產生的圖形介面。

【0231】 在一些實施例中，一或多個處理器2302各包括一或多個處理器核心2307以處理指令，當執行該等指令時將執行用於系統和使用者的軟體的操作。在一些實施例中，一或多個處理器核心2307之各者被配置成處理特定指令集2309。在一些實施例中，指令集2309可有助於複雜指令集計算(CISC)、精簡指令集計算(RISC)、或透過超長指令字(VLIW)的計算。多個處理器核心2307可以各自處理不同的指令集2309，其可包括有助於其他指令集之仿真的指令。處理器核心2307亦可包括其他的處理裝置，諸如數位信號處理器(DSP)。

【0232】 在一些實施例中，處理器2302包括快取記憶體2304。取決於結構，處理器2302可具有單一內部快取或者多階內部快取。在一些實施例中，在處理器2302的各種組件之間共用快取記憶體。在一些實施例中，處理器2302亦使用外部快取(例如，第3階(L3)快取或最後一階快取(LLC))(未示出)，其可使用已知的快取一致性技術在處理器核心2307之間共用。處理器2302中還包括暫存器檔案2306，其可包括用於儲存不同類型之資料的不同類型的暫存器(例如，整數暫存器、浮點暫存器、狀態暫存器、及指令指標暫存器)。某些暫存器可以是通用暫存器，而其他的暫存器可以專用於處理器2302的設計。

【0233】 在一些實施例中，處理器2302與處理器匯流排2310耦接，以在處理器2302和系統2300中的其他組件之間傳輸諸如位址、資料、或控制信號的通訊信號。在一個

實施例中，系統2300使用示例性「集線器」系統架構，包括記憶體控制器集線器2316及輸入輸出(I/O)控制器集線器2330。記憶體控制器集線器2316有助於記憶體裝置和系統2300之其他組件之間的通訊，而I/O控制器集線器(ICH)2330提供透過本地I/O匯流排至I/O裝置的連接。在一個實施例中，記憶體控制器集線器2316的邏輯被整合在處理器內。

【0234】記憶體裝置2320可以是動態隨機存取記憶體(DRAM)裝置、靜態隨機存取記憶體(SRAM)裝置、快取記憶體裝置、相變記憶體裝置、或者具有合適性能以作為處理記憶體的一些其他記憶體裝置。在一個實施例中，記憶體裝置2320可操作為用於系統2300的系統記憶體，以儲存用於當一或多個處理器2302執行應用程式或者處理時使用的資料2322及指令2321。記憶體控制器集線器2316亦與可選的外部圖形處理器2312耦接，外部圖形處理器2312可與處理器2302中的一或多個圖形處理器2308通訊，以執行圖形和媒體操作。

【0235】在一些實施例中，ICH 2330使週邊設備可以透過高速I/O匯流排連接至記憶體裝置2320及處理器2302。I/O週邊設備包括，但不限於，音訊控制器2346、韌體介面2328、無線收發器2326(例如，Wi-Fi、藍牙)、資料儲存裝置2324(例如，硬碟驅動機、快取記憶體等)、及用於將舊有(例如，個人系統2(PS/2))裝置耦接至系統的舊有I/O控制器2340。一或多個通用序列匯流排(USB)控制器

2342連接諸如鍵盤和滑鼠2344組合的輸入裝置。網路控制器2334亦可耦接至ICH 2330。在一些實施例中，高效能網路控制器(未示出)耦接至處理器匯流排2310。應當理解的是，所示的系統2300是示例性的而非限制性的，因為亦可使用不同配置的其他類型的資料處理系統。例如，I/O控制器集線器2330可被整合在一或多個處理器2302之內，或者記憶體控制器集線器2316及I/O控制器集線器2330可被整合進個別的外部圖形處理器，諸如外部圖形處理器2312。

【0236】圖24是具有一或多個處理器核心2402A-2402N、整合式記憶體控制器2414、及整合式圖形處理器2408之處理器2400的實施例的方塊圖。具有與本文任何其他圖式之元件相同標號(或名稱)的圖24的該些元件，可以類似於本文他處所描述的任何方式操作或發生功用，但不限於此。處理器2400可包括高達且包括由虛線框表示的額外核心2402N的額外核心。處理器核心2402A-2402N之各者包括一或多個內部快取單元2404A-2404N。在一些實施例中，每個處理器核心亦可存取一或多個共用快取單元2406。

【0237】內部快取單元2404A-2404N及共用快取單元2406表示處理器2400內的快取記憶體階層。快取記憶體階層可包括在每個處理器核心內的至少一階的指令和資料快取，以及一或多階的共用中階快取，諸如第2階(L2)、第3階(L3)、第4階(L4)、或其他階的快取，其中在外部記憶體

之前的最高階快取被歸類為 LLC。在一些實施例中，快取一致性邏輯維持各種快取單元 2406 及 2404A-2404N 之間的一致性。

【0238】在一些實施例中，處理器 2400 亦可包括一組一或多個匯流排控制器單元 2416 及系統代理核心 2410。一或多個匯流排控制器單元 2416 管理一組週邊匯流排，諸如一或多個週邊組件互連匯流排（例如，PCI、PCI Express）。系統代理核心 2410 為各種處理器組件提供管理功能。在一些實施例中，系統代理核心 2410 包括一或多個整合式記憶體控制器 2414，以管理對各種外部記憶體裝置（未示出）的存取。

【0239】在一些實施例中，一或多個處理器核心 2402A-2402N 包括對同時多線程的支持。在這種實施例中，系統代理核心 2410 包括用於在多線程處理期間協調和操作核心 2402A-2402N 的組件。系統代理核心 2410 還可包括電力控制單元 (PCU)，其包括用以調節處理器核心 2402A-2402N 及圖形處理器 2408 之電力狀態的邏輯和組件。

【0240】在一些實施例中，處理器 2400 還包括圖形處理器 2408 以執行圖形處理操作。在一些實施例中，圖形處理器 2408 與該組共用快取單元 2406、以及包括一或多個整合式記憶體控制器 2414 的系統代理核心 2410 耦接。在一些實施例中，顯示控制器 2411 與圖形處理器 2408 耦接，以驅動圖形處理器輸出至一或多個耦接的顯示器。在一些實施

例中，顯示控制器 2411 可以是經由至少一個互連而與圖形處理器耦接的單獨模組，或者可被整合在圖形處理器 2408 或系統代理核心 2410 之內。

【0241】在一些實施例中，基於環形之互連單元 2412 被用來耦接處理器 2400 的內部組件。然而，可使用替代的互連單元，諸如，點對點互連、切換式互連或其他技術，包括此領域中熟知的技術。在一些實施例中，圖形處理器 2408 經由 I/O 鏈路 2413 與環形互連 2412 耦接。

【0242】示例性 I/O 鏈路 2413 表示多種 I/O 互連中之至少一者，包括有助於各種處理器組件與高效能嵌入式記憶體模組 2418 (諸如，eDRAM 模組) 之間的通訊之封裝上 I/O 互連。在一些實施例中，處理器核心 2402A-2402N 之各者及圖形處理器 2408 使用嵌入式記憶體模組 2418 作為共用最後一階快取記憶體。

【0243】在一些實施例中，處理器核心 2402A-2402N 為執行同一指令集架構的均質核心。在另一實施例中，就指令集架構 (ISA) 而言，處理器核心 2402A-2402N 是異質的，其中處理器核心 2402A-2402N 之其中一或多者執行第一指令集，而其他核心中之至少一者執行第一指令集的子集或不同的指令集。在一個實施例中，就微架構而言，處理器核心 2402A-2402N 是異質的，其中具有相對較高功耗的一或多個核心與具有較低功耗的一或多個核心耦接。此外，處理器 2400 可在一或多個晶片上實現，或者實現為除了其他組件之外還具有所示組件的 SoC 積體電路。

【0244】圖25是圖形處理器2500的方塊圖，圖形處理器2500可以是分離式(discrete)圖形處理單元、或者可以是與複數個處理核心整合的圖形處理器。在一些實施例中，圖形處理器經由至圖形處理器上的暫存器的記憶體映射I/O介面以及以放置到處理器記憶體中的命令來通訊。在一些實施例中，圖形處理器2500包括記憶體介面2514，用以存取記憶體。記憶體介面2514可以是至本地記憶體、一或多個內部快取、一或多個共用外部快取、及/或系統記憶體的介面。

【0245】在一些實施例中，圖形處理器2500亦包括顯示控制器2502，以驅動顯示輸出資料至顯示裝置2520。顯示控制器2502包括一或多個疊覆平面的硬體，用於多個視訊層或使用者介面元件的顯示及合成。在一些實施例中，圖形處理器2500包括視訊編解碼引擎2506，用以將媒體編碼至一或多個媒體編碼格式、自一或多個媒體編碼格式解碼媒體、或者在一或多個媒體編碼格式之間轉碼媒體，該一或多個媒體編碼格式包括，但不限於，諸如MPEG-2的動畫專家群(MPEG)格式、諸如H.264/MPEG-4 AVC的先進視訊編碼(AVC)格式、以及動畫與電視工程師協會(SMPTE) 421M/VC-1、與諸如JPEG的聯合影像專家群(JPEG)格式、以及動畫JPEG(MJPEG)格式。

【0246】在一些實施例中，圖形處理器2500包括區塊影像轉移(BLIT)引擎2504，用以執行包括，例如，位元邊界(bit-boundary)區塊轉移的二維(2D)光柵化操作。然而，

在一個實施例中，使用圖形處理引擎(GPE)2510的一或多個組件來執行2D圖形操作。在一些實施例中，GPE 2510是用於執行包括三維(3D)圖形操作及媒體操作之圖形操作的計算引擎。

【0247】在一些實施例中，GPE 2510包括3D管線2512用於執行3D操作，諸如使用作用於3D基元形狀(例如，矩形、三角形等)的處理函數來渲染三維影像及場景。3D管線2512包括可編程及固定功能元件，其在元件內執行各種任務及/或產生執行線程至3D/媒體子系統2515。儘管3D管線2512可被用來執行媒體操作，但GPE 2510的實施例亦包括媒體管線2516，其具體地被用來執行諸如視訊後處理及影像增強的媒體操作。

【0248】在一些實施例中，媒體管線2516包括固定功能或可編程邏輯單元，以代替或代表視訊編解碼引擎2506執行一或多個專用媒體操作，諸如視訊解碼加速、視訊解交錯(de-interlacing)、及視訊編碼加速。在一些實施例中，媒體管線2516還包括線程產生(thread spawning)單元，以產生用於在3D/媒體子系統2515上執行的線程。產生的線程在3D/媒體子系統2515中包括的一或多個圖形執行單元上執行針對媒體操作的計算。

【0249】在一些實施例中，3D/媒體子系統2515包括用於執行由3D管線2512及媒體管線2516所產生的線程的邏輯。在一個實施例中，管線傳送線程執行請求至3D/媒體子系統2515，其包括線程分派邏輯，用於仲裁及分派各種

請求至可用的線程執行資源。執行資源包括用以處理3D及媒體線程的圖形執行單元之陣列。在一些實施例中，3D/媒體子系統2515包括用於線程指令和資料的一或多個內部快取。在一些實施例中，該子系統亦包括共用記憶體，其包括暫存器和可定址記憶體，用以在線程之間共用資料以及儲存輸出資料。

示例性附加的圖形處理引擎

【0250】圖26是依據一些實施例之圖形處理器的圖形處理引擎2610的方塊圖。在一個實施例中，圖形處理引擎(GPE)2610是圖25中所示之GPE 2510的一個版本。具有與本文任何其他圖式之元件相同標號(或名稱)的圖26的元件，可以類似於本文他處所描述的任何方式操作或發生功用，但不限於此。例如，示出圖25的3D管線2512和媒體管線2516。媒體管線2516在GPE 2610的一些實施例中是可選的，並且可不被明確地包括在GPE 2610內。例如並且在至少一個實施例中，單獨的媒體及/或影像處理器被耦接到GPE 2610。

【0251】在一些實施例中，GPE 2610耦接或包括命令串流器2603，其提供命令串流至3D管線2512及/或媒體管線2516。在一些實施例中，命令串流器2603與記憶體耦接，記憶體可以是系統記憶體，或者是內部快取記憶體及共用快取記憶體中的一或多個。在一些實施例中，命令串流器2603從記憶體接收命令，並將該些命令發送至3D管線

2512及/或媒體管線2516。該些命令是提取自環形緩衝器的指令(directives)，該環形緩衝器儲存用於3D管線2512及媒體管線2516的命令。在一個實施例中，環形緩衝器還可包括批次命令緩衝器，其儲存多個命令的批次。用於3D管線2512的命令還可以包括對儲存在記憶體中的資料的參照，諸如但不限於用於3D管線2512的頂點和幾何資料及/或用於媒體管線2516的影像資料和記憶體物件。3D管線2512及媒體管線2516透過經由在個別管線內的邏輯執行操作或者透過將一或多個執行線程分派給圖形核心陣列2614來處理命令和資料。

【0252】在各種實施例中，3D管線2512可以透過處理指令並且將執行執行線程分派給圖形核心陣列2614來執行一或多個著色器程式，諸如頂點著色器、幾何著色器、像素著色器、片段著色器、計算著色器、或其他著色器程式。圖形核心陣列2614提供了一統一的執行資源塊。圖形核心陣列2614內的多用途執行邏輯(例如，執行單元)包括對各種3D API著色器語言的支持，並且可以執行與多個著色器相關聯多個同時的執行線程。

【0253】在一些實施例中，圖形核心陣列2614也包括執行邏輯，用以執行媒體功能，諸如視訊及/或影像處理。在一個實施例中，執行單元額外地包括可編程以執行除了圖形處理操作之外的平行通用計算操作的通用邏輯。通用邏輯可以平行地或者與圖23的處理器核心2307內的通用邏輯、圖24中的核心2402A-2402N、或本文所述的任何

其他的處理器一起執行處理操作。

【0254】由在圖形核心陣列2614上執行的線程所產生的輸出資料可以將資料輸出到統一返回緩衝器(URB)2618中的記憶體。URB 2618可以儲存用於多個線程的資料。在一些實施例中，URB 2618可被用來在圖形核心陣列2614上執行的不同線程之間傳送資料。在一些實施例中，URB 2618還可被用於圖形核心陣列上的線程與共用功能邏輯2620內的固定功能邏輯之間的同步。

【0255】在一些實施例中，圖形核心陣列2614是可縮放的，使得陣列包括可變數量的圖形核心，每個圖形核心具有依據GPE 2610之目標功率和性能等級的可變數量的執行單元。在一個實施例中，執行資源是可動態縮放的，使得可依據需要啟用或禁用執行資源。

【0256】圖形核心陣列2614與共用功能邏輯2620耦接，共用功能邏輯2620包括在圖形核心陣列中的圖形核心之間共用的多個資源。共用功能邏輯2620內的共用功能是硬體邏輯單元，其為圖形核心陣列2614提供專門的輔助功能。在各種實施例中，共用功能邏輯2620包括但不限於取樣器2621、數學2622、及線程間(inter-thread)通訊(ITC)2623邏輯。此外，一些實施例實現共用功能邏輯2620內的一或多個快取2625。在給定的專門功能的需求不足以包含在圖形核心陣列2614內的情況下實施共用功能。相反的，將該專門功能的單個實例實現為共用功能邏輯2620中的獨立實體，並且在圖形核心陣列2614內的執行資

源間共用。在各種實施例之間，在圖形核心陣列2614之間共用並且被包括在圖形核心陣列2614內的精確功能組是不同的。

【0257】圖27是圖形處理器2700的另一實施例的方塊圖。具有與本文任何其他圖式之元件相同標號(或名稱)的圖27的元件，可以類似於本文他處所描述的任何方式操作或發生功用，但不限於此。

【0258】在一些實施例中，圖形處理器2700包括環形互連2702、管線前端2704、媒體引擎2737、及圖形核心2780A-2780N。在一些實施例中，環形互連2702將圖形處理器耦接至其他處理單元，包括其他圖形處理器或一或多個通用處理器核心。在一些實施例中，圖形處理器是整合在多核處理系統內的許多處理器的其中一者。

【0259】在一些實施例中，圖形處理器2700透過環形互連2702接收一批命令。傳入的命令由管線前端2704中的命令串流器2703解譯。在一些實施例中，圖形處理器2700包括可縮放執行邏輯，用以經由圖形核心2780A-2780N執行3D幾何處理和媒體處理。針對3D幾何處理命令，命令串流器2703提供命令給幾何管線2736。針對至少一些媒體處理命令，命令串流器2703提供命令給視訊前端2734，其與媒體引擎2737耦接。在一些實施例中，媒體引擎2737包括用於視訊和影像後處理的視訊品質引擎(VQE)2730和用以提供硬體加速的媒體資料編碼及解碼的多格式編碼/解碼(MFX)2733引擎。在一些實施例中，幾何管線2736和媒

體引擎 2737 各針對由至少一個圖形核心 2780A 提供的線程執行資源產生執行線程。

【0260】 在一些實施例中，圖形處理器 2700 包括可縮放線程執行資源，其主要部分為模組化核心 2780A-2780N (有時被稱為核心片 (slice))，各具有多個子核心 2750A-2750N、2760A-2760N (有時被稱為核心子片)。在一些實施例中，圖形處理器 2700 可以具有任何數量的圖形核心 2780A 至 2780N。在一些實施例中，圖形處理器 2700 包括圖形核心 2780A，其具有至少第一子核心 2750A 和第二子核心 2760A。在其他實施例中，圖形處理器是具有單子核心 (例如，2750A) 的低功率處理器。在一些實施例中，圖形處理器 2700 包括多個圖形核心 2780A-2780N，各者包括一組第一子核心 2750A-2750N 及一組第二子核心 2760A-2760N。該組第一子核心 2750A-2750N 中的每個子核心包括至少第一組執行單元 2752A-2752N 及媒體/紋理取樣器 2754A-2754N。該組第二子核心 2760A-2760N 中的每個子核心包括至少第二組執行單元 2762A-2762N 及取樣器 2764A-2764N。在一些實施例中，每個子核心 2750A-2750N、2760A-2760N 共用一組共用資源 2770A-2770N。在一些實施例中，共用資源包括共用快取記憶體和像素操作邏輯。其他的共用資源也可以被包括在圖形處理器的各種實施例中。

示例性附加的執行單元

【0261】圖28示出線程執行邏輯2800，其包括在GPE的一些實施例中採用的處理元件之陣列。具有與本文任何其他圖式之元件相同標號(或名稱)的圖28的元件，可以類似於本文他處所描述的任何方式操作或發生功用，但不限於此。

【0262】在一些實施例中，線程執行邏輯2800包括著色器處理器2802、線程分派器2804、指令快取2806、包括複數個執行單元2808A-2808N的可縮放執行單元陣列、取樣器2810、資料快取2812、及資料埠2814。在一個實施例中，可縮放執行單元陣列可以依據工作負載的計算需求透過啟用或禁用一或多個執行單元(例如，執行單元2808A、2808B、2808C、2808D、至2808N-1及2808N中的任一個)來動態地縮放。在一個實施例中，所包含的組件係經由鏈接到每個組件的互連結構進行互連。在一些實施例中，線程執行邏輯2800包括一或多個連接，其經由指令快取2806、資料埠2814、取樣器2810、及執行單元2808A-2808N中的一或多者連接到記憶體，諸如系統記憶體或快取記憶體。在一些實施例中，每個執行單元(例如，2808A)是獨立的可編程通用計算單元，其能夠執行多個同時的硬體線程，同時為每個線程平行處理多個資料元件。在各種實施例中，執行單元2808A-2808N的陣列是可縮放的，以包括任意數量的單獨執行單元。

【0263】在一些實施例中，執行單元2808A-2808N主要被用來執行著色器程式。著色器處理器2802可以處理各

種著色器程式並透過線程分派器 2804 分派與著色器程式相關聯的執行線程。在一個實施例中，線程分派器包括用以仲裁來自圖形和媒體管線的線程啟動請求並且在執行單元 2808A-2808N 中的一或多個執行單元上將請求的線程實例化的邏輯。例如，幾何管線(例如，圖 27 的 2736)可將頂點、曲面細分、或幾何著色器分派給線程執行邏輯 2800(圖 28)進行處理。在一些實施例中，線程分派器 2804 也可以處理來自正在執行的著色器程式的執行時線程產生請求。

【0264】在一些實施例中，執行單元 2808A-2808N 支援包括對許多標準 3D 圖形著色器指令原生支援的指令集，使得來自圖形程式庫(例如，Direct 3D 及 OpenGL)的著色器程式以最小轉譯執行。執行單元支援頂點及幾何處理(例如，頂點程式、幾何程式、頂點著色器)、像素處理(例如，像素著色器、片段著色器)及通用處理(例如，計算和媒體著色器)。執行單元 2808A-2808N 之各者能夠進行多發佈(multi-issue)單指令多資料(SIMD)執行，且多線程的運算在面對較高延遲的記憶體存取時能實現高效的執行環境。每個執行單元內的各個硬體線程具有專用高頻寬暫存器檔案及相關聯的獨立線程狀態。執行是每時脈多發佈至能夠進行整數、單精度和雙精度浮點運算、SIMD 分支功能、邏輯運算、超越(transcendental)運算、及其他雜項運算的管線。在等待來自記憶體或共用功能之一的資料時，執行單元 2808A-2808N 內的相依邏輯導致等待線程休眠直

到請求的資料已被返回。當等待線程正在休眠時，硬體資源可專用於處理其他線程。例如，在與頂點著色器操作相關聯的延遲期間，執行單元可以執行用於像素著色器、片段著色器、或包括不同頂點著色器之另一類型的著色器程式的操作。

【0265】執行單元2808A-2808N中的每個執行單元對資料元件的陣列進行操作。資料元件的數量是「執行大小」或用於指令的通道數量。執行通道是用於執行指令內的資料元件存取、遮蔽、及流程控制的邏輯單元。通道的數量可以獨立於用於特定圖形處理器之物理算術邏輯單元(ALU)或浮點單元(FPU)的數量。在一些實施例中，執行單元2808A-2808N支援整數及浮點資料類型。

【0266】執行單元指令集包括SIMD指令。可將各種資料元件作為緊縮資料類型儲存於暫存器中，並且執行單元將依據元件的資料大小來處理各種元件。例如，當對256位元寬的向量進行操作時，將向量的256位元儲存在暫存器中，並且執行單元將該向量作為四個單獨的64位元緊縮資料元件(四字(QW)大小資料元件)、八個單獨的32位元緊縮資料元件(雙字(DW)大小資料元件)、十六個單獨的16位元緊縮資料元件(字(W)大小資料元件)、或三十二個單獨的8位元資料元件(位元組(B)大小資料元件)進行操作。然而，不同的向量寬度和暫存器大小是可能的。

【0267】一或多個內部指令快取(例如，2806)被包括在線程執行邏輯2800中，以快取用於執行單元的快取線程

指令。在一些實施例中，包括一或多個資料快取(例如，2812)以在線程執行期間快取線程資料。在一些實施例中，包括取樣器2810以提供用於3D操作的紋理取樣和用於媒體操作的媒體取樣。在一些實施例中，取樣器2810包括專門的紋理或媒體取樣功能，以在將取樣資料提供給執行單元之前，在取樣處理期間處理紋理或媒體資料。

【0268】在執行期間，圖形和媒體管線經由線程產生及分派邏輯將線程啟動請求發送給線程執行邏輯2800。一旦一組幾何物件已被處理並且被光柵化成為像素資料，將調用著色器處理器2802內的像素處理器邏輯(例如，像素著色器邏輯、片段著色器邏輯等)，以進一步計算輸出資料並使結果被寫入到輸出表面(例如，色彩緩衝器、深度緩衝器、模板緩衝器等)。在一些實施例中，像素著色器或片段著色器計算各種頂點屬性的值，該些值將在經光柵化的物件上進行內插。在一些實施例中，著色器處理器2802內的像素處理器邏輯接著執行應用程式設計介面(API)提供的像素或片段著色器程式。為了執行著色器程式，著色器處理器2802經由線程分派器2804將線程分派至執行單元(例如，2808A)。在一些實施例中，像素著色器2802使用取樣器2810中的紋理取樣邏輯來存取儲存在記憶體中的紋理圖中的紋理資料。對紋理資料和輸入幾何資料的算術運算計算用於每個幾何片段的像素色彩資料，或者從進一步處理丟棄一或多個像素。

【0269】在一些實施例中，資料埠2814提供記憶體存

取機制，用於線程執行邏輯2800輸出經處理的資料至記憶體，以在圖形處理器輸出管線上處理。在一些實施例中，資料埠2814包括或耦接到一或多個快取記憶體(例如，資料快取2812)，以經由資料埠快取用於記憶體存取的資料。

【0270】圖29是示出依據一些實施例之圖形處理器指令格式2900的方塊圖。在一或多個實施例中，圖形處理器執行單元支援具有多種格式之指令的指令集。實線框繪示了通常包括在執行單元指令中的組件，而虛線則包括可選的組件或者僅包括在指令之子集中的組件。在一些實施例中，所描述和繪示的指令格式2900是巨集指令，因為它們是供應給執行單元的指令，而不是在指令被處理之後即由指令解碼所產生的微操作。

【0271】在一些實施例中，圖形處理器執行單元原生地支援128位元指令格式2910的指令。根據所選的指令、指令選項、和運算元的數量，64位元壓緊指令格式2930可用於某些指令。原生的128位元指令格式2910提供對所有指令選項的存取，而一些選項及操作在64位元格式2930中受到限制。64位元格式2930中可用的原生指令隨實施例而不同。在一些實施例中，使用索引欄位2913中的一組索引值來部分地壓緊指令。執行單元基於索引值硬體參考一組壓緊表，並且使用壓緊表輸出來重構128位元格式2910的原生指令。

【0272】針對各個格式，指令操作碼2912定義執行單

元將要執行的操作。執行單元跨各個運算元的多個資料元件平行地執行各個指令。例如，回應於添加指令，執行單元跨表示紋理元件或圖片元件的各個色彩通道執行同時的添加操作。按照預設，執行單元跨運算元的全部資料通道執行每個指令。在一些實施例中，指令控制欄位 2914 致能對諸如通道選擇(例如，預測)及資料通道順序(例如，調換(swizzle))之某些執行選項的控制。針對 128 位元指令格式 2910，執行大小(exec-size)欄位 2916 限制了將被平行執行的資料通道的數量。在一些實施例中，執行大小欄位 2916 不適用於在 64 位元壓緊指令格式 2930 中使用。

【0273】一些執行單元指令具有高達三個運算元，包括兩個來源運算元 src0 2920、src1 2922 以及一個目的地 2918。在一些實施例中，執行單元支援雙目的地指令，其中目的地之其中一者是隱含的。資料操縱指令可以具有第三來源運算元(例如，SRC2 2924)，其中指令操作碼 2912 判定來源運算元的數量。指令的最後一個來源運算元可以是與指令一起傳遞的立即(例如，硬編碼的)值。

【0274】在一些實施例中，128 位元指令格式 2910 包括存取/定址模式欄位 2926，其，指定例如是使用直接暫存器定址模式或是使用間接暫存器定址模式。當使用直接暫存器定址模式時，由指令中的位元直接地提供一或多個運算元的暫存器位址。

【0275】在一些實施例中，128 位元指令格式 2910 包括存取/定址模式欄位 2926，其指定用於指令的定址模式

及/或存取模式。在一個實施例中，存取模式被用以定義指令的資料存取對齊。一些實施例支援存取模式包括16位元組對齊的存取模式及1位元組對齊的存取模式，其中存取模式的位元組對齊決定指令運算元的存取對齊。例如，當在第一模式時，指令可將位元組對齊的定址用於來源及目的地運算元，且在第二模式時，指令可將16位元組對齊的定址用於所有的來源及目的地運算元。

【0276】在一個實施例中，存取/定址模式欄位2926的定址模式部分決定指令是使用直接或間接定址。當使用直接暫存器定址模式時，指令中的位元直接地提供一或多個運算元的暫存器位址。當使用間接暫存器定址模式時，可基於定址暫存器值及指令中的定址立即欄位來計算一或多個運算元的暫存器位址。

【0277】在一些實施例中，基於操作碼2912位元欄位將指令分組以簡化操作碼解碼2940。針對8位元操作碼，位元4、5、及6允許執行單元決定操作碼的類型。所示之精確的操作碼分組僅是示例性的。在一些實施例中，移動及邏輯操作碼群組2942包括資料移動及邏輯指令(例如，移動(mov)、比較(cmp))。在一些實施例中，移動及邏輯群組2942共用五個最高有效位元(MSB)，其中移動(mov)指令是0000xxxxb的形式，且邏輯指令是0001xxxxb的形式。流量控制指令群組2944(例如，呼叫、跳越(jmp))包括0010xxxxb(例如，0x20)之形式的指令。雜項指令群組2946包括指令的混合，包括0011xxxxb(例如，0x30)之形

式的同步指令(例如，等待、發送)。平行數學指令群組 2948 包括 0100xxxxb(例如，0x40)之形式的在分量方面(component-wise)的算術指令(例如，加、乘(mul))。平行數學群組 2948 跨資料通道平行地執行算數運算。向量數學群組 2950 包括 0101xxxxb(例如，0x50)之形式的算數指令(例如，dp4)。向量數學群組對向量運算元執行諸如點積計算的算術。

示例性附加的圖形管線

【0278】圖 30 是圖形處理器 3000 的另一實施例的方塊圖。具有與本文任何其他圖式之元件相同標號(或名稱)的圖 30 的元件，可以類似於本文他處所描述的任何方式操作或發生功用，但不限於此。

【0279】在一些實施例中，圖形處理器 3000 包括圖形管線 3020、媒體管線 3030、顯示引擎 3040、線程執行邏輯 3050、及渲染輸出管線 3070。在一些實施例中，圖形處理器 3000 是在包括一或多個通用處理核心之多核處理系統內的圖形處理器。圖形處理器係透過至一或多個控制暫存器(未示出)的暫存器寫入或經由環形互連 3002 發佈至圖形處理器 3000 的命令而被控制。在一些實施例中，環形互連 3002 將圖形處理器 3000 耦接到其他處理元件，諸如其他圖形處理器或通用處理器。來自環形互連 3002 的命令係由命令串流器 3003 解譯，其將指令提供至圖形管線 3020 或媒體管線 3030 的個別元件。

【0280】在一些實施例中，命令串流器3003指示頂點提取器3005的操作，頂點提取器3005從記憶體讀取頂點資料並且執行由命令串流器3003提供的頂點處理命令。在一些實施例中，頂點提取器3005將頂點資料提供給頂點著色器3007，其對各個頂點執行座標空間轉換及打光操作。在一些實施例中，頂點提取器3005及頂點著色器3007透過經由線程分派器3031將執行線程分派至執行單元3052A-3052B來執行頂點處理指令。

【0281】在一些實施例中，執行單元3052A-3052B為向量處理器的陣列，其具有用於執行圖形和媒體操作的指令集。在一些實施例中，執行單元3052A-3052B具有特定用於各個陣列或者在陣列之間共用的附加的L1快取3051。該快取可被配置成為資料快取、指令快取、或單一快取，其被分割成在不同的分區中包含資料及指令。

【0282】在一些實施例中，圖形管線3020包括曲面細分元件，用以執行3D物件之硬體加速的曲面細分。在一些實施例中，可編程的輪廓著色器3011配置曲面細分操作。可編程的域著色器3017提供曲面細分輸出的後端評估。曲面細分器3013在輪廓著色器3011的方向上操作，並且包含專用邏輯以基於被提供作為圖形管線3020之輸入的粗略幾何模型來產生一組細微的幾何物件。在一些實施例中，若未使用曲面細分，則可繞過曲面細分元件(例如，輪廓著色器3011、曲面細分器3013、及域著色器3017)。

【0283】在一些實施例中，完整的幾何物件可透過幾

何著色器 3019 經由被分派至執行單元 3052A-3052B 的一或多個線程來處理，或者可直接前進到剪裁器 3029。在一些實施例中，幾何著色器對整個幾何物件進行操作，而不是如在圖形管線的先前階段中對頂點或是頂點的面片 (patches) 進行操作。若曲面細分被停用，則幾何著色器 3019 從頂點著色器 3007 接收輸入。在一些實施例中，幾何著色器 3019 可透過幾何著色器程式被編程，以在曲面細分單元被禁用的情況下執行幾何曲面細分。

【0284】在光柵化之前，剪裁器 3029 處理頂點資料。剪裁器 3029 可以是固定功能剪裁器或是具有剪裁及幾何著色器功能的可編程剪裁器。在一些實施例中，渲染輸出管線 3070 中的光柵化及深度測試元件 3073 分派像素著色器以將幾何物件轉換成它們的每個像素表示。在一些實施例中，像素著色器邏輯包括在線程執行邏輯 3050 中。在一些實施例中，應用程式可繞過光柵化及深度測試元件 3073，且經由串流輸出單元 3023 存取未被光柵化的頂點資料。

【0285】圖形處理器 3000 具有允許資料及訊息在處理器的主要組件之間傳遞的互連匯流排、互連結構、或一些其他互連機構。在一些實施例中，執行單元 3052A-3052B 及相關聯的快取 3051、紋理及媒體取樣器 3054、及紋理 / 取樣器快取 3058 經由資料埠 3056 互連以執行記憶體存取，並與處理器的渲染輸出管線組件通訊。在一些實施例中，取樣器 3054、快取 3051、3058 及執行單元 3052A-3052B 各具獨立的記憶體存取路徑。

【0286】 在一些實施例中，渲染輸出管線3070包含光柵化及深度測試元件3073，其將基於頂點的物件轉換成相關聯的基於像素的表示。在一些實施例中，光柵化邏輯包括分窗器(windower)/遮蔽器(masker)單元以執行固定功能三角形及線光柵化。相關聯的渲染快取3078及深度快取3079在一些實施例中也是可用的。像素操作組件3077對資料執行基於像素的操作，然而在一些實例中，與2D操作相關聯的像素操作(例如，具有混色的位元區塊影像傳送)是由2D引擎3041執行，或者在顯示時間由顯示控制器3043使用疊加顯示平面代替。在一些實施例中，共用L3快取3075可用於所有的圖形組件，允許在不使用主系統記憶體的情況下共用資料。

【0287】 在一些實施例中，圖形處理器媒體管線3030包括媒體引擎3037及視訊前端3034。在一些實施例中，視訊前端3034接收來自命令串流器3003的管線命令。在一些實施例中，媒體管線3030包括獨立的命令串流器。在一些實施例中，視訊前端3034在將媒體命令傳送至媒體引擎3037之前處理該等命令。在一些實施例中，媒體引擎3037包括線程產生(spawning)功能性以產生線程用於經由線程分派器3031分派至線程執行邏輯3050。

【0288】 在一些實施例中，圖形處理器3000包括顯示引擎3040。在一些實施例中，顯示引擎3040在處理器3000外部，並且經由環形互連3002、或一些其他互連匯流排或結構與圖形處理器耦接。在一些實施例中，顯示引擎3040

包括2D引擎3041及顯示控制器3043。在一些實施例中，顯示引擎3040包含能夠獨立於3D管線進行操作的專用邏輯。在一些實施例中，顯示控制器3043與顯示裝置(未示出)耦接，該顯示裝置可以是如在膝上型電腦中的系統整合式顯示裝置，或者是透過顯示裝置連接器附接的外部顯示裝置。

【0289】在一些實施例中，圖形管線3020及媒體管線3030可被配置成基於多個圖形及媒體程式介面執行操作，並且不特定於任一個應用程式介面(API)。在一些實施例中，用於圖形處理器的驅動軟體將特定於特定圖形或媒體程式庫的API呼叫轉譯成可由圖形處理器處理的命令。在一些實施例中，提供對開放圖形程式庫(OpenGL)、開放計算語言(OpenCL)、及/或Vulkan圖形及計算API的支援，這些均來自科納斯組織(Khronos Group)。在一些實施例中，亦可提供對來自微軟公司的Direct3D程式庫的支援。在一些實施例中，可支援這些程式庫的組合。亦可提供對開放源電腦視覺程式庫(OpenCV)的支援。還將支援具有相容3D管線的未來API，若能將該未來API的管線映射到圖形處理器的管線。

圖形管線程式設計

【0290】圖31A是示出依據一些實施例之圖形處理器命令格式3100的方塊圖。圖31B是示出依據實施例之圖形處理器命令序列3110的方塊圖。圖31A中的實線框繪示了

圖形命令中通常包括的元件，而虛線則包括可選的或僅包括在圖形命令之子集中的元件。圖 31A 的示例性圖形處理器命令格式 3100 包括用以識別命令之目標客戶端 3102 的資料欄位、命令操作碼(opcode) 3104、及用於命令的相關資料 3106。子操作碼 3105 及命令大小 3108 亦包括在一些命令中。

【0291】 在一些實施例中，客戶端 3102 指定圖形裝置之處理命令資料的客戶端單元。在一些實施例中，圖形處理器命令剖析器檢驗每個命令的客戶端欄位以調節命令的進一步處理，並將命令資料路由到適當的客戶端單元。在一些實施例中，圖形處理器客戶端單元包括記憶體介面單元、渲染單元、2D 單元、3D 單元、及媒體單元。每個客戶端單元具有處理命令的相對應的處理管線。一旦命令由客戶端單元接收，該客戶端單元讀取操作碼 3104，及若存在的話，讀取子操作碼 3105 以判定待執行的操作。該客戶端單元使用在資料欄位 3106 中的資訊來執行命令。針對某些命令，需要顯式命令大小 3108 來指定命令的大小。在一些實施例中，命令剖析器基於命令操作碼自動地判定命令中之至少一些命令的大小。在一些實施例中，經由雙字的倍數來對齊命令。

【0292】 圖 31B 中的流程圖示出示例性圖形處理器命令序列 3110。在一些實施例中，表徵圖形處理器之實施例的資料處理系統之軟體或韌體使用所示之命令序列的版本來設定、執行、及終止一組圖形操作。僅出於示例之目的

示出及描述範例命令序列，因為實施例不限於這些特定命令或此命令序列。此外，命令可作為命令序列中的命令批次而被發佈，使得圖形處理器將至少部分地同時處理命令序列。

【0293】在一些實施例中，圖形處理器命令序列3110可以管線排清命令3112開始，以使任何作用中的圖形管線完成用於管線的當前未決的命令。在一些實施例中，不會同時操作3D管線3122及媒體管線3124。執行管線排清以使作用中的圖形管線完成任何未決的命令。回應於管線排清，用於圖形處理器的命令剖析器將暫停命令處理直到作用中的製圖引擎完成未決的操作，且相關的讀取快取是無效的。可選地，渲染快取中被標記為「髒(dirty)」的任何資料可被排清至記憶體。在一些實施例中，管線排清命令3112可被用於管線同步或在將圖形處理器置於低功率狀態之前使用。

【0294】在一些實施例中，當命令序列要求圖形處理器顯式地在管線之間切換時，使用管線選擇命令3113。在一些實施例中，管線選擇命令3113在發佈管線命令之前在執行上下文內僅被需要一次，除非該上下文是用於發佈針對兩個管線的命令。在一些實施例中，在經由管線選擇命令3113的管線切換之前立即需要管線排清命令3112。

【0295】在一些實施例中，管線控制命令3114配置圖形管線以供操作，並且被用以編程3D管線3122和媒體管線3124。在一些實施例中，管線控制命令3114配置用於作用

中的管線的管線狀態。在一個實施例中，管線控制命令 3114 被用於管線同步，並且在處理一批命令之前被用以清除作用中的管線內的一或多個快取記憶體的资料。

【0296】在一些實施例中，返回緩衝狀態命令 3116 被用以配置一組返回緩衝器以供個別的管線寫入資料。一些管線操作需要一或多個返回緩衝器的分配、選擇、或組態，該些操作在處理期間將中間資料寫入該一或多個返回緩衝器中。在一些實施例中，圖形處理器亦使用一或多個返回緩衝器來儲存輸出資料，並執行跨線程通訊。在一些實施例中，返回緩衝狀態 3116 包括選擇要用於一組管線操作的返回緩衝器的大小及數量。

【0297】命令序列中的剩餘命令基於用於操作的作用中的管線而不同。基於管線判定 3120，命令序列係針對從 3D 管線狀態 3130 開始的 3D 管線 3122 設計，或者針對從媒體管線狀態 3140 開始的媒體管線 3124 設計。

【0298】用於配置 3D 管線狀態 3130 的命令包括 3D 狀態設定命令，用於頂點緩衝狀態、頂點元件狀態、恆定色彩狀態、深度緩衝狀態、以及在處理 3D 基元命令之前要被配置的其他狀態變數。這些命令的值係至少部分地依據使用中的特定 3D API 來判定。在一些實施例中，3D 管線狀態 3130 命令亦能夠選擇性地停用或者繞過某些管線元件，若那些元件將不被使用。

【0299】在一些實施例中，3D 基元 3132 命令係用以提交 3D 基元以由 3D 管線處理。經由 3D 基元 3132 命令被傳遞

至圖形處理器的命令及相關聯參數被轉送至圖形管線中的頂點提取功能。頂點提取功能使用3D基元3132命令資料來產生頂點資料結構。頂點資料結構被儲存在一或多個返回緩衝器中。在一些實施例中，3D基元3132命令被用以經由頂點著色器對3D基元執行頂點操作。為了處理頂點著色器，3D管線3122將著色器執行線程分派至圖形處理器執行單元。

【0300】在一些實施例中，3D管線3122係經由執行3134命令或事件而被觸發。在一些實施例中，暫存器寫入觸發命令執行。在一些實施例中，執行係經由命令序列中的「前進(go)」或「啟動(kick)」命令而被觸發。在一個實施例中，命令執行係使用管線同步命令來從圖形管線中排清命令序列而被觸發。3D管線將針對3D基元執行幾何處理。一旦操作完成，所得到的幾何物件將被光柵化，且像素引擎將所得到的像素上色。控制像素著色及像素後端操作的額外命令亦可被包括用於那些操作。

【0301】在一些實施例中，圖形處理器命令序列3110在執行媒體操作時遵循媒體管線3124路徑。一般而言，用於媒體管線3124之編程的具體用途和方式取決於待執行的媒體或計算操作。具體的媒體解碼操作可在媒體解碼期間被卸載至媒體管線。在一些實施例中，亦可繞過媒體管線，且可使用由一或多個通用處理核心所提供的資源來整體或部分地執行媒體解碼。在一個實施例中，媒體管線亦包括用於通用圖形處理器單元(GPGPU)操作的元件，其中

圖形處理器被用以使用並非顯式地與圖形基元之渲染相關的計算著色器程式來執行SIMD向量操作。

【0302】在一些實施例中，以與3D管線3122類似的方式組態媒體管線3124。將一組組態媒體管線狀態3140的命令分配或放置在命令序列中的媒體物件命令3142之前。在一些實施例中，媒體管線狀態命令3140包括用以組態將被用來處理媒體物件之媒體管線元件的資料。這包括用以組態媒體管線內之視訊解碼及視訊編碼邏輯，諸如編碼或解碼格式的資料。在一些實施例中，媒體管線狀態命令3140亦支援使用一或多個指標指向「間接」狀態元件，其包含一批狀態設定。

【0303】在一些實施例中，媒體物件命令3142將指標供應至媒體物件以供媒體管線處理。媒體物件包括記憶體緩衝器，其包含待處理的視訊資料。在一些實施例中，在發佈媒體物件命令3142之前，所有的媒體管線狀態必須是有效的。一旦組態了管線狀態且將媒體物件命令3142排進佇列，媒體管線3124經由執行命令3144或等效的執行事件(例如，暫存器寫入)而被觸發。來自媒體管線3124的輸出可接著由3D管線3122或媒體管線3124所提供的操作進行後處理。在一些實施例中，以與媒體操作相似的方式組態及執行GPGPU操作。

圖形軟體結構

【0304】圖32示出依據一些實施例之用於資料處理系

統 3200 的示例性圖形軟體結構。在一些實施例中，軟體結構包括 3D 圖形應用程式 3210、作業系統 3220、及至少一個處理器 3230。在一些實施例中，處理器 3230 包括圖形處理器 3232 及一或多個通用處理器核心 3234。圖形應用程式 3210 及作業系統 3220 各在資料處理系統之系統記憶體 3250 中執行。

【0305】 在一些實施例中，3D 圖形應用程式 3210 包含一或多個著色器程式，其包括著色器指令 3212。著色器語言指令可以是高階著色器語言，諸如高階著色器語言 (HLSL) 或是 OpenGL 著色器語言 (GLSL)。應用程式亦包括適於通用處理器核心 3234 執行之機器語言形式的可執行指令 3214。應用程式亦包括由頂點資料所界定的圖形物件 3216。

【0306】 在一些實施例中，作業系統 3220 為來自微軟公司的 Microsoft® Windows® 作業系統、專有的類 UNIX 作業系統、或使用 Linux 核心之變體的開源類 UNIX 作業系統。作業系統 3220 可支援圖形 API 3222，諸如 Direct3D API、OpenGL API、或 Vulkan API。當使用 Direct3D API 時，作業系統 3220 使用前端著色器編譯器 3224 來將 HLSL 之形式的任何著色器指令 3212 編譯成較低階的著色器語言。該編譯可以是即時 (JIT) 編譯或者該應用程式可執行著色器預編譯。在一些實施例中，高階著色器在 3D 圖形應用程式 3210 的編譯期間被編譯成低階著色器。在一些實施例中，以中間形式提供著色器指令 3212，諸如由 Vulkan API

使用的標準可攜式中間表示 (Standard Portable Intermediate Representation, SPIR)的版本。

【0307】在一些實施例中，使用者模式圖形驅動器 3226 包含後端著色器編譯器 3227，以將著色器指令 3212 轉換成硬體特定表示。當使用 OpenGL API 時，GLSL 高階語言形式的著色器指令 3212 被傳遞至使用者模式圖形驅動器 3226 用於編譯。在一些實施例中，使用者模式圖形驅動器 3226 使用作業系統核心模式功能 3228 來與核心模式圖形驅動器 3229 通訊。在一些實施例中，核心模式圖形驅動器 3229 與圖形處理器 3232 通訊以分派命令及指令。

IP 核心實現

【0308】至少一個實施例的一或多個態樣可由儲存在機器可讀取媒體上的代表性程式碼實施，該機器可讀取媒體表示及/或定義在諸如處理器之積體電路內的邏輯。舉例來說，機器可讀取媒體可包括表示處理器內之各種邏輯的指令。當由機器讀取時，指令可使機器製造邏輯來執行本文所述的技術。被稱為「IP 核心」的此種表示為用於積體電路的可重複使用的邏輯單元，其可被儲存在有形的、機器可讀取媒體上作為描述積體電路之結構的硬體模型。硬體模型可被供應給各種客戶或製造設施，其將硬體模型加載在製造積體電路的製造機器上。可製造積體電路使得電路執行與本文所述之任何實施例相關聯地描述的操作。

【0309】圖 33 是示出依據實施例之可被用來製造積體

電路以執行運算的IP核心開發系統3300的方塊圖。可使用IP核心開發系統3300來產生可被整合進較大型設計或被用來建構整個積體電路(例如，SOC積體電路)的模組化的、可重複使用的設計。設計設施3330可以高階程式語言(例如，C/C++)來產生IP核心設計的軟體模擬3310。軟體模擬3310可被用來使用模擬模型3312設計、測試、及驗證IP核心的行為。模擬模型3312可包括功能、行為、及/或時序模擬。接著可從模擬模型3312建立或合成暫存器傳送層級(RTL)設計3315。RTL設計3315為模型化硬體暫存器之間的數位信號流之積體電路的行為的抽象化，包括使用模型化的數位信號執行的相關聯的邏輯。除了RTL設計3315之外，亦可建立、設計、或合成在邏輯層級或電晶體層級的較低階設計。因此，初始設計和模擬的特定細節可不同。

【0310】RTL設計3315或等效物可進一步由設計設施合成為硬體模型3320，其可以是硬體描述語言(HDL)格式或實體設計資料的某些其他表示。可進一步模擬或測試HDL來驗證該IP核心設計。可使用非揮發性記憶體3340(例如，硬碟、快閃記憶體、或任何非揮發性儲存媒體)來儲存IP核心設計用以傳遞至第三方製造設施3365。替代地，可經由有線連接3350或無線連接3360來(例如，透過網際網路)傳輸IP核心設計。接著製造設施3365可製造至少部分依據IP核心設計的積體電路。所製造的積體電路可被配置成依據本文所述之至少一個實施例執行操作。

示例性系統單晶片積體電路

【0311】圖 34-36 依據本文所述的各種實施例示出可使用一或多個 IP 核心來製造的示例性積體電路及相關聯的圖形處理器。除了示出的以外，還可包括其他的邏輯和電路，其包括附加的圖形處理器/核心、週邊介面控制器、或通用處理器核心。

【0312】圖 34 是示出依據實施例之可使用一或多個 IP 核心來製造的示例性系統單晶片積體電路 3400 的方塊圖。示例性積體電路 3400 包括一或多個應用程式處理器 3405 (例如，CPU)、至少一個圖形處理器 3410、且可額外包括影像處理器 3415 及/或視訊處理器 3420，其中任一者可為來自相同或多個不同設計設施的模組化 IP 核心。積體電路 3400 包括週邊或匯流排邏輯，其包括 USB 控制器 3425、UART 控制器 3430、SPI/SDIO 控制器 3435、及 I²S/I²C 控制器 3440。另外，積體電路可包括顯示裝置 3445，其耦接至高解析度多媒體介面 (HDMI) 控制器 3450 及行動產業處理器介面 (MIPI) 顯示介面 3455 中的一或多者。儲存可由包括快閃記憶體及快閃記憶體控制器的快閃記憶體子系統 3460 提供。記憶體介面可經由記憶體控制器 3465 提供，用於存取 SDRAM 或 SRAM 記憶體裝置。一些積體電路另外包括嵌入式安全引擎 3470。

【0313】圖 35 是示出依據實施例之可使用一或多個 IP 核心來製造的系統單晶片積體電路的示例性圖形處理器 3510 的方塊圖。圖形處理器 3510 可以是圖 34 之圖形處理器

3410的變形。圖形處理器3510包括頂點處理器3505及一或多個片段處理器3515A-3515N(例如, 3515A、3515B、3515C、3515D、至3515N-1、及3515N)。圖形處理器3510可透過獨立的邏輯執行不同的著色器程式,使得頂點處理器3505被最佳化以執行頂點著色器程式的操作,而一或多個片段處理器3515A-3515N執行片段的片段(例如,像素)著色操作或像素著色器程式。頂點處理器3505執行3D圖形管線的頂點處理階段,並產生基元和頂點資料。片段處理器3515A-3515N使用頂點處理器3505產生的基元和頂點資料來產生被顯示在顯示裝置上的幀緩衝器。在一個實施例中,將片段處理器3515A-3515N最佳化以執行OpenGL API中提供的片段著色器程式,其可被用來執行與Direct 3D API中提供的像素著色器程式類似的操作。

【0314】圖形處理器3510另外包括一或多個記憶體管理單元(MMU)3520A-3520B、快取3525A-3525B、及電路互連3530A-3530B。一或多個MMU 3520A-3520B針對圖形處理器3510,包括針對頂點處理器3505及/或片段處理器3515A-3515N,提供虛擬至實體位址映射,除了儲存在一或多個快取3525A-3525B中的頂點或影像/紋理資料之外,其還可參考儲存在記憶體中的頂點或影像/紋理資料。在一個實施例中,一或多個MMU 3520A-3520B可以與系統內的其他MMU同步,該系統包括與圖34之一或多個應用處理器3405、影像處理器3415、及/或視訊處理器3420相關聯的一或多個MMU,使得每個處理器3405-3420可參與共

享或統一的虛擬記憶體系統。依據實施例，一或多個電路互連 3530A-3530B 使圖形處理器 3510 能夠透過 SoC 的內部匯流排或者透過直接連接與 SoC 內的其他 IP 核心介接。

【0315】圖 36 是示出依據實施例之可使用一或多個 IP 核心來製造的系統單晶片積體電路之另外的示例性圖形處理器 3610 的方塊圖。圖形處理器 3610 可以是圖 34 之圖形處理器 3410 的變形。圖形處理器 3610 包括圖 35 之積體電路 3510 的一或多個 MMU 3520A-3520B、快取 3525A-3525B、及電路互連 3530A-3530B。

【0316】圖形處理器 3610 包括一或多個著色器核心 3615A-3615N(例如，3615A、3615B、3615C、3615D、3615E、3615F、至 3615N-1、及 3615N)，其提供統一的著色器核心架構，其中單核心或類型或核心可執行所有類型的可編程著色器碼，包括實現頂點著色器、片段著色器、及/或計算著色器的著色器程式碼。實施例和實現方式中存在的著色器的確切數量可以不同。此外，圖形處理器 3610 包括核心間(inter-core)任務管理器 3605，其充當線程分派器以將執行線程分派給一或多個著色器核心 3615A-3615N 及拼貼(tiling)單元 3618 以加速用於基於圖磚渲染的拼貼操作，其中針對場景的渲染操作在影像空間中被細分，例如利用場景內的本地空間一致性或是最佳化內部快取的利用。

【0317】以下子句及/或範例與特定實施例或其範例有關。可在一或多個實施例中的任何地方使用範例中的細

節。不同實施例或範例的各種特徵可與所包括的某些特徵以及被排除在外的其他特徵進行各種組合，以適應各種不同的應用。範例可包括諸如方法、執行該方法之動作的機構、包括指令的至少一個機器可讀取媒體，當由機器執行時該些指令，使該機器執行該方法的動作、或者依據本文所述之實施例和範例的裝置或系統的標的。各種組件可以是用於執行所述運算或功能的機構。

【0318】 本文所述之實施例是指被配置成執行某些運算或具有預定功能的硬體的特別配置，諸如特定應用積體電路(ASIC)。此種電子裝置通常包括耦接至一或多個其他組件的一或多個處理器的集合，諸如一或多個儲存裝置(非暫態機器可讀取儲存媒體)、使用者輸入/輸出裝置(例如，鍵盤、觸控螢幕、及/或顯示器)、及網路連接。處理器的集合與其他組件的耦接通常透過一或多個匯流排及橋接器(也稱為匯流排控制器)。儲存裝置和乘載網路流量的信號分別代表一或多個機器可讀取儲存媒體和機器可讀取通訊媒體。因此，給定的電子裝置的儲存裝置通常儲存用於在該電子裝置的一或多個處理器的集合上執行的程式碼及/或資料。

【0319】 當然，可使用軟體、韌體、及/或硬體的不同組合來實現實施例的一或多個部分。在整個詳細描述中，為了解釋的目的，闡述了許多具體細節以便提供對本發明的透徹理解。然而，對於本領域之技術人員而言將顯而易見的是，可在沒有這些具體細節中的一些的情況下實

施這些實施例。在某些情況下，未詳細闡述公知的結構和功能以避免模糊實施例的發明標的。因此，本發明的範圍和精神應依據所附的申請專利範圍來判斷。

【符號說明】

【0320】

100：計算系統
101：處理子系統
102：處理器
104：系統記憶體
105：記憶體集線器
106：通訊鏈路
107：I/O集線器
108：輸入裝置
110A：顯示裝置
110B：顯示裝置
111：I/O子系統
112：平行處理器
113：通訊鏈路
114：系統儲存單元
116：I/O開關
118：網路配接器
119：無線網路配接器
120：附加裝置

第 107105950 號

民國 111 年 4 月 11 日修正

200：平行處理器
202：平行處理單元
204：I/O單元
206：主機介面
208：前端
210：排程器
212：處理陣列
214A-214N：叢集
216：記憶體交叉
218：記憶體介面
220A-220N：分割單元
221：L2快取
222：平行處理器記憶體
224A-224N：記憶體單元
225：幀緩衝器介面
226：光柵操作單元(ROP)
232：管線管理器
234：圖形多處理器
236：紋理單元
240：資料交叉
242：預光柵操作單元(PreROP)
245：記憶體管理單元(MMU)
248：L1快取
252：指令快取

第 107105950 號

民國 111 年 4 月 11 日修正

254：指令單元

256：地址映射單元

258：暫存器檔案

262：通用圖形處理單元(GPGPU)核心

266：加載/儲存單元

268：記憶體及快取互連

270：共用記憶體

272：快取記憶體

325：圖形多處理器

327：互連結構

330：共通指令快取

332A-332B：指令單元

334A-334B：暫存器檔案

336A-336B：GPGPU核心

337A-337B：GPGPU核心

338A-338B：GPGPU核心

340A-340B：加載/儲存單元

342：紋理及/或資料快取記憶體

344A-344B：紋理單元

346：共用記憶體

350：圖形多處理器

352：互連結構

354：指令快取

356A-356D：執行資源

358A-358B：紋理及/或資料快取記憶體

360A-360D：紋理單元

362：共用記憶體

401-402：處理器記憶體

405-406：多核心處理器

407：多核心處理器

410-413：圖形處理單元(GPU)

420-423：GPU記憶體

425：代理電路

430A-430B：記憶體互連

431-432、N：圖形處理引擎

433-434、M：圖形記憶體

435：介面

436：加速器整合電路

437：介面

438：快取

439：MMU

440：高速鏈路

440A-440D：高速鏈路

441：系統記憶體

442A-442B：高速鏈路

443：高速鏈路

444：提取

445：暫存器

446：圖形加速模組
447：中斷管理電路
448：上下文管理電路
450A-450D：GPU記憶體互連
456：共用快取
464：一致性匯流排
460A-460D：核心
461A-461D：轉譯後備緩衝區
462A-462D：快取
480：應用程式
481：GPU調用
482：應用程式有效位址空間
483：程序元件
484：工作描述符(WD)
485：OS虛擬位址空間
486：區段/頁表
490：加速器整合片
491：WD提取單元
492：中斷事件
493：有效位址
494A-494E：偏置/一致性管理電路
495：作業系統
496：系統超管理器
498：超管理器真實位址空間

499：程序元件表

500：圖形處理管線

502：資料組合器

504：頂點處理單元

506：基元組合器

508：曲面細分控制處理單元

510：曲面細分單元

512：曲面細分評估處理單元

514：基元組合器

516：幾何處理單元

518：基元組合器

520：視埠縮放、剔除及剪裁單元

522：光柵器

524：片段/像素處理單元

526：光柵操作單元

528：記憶體介面

602：機器學習應用

604：機器學習框架

606：計算框架

608：GPGPU驅動器

610：GPGPU硬體

700：高度平行通用圖形處理單元

702：主機介面

704：全域排程器

第 107105950 號

民國 111 年 4 月 11 日修正

706A-706H：計算叢集

708：快取記憶體

709：I/O集線器

710：GPU鏈路

712A-712B：記憶體控制器

714A-714B：記憶體

800：多GPU計算系統

802：處理器

804：主機介面開關

806A-806D：GPGPU

816：P2P GPU鏈路

902：輸入

904：卷積層

906：卷積層

908：完全連接層

1000：遞迴神經網路

1002：輸入層

1004：隱藏層

1005：反饋機制

1006：輸出層

1102：訓練資料集

1104：訓練框架

1106：未經訓練的神經網路

1108：經訓練的神經網路

第 107105950 號

民國 111 年 4 月 11 日修正

1112：新資料

1114：結果

1202：模型平行處理

1204：資料平行處理

1206：組合的模型及資料平行處理

1300：推論系統單晶片(SOC)

1302：媒體處理器

1304：視覺處理器

1305：晶載記憶體

1306：GPGPU

1308：多核心處理器

1400：動態精度浮點單元

1402：控制單元

1404：內部暫存器

1406：指數塊

1408：有效數塊

1412：精度追蹤邏輯

1414：運算元暫存器

1416：比較器

1418：動態精度乘法器

1422：數值變換單元

1424：中間暫存器

1426：動態精度指數加法器

1428：移位單元

1434：誤差累加器
1438：動態精度有效數加法器
1448：累加暫存器
1502：輸入緩衝器
1504：溢位乘法器
1506：乘法器
1508：精度暫存器
1510：輸出緩衝器
1512：8位元輸入緩衝器
1514：溢位加法器
1516：加法器
1518：精度暫存器
1520：8位元輸出緩衝器
1522：24位元/32位元輸入緩衝器
1524：溢位加法器
1526：加法器
1528：精度暫存器
1530：24位元/32位元輸出緩衝器
1600：動態精度處理系統
1606A-1606D：操作線程
1608A-1608D：動態浮點單元
1700：邏輯
1801A-1801F：64位元組塊
1802：剝離迴圈

1804：向量化的主迴圈

1806：剩餘迴圈

1808A-1808C：FP16 SIMD8單元

1812：執行遮罩

1814：執行遮罩

1816：執行遮罩

1900：線程處理系統

1902：SIMD線程

1904：SIMD線程

1906：SIMT線程組

1910：SIMD4槽

1912：SIMD4槽

1920：SIMD8浮點單元

1922A-1922H：動態浮點單元

2000：邏輯

2100：深度神經網路

2102A-2102N：神經網路層

2104A-2104B：FMA邏輯

2106A-2106B：正規化及變換邏輯

2200：邏輯

2300：處理系統

2302：處理器

2304：快取記憶體

2306：暫存器檔案

第 107105950 號

民國 111 年 4 月 11 日修正

2307：處理器核心

2308：圖形處理器

2309：指令集

2310：處理器匯流排

2312：外部圖形處理器

2316：記憶體控制器集線器

2320：記憶體裝置

2321：指令

2322：資料

2324：資料儲存裝置

2326：無線收發器

2328：韌體介面

2330：I/O控制器集線器

2334：網路控制器

2340：舊有 I/O 控制器

2342：通用序列匯流排(USB)控制器

2344：鍵盤和滑鼠

2346：音訊控制器

2400：處理器

2402A-2402N：核心

2404A-2404N：內部快取單元

2406：共用快取單元

2408：圖形處理器

2410：系統代理核心

2411：顯示控制器
2412：基於環形之互連單元
2414：整合式記憶體控制器
2416：匯流排控制器單元
2418：嵌入式記憶體模組
2500：圖形處理器
2502：顯示控制器
2504：區塊影像轉移(BLIT)引擎
2506：視訊編解碼引擎
2510：圖形處理引擎(GPE)
2512：3D管線
2514：記憶體介面
2515：3D/媒體子系統
2516：媒體管線
2520：顯示裝置
2603：命令串流器
2610：圖形處理引擎
2614：圖形核心陣列
2618：統一返回緩衝器(URB)
2620：共用功能邏輯
2621：取樣器
2622：數學
2623：線程間通訊(ITC)
2625：快取

第 107105950 號

民國 111 年 4 月 11 日修正

2700：圖形處理器
2702：環形互連
2703：命令串流器
2704：管線前端
2730：視訊品質引擎(VQE)
2733：多格式編碼/解碼(MFX)
2734：視訊前端
2736：幾何管線
2737：媒體引擎
2750A-2750N：子核心
2752A-2752N：執行單元
2754A-2754N：媒體/紋理取樣器
2760A-2760N：子核心
2762A-2762N：執行單元
2764A-2764N：取樣器
2770A-2770N：共用資源
2780A-2780N：圖形核心
2800：線程執行邏輯
2802：著色器處理器
2804：線程分派器
2806：指令快取
2808A-2808N：執行單元
2810：取樣器
2812：資料快取

2814：資料埠

2900：圖形處理器指令格式

2910：128位元指令格式

2912：指令操作碼

2913：索引欄位

2914：指令控制欄位

2916：執行大小欄位

2918：目的地

2926：存取/定址模式欄位

2930：64位元壓縮指令格式

2940：操作碼解碼

2942：移動及邏輯操作碼群組

2944：流量控制指令群組

2946：雜項指令群組

2948：平行數學指令群組

2950：向量數學群組

3000：圖形處理器

3002：環形互連

3003：命令串流器

3005：頂點提取器

3007：頂點著色器

3011：輪廓著色器

3013：曲面細分器

3017：域著色器

3019：幾何著色器
3020：圖形管線
3023：串流輸出單元
3029：剪裁器
3030：媒體管線
3031：線程分派器
3034：視訊前端
3037：媒體引擎
3040：顯示引擎
3041：2D引擎
3043：顯示控制器
3050：線程執行邏輯
3051：L1快取
3052A-3052B：執行單元
3054：紋理及媒體取樣器
3056：資料埠
3058：紋理/取樣器快取
3070：渲染輸出管線
3073：光柵化及深度測試元件
3075：共用L3快取
3077：像素操作組件
3078：渲染快取
3079：深度快取
3100：圖形處理器命令格式

第 107105950 號

民國 111 年 4 月 11 日修正

3102：客戶端

3104：命令操作碼(opcode)

3105：子操作碼

3106：相關資料

3108：命令大小

3200：資料處理系統

3210：3D圖形應用程式

3212：著色器指令

3214：可執行指令

3216：圖形物件

3220：作業系統

3222：圖形API

3224：前端著色器編譯器

3226：使用者模式圖形驅動器

3227：後端著色器編譯器

3228：作業系統核心模式功能

3229：核心模式圖形驅動器

3230：處理器

3232：圖形處理器

3234：通用處理器核心

3250：系統記憶體

3300：IP核心開發系統

3310：軟體模擬

3312：模擬模型

3315：暫存器傳送層級(RTL)設計

3320：硬體模型

3330：設計設施

3340：非揮發性記憶體

3350：有線連接

3360：無線連接

3365：製造設施

3400：系統單晶片積體電路

3405：應用程式處理器

3410：圖形處理器

3415：影像處理器

3420：視訊處理器

3425：USB控制器

3430：UART控制器

3435：SPI/SDIO控制器

3440：I²S/I²C控制器

3445：顯示裝置

3450：高解析度多媒體介面(HDMI)控制器

3455：行動產業處理器介面(MIPI)顯示介面

3460：快閃記憶體子系統

3465：記憶體控制器

3470：嵌入式安全引擎

3505：頂點處理器

3510：圖形處理器

第 107105950 號

民國 111 年 4 月 11 日修正

3515A-3515N：片段處理器

3520A-3520B：記憶體管理單元(MMU)

3525A-3525B：快取

3530A-3530B：電路互連

3605：核心間任務管理器

3610：圖形處理器

3615A-3615N：著色器核心

3618：拼貼單元

【發明申請專利範圍】

【第1項】

一種通用圖形處理單元，包括：

動態精度浮點單元，包括控制單元，其具有精度追蹤硬體邏輯以追蹤相對於目標精度之用於計算資料的精度的可用位元數，其中該動態精度浮點單元包括計算邏輯，用來以多個精度輸出資料，以及其中該動態精度浮點單元包括有效數(significand)塊，用來執行浮點計算的有效數部分，該有效數塊包括動態精度加法器，可配置成以多個精度加(add)或減(subtract)輸入資料。

【第2項】

如請求項1之通用圖形處理單元，其中該動態精度浮點單元包括一組暫存器，用來以多個精度儲存輸入資料和中間資料。

【第3項】

如請求項2之通用圖形處理單元，其中該組暫存器包括誤差累加器，用來追蹤一組浮點運算過程中累積的誤差。

【第4項】

如請求項1之通用圖形處理單元，其中該有效數塊包括動態精度乘法器，可配置成以多個精度加或乘(multiply)或除(divide)輸入資料。

【第5項】

如請求項4之通用圖形處理單元，其中該動態精度浮

第 107105950 號

民國 111 年 4 月 11 日修正

點單元包括指數塊，用來執行浮點計算的指數部分，該指數塊包括動態精度加法器，可配置成以多個精度加或減輸入資料的指數。

【第6項】

如請求項5之通用圖形處理單元，其中該指數塊及該有效數塊用執行第一浮點運算，以輸出具有16位元精度的第一輸出值。

【第7項】

如請求項6之通用圖形處理單元，其中該指數塊及該有效數塊執行第二運算，以輸出具有32位元精度的第二輸出值。

【第8項】

如請求項7之通用圖形處理單元，其中該指數塊及該有效數塊對具有32位元值的輸入資料執行第三浮點運算，以輸出具有32位元資料類型的第三輸出值，該第三輸出值以16位元精度產生。

【第9項】

如請求項8之通用圖形處理單元，其中該指數塊包括8位元乘法器，並且其中該指數塊及該有效數塊被配置以執行雙8位元整數運算。

【第10項】

一種在通用圖形處理單元之硬體內執行可變精度操作的方法，該方法包括：

接收以第一精度執行數值運算的請求；

使用與低於該第一精度之第二精度相關聯的位元數來執行該數值運算；

以第二精度產生中間結果；

確定該中間結果的精度損失；以及

當該中間結果的該精度損失低於閾值時，以該第二精度輸出結果。

【第11項】

如請求項10之方法，其中該閾值可透過軟體邏輯進行配置。

【第12項】

如請求項10之方法，其中該閾值是預設的硬體值。

【第13項】

如請求項10之方法，還包括：

當該中間結果的該精度損失高於該閾值時，計算結果的剩餘位元；以及

以該第一精度輸出該結果。

【第14項】

如請求項13之方法，其中使用與低於該第一精度之該第二精度相關聯的位元數來執行該數值運算包括使用第一組邏輯單元來執行該數值運算，以及當該中間結果的該精度損失高於該閾值時計算結果的剩餘位元包括使用第二組邏輯單元來計算結果的剩餘位元。

【第15項】

一種資料處理系統，包括：

非暫時性機器可讀取媒體，用來儲存由該資料處理系統之一或多個處理器執行的指令；以及

通用圖形處理單元，包含動態精度浮點單元，其包括控制單元，具有精度追蹤硬體邏輯以追蹤相對於目標精度之用於計算資料的精度的可用位元數，其中該動態精度浮點單元包括計算邏輯，用來以多個精度輸出資料，以及其中該動態精度浮點單元包括有效數(significand)塊，用來執行浮點計算的有效數部分，該有效數塊包括動態精度加法器，可配置成以多個精度加(add)或減(subtract)輸入資料。

【第16項】

如請求項15之資料處理系統，其中該動態精度浮點單元包括一組暫存器，用來以多個精度儲存輸入資料和中間資料。

【第17項】

如請求項16之資料處理系統，其中該組暫存器包括誤差累加器，用來追蹤一組浮點運算過程中累積的誤差。

【第18項】

如請求項15之資料處理系統，其中該有效數塊包括動態精度乘法器，可配置成以多個精度加或乘(multiply)或除(divide)輸入資料。

【發明圖式】

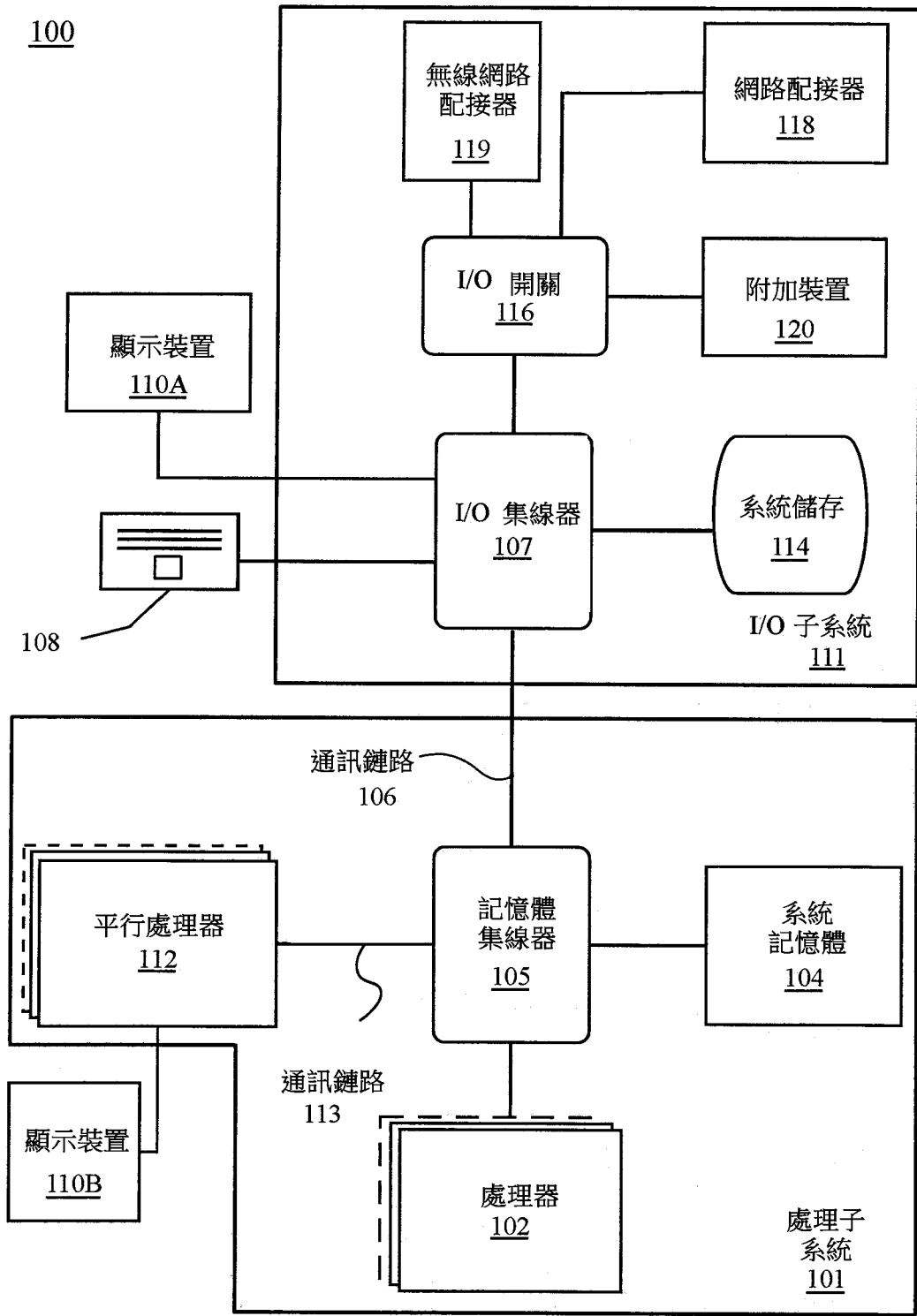


圖 1

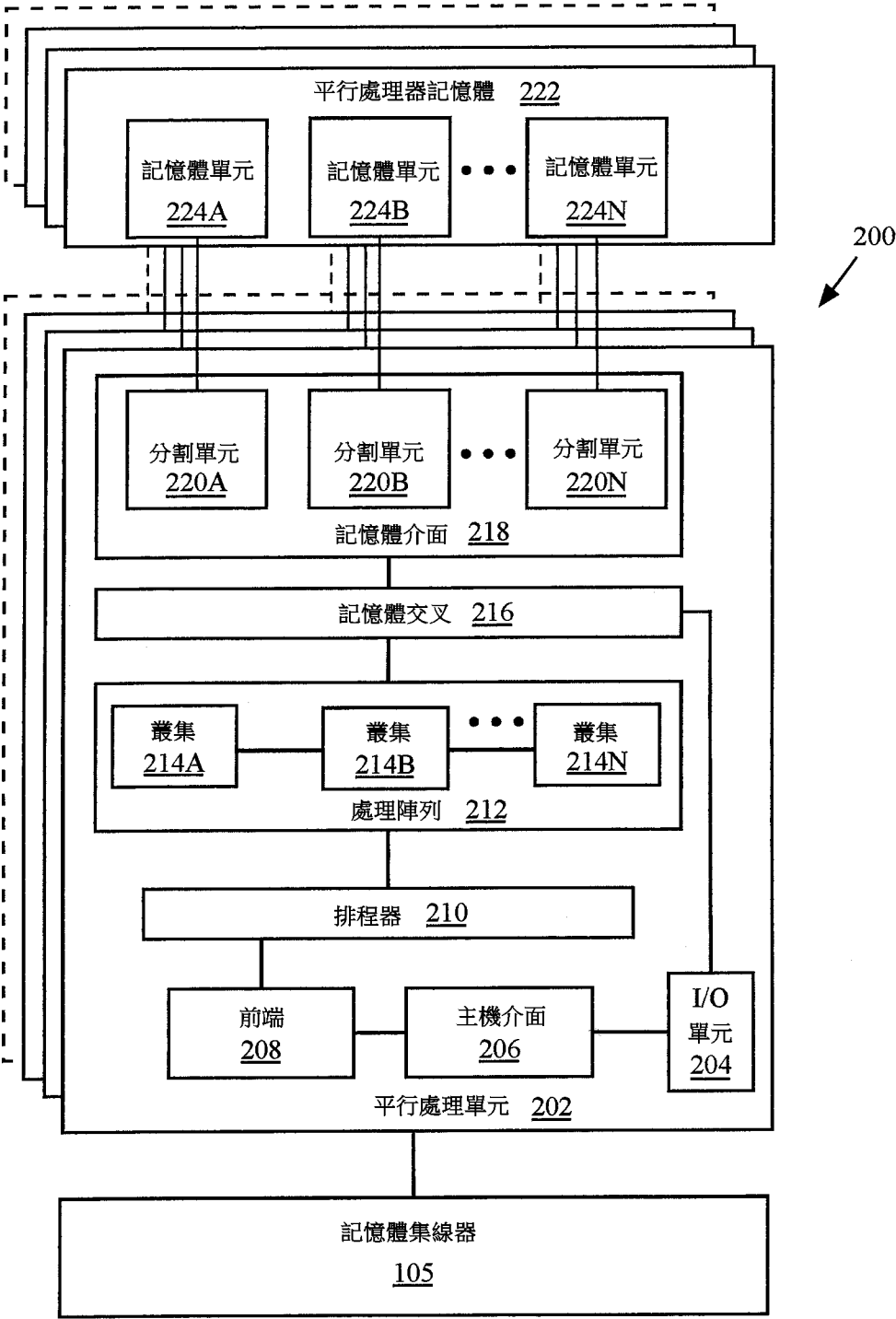


圖 2A

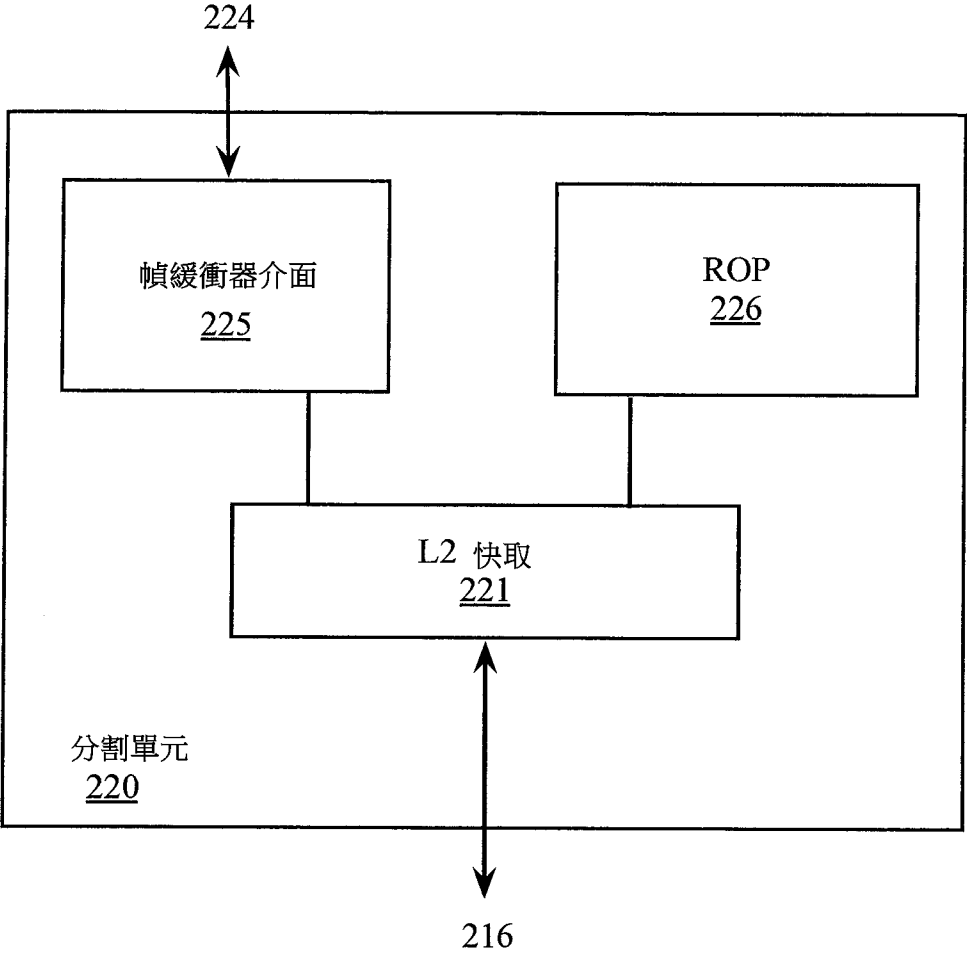


圖 2B

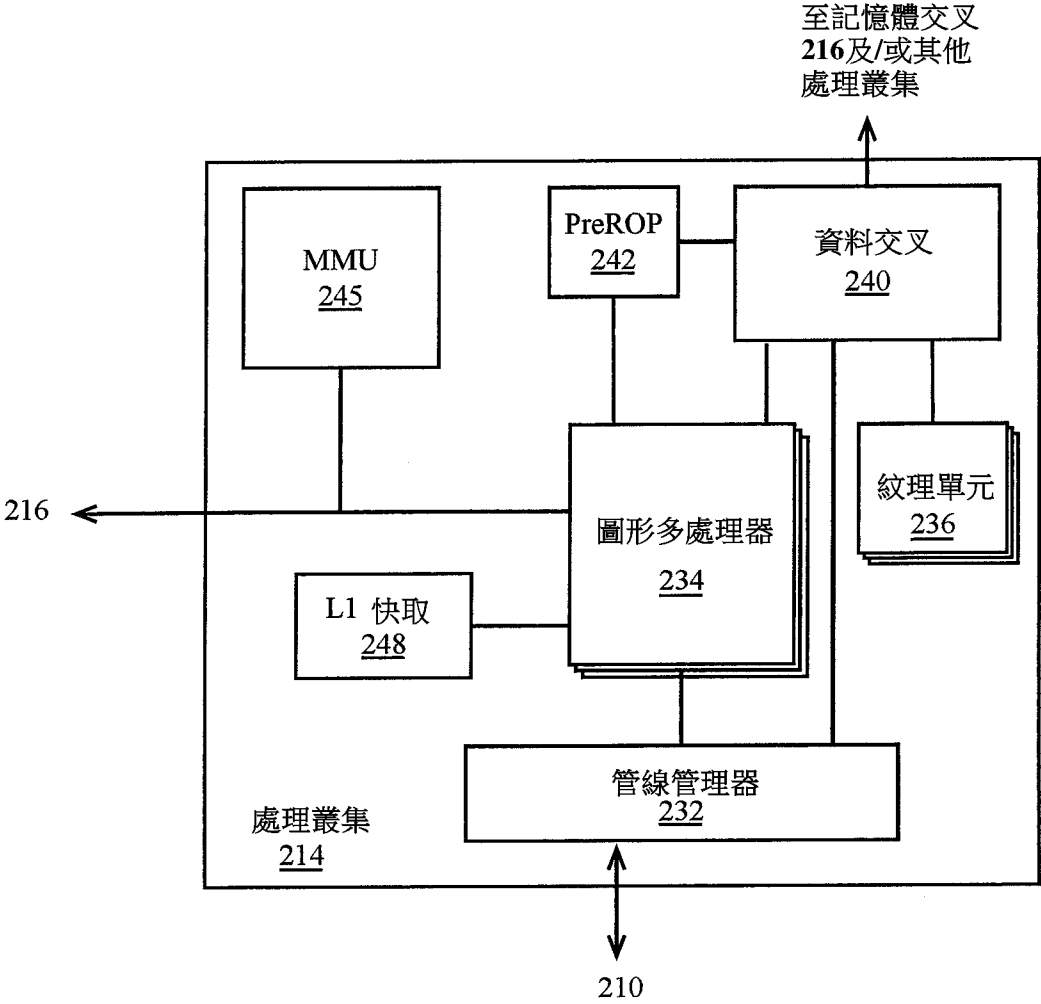


圖 2C

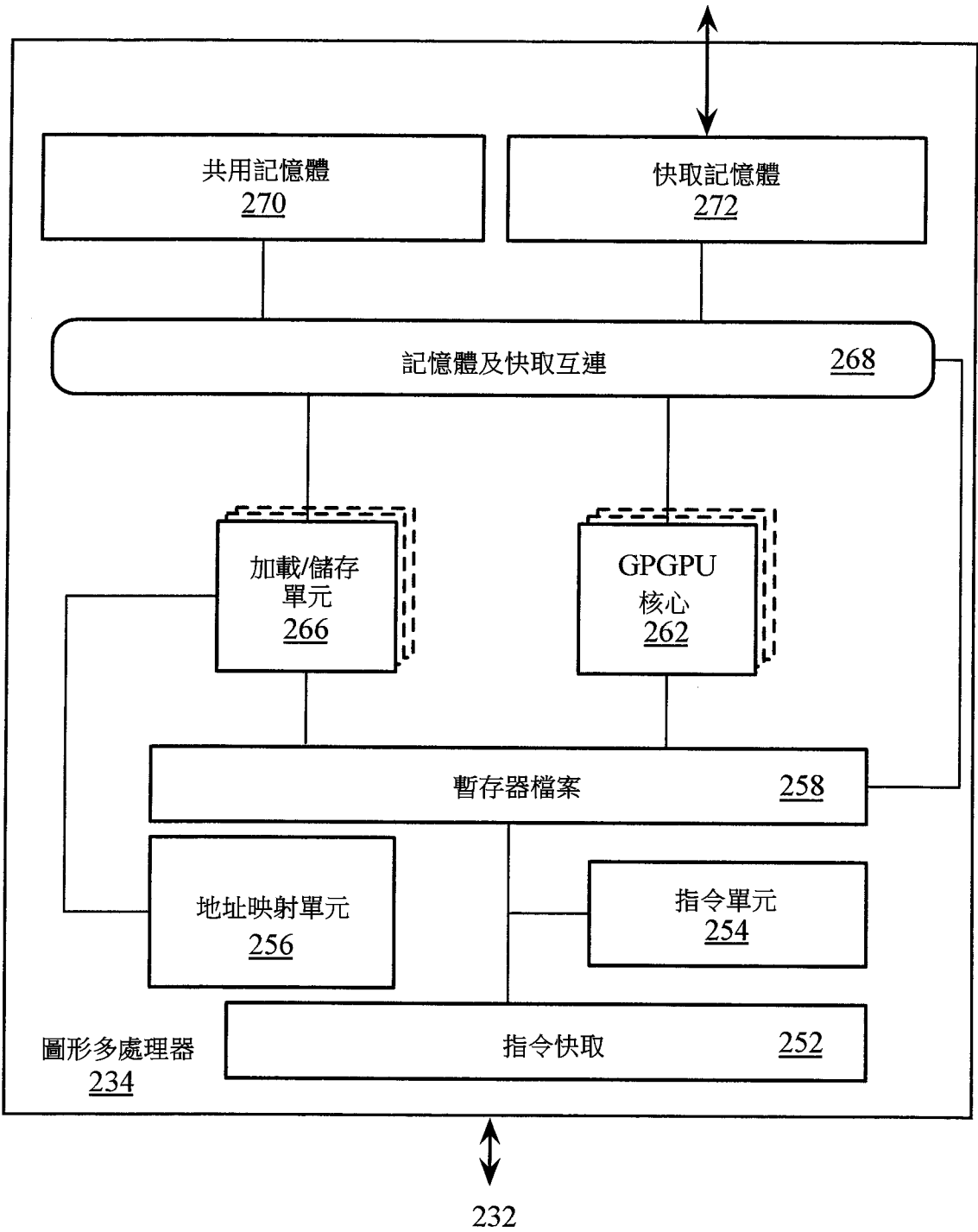


圖 2D

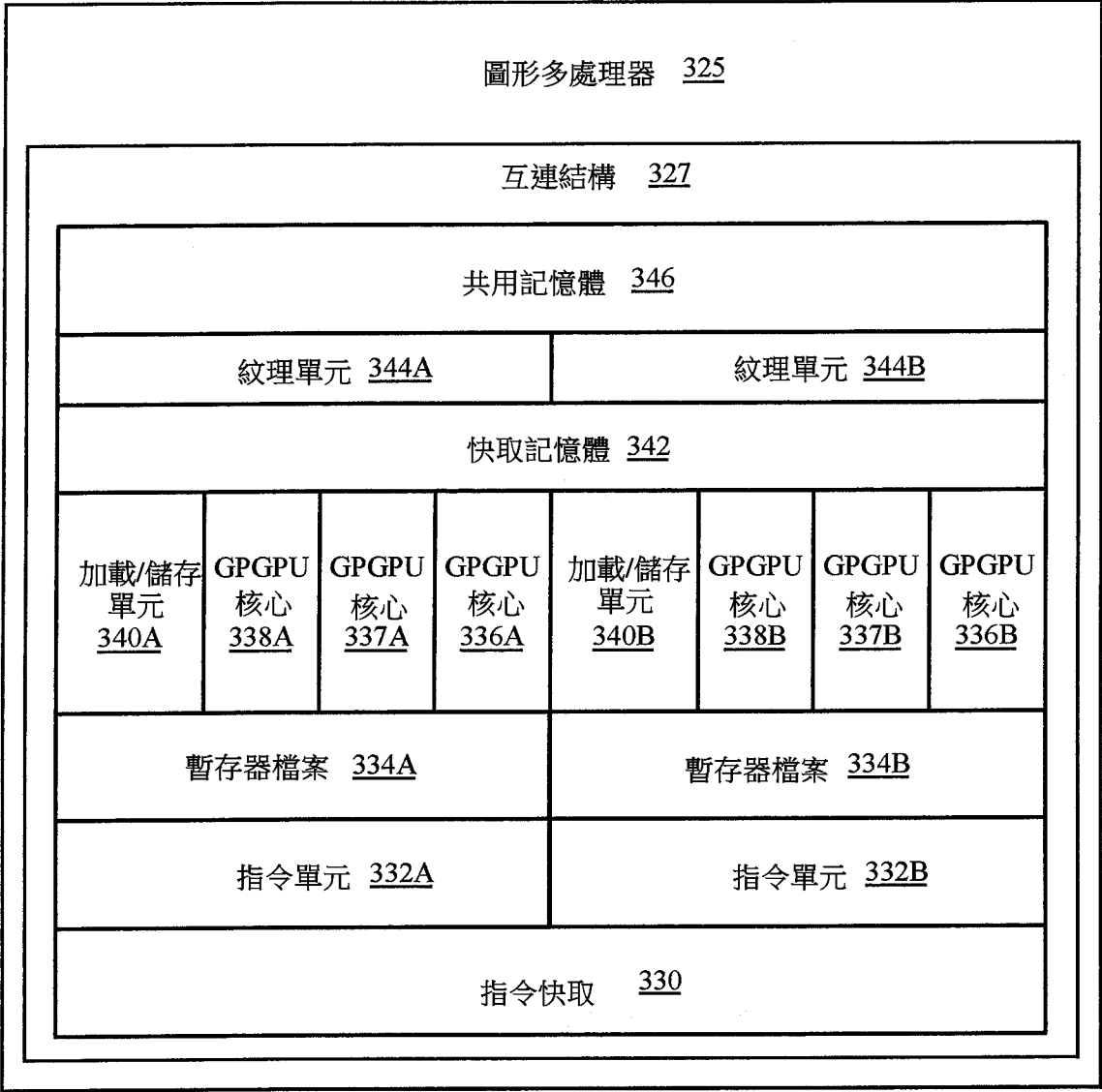


圖 3A

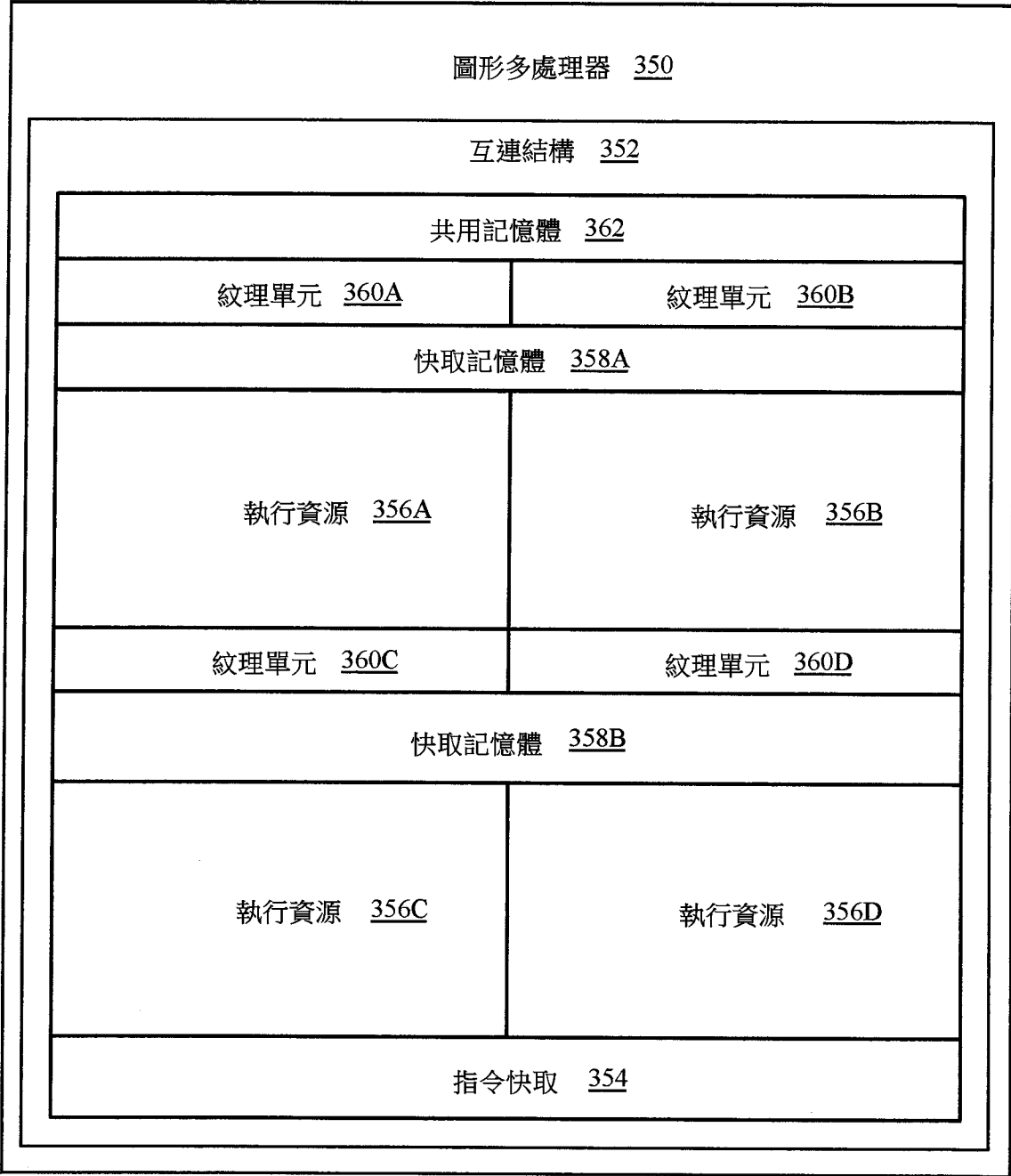


圖 3B

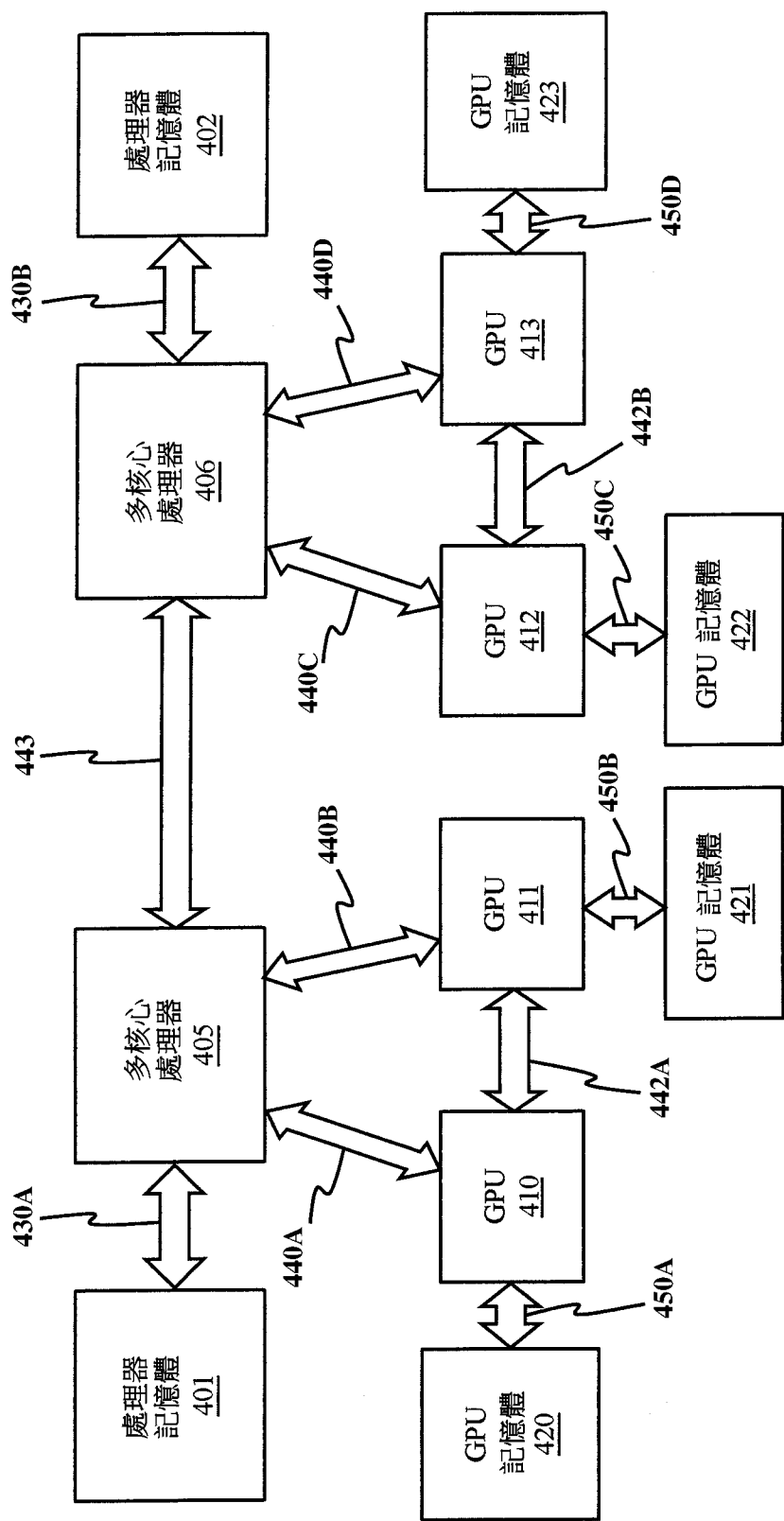


圖 4A

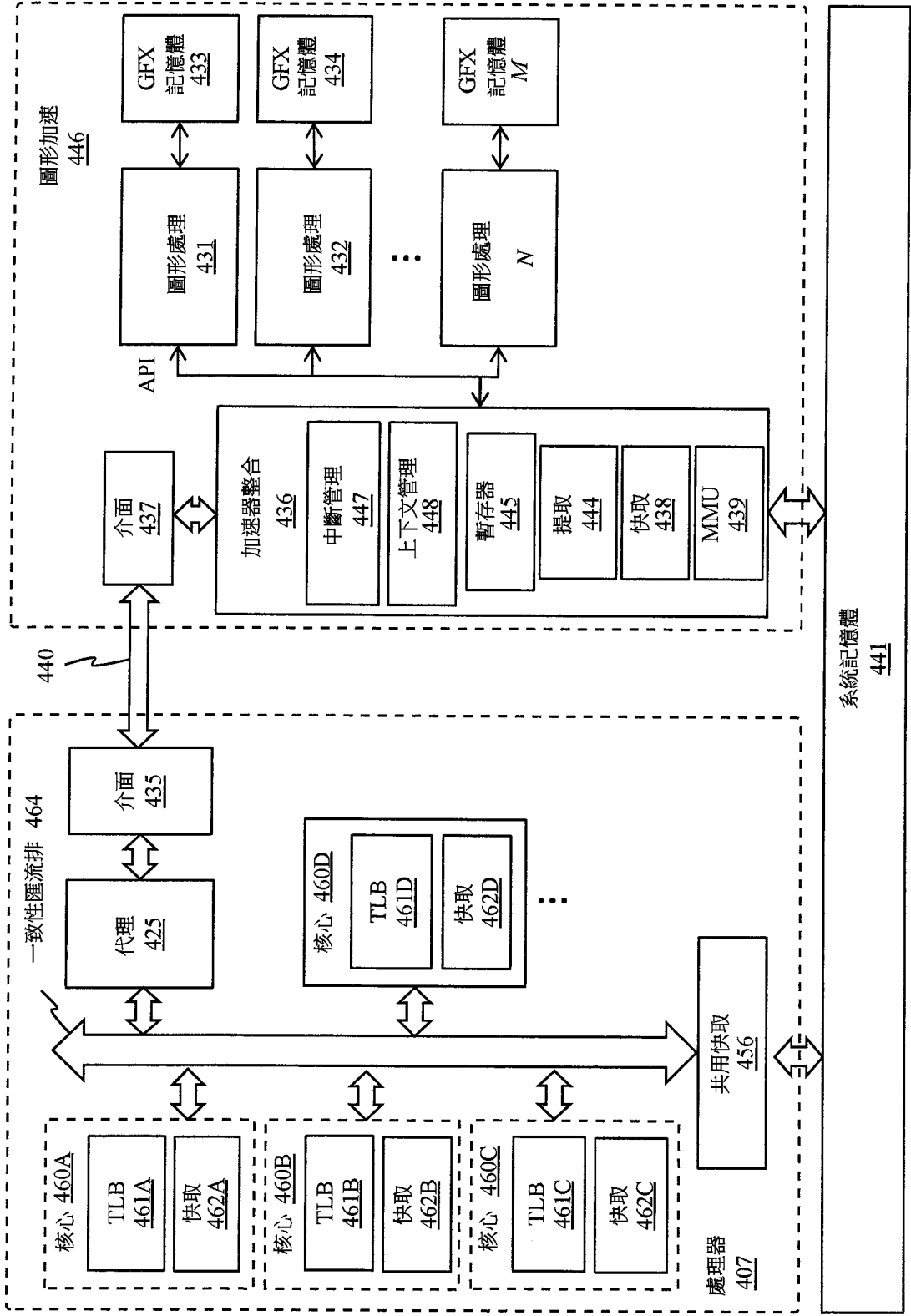


圖 4B

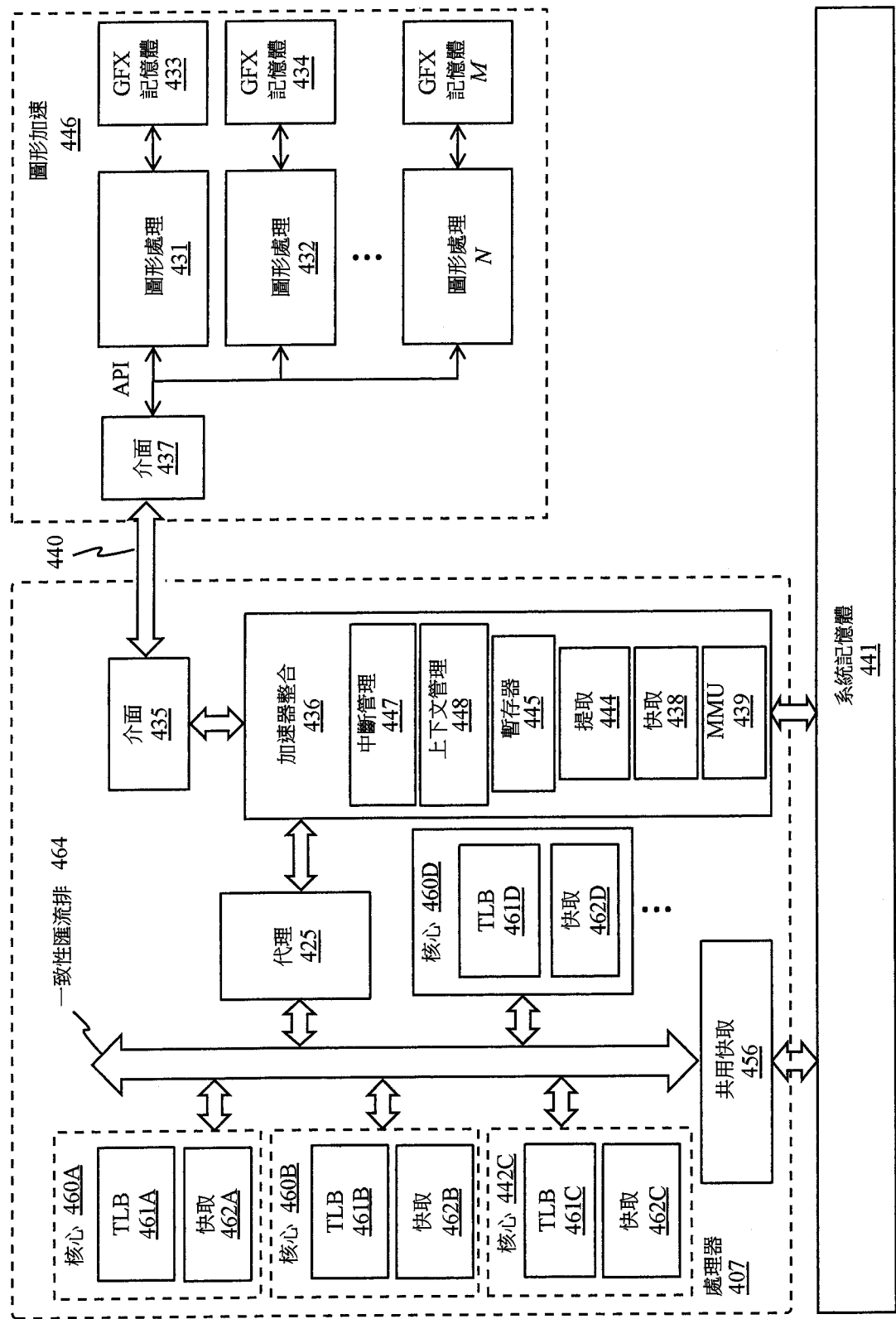


圖 4C

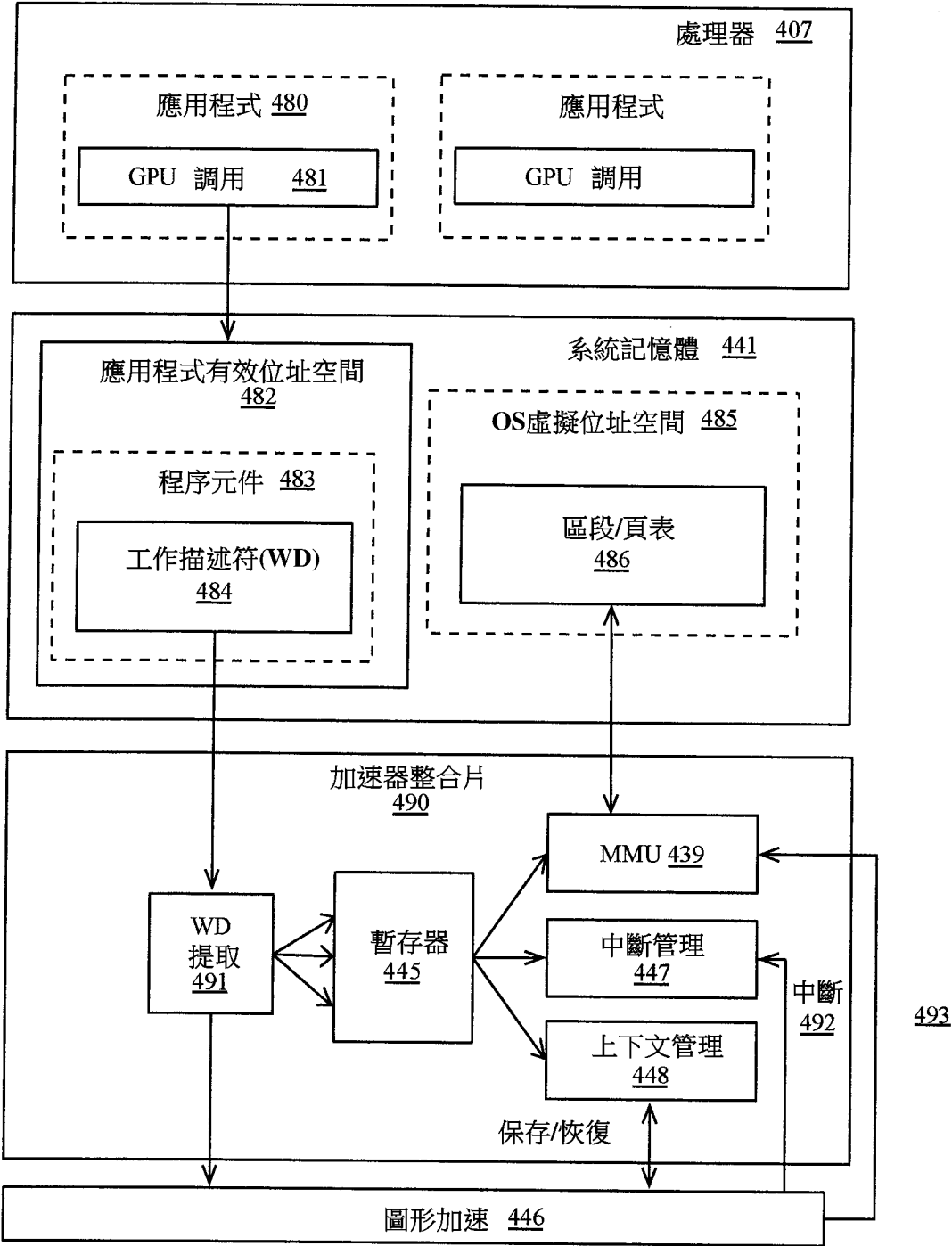


圖 4D

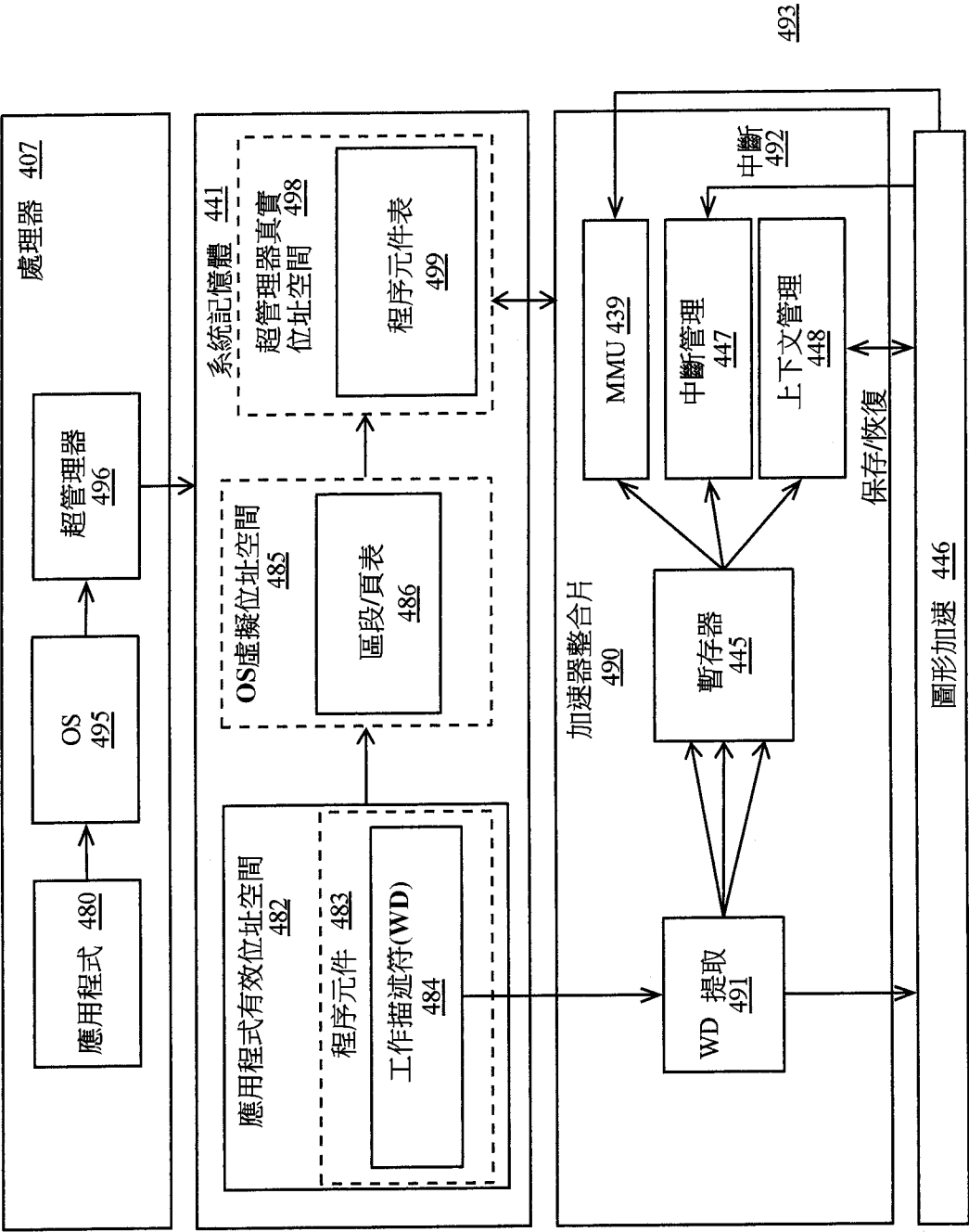


圖 4E

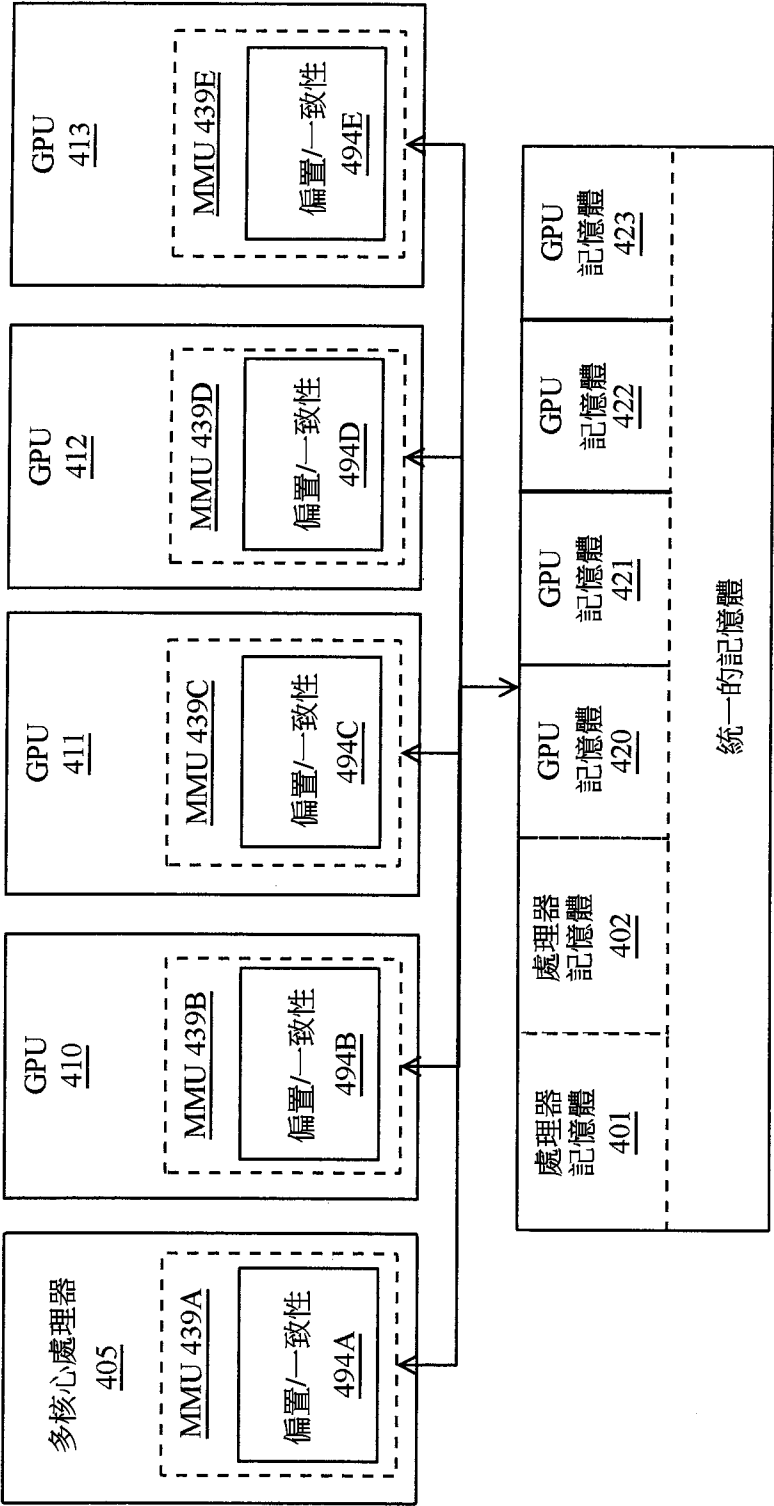


圖 4F

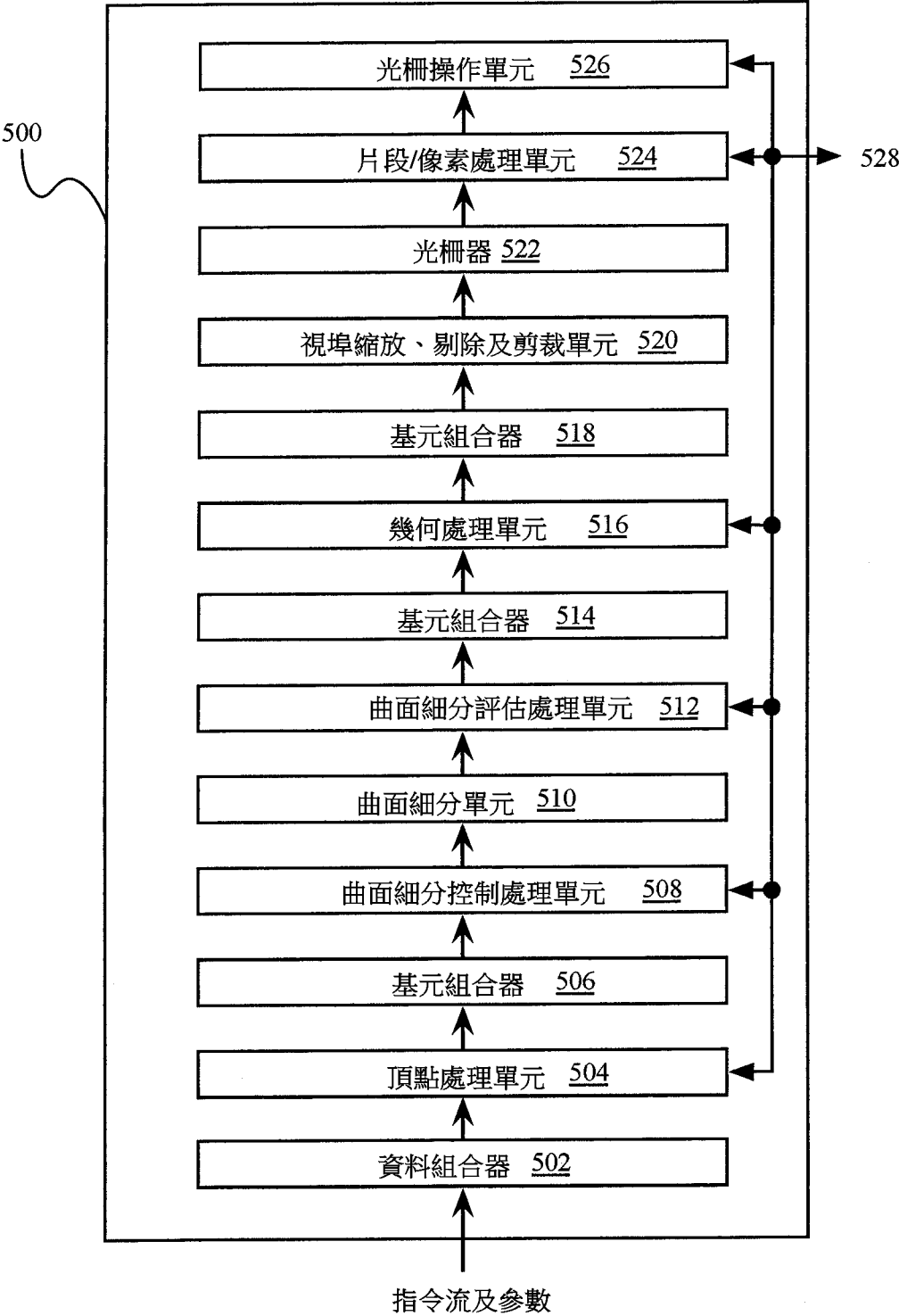


圖 5

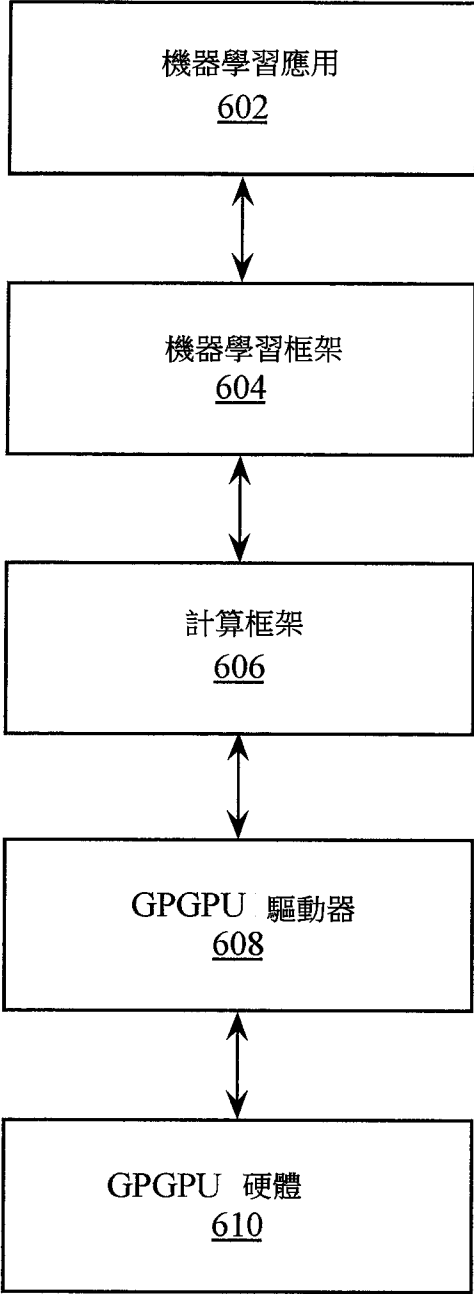


圖 6

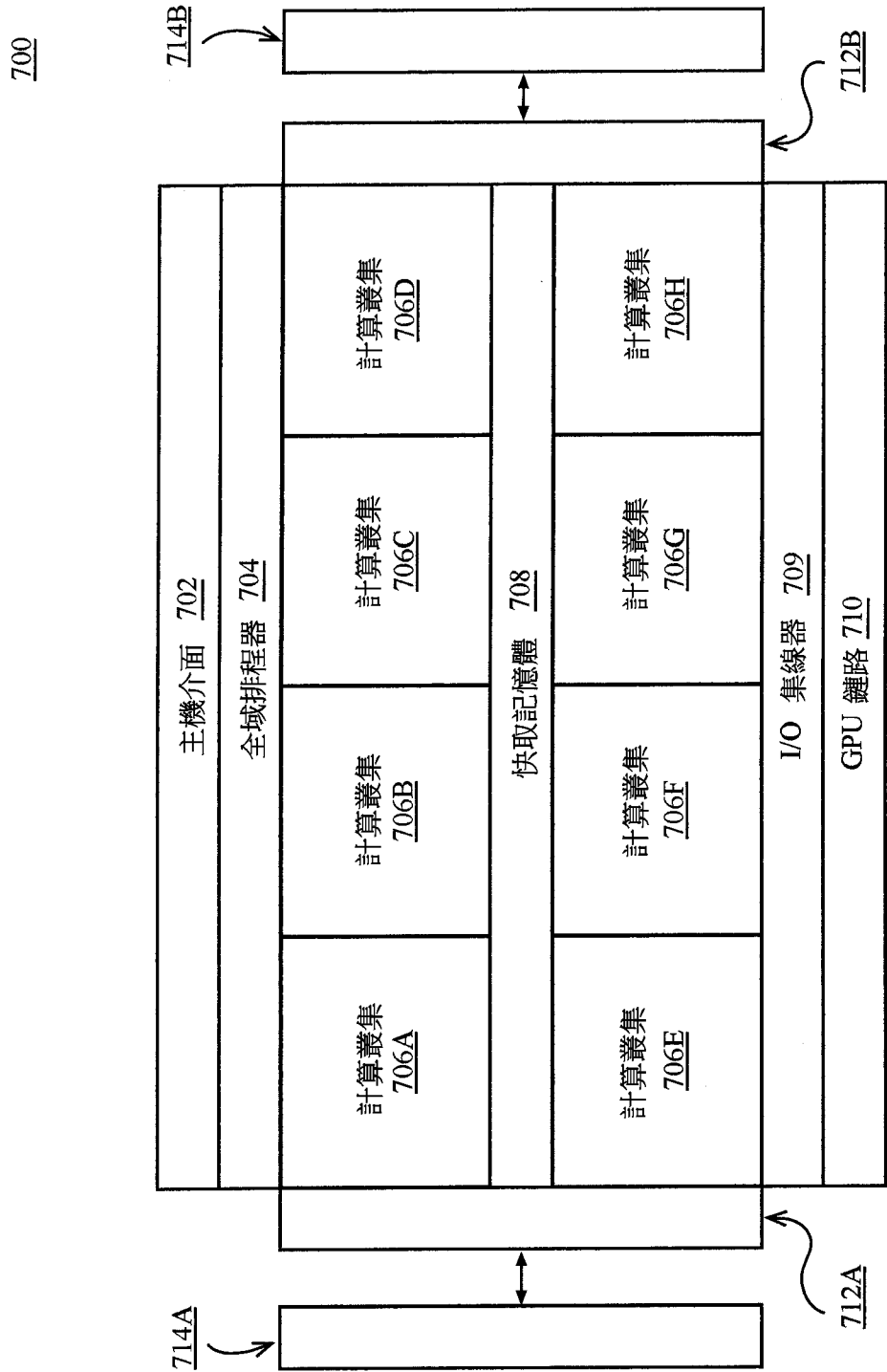


圖 7

800

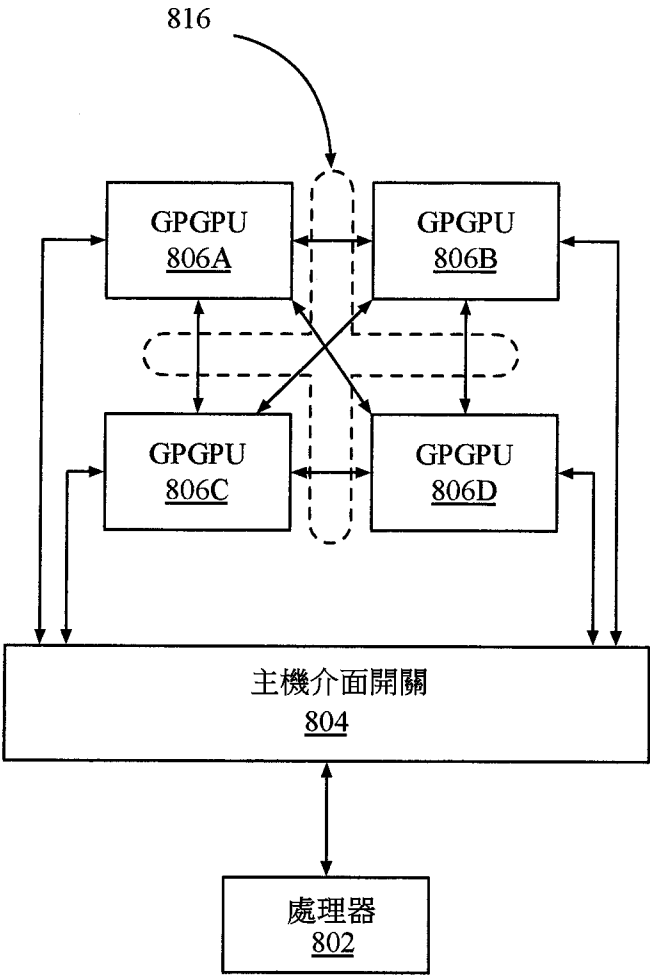


圖 8

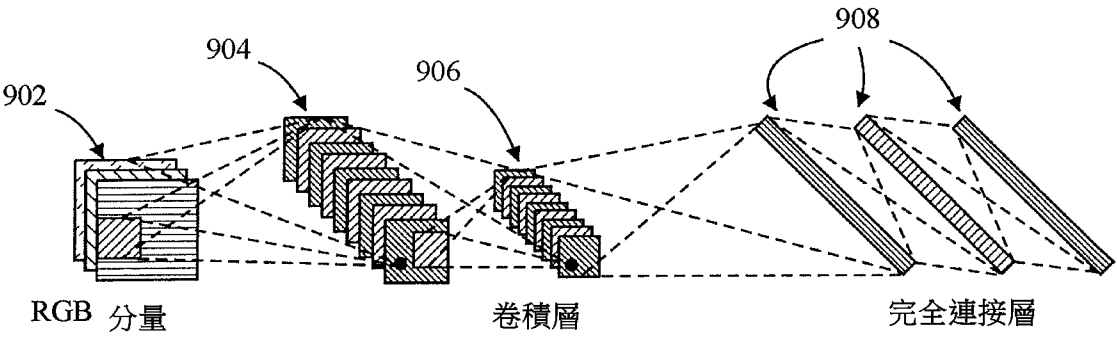


圖 9A

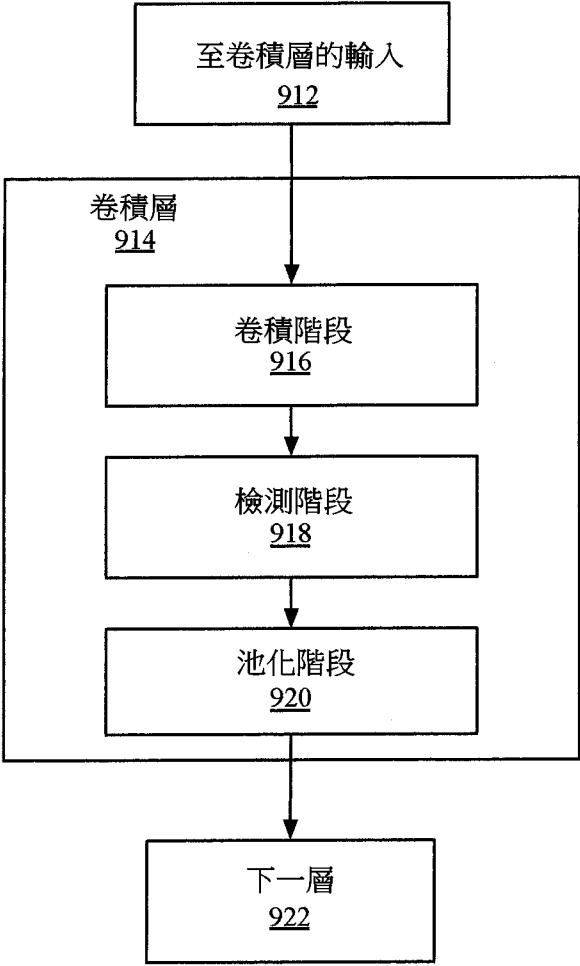


圖 9B

1000

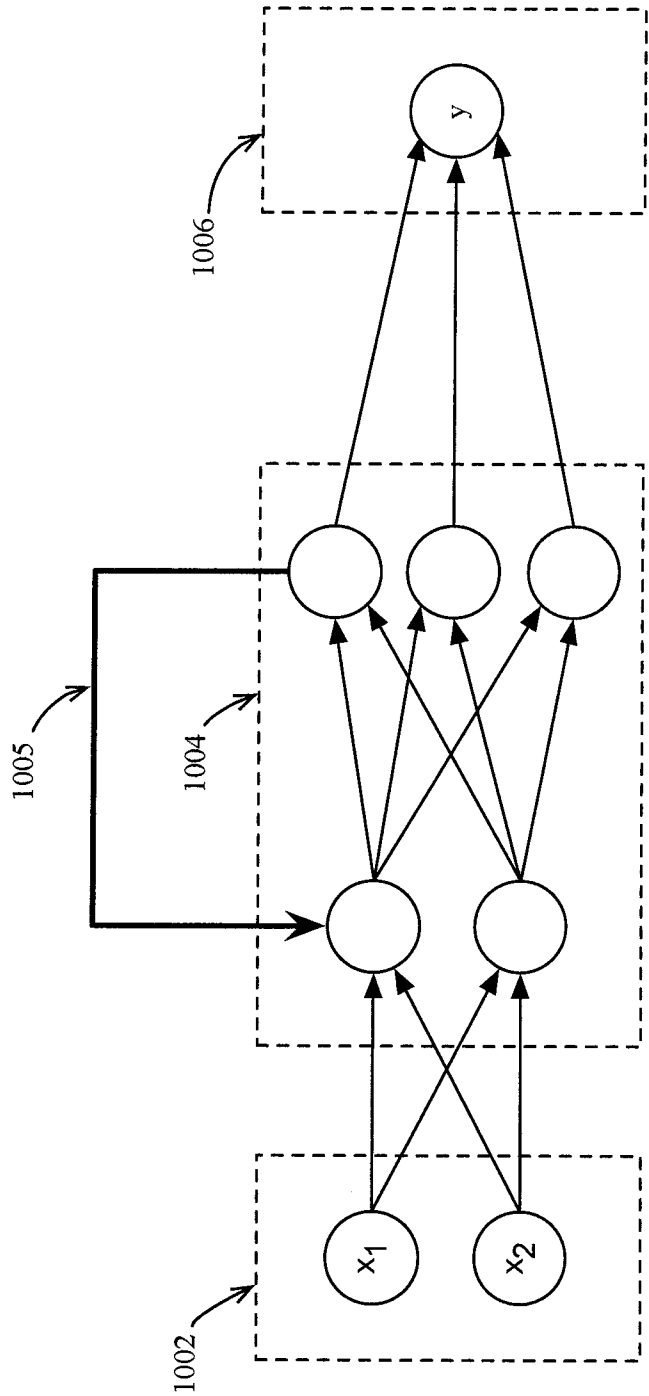


圖 10

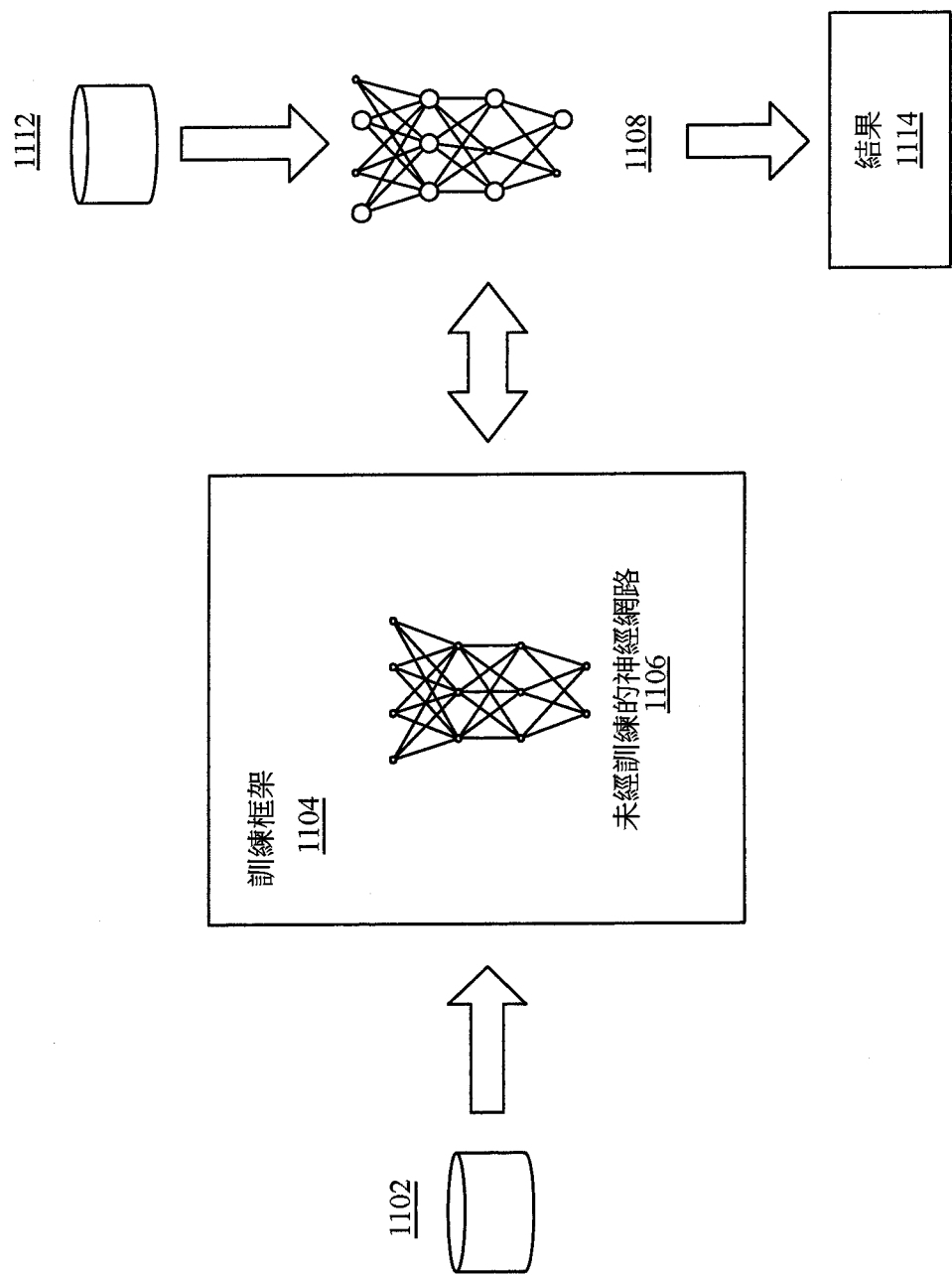


圖 11

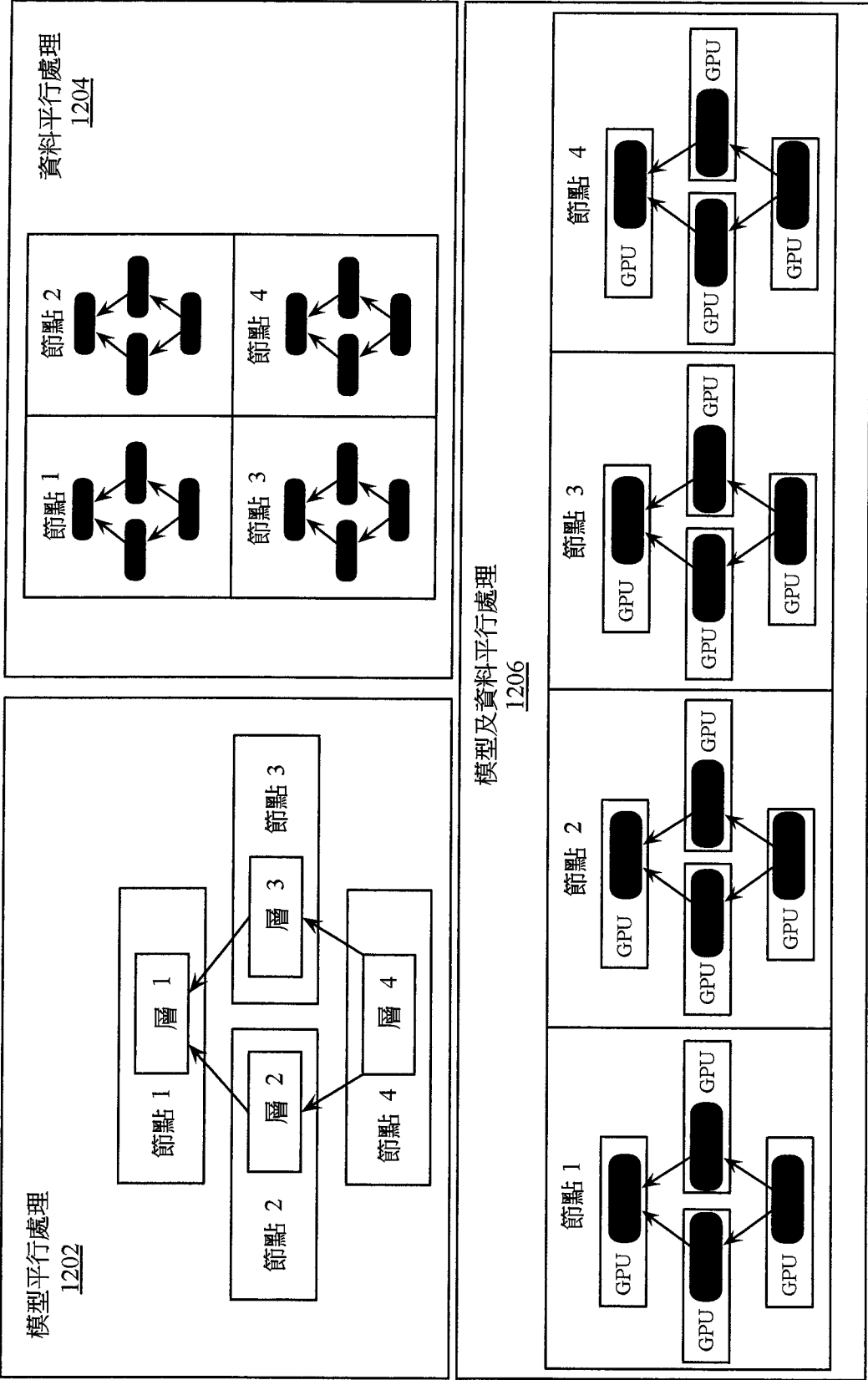


圖 12

1300

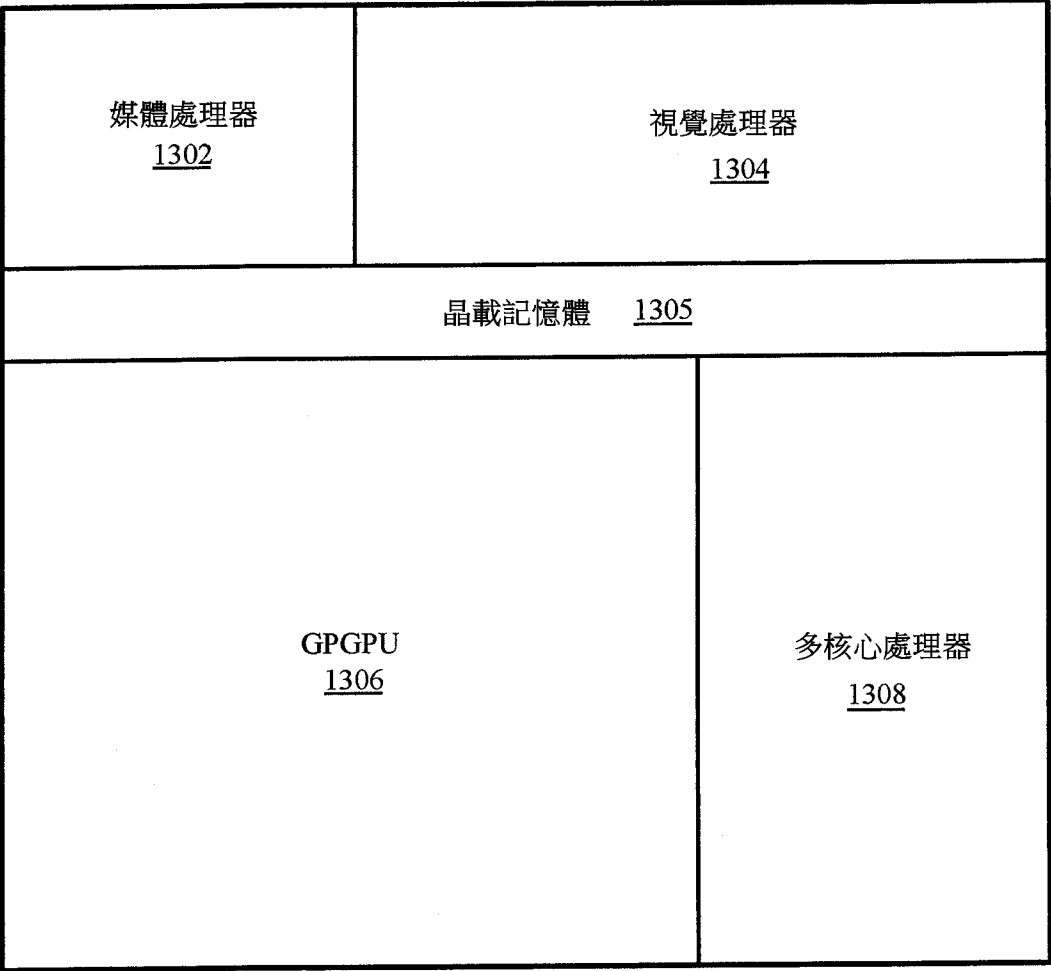


圖 13

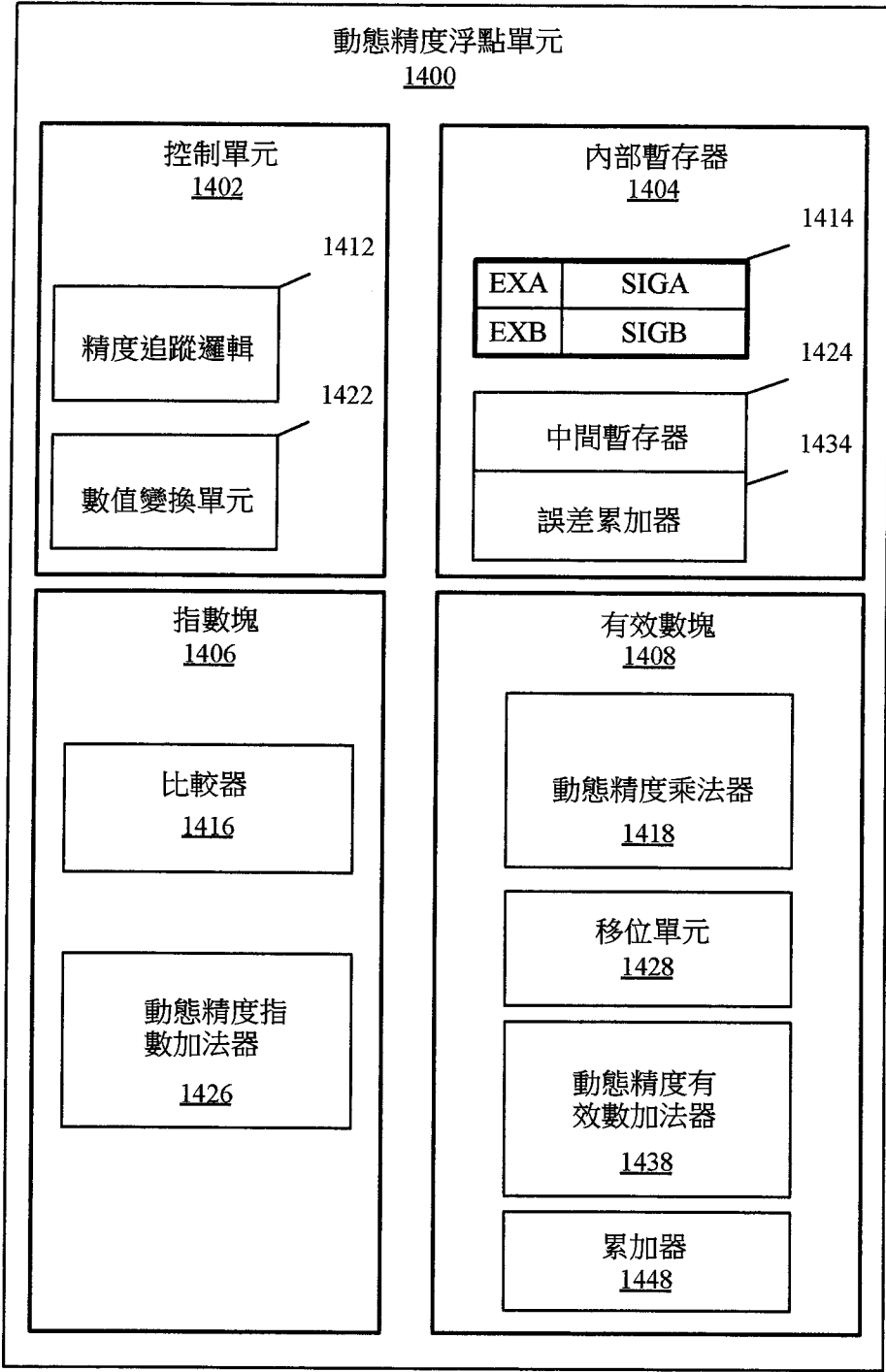


圖 14

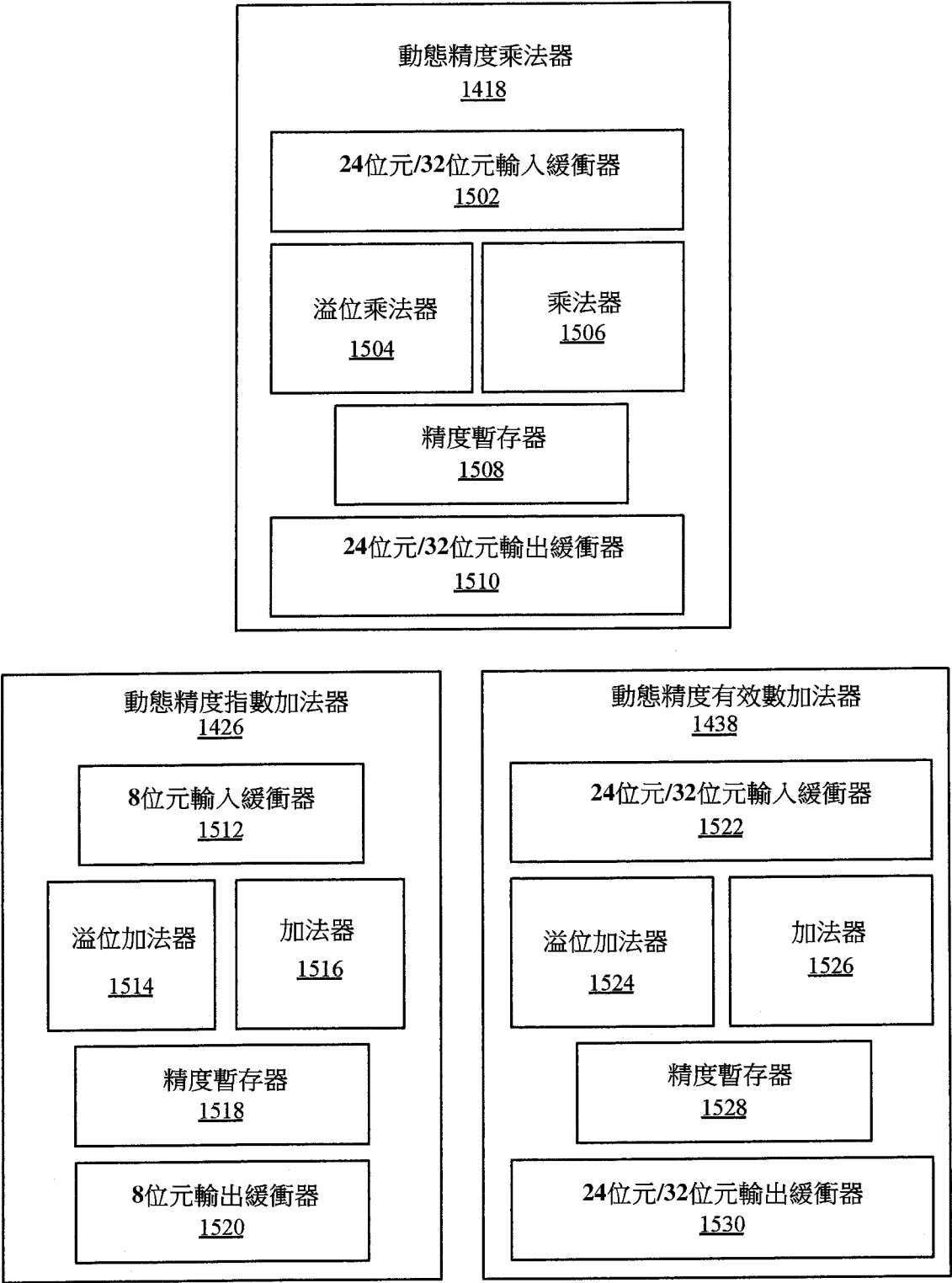


圖 15

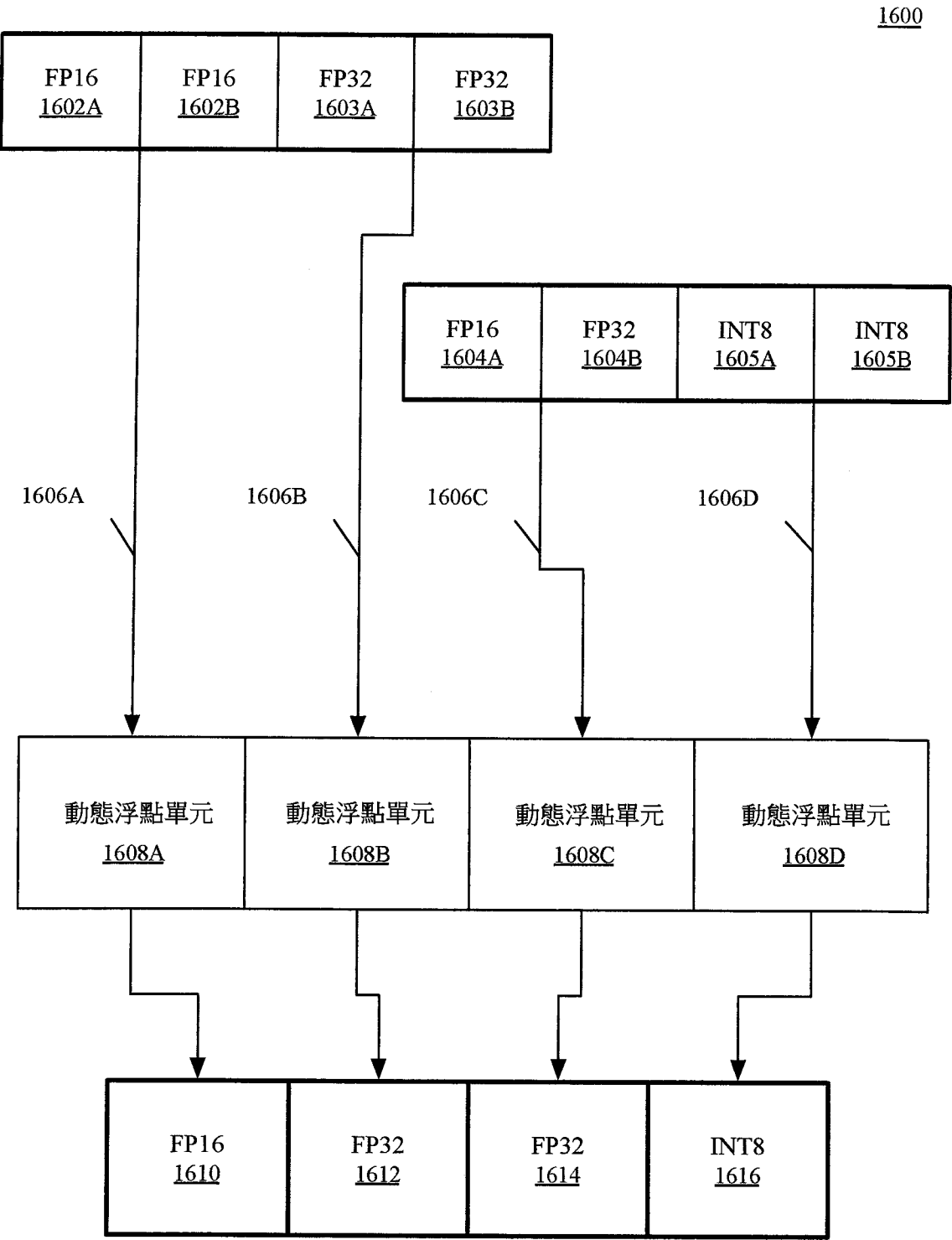


圖 16

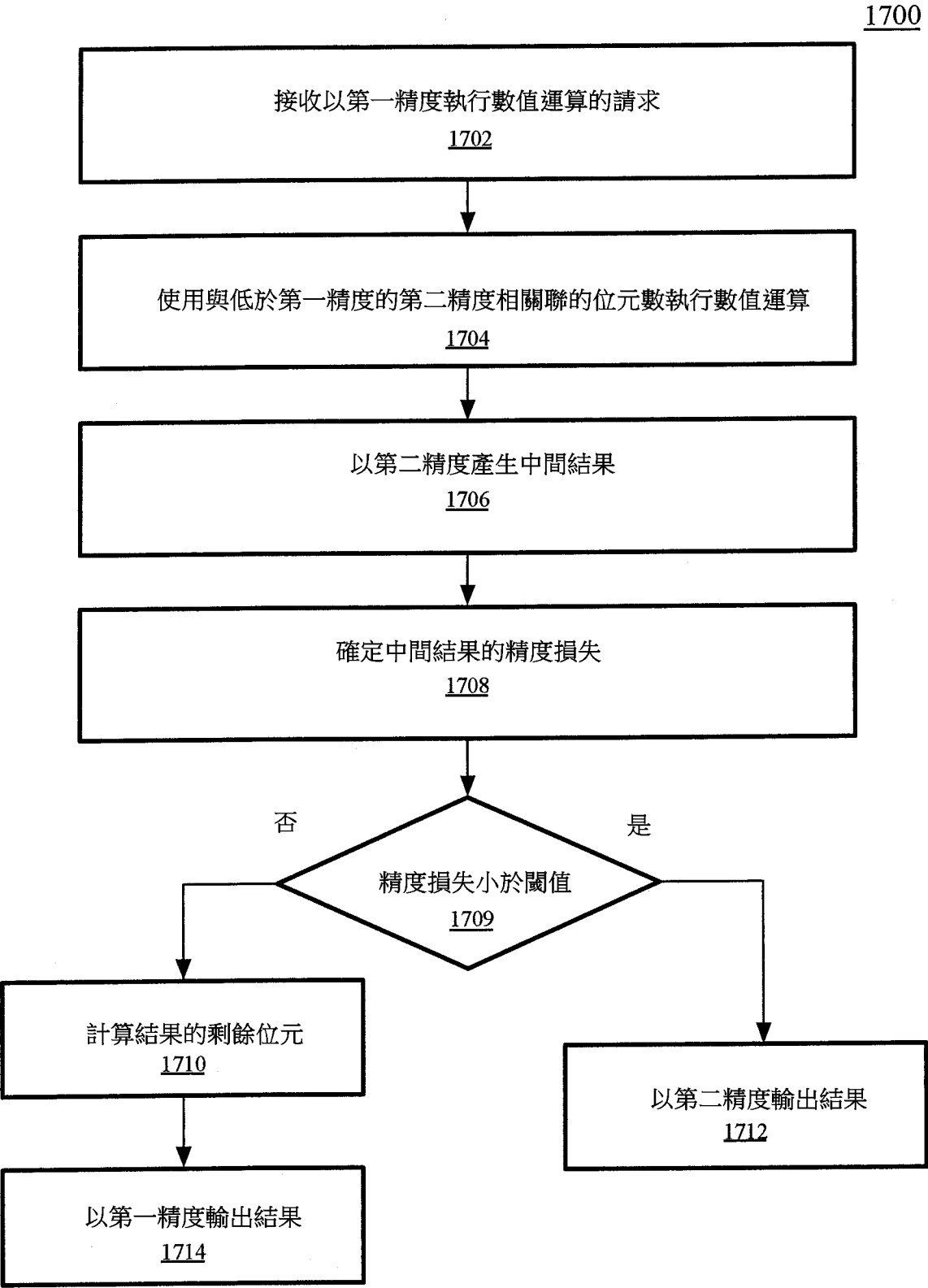


圖 17

1900

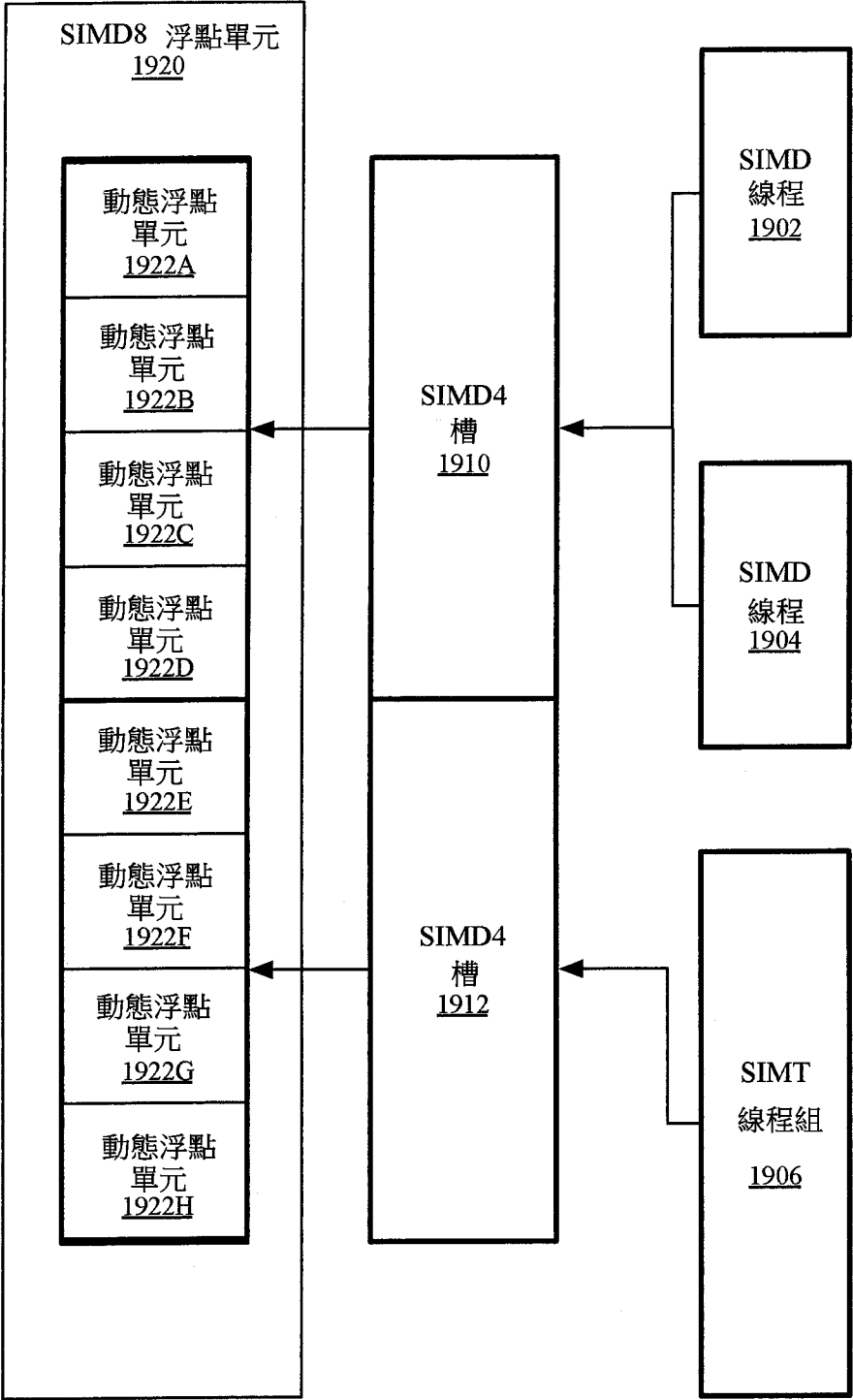


圖 19

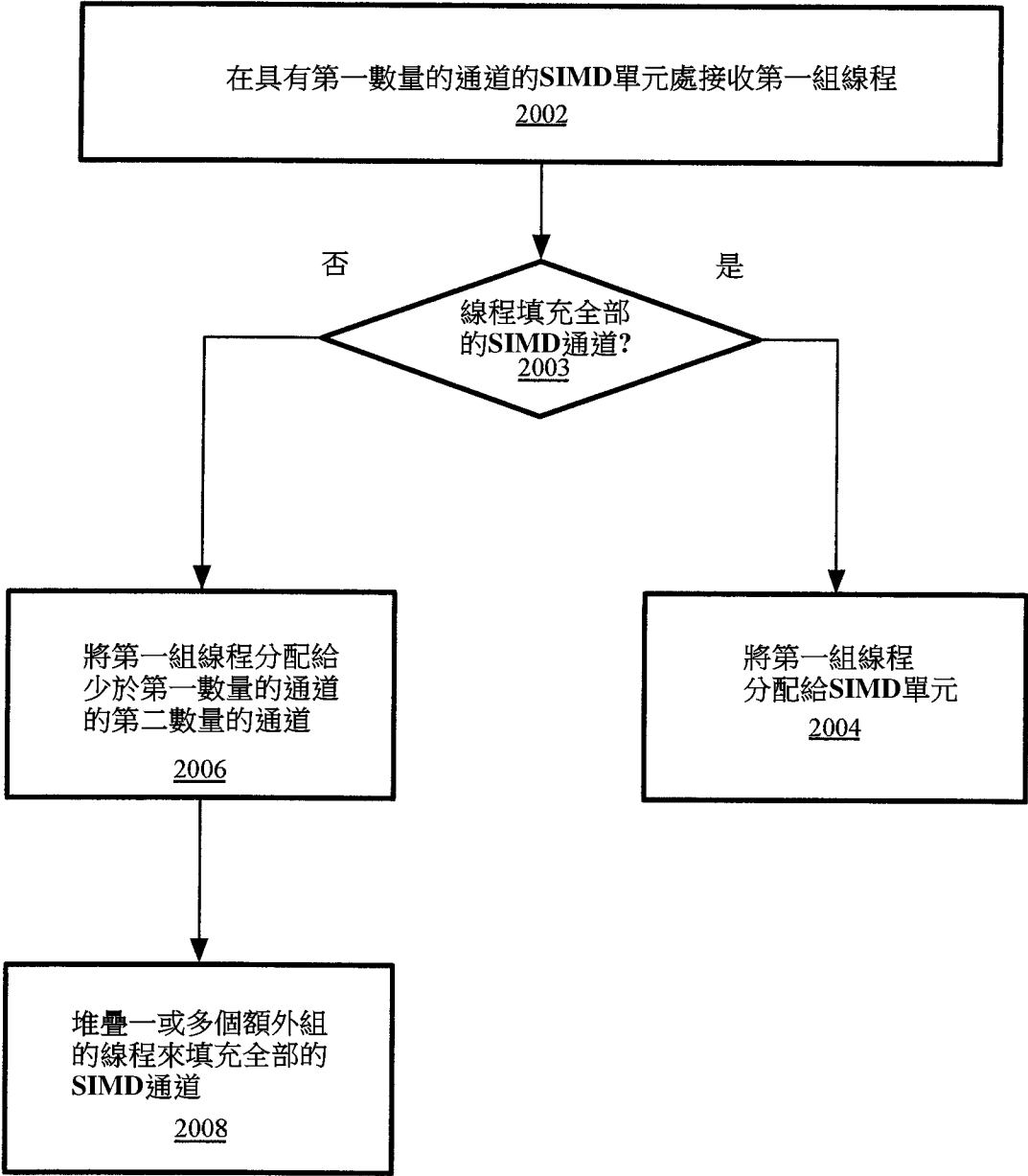


圖 20

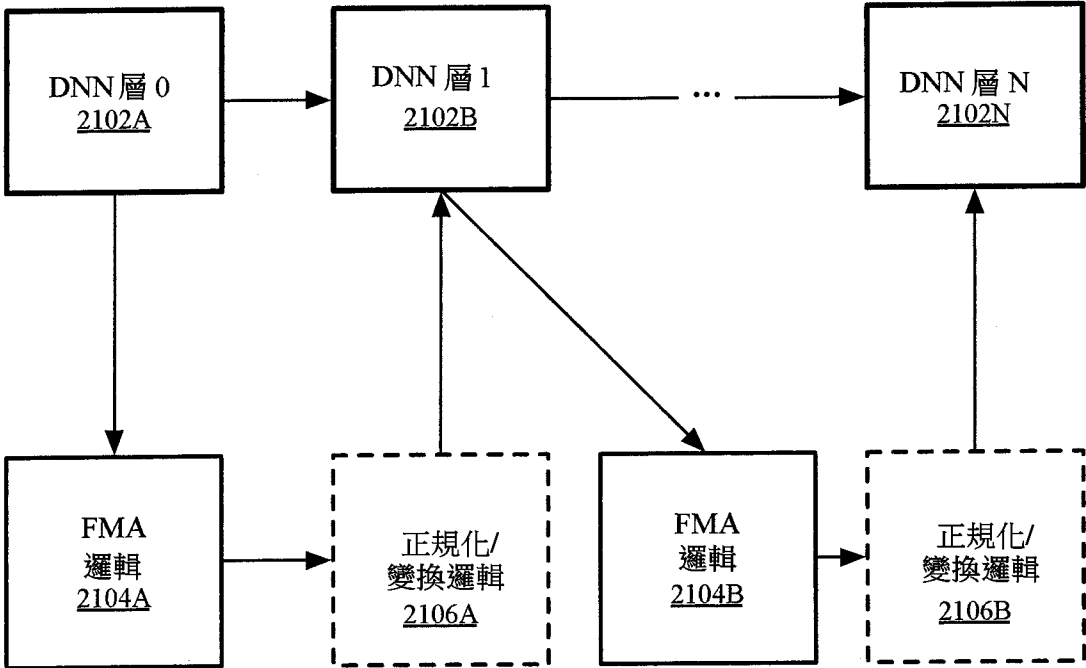


圖 21

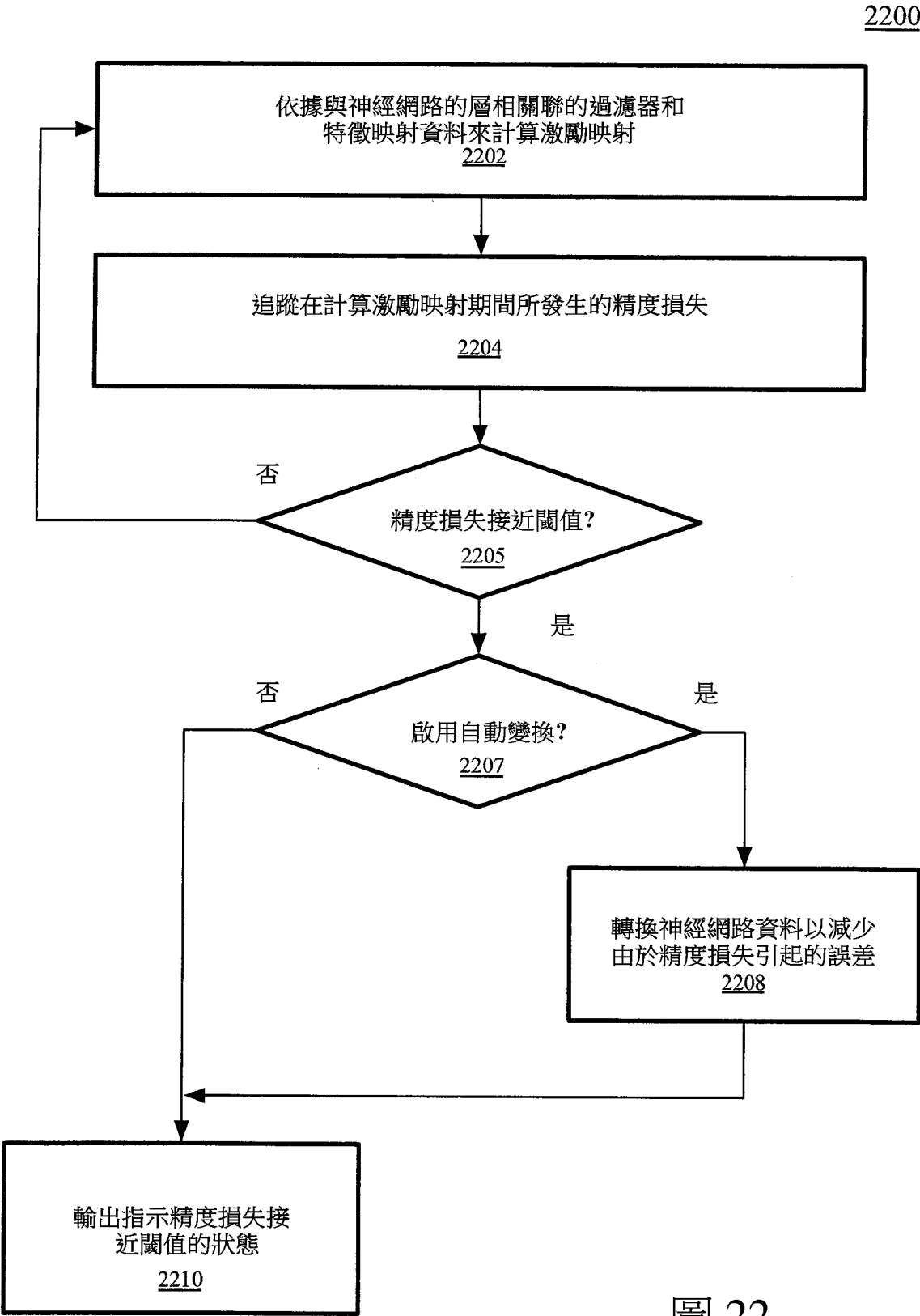
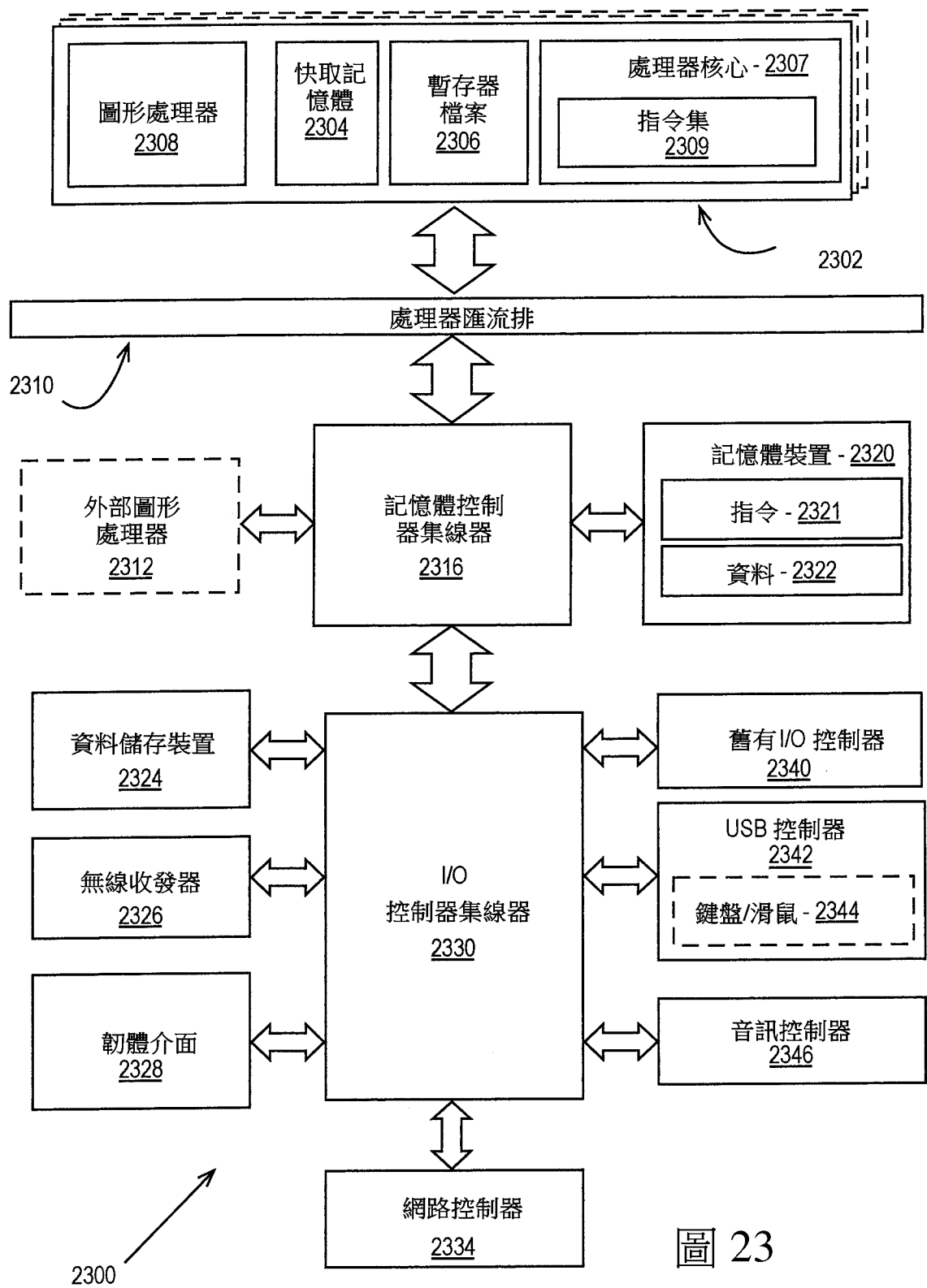


圖 22



2400 ↗

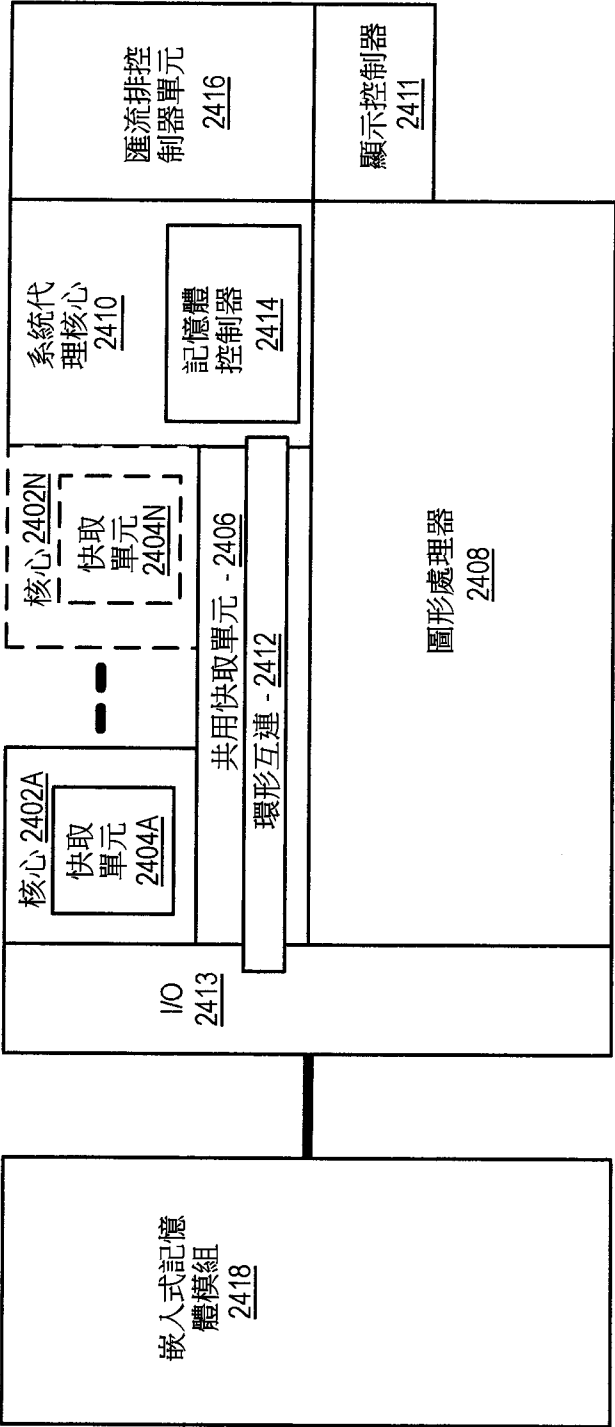


圖 24

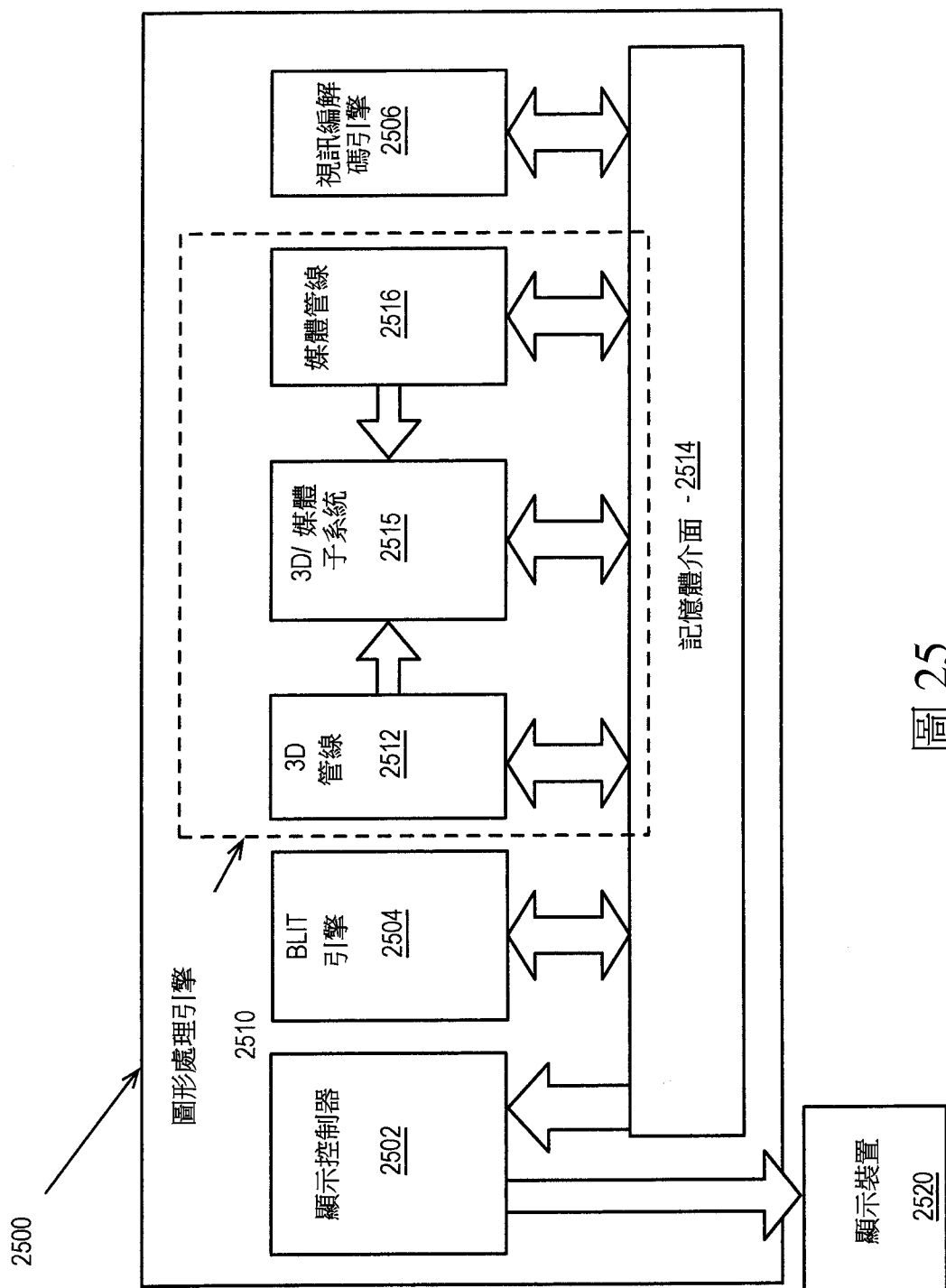


圖 25

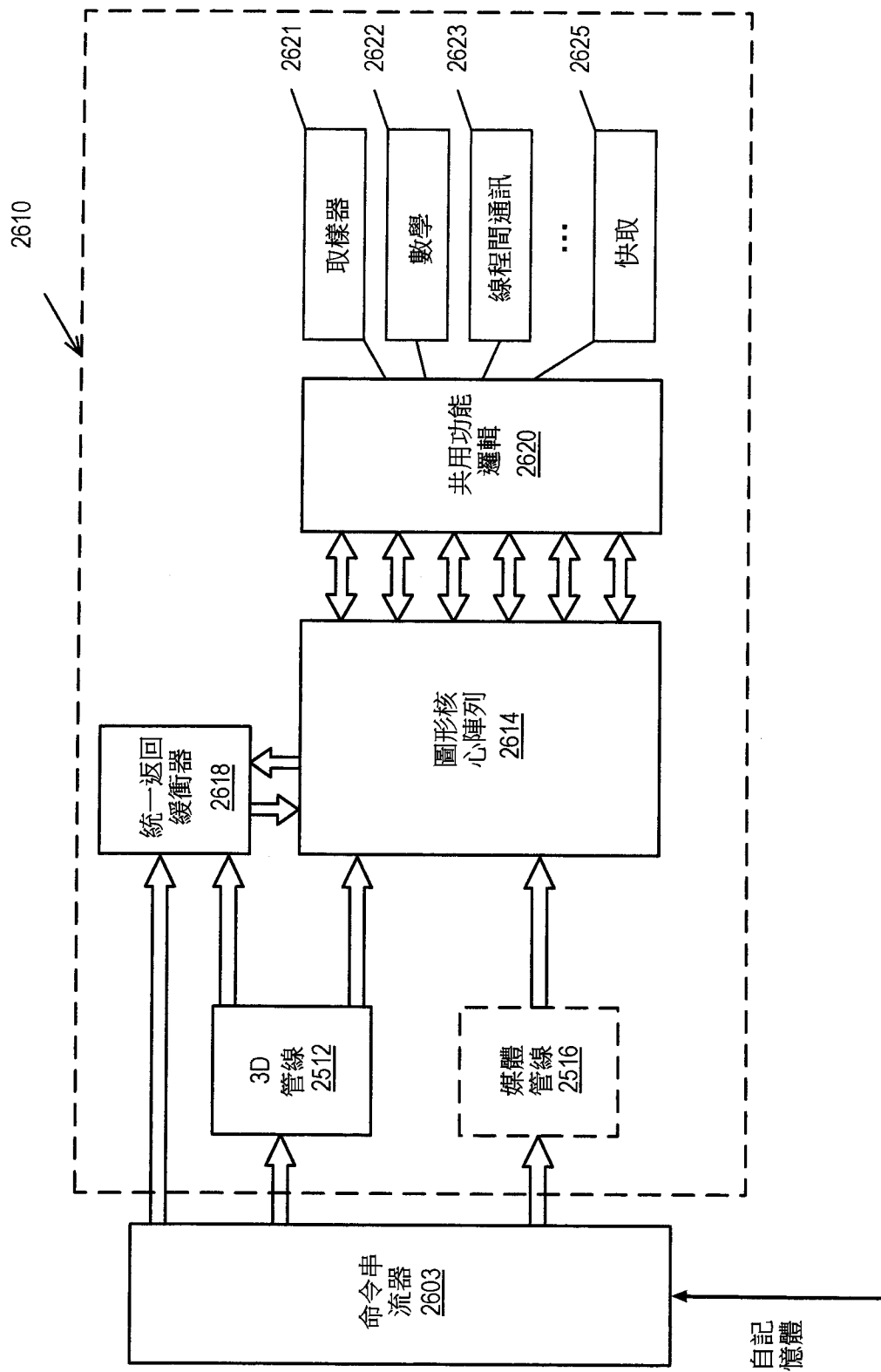


圖 26

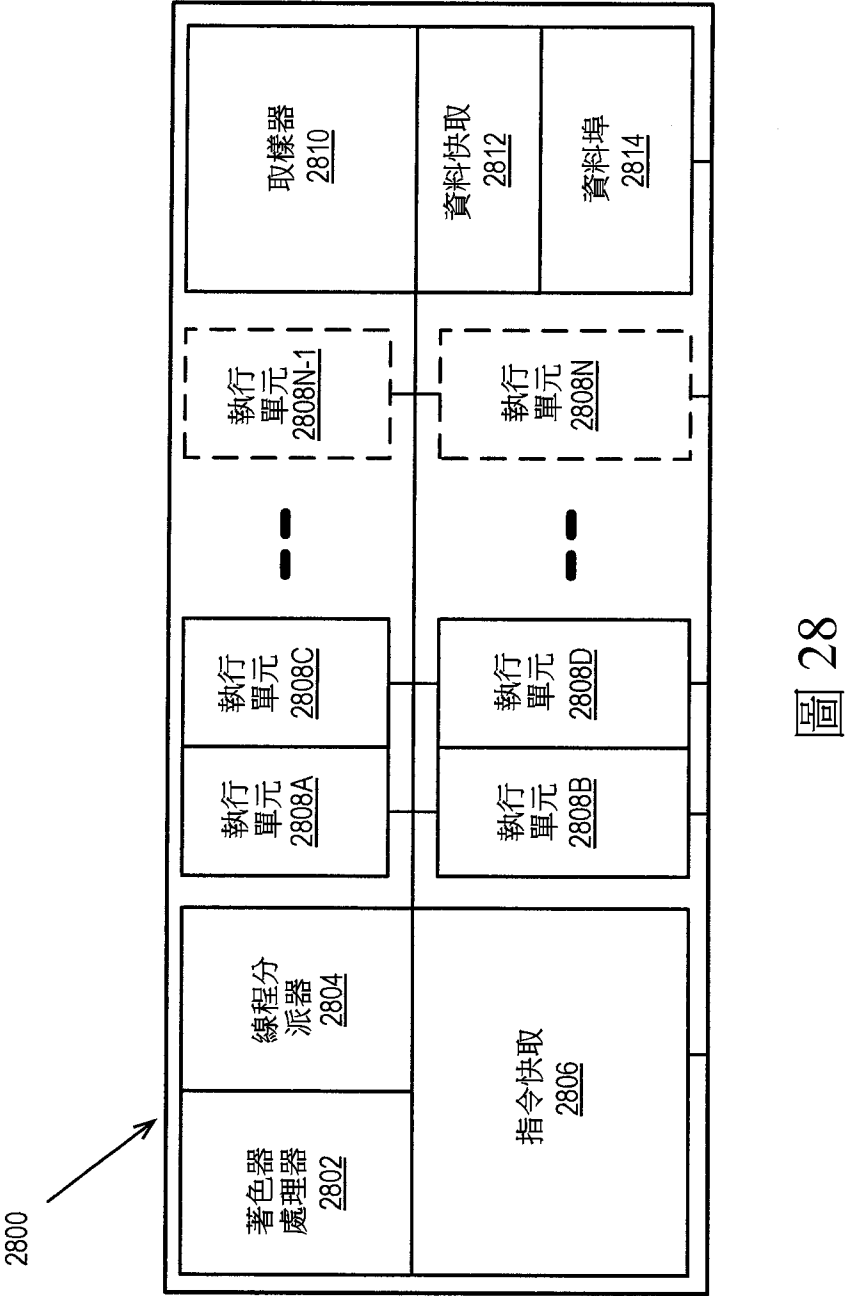


圖 28

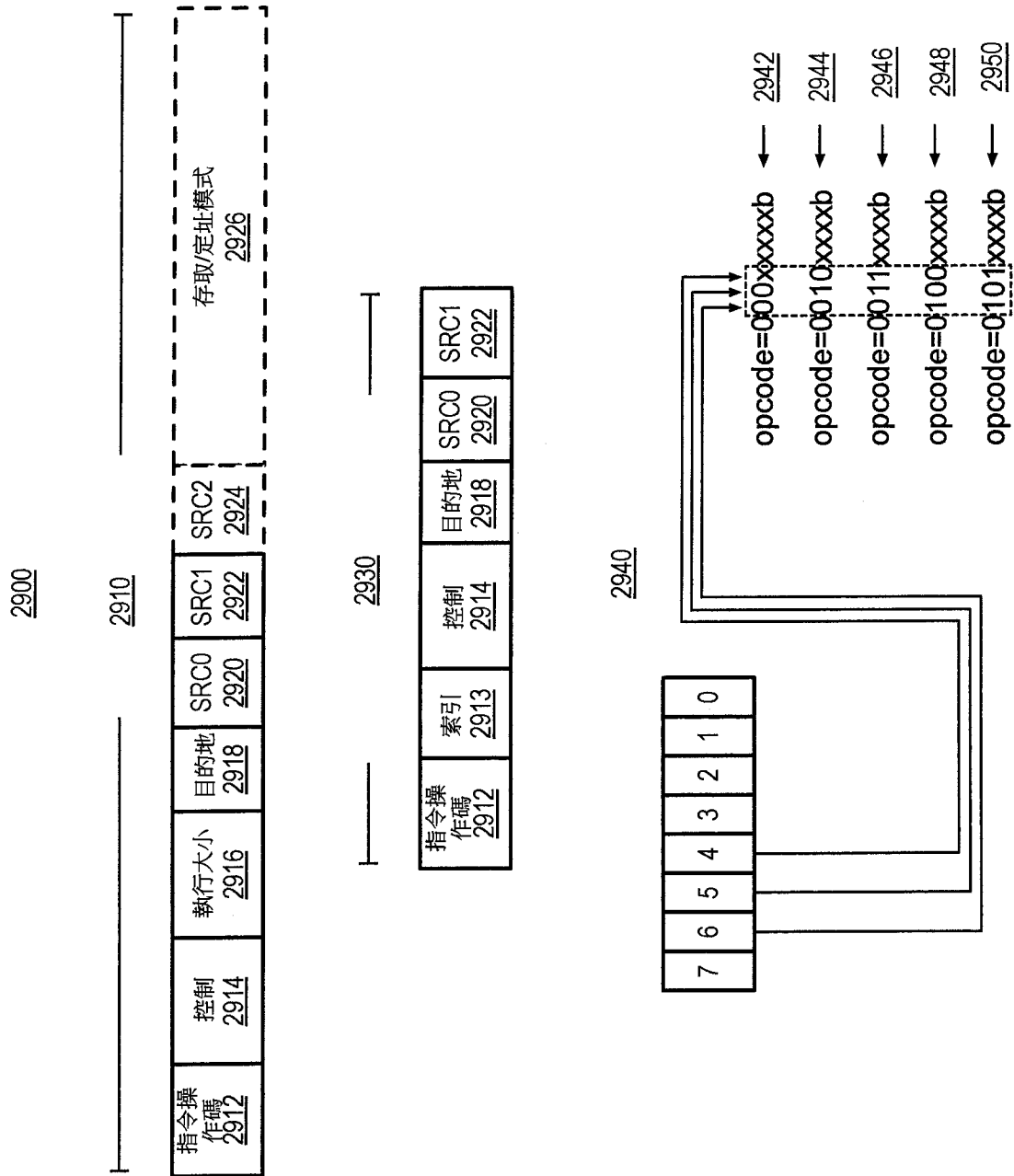


圖 29

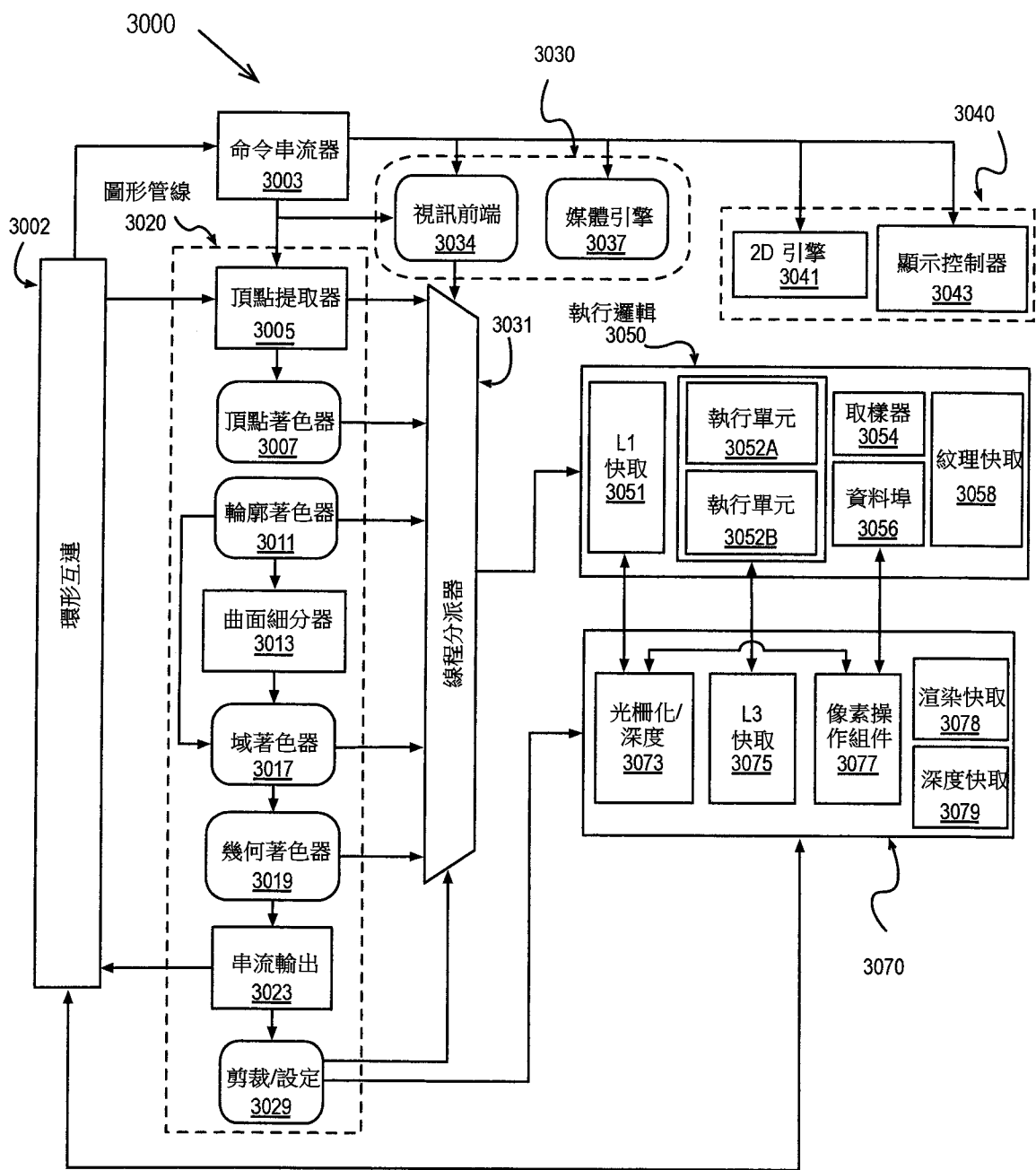


圖 30

圖 31A

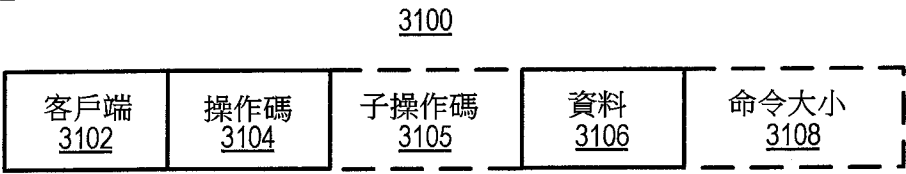
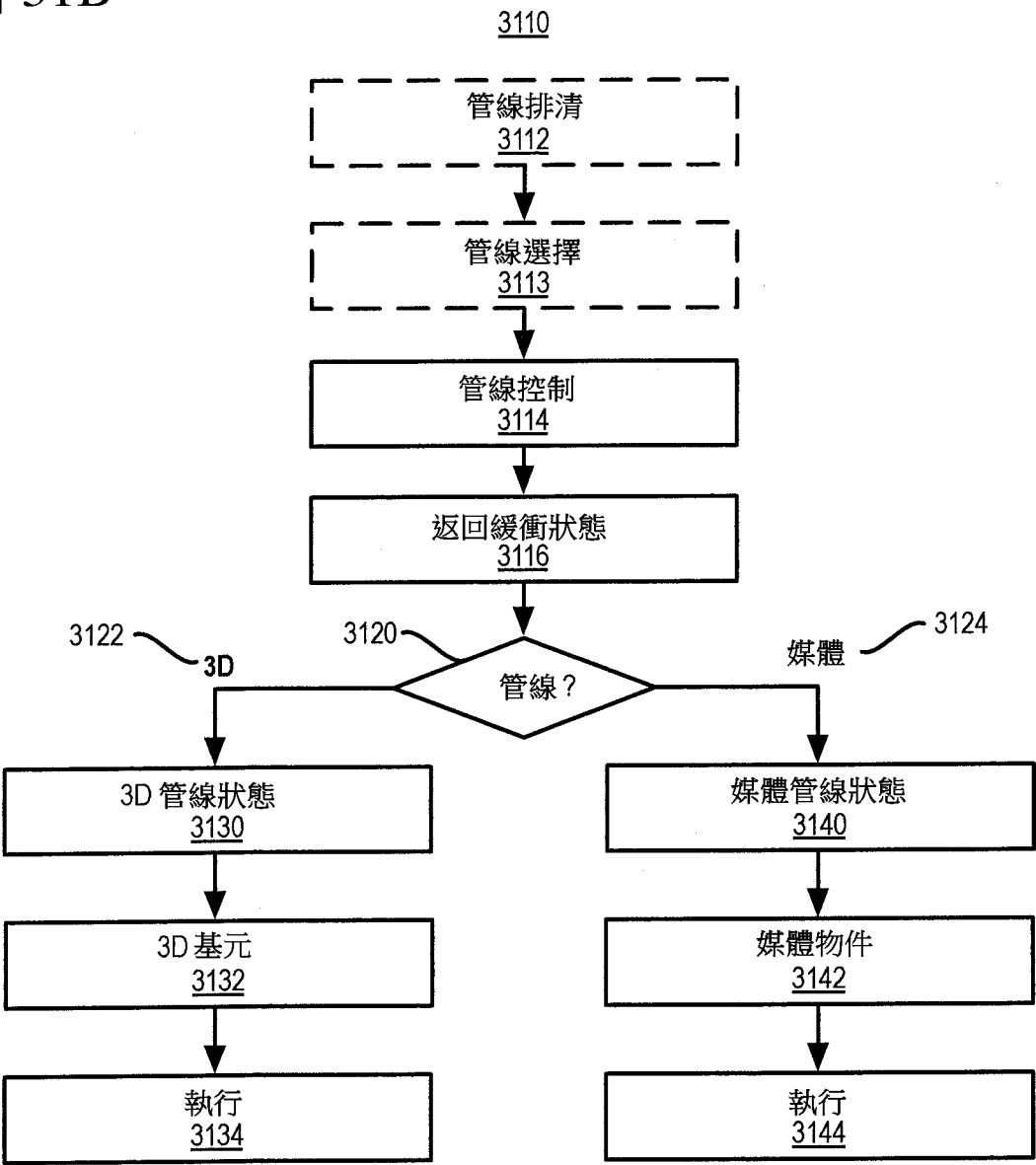


圖 31B



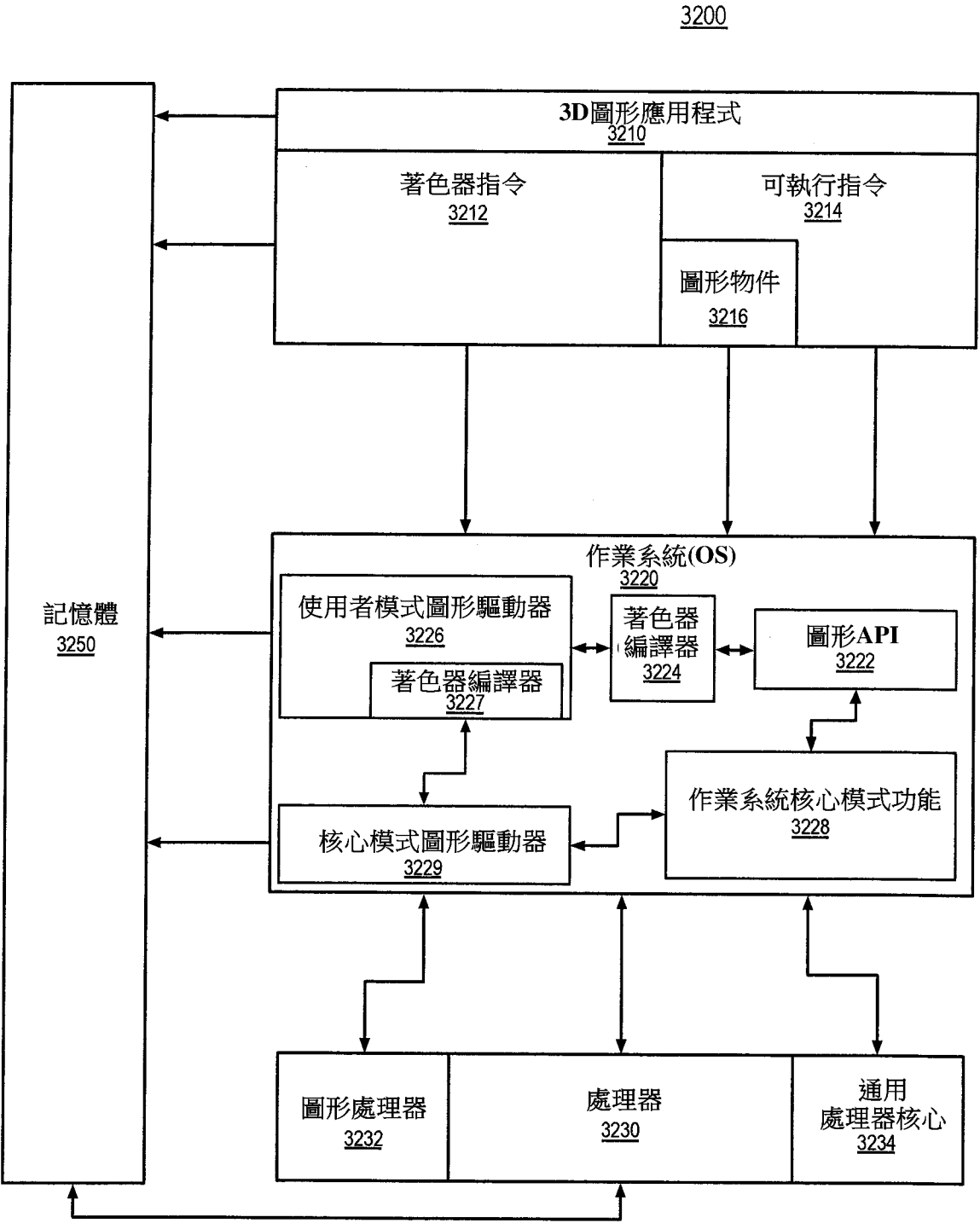


圖 32

3300

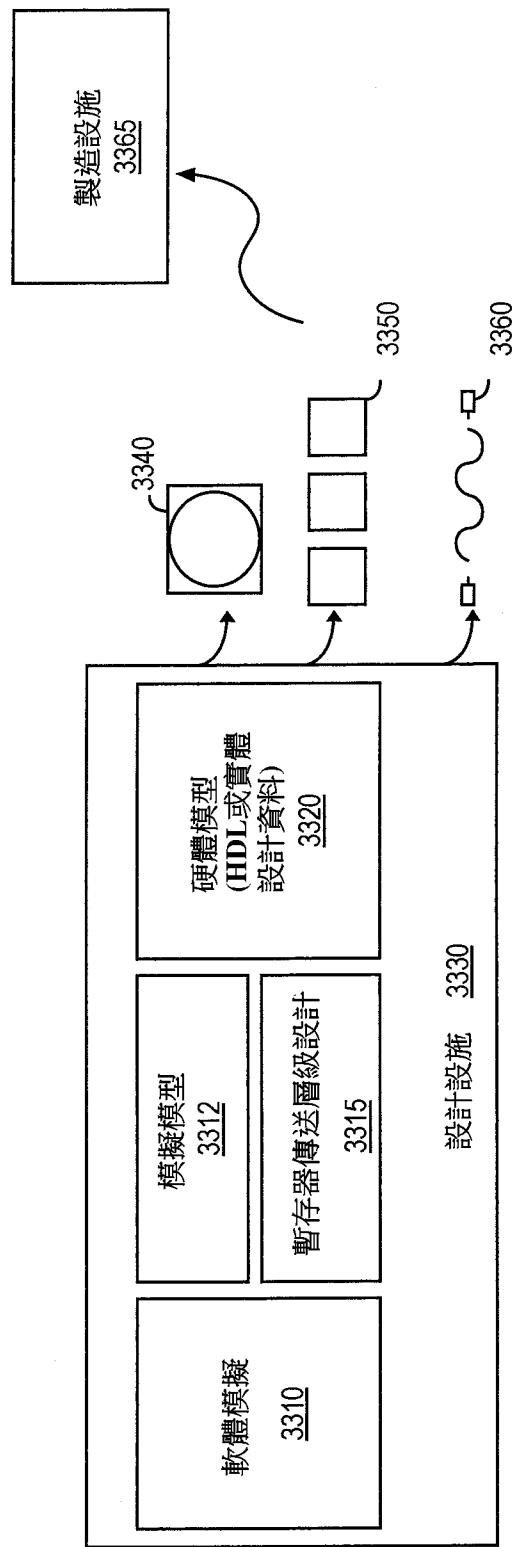


圖 33

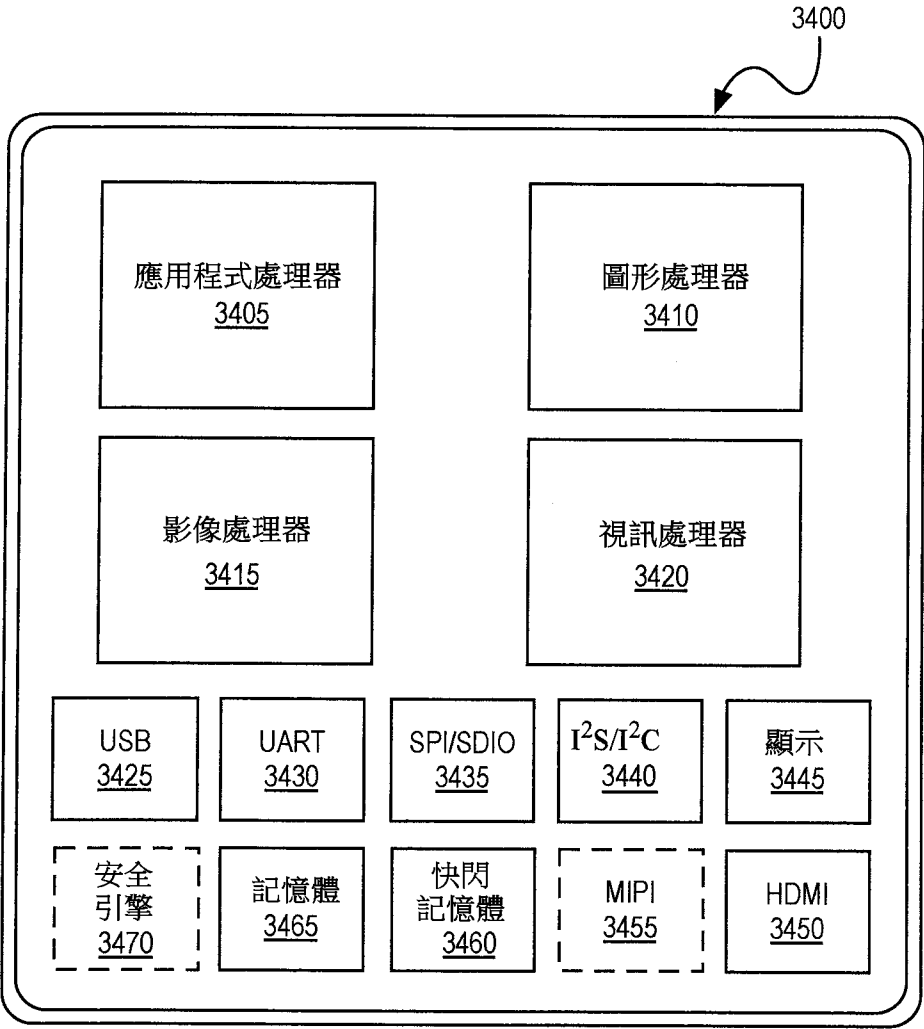


圖 34

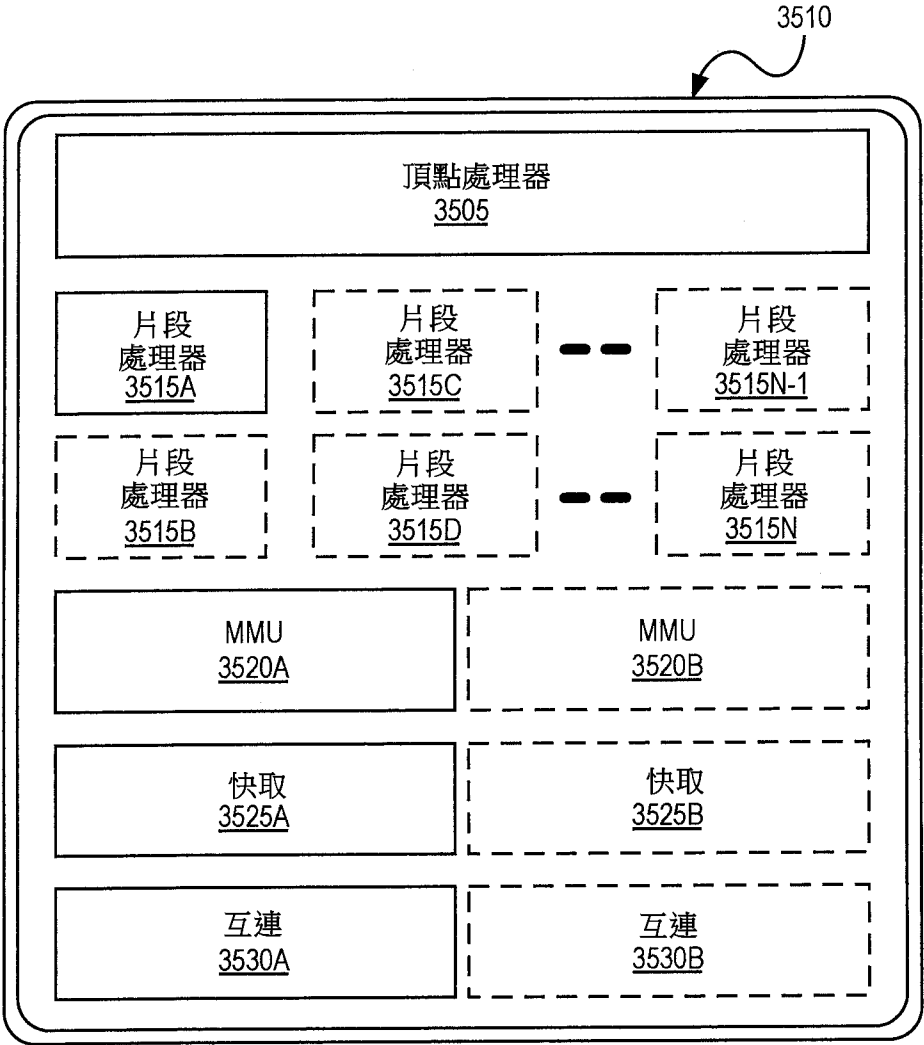


圖 35

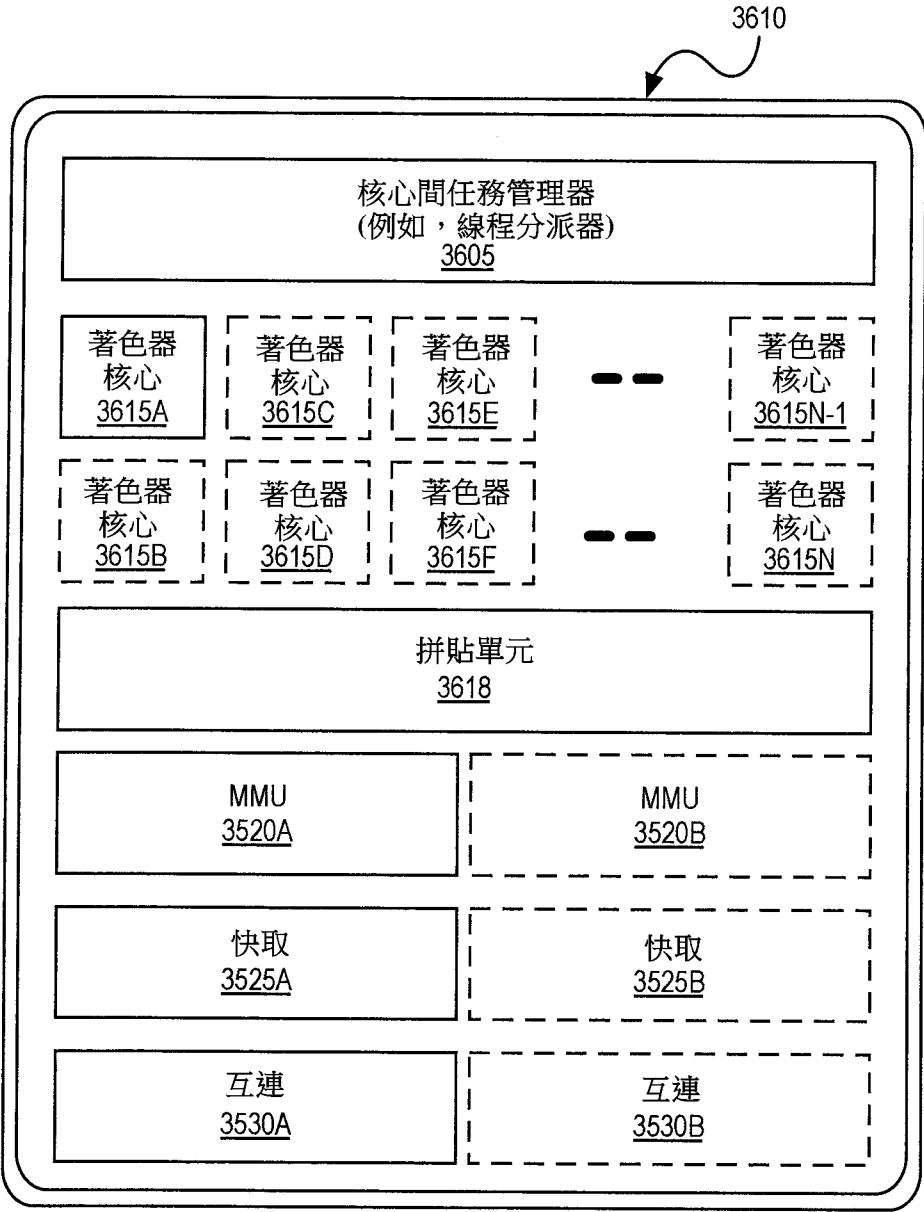


圖 36