I402766

(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 I402766B1**

(45)公告日： 中華民國 102 (2013) 年 07 月 21 日

(21)申請案號：097147390 (22)申請日： 中華民國 97 (2008) 年 12 月 05 日

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

(30)優先權：2007/12/07 美國 11/952,858

(71)申請人：輝達公司 (美國) NVIDIA CORPORATION (US)
美國

(72)發明人：歐巴馬 史都華 OBERMAN, STUART (US)；蕭銘 SIU, MING (US)；覃諾博 大衛 TANNENBAUM, DAVID (US)

(74)代理人：蔡濱陽

(56)參考文獻：

US 4972362 US 5487022

US 6061781 US 2005/0235134A1

審查人員：游象甫

申請專利範圍項數：17 項 圖式數：12 共 0 頁

(54)名稱

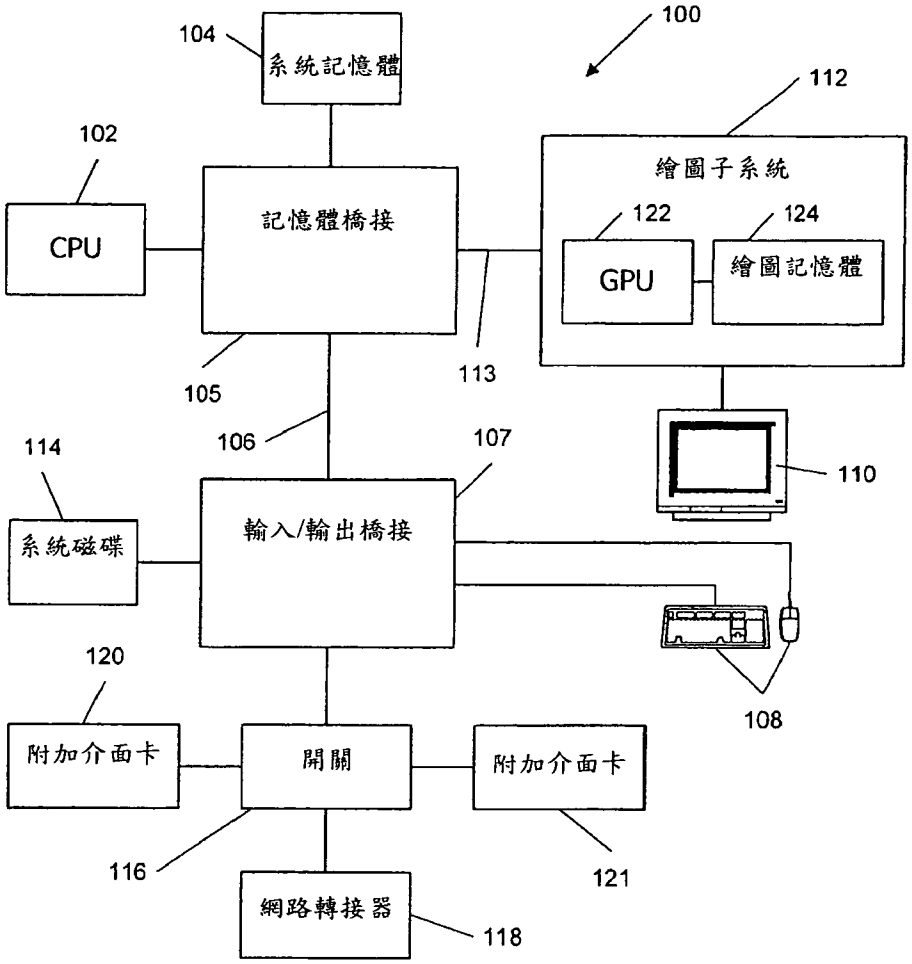
繪圖處理器

GRAPGICS PROCESSOR

(57)摘要

一功能單元加至一繪圖處理器以提供倍精確度算術運算的直接支援，除了可使用於繪圖的單精確度功能單元之外。該倍精確度功能單元可使用至少倍精確度寬度的資料路徑及/或邏輯電路，執行倍精確度輸入的許多不同運算，包括融合乘加算術運算。該等倍精確度與單精確度功能單元可受到一共用指令發送電路的控制，且一核心中包括的倍精確度功能單元的副本數目可少於單精確度功能單元的副本數目，藉此減少增加晶片面積的倍精確度支援的影響。

A functional unit is added to a graphics processor to provide direct support for double-precision arithmetic, in addition to the single-precision functional units used for rendering. The double-precision functional unit can execute a number of different operations, including fused multiply-add, on double-precision inputs using data paths and/or logic circuits that are at least double-precision width. The double-precision and single-precision functional units can be controlled by a shared instruction issue circuit, and the number of copies of the double-precision functional unit included in a core can be less than the number of copies of the single-precision functional units, thereby reducing the effect of adding support for double-precision on chip area.



第一圖

100・・・電腦系統

102・・・中央處理單元

104・・・系統記憶體

105・・・記憶體橋接

106・・・匯流排或通訊路徑

107・・・輸入/輸出橋接

108・・・使用者輸入裝置

110・・・顯示裝置

112・・・繪圖子系統

113・・・匯流排或通訊路徑

114・・・系統磁碟

116・・・開關

118・・・網路轉接器

120・・・附加介面卡

121・・・附加介面卡

122・・・繪圖處理單元(GPU)

124・・・繪圖記憶體

公告本

101 年 8 月 29 日修正替換頁

發明專利說明書

（本說明書格式、順序，請勿任意更動，※記號部分請勿填寫）

※ 申請案號：097147390

※ 申請日：97 年 12 月 5 日　　※IPC 分類：G06T 1/20 (2006.01)

一、發明名稱：（中文/英文）

繪圖處理器

GRAPGICS PROCESSOR

二、中文發明摘要：

一功能單元加至一繪圖處理器以提供倍精確度算術運算的直接支援，除了可使用於繪圖的單精確度功能單元之外。該倍精確度功能單元可使用至少倍精確度寬度的資料路徑及/或邏輯電路，執行倍精確度輸入的許多不同運算，包括融合乘加算術運算。該等倍精確度與單精確度功能單元可受到一共用指令發送電路的控制，且一核心中包括的倍精確度功能單元的副本數目可少於單精確度功能單元的副本數目，藉此減少增加晶片面積的倍精確度支援的影響。

三、英文發明摘要：

A functional unit is added to a graphics processor to provide direct support for double-precision arithmetic, in addition to the single-precision functional units used for rendering. The double-precision functional unit can execute a number of different operations, including fused multiply-add, on double-precision inputs using data paths and/or logic circuits that are at least double-precision width. The double-precision and single-precision functional units can be controlled by a shared instruction issue circuit, and the

number of copies of the double-precision functional unit included in a core can be less than the number of copies of the single-precision functional units, thereby reducing the effect of adding support for double-precision on chip area.

四、指定代表圖：

(一)本案指定代表圖為：第（一）圖。

(二)本代表圖之元件符號簡單說明：

100	電腦系統	113	匯流排或通訊路徑
102	中央處理單元	114	系統磁碟
104	系統記憶體	116	開關
105	記憶體橋接	118	網路轉接器
106	匯流排或通訊路徑	120	附加介面卡
107	輸入/輸出橋接	121	附加介面卡
108	使用者輸入裝置	122	繪圖處理單元 (GPU)
110	顯示裝置		
112	繪圖子系統	124	繪圖記憶體

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

六、發明說明：

【發明背景】

本發明大體上係關於一種繪圖處理器，且更特別地係，關於一繪圖處理器的倍精確度融合乘加功能單元。

繪圖處理器普遍使用在電腦系統，以加速從二維或三維幾何資料提供影像。這類處理器典型具有高度平行與高度傳送量的設計，允許數千個基本圖元可平行處理，以即時提供複雜的實際動畫影像。高階繪圖處理器提供比典型中央處理單元(CPU, “Central Processing Unit”)更強的計算能力。

最近，已有興趣於有效利用繪圖處理器的能力以加速與影像顯示無關的各種計算。「一般目的」的繪圖處理器可用來執行科學、財務、商業及其他領域的計算。

調適用於一般目的計算的繪圖處理器之一困難度在於繪圖處理器通常設計用於相當低的數值精確度。高品質影像可使用 32 位元(單精確度)或甚至 16 位元(半精確度)浮點數值顯示，且功能單元及內部管線可組態成支援這些資料寬度。對照下，許多一般目的計算需要更高的數值精確度，例如 64 位元(倍精確度)。

為了要支援較高的精確度，一些繪圖處理器使用軟體技術，以一序列機器指令與 32 位元或 16 位元功能單元，執行倍精確度計算。此方式會減慢傳送量；例如，可能需要一百或更多的機器指令完成單一 64 位元乘法運算。此長序列顯著減少繪圖處理器的倍精確度傳送量。在一代表性的情況中，一般估計該繪圖處理器將能夠以大約一高階雙核心 CPU 晶片可能傳送量的 1/5 完成倍精確度計算。(相較下，相同繪圖處理器能夠以大約雙核心 CPU 傳送量的 15-20 倍完成單精確度計算。)因為以軟體為主之解決方案係遠較慢，所以現行的繪圖處理

器很少用於倍精確度計算。

另一解決方案只是使繪圖處理器的所有算術電路足夠處理倍精確度運算元。相較於單一速度傳送量，此將會增加繪圖處理器的倍精確度運算之傳送量。然而，繪圖處理器典型具相當多的每一算術電路的副本以支援平行運算，且增加每一此電路的尺寸將會實質增加晶片面積、成本與功率消耗。

仍然另一解決方案(如在2006年2月21日申請的共同擁有、共同美國申請專利案第11/359,353號)係有效利用單精確度算術電路以執行倍精確度運算。在此方式中，單精確度功能單元中包括的特殊硬體是用來反覆地執行一倍精確度運算。此方式認為較快於以軟體為主之解決方案(可減少傳送量，例如藉由相對於單精確度傳送量的4因數，而不係藉由100的因數)，但是此明顯使晶片設計複雜化。此外，若太多指令需要相同功能單元，單精確度與倍精確度運算之間共用相同功能單元將使此功能單元在管線中造成瓶頸。

【發明內容】

本發明的具體實施例是在一繪圖處理器中直接支援倍精確度算術。除了單精確度功能單元用於繪圖之外，提供一多用途倍精確度功能單元。倍精確度功能單元可執行許多不同運算，包括在使用資料路徑的倍精確度輸入上的融合乘加、及/或至少倍精確度寬度的邏輯電路。一共用指令發送電路可控制倍精確度與單精確度功能單元，且一核心中包括的倍精確度功能單元的副本數目可少於單精確度功能單元的副本數目，藉此減少增加對晶片區域的倍精確度支援之效果。

根據本發明之一態樣，一繪圖處理器具有一繪圖管

線，其可調適產生影像資料。在單精確度運算元上操作的繪圖管線包括一處理核心，其可調適執行許多同時發生的執行緒。該處理核心包括一多用途倍精確度功能單元，其可調適選擇性執行一組倍精確度輸入運算元之許多倍精確度運算之一。多用途倍精確度功能單元包括至少一算術邏輯電路，且倍精確度功能單元的所有算術邏輯電路可充分廣泛以倍精確度運算。在一些具體實施例中，倍精確度功能單元可適應，使得每一倍精確度運算可完成相同數目的時脈週期，而且該單元亦可調適，使得完成任一倍精確度運算所需的時間(例如，時脈週期數目)不會受到下限溢位或上限溢位條件的影響。

倍精確度運算的各種運算與組合可被支援。在一具體實施例中，倍精確度運算包括：一加算，其可將兩個倍精確度運算元相加；一乘算，其可將兩個倍精確度運算元相乘；及一融合乘加運算，其係先計算一第一倍精確度運算元與一第二倍精確度運算元的乘積，然後將一第三倍精確度運算元與該乘積進行加算。支援的其他倍精確度運算包括：一倍精確度比較(DSET)操作，其可執行一第一運算元與一第二運算元之比較測試，及產生一布林(Boolean)結果以指示是否滿足該比較測試；一倍精確度最大(DMAX, “Double-precision Maximum”)運算，其可傳回兩個倍精確度輸入運算元之較大一者；或一倍精確度最小(DMIN, “Double-precision Minimum”)運算，其可傳回兩個倍精確度輸入運算元之較小一者。此外，亦支援將一運算元從一倍精確度格式轉換成一非倍精確度格式(或相反的轉換)之格式化轉換運算。

根據本發明之另一態樣，一繪圖處理器包括一繪圖管線，其調適成產生影像資料。繪圖管線包括一處理核心，其調適成執行多個同時發生的執行緒。處理核心包

括單精確度功能單元，其調適成在一或多個單精確度運算元上執行一算術運算；及一倍精確度融合乘加(DFMA, "Double-precision Fused Multiply add")功能單元，其調適成在一組倍精確度輸入運算元上執行一融合乘加法運算，並提供一倍精確度結果。DFMA 功能單元可有利地包括一 DFMA 管線，該管線具有足夠資料路徑寬度透過 DFMA 管線，在單一通過中執行融合乘加法運算。例如，該 DFMA 功能單元可包括一乘法器，其調適成在單循環中，計算兩個倍精確度尾數的乘積；及一加法器，其調適成在單循環中，計算兩個倍精確度尾數的加總。

DFMA 功能單元亦可組態成執行其他運算。例如，在一些具體實施例中，該 DFMA 組態成執行一對倍精確度輸入運算元的乘法運算，並提供一倍精確度結果。在一些具體實施例中，乘法運算與融合乘加法運算是每一者在相同時脈週期數內完成。同樣地，DFMA 功能單元可組態成執行一對倍精確度輸入運算元的加法運算，並提供一倍精確度結果。在一具體實施例中，加法運算與融合乘加法運算是每一者在相同時脈週期數目內完成。

在一些具體實施例中，處理核心包括第一功能單元的副本數目(P)，該第一功能單元係調適成平行操作；及 DFMA 功能單元的副本數目(N)，該數目 P 係大於該數目 N。在一具體實施例中，此數目 N 是 1。

處理核心可包括一輸入管理器電路，其係調適成在不同(例如，連續)時脈週期上，收集有關 DFMA 功能單元的 P 組倍精確度輸入運算元，及將該等 P 組倍精確度運算元之不相同者傳遞至 DFMA 功能單元。輸入管理器電路亦可調適成收集第一功能單元的 P 組單精確度輸入運算元，及同時將該等 P 組單精確運算元之不相同者傳遞至該第一功能單元的 P 副本之每一者。

下列連同附圖的詳細描述將提供對本本發明的本質與優點的更佳瞭解。

【實施方式】

本發明的具體實施例提供繪圖處理器，其包括專屬的倍精確度(例如，64 位元)功能單元。在一具體實施例中，一倍精確度功能單元可執行加算、乘算、及融合乘加運算、以及倍精確度比較與針對倍精確度格式的格式轉換。

I. 系統概觀

A. 電腦系統概觀

第一圖為根據本發明之一具體實施例的一電腦系統 100 之區塊圖。該電腦系統 100 包括一中央處理單元(CPU, “Central Processing Unit”)102 與一系統記憶體 104，彼此係經由一匯流排路徑進行通訊，該路徑包括一記憶體橋接 105。記憶體橋接 105(可為例如一習知北橋(Northbridge)晶片)是經由一匯流排或其他通訊路徑 106(例如，一 HyperTransport 鏈路)連接至一輸入/輸出(Input/Output，I/O)橋接 107。I/O 橋接 107(可為例如一習知南橋(Southbridge)晶片)係從一或多個使用者輸入裝置 108(例如，鍵盤、滑鼠)接收使用者輸入，並經由匯流排 106 與記憶體橋接 105 將輸入轉送至 CPU 102。視景輸出是在一像素式顯示裝置 110(例如，一習知 CRT 式或 LCD 式之監視器)上提供，其操作是在經由一匯流排或其他通訊路徑 113，在耦合至記憶體橋接 105 的一繪圖子系統 112 控制下操作，例如一 PCI Express (PCI-E)或加速繪圖埠(AGP, “Accelerated Graphics Port”)鏈路。一系統磁碟 114 亦連接至 I/O 橋接 107。一開關 116

提供在 I/O 橋接 107 與其他組件(例如一網路轉接器 118 和各種附加介面卡 120、121)之間的連接。其他組件(未在圖明確顯示)(包括 USB 或其他連接埠連接、CD 光碟機、DVD 光碟器等)亦可連接至 I/O 橋接 107。在各種組件之中的匯流排連接可使用匯流排協定實施，例如周邊組件互連(PCI, "Peripheral Component Interconnect")、PCI-E、AGP、HyperTransport 或任何其他匯流排或點對點通訊協定，且不同裝置之間的連接可使用在技術中熟知的不同協定。

繪圖處理子系統 112 包括一繪圖處理單元(GPU, "Graphics Processing Unit")122 與一繪圖記憶體 124，其可例如使用一或多個積體電路裝置加以實施，例如可程式處理器、特殊應用積體電路(ASIC, "Application Specific Integrated Circuit")與記憶體裝置。GPU 122 可組態成執行各種工作，以進行經由記憶體橋接 105 和匯流排 113，藉由 CPU 102 及/或系統記憶體 104 供應的繪圖資料產生像素資料；與繪圖記憶體 124 進行互動，以儲存及更新像素資料等相關工作。例如，GPU 122 可從由 CPU 102 執行的各種程式提供的 2D 或 3D 場景資料產生像素資料。GPU 122 亦可將經由記憶體橋接 105 接收的像素資料儲存至繪圖記憶體 124，以供進行或無需進一步處理。GPU 122 亦包括一掃出模組，其組態成將像素資料從繪圖記憶體 124 傳遞至顯示裝置 110。

GPU 122 亦組態成執行資料處理工作的一般目的計算，包括有關繪圖應用(例如，電視遊樂器等的物理模型)的工作、以及無關繪圖應用的工作。對於一般目的計算而言，GPU 122 有利地從系統記憶體 104 或繪圖記憶體 124 讀入資料，執行一或多個程式以處理資料，及將輸出資料寫回至系統記憶體 104 或繪圖記憶體 124。除了

在繪圖操作期間使用的其他單精確度功能單元之外，GPU 122 可有利地包括用於一般目的計算的一或多個倍精確度融合乘加單元(未在第一圖顯示)。

CPU 102 的操作如同系統 100 的主處理器，用於控制及協調其他系統組件的操作。特別地係，CPU 102 係送出控制 GPU 122 操作的命令。在一些具體實施例中，CPU 102 將 GPU 122 的命令流寫至一命令緩衝器，該緩衝器可在系統記憶體 104、繪圖記憶體 124、或可由 CPU 102 和 GPU 122 二者存取的另一儲存位置。GPU 122 是從命令緩衝器讀取命令流，並與 CPU 102 的操作非同步地執行命令。命令亦可包括習知繪圖命令，其用於產生影像；以及一般目的計算命令，其允許 CPU 102 執行應用程式，以有效利用 GPU 122 的計算能力，作為可能無關影像產生的資料處理。

應該明白，在此顯示的系統只是說明，且不同的變化與修改是可能的。匯流排拓撲(Bus Topology)(包括橋接的數目與配置)可視需要予以修改。例如，在一些具體實施例中，系統記憶體 104 係直接連接至 CPU 102 而不是透過一橋接，且其他裝置可經由記憶體橋接 105 和 CPU 102 而與系統記憶體 104 進行通訊。在其他替代拓撲中，繪圖子系統 112 連接至 I/O 橋接 107 而不是連接至記憶體橋接 105。在仍然其他具體實施例中，I/O 橋接 107 與記憶體橋接 105 可整合在單晶片內。在此顯示的特定組件是選擇性；例如，可支援任何數量的附加介面卡或周邊裝置。在一些具體實施例中，可免除開關 116，且網路轉接器 118 與附加介面卡 120、121 係直接連接至 I/O 橋接 107。

GPU 122 連接至系統 100 的其餘部分亦會改變。在一些具體實施例中，繪圖系統 112 實施作為一附加介面

卡，其可插入系統 100 的一擴充槽。在其他具體實施例中，一 GPU 是在單晶片上與一匯流排橋接整合，該匯流排橋接例如有記憶體橋接 105 或 I/O 橋接 107。在仍然其他具體實施例中，GPU 122 的一些或所有元件可與 CPU 102 一起整合。

一 GPU 可具有任何數量的本機繪圖記憶體，包括沒有本機記憶體，並可以任何組合方式使用本機記憶體與系統記憶體。例如，在統一記憶體架構(UMA, "Unified Memory Architecture")具體實施例中，沒有提供專屬的繪圖記憶體裝置，且 GPU 專門或幾乎專門使用系統記憶體。在 UMA 具體實施例中，GPU 可整合在一匯流排橋接晶片或提供作為一分立晶片，具有一高速匯流排(例如，PCI-E)，該匯流排連接 GPU 至橋接晶片與系統記憶體。

亦應該瞭解，任何數量的 GPU 可包括在一系統，例如，藉由在單一繪圖介面卡上包括多個 GPU、或藉由連接多個繪圖介面卡至匯流排 113。多個 GPU 可平行操作，以產生可供相同顯示裝置或不同顯示裝置的影像，或一 GPU 可操作產生影像，而另一 GPU 可執行一般目的計算，包括如下述的倍精確度計算。

此外，具體實施本發明之態樣的 GPU 可合併在各種裝置，包括一般用途電腦系統、電視遊樂器控制台及其他特殊目的電腦系統、DVD 播放器、手持式裝置，例如行動電話或個人數位助理等。

B. 繪圖管線概觀

第二圖為根據本發明之一具體實施例之可在第一圖的 GPU 122 中實施的一繪圖管線 200 之區塊圖。在此具體實施例中，繪圖管線 200 係使用一架構實施，其中

使用相同平行處理硬體(在此稱為「多執行緒核心陣列(Multithreaded Core Array)」202)執行任何適用的繪圖相關程式(例如，頂點著色器(Vertex Shader)、幾何著色器(Geometry Shader)及/或像素著色器(Pixel Shader))，與一般目的計算程式。

除了多執行緒核心陣列 202 之外，繪圖管線 200 包括一前端 204 與資料組合器 206、一設定模組 208、光柵化器 210、一彩色組合模組 212、與一光柵操作模組(ROP, "Raster Operations Module")214，其每一者可使用習知的積體電路技術或其他技術實施。

針對繪圖操作，前端 204 係例如從第一圖的 CPU 102 接收狀態資訊(STATE)、命令(CMD)、與幾何資料(GDATA)。在一些具體實施例中，而不是直接提供幾何資料，CPU 102 提供幾何資料儲存的系統記憶體 104 之位置參考；資料組合器 206 是從系統記憶體 104 取回資料。針對繪圖操作，狀態資訊、命令與幾何資料可為通常一習知本質，並可用來定義想要的描繪影像或一些影像，包括場景的幾何、照明、暗影、結構、動作、及/或照像機參數。

狀態資訊與繪圖命令係定義繪圖管線 200 的不同階段之處理參數與動作。前端 204 係將狀態資訊與繪圖命令經由一控制路徑(未在圖明確顯示)導向繪圖管線 200 的其他組件。如在技術中所熟知的，這些組件可藉由儲存或更新於處理期間存取在各種控制暫存器中的值，以回應接收的狀態資訊，並可藉由處理在管線中接收的資料，以回應繪圖命令。

前端 204 係將幾何資料導向資料組合器 206。資料組合器 206 格式化幾何資料，並準備將此幾何資料傳遞至在多執行緒核心陣列 202 中的一幾何模組 218。

幾何模組218使用回應前端204提供的狀態資訊所選取的程式，以導引多執行緒核心陣列202中的可程式處理引擎(未在圖明確顯示)在頂點資料上執行頂點及/或幾何著色器程式。頂點及/或幾何著色器程式可藉由如在技術中熟知的繪圖應用程式予以指定，且不同著色器程式可應用至不同頂點及/或基本圖元。在一些具體實施例，頂點著色器程式與幾何著色器程式係使用多執行緒核心陣列202中的相同可程式處理核心執行。因此，在特定時間，一給定的處理核心可操作為一頂點著色器，負責接收及執行頂點程式指令，且在其他時間，相同處理核心可操作為一幾何著色器，負責接收及執行幾何程式指令。處理核心可為多執行緒，且執行不同類型著色器程式的不同執行緒可在多執行緒核心陣列202中同時執行。

在頂點及/或幾何著色器程式執行之後，幾何模組218可將處理過的幾何資料(GDATA')傳遞至設定模組208。通常為習知設計的設定模組208可從每一基本圖元的裁剪空間或螢幕空間坐標產生幹邊方程；該等幹邊方程可有利地用來決定螢幕空間的一點是在基本圖元的內部或外部。

設定模組208提供每一基本圖元(PRIM)至光柵化器210。通常為習知設計的光柵化器210可例如使用習知的掃描轉換演算法，決定基本圖元涵蓋哪些(如果有的話)像素。在此的使用，一「像素」(或「圖點(Fragment)」)通常是指將決定單色值的2D螢幕空間的區域；像素的數目與配置可為繪圖管線200的可組態參數，並可與一特定顯示裝置的螢幕解析度有關或無關。

在決定一基本圖元涵蓋哪些像素之後，光柵化器210將基本圖元(PRIM)、連同基本圖元所涵蓋像素的一

連串螢幕坐標(X, Y)提供給一彩色組合模組 212。彩色組合模組 212 係使從光柵化器 210 接收的基本圖元和涵蓋資訊與基本圖元的頂點屬性(例如，彩色組件、結構坐標、表面正常)有關聯性，並產生平面方程式(或其他適當方程式)，以將一些或全部屬性定義為螢幕坐標空間的位置函數。

這些屬性方程式可有利地使用在像素著色器程式中，以計算基本圖元中任何位置的屬性值；習知的技術可用來產生方程式。例如，在一具體實施例中，彩色組合模組 212 可對每一屬性 U，產生式子 $U= Ax + By + C$ 平面方程式的係數 A、B、和 C。

彩色組合模組 212 係將每一基本圖元之屬性方程式(EQS，其可包括例如平面方程式係數 A、B 和 C)提供給多執行緒核心陣列 202 中的一像素模組 224，該基本圖元涵蓋一像素的至少一取樣位置、及涵蓋像素的一連串螢幕坐標(X, Y)。像素模組 224 係導引多執行緒核心陣列 202 中的可程式處理引擎(未在圖明確顯示)在基本圖元所涵蓋的每一像素上執行一或多個像素著色器程式，且程式係回應前端 204 提供的狀態資訊而選取。正如頂點著色器程式與幾何著色器程式，繪圖應用程式可指定用於任何給定組像素的像素著色器程式。

像素著色器程式可使用相同可程式的處理引擎在多執行緒核心陣列 202 中有利地執行，其亦可執行頂點及/或幾何著色器程式。因此，在特定時間，一給定的處理引擎可操作為一頂點著色器，負責接收及執行頂點程式指令；在其他時間，相同處理引擎可操作為一幾何著色器，負責接收及執行幾何程式指令；且在仍然其他時間，相同處理引擎可操作為一像素著色器，負責接收及執行像素著色器程式指令。

一旦完成一像素或一群像素的處理，像素模組 224 提供處理的像素(PDATA)至 ROP 214。通常可為習知設計的 ROP 214 係整合從像素模組 224 接收的像素值與訊框緩衝器 226 中建構的影像像素，該訊框緩衝器可例如位在繪圖記憶體 124。在一些具體實施例中，ROP 214 可遮罩像素，或將新的像素與先前寫至繪圖影像的像素予以混合。深度緩衝器、ALPHA 緩衝器與模板緩衝器亦可用來決定每一進入像素對繪圖影像的貢獻(如果有的話)。對應至每一進入像素值與任何先前儲存像素值的適當組合之像素資料 PDATA'係寫回至訊框緩衝器 226。一旦完成影像，訊框緩衝器 226 可掃瞄至一顯示裝置及/或做進一步處理。

對於一般目的計算而言，像素模組 224(或藉由幾何模組 218)可控制多執行緒核心陣列。前端 204 係例如從第一圖的 CPU 102 接收狀態資訊(STATE)與處理命令(CMD)，並經由一控制路徑(未在圖明確顯示)傳遞至一工作分配單元，該工作分配單元可合併例如在彩色組合模組 212 或像素模組 224。該工作分配單元可在構成多執行緒核心陣列 202 的處理核心之中分配處理工作。可使用各種工作分配演算法。

每一處理工作可有利地包括執行許多執行緒，其中每一執行緒係執行相同程式。程式可有利地包括一些指令，以從「全域記憶體(例如系統記憶體 104、繪圖記憶體 124、或 GPU 122 和 CPU 102 二者可存取的任何其他記憶體)」讀取輸入資料；執行各種運算，包括輸入資料進行至少一些倍精確度運算，以產生輸出資料；及將輸出資料寫至全域記憶體。特定處理工作對本發明不是具決定性。

應該明白，在此描述的繪圖管線只是說明，且各種

變化與修改是可能的。管線可包括不同於顯示的一些單元，且處理事件的順序可與在此描述不同。此外，在此描述的一些或全部模組的多個實例可平行操作。在一此具體實施例中，多執行緒核心陣列 202 包括可平行操作的兩或多個幾何模組 218、與相等數目的像素模組 224。每一幾何模組與像素模組係共同控制多執行緒核心陣列 202 中的不同處理引擎子集。

C. 核心概觀

多執行緒核心陣列 202 可有利地包括一或多個處理核心，該等處理核心適合平行執行大量處理執行緒，其中術語「執行緒」是指在一特定輸入資料集上執行的一特定程式之實例。例如，一執行緒可為下列程式的一實例：執行在單頂點屬性的一頂點著色器程式、執行在一給定基本圖元與像素的一像素著色器程式、或一般目的計算程式。

第三圖為根據本發明之一具體實施例的一執行核心 300 之區塊圖。可例如在上述多執行緒核心陣列 202 中實施的執行核心 300 係組態成執行任意序列的指令，用以執行各種計算。在一些具體實施例中，相同執行核心 300 在所有繪圖著色階段可用來執行著色器程式，包括頂點著色器、幾何著色器、及/或像素著色器程式、以及一般目的計算程式。

執行核心 300 包括一拾取與分發單元 302、一發送單元 304、一倍精確度融合乘加 (DFMA, “Double-Precision Fused Multiply-Add”)單元 320、其他功能單元(FU, “Functional Unit”)322 的數目(N)、與一暫存器檔案 324。每一功能單元 320、322 組態成執行指定的運算。在一具體實施例中，除了如下述的其他倍精

確度運算之外，DFMA 單元 320 可有利地實施一倍精確度融合乘加運算。應該瞭解，任何數目的 DFMA 單元 320 可包在一核心 300 中。

其他功能單元 322 可為通常習知的設計，並可支援多種運算，例如單精確度加算、乘算、位元邏輯(Bitwise Logic)運算、比較運算、格式轉換操作、結構篩選、記憶體存取(例如，載入及儲存操作)、超越函數的近似、插值法等。功能單元 320、322 可經管線處理，以允許在結束一先前指令之前，發送一新指令，如技術中已熟知者。可提供功能單元的任何組合。

在執行核心 300 的運算時，拾取與分發單元 302 係從一指令儲存裝置(未在圖顯示)獲得指令，將該等指令解碼，及將解碼指令(當作與運算元參考或運算元資料有關的運算碼)分發至發送單元 304。對於每一指令而言，發送單元 304 可例如從暫存器檔案 324 獲得任何參考的運算元。當一指令的所有運算元準備好時，發送單元 304 可藉由傳送運算碼與運算元，將指令發送至 DFMA 單元 320 或另一功能單元 322。發送單元 304 係有利地使用運算碼以選擇適當的功能單元執行一給定的指令。拾取與分發單元 302 與發送單元 304 可使用習知的微處理器架構與技術加以實施，並將省略詳細描述，不致對本發明之瞭解造成失焦。

DFMA 單元 320 及其他功能單元 322 係接收運算碼與相關的運算元，並執行運算元的指定運算。結果資料是以結果值的形式加以提供，該結果值可經由一資料傳輸路徑 326 轉送至暫存器檔案 324(或另一目的地)。在一些具體實施例中，暫存器檔案 324 包括一本機暫存器檔案，具有配置給特定執行緒的區段；以及一全域暫存器檔案，此允許資料在不同執行緒之間共用。在程式執行

期間，暫存器檔案 324 可用來儲存輸入資料、中間結果等。暫存器檔案 324 的一特定實施對本發明不是具決定性的。

在一具體實施例中，核心 300 是多執行緒，並可例如藉由維持與每一執行緒有關的目前狀態資訊，同時執行多達最大數目執行緒(例如，384、768)。核心 300 係有利地設計成可在不同執行緒之間快速切換，以便，例如，來自一頂點執行緒的一程式指令可在一時脈週期上發送，且該指令之後係來自一不同頂點執行緒或一不同類型執行緒(例如一幾何執行緒或一像素執行緒等)的一程式指令。

應該明白，第三圖的執行核心只是說明，且不同的變化與修改是可能的。任何數目核心可包括在一處理器中，且任何數目的功能單元可包括在一核心中。拾取與分發單元 302 與發送單元 304 可實施任何想要的微架構，可視需要包括具順序或無順序指令發送、預測執行模式、單指令多資料(SIMD, “Single-Instruction, Multiple Data”)等的純量、超純量或向量架構。在一些結構中，發送單元可接收及/或發送一長指令字，包括用於多功能單元的運算碼與運算元、或用於一功能單元的運算碼及/或運算元。在一些結構中，執行核心包括可平行操作的每一功能單元的多個實例，例如，用於執行 SIMD 指令。執行核心亦可包括一序列的管線功能單元，其中來自一管線級中功能單元的結果係轉送至在稍後管線級中的功能單元，而不是直接至一暫存器檔案；此組態中的功能單元可藉由單一長指令字或個別指令來控制。

此外，閱讀此說明的熟諳此項技術人士應該明白，DFMA 單元 320 能夠使用任何微處理器如同一功能單元加以實施，並未限於繪圖處理器或任何特定處理器或執

行核心結構。例如，DFMA 單元 320 可在一般目的平行處理單元或一 CPU 中實施。

C. DFMA 單元概觀

根據本發明之一具體實施例，執行核心 300 包括一可執行三類型運算的 DFMA 單元 320：倍精確度算術運算、比較運算、及在倍精確度與其他格式之間的格式轉換。

DFMA 單元 320 可有利地使用其他浮點與固定點格式以處理倍精確度浮點格式的輸入與輸出，供轉換運算；用於不同運算的運算元可為不同格式。在描述 DFMA 單元 320 的一具體實施例之前，將定義一些代表性格式。

如在此使用的「fp32」是指標準 IEEE 754 單精確度浮點格式，其中一正常浮點數字是以一符號位元、八個指數位元、與 23 個有效數字位元表示。指數是以 127 向上偏移，以便範圍 2^{-126} 至 2^{127} 中的指數可使用從 1 至 254 的整數表示。對於「正常」數字而言，23 個有效數字位元係解釋為隱含 1 作為整數部分的 24 位元尾數之分數部分。(在此使用術語「有效數字」是指當隱含前導 1 時，而「尾數」是用來表示(若適用)前導 1 已明確宣告)。

如在此使用的「fp64」是指標準 IEEE 754 倍精確度浮點格式，其中一正常浮點數字是以符號位元、11 個指數位元與 52 個有效數字位元表示。指數是以 1023 向上偏移，以便在範圍 2^{-1022} 至 2^{1023} 中的指數可使用從 1 至 2046 的整數表示。對於「正常」數字而言，52 個有效數字位元係解釋為隱含 1 作為整數部分的 53 位元尾數之分數部分。

如在此使用的「fp16」是稱為普遍使用在繪圖的「半精確度」浮點格式，其中一正常浮點數字係藉由一符號

位元表示，5 個指數位元與 10 個有效數字位元。指數是向上偏移 15，以便在範圍 2^{-14} 至 2^{15} 中的指數可使用從 1 至 30 的整數來表示。對於「正常」數字而言，10 個有效數字位元係解釋為隱含 1 作為整數部分的 11 位元尾數之分數部分。

在 fpl6、fp32 和 fp64 格式中，指數位元中全是零的數字是稱為反向規格化數字(或簡稱「反向規格化」)，並解釋為在尾數中不含有前導 1；此數字可代表例如計算中的下限溢位。指數位元中的全是 1 的(正或負)數字、及有效數字中的零是稱為(正或負)INF；此數字可代表例如計算中的上限溢位。指數位元中全是 1 的數字與有效數字位元中的一非零數字是稱為非數字(NaN, "Not a Number")，並可用來例如代表未定義的數值。零亦認為一特別數字，並可藉由設定成零的所有指數與有效數字位元表示。零可帶有任一符號；如此，允許使用正與負零二者。

固定點格式在此係藉由起始「s」或「u」指定，表示格式是否為有符號或無符號，且一數字係表示總位元數目(例如，16、32、64)；因此，s32 是指符號 32 位元格式，u64 是指一無符號 64 位元格式等。對於有符號格式而言，可有利地使用 2 的補數表示法。在此使用的所有格式中，最高有效位元(Most Significant Bit，MSB)是在位元欄位的左邊，且最低有效位元(Least Significant Bit，LSB)是在欄位的右邊。

應可瞭解，在此定義及參考的這些格式只是為說明目的，且一 DFMA 單元可支援這些格式或不同格式的任何組合，不致脫離本發明的範疇。特別地係，應可瞭解，「單精確度」和「倍精確度」是指任何兩不同浮點格式，未限於現階段定義的標準；一個倍精確度格式(例如，

fp64)是指使用比一相關單精確度格式(例如，fp32)更多位元數目的任何格式，以代表較大範圍的浮點數字、及/或代表較高精確度的浮點數值。同樣地，「半精確度」通常是指使用比一相關單精確度格式更少位元之一格式，以代表較小範圍的浮點數字、及/或代表具較低精確度的浮點數字。

根據本發明的DFMA單元320之一具體實施例現將描述。第四圖為列出藉由DFMA單元320的此具體實施例執行倍精確算術、比較運算與格式轉換運算的表400。

區段402列出算術運算。加算(DADD)是加算兩fp64輸入A和C，並傳回fp64加總A+C。乘算(DMUL)是乘算兩fp64輸入A和B，並傳回fp64乘積A*B。融合乘加(DFMA)係接收三個fp64輸入A、B和C，並計算A*B+C。運算是「融合」在於A*B的結果加至C之前，乘積A*B並未被捨入；使用精確值A*B改善準確度，並遵從即將來臨關於浮點算術運算的IEEE 754R標準。

區段404列出比較運算。一最大運算(DMAX)係傳回fp64運算元A和B之較大一者，且一最小運算(DMIN)傳回兩者之較小一者。二位元測試運算(DSET)係執行倍精確度運算元A和B的許多二位元關係測試之一者，並傳回一布林值以指出是否滿足該測試。在此具體實施例中，可測試的二位元關係包括大於(A > B)、小於(A < B)、等於(A = B)、及無序(A ? B，若A或B是NaN，則此關係式為真)、以及逆反性(例如，A ≠ B)、及各種組合測試(例如A ≥ B、A <> B、A ?= B等)。

區段406係列出格式轉換與捨入運算。在此具體實施例中，DFMA單元320可將fp64格式數字轉換成其他64位元或32位元格式的數字，反之亦然。D2F運算將運算元A從fp64轉換成fp32，F2D運算將運算元A從

fp32 轉換成 fp64。D2I 運算將運算元 A 從 fp64 轉換成 s64、u64、s32 和 u32 格式之任一者；應可瞭解，確定的運算碼可用來識別目標格式。I2D 運算將整數運算元 C 從任一 s64、u64、s32 和 u32 格式轉換成 fp64 格式；再次，應可瞭解，確定的運算碼可用來識別來源格式。在此具體實施例中，DFMA 單元 320 可支援來回於倍精確度格式之所有轉換；其他功能單元可執行其他格式轉換(例如，在 fp32 和 fpl6 之間；在 fp32 和整數格式之間等)。

D2D 運算是用來將捨入運算(例如 IEEE 捨入模式)應用至一 fp64 運算元。這些運算是將 fp64 運算元捨入至以 fp64 格式表示的一整數值。在一具體實施例中，支援的 D2D 運算包括截掉(捨入至零)、捨入至正+INF(ceil)、捨入至負-INF(floor)、與最近數值(向上或向下捨入至最近整數)。

在此具體實施例中，DFMA 單元 320 不提供更進階算術函數的直接硬體支援，例如除算、餘數或平方根。然而，DFMA 單元 320 可用來加速這些運算的以軟體為主之實施。例如，除算的一通用方法係估算一商數 q=a/b，然後使用 t=q*b-a 以測試該估算。若 t 是零，商數 q 已正確決定。若撤銷，則大小 t 是用來修改估算的商數 q，並重複測試直到 t 變成零。可使用單一 DFMA 運算(使用 A=q, B=b, C=-a)正確計算每一反復的測試結果 t。同樣地，對於平方根而言，一通用方法係估算 $i=a^{1/2}$，然後計算 t =r*r-a 以測試該估算，且若 t 不是零，則修改 r。再次，可使用單一 DFMA 運算(使用 A= B=r，C=-a)正確計算每一反復的測試結果 t。

第 2 和 3 節係描述可執行在第四圖中顯示所有操作的 DFMA 單元 320。第 2 節係描述 DFMA 單元 320 的一

電路結構，且第 3 節係描述電路結構如何用來執行在第四圖中列出的操作。應該瞭解，在此描述的 DFMA 單元 320 只是說明，並可使用電路區塊的適當組合以支援其他或不同的功能組合。

2. DFMA 單元結構

第五圖為根據本發明之一具體實施例的 DFMA 單元 320 之簡化區塊圖，其支援在第四圖中顯示的所有操作。在此具體實施例中，DFMA 單元 320 係實施用於所有操作的一多級管線。在每一處理器週期上，DFMA 單元 320 可經由運算元輸入路徑 502、504、506 接收(例如，從第三圖的發送單元 304)三個新運算元(A_0、B_0、C_0)；及一運算碼，指出是否經由運算碼路徑 508 執行運算。在此具體實施例中，運算可為第四圖中顯示的任何運算。除了運算之外，運算碼可有利地指示運算元的輸入格式、以及結果使用的輸出格式，此可與或不與輸入格式相同。應該注意，第四圖顯示的運算可具有與此運算有關的多個運算碼；例如，對於具 s64 輸出的 D2I 運算而言，可為一運算碼，且對於具 s32 輸出的 D2I 運算而言，可為一不同的運算碼等。

DFMA 單元 320 係透過其所有管線級以處理每一運算，並在信號路徑 510 上產生一 64 位元(或對於特定格式轉換運算為 32 位元)結果值(OUT)，及在信號路徑 512 上產生一對應的條件碼(COND)。這些信號可傳遞例如至如第三圖所示的暫存器檔案 324、發送裝置 304、或一處理器核心的其他元件，此係取決於所使用的架構。在一具體實施例中，一管線級係對應至一處理器週期；在其他具體實施例中，一管線級可包括多個處理器週期。此外，管線中的不同路徑可有利地平行操作。

第 2.A 節係提供 DFMA 管線的概觀，且第 2.B-1 節係詳細描述每一區段的電路區塊。

A. DFMA 管線

管線的最初瞭解係關於如何在 DFMA 運算期間使用電路區塊。運算元準備區塊 514 係執行運算元格式化(用於不是 fp64 格式的運算元)與特別數字偵測；運算元準備區塊 514 亦從輸入 fp64 運算元擷取尾數(Am、Bm、Cm)、指數(Ae、Be、Ce)與符號位元(As、Bs、Cs)。在一具體實施例中，沒有不合法的運算元組合；可只忽略未使用在一特定運算的任何運算元。

尾數路徑 516 係計算尾數 Am 和 Bm 的乘積。同時，指數路徑 518 係使用指數 Ae 和 Be 決定在乘積 A*B 與運算元 C 之間的相對對準，並將運算元 C(C_align)的一對齊尾數供應給尾數路徑 516。尾數路徑 516 將 C_align 加至乘積 Am*Bm，然後將該結果予以正規化。基於該正規化，尾數路徑 516 將一對準信號(ALIGN_NORM)提供回給指數路徑 518，其使用 ALIGN_NORM 信號連同指數 Ae、Be 和 Ce 決定最後結果的指數。

符號路徑 520 係接收來自運算元準備區塊 514 的符號位元 As、Bs 和 Cs，並決定結果的符號。尾數路徑 516 係偵測結果為零的情況，並將零結果(R_ZERO)發信通知給符號路徑 520。

輸出區段 522 係接收來自尾數路徑 516 的一結果尾數 Rm、來自指數路徑 518 的一結果指數 Re、及來自符號路徑 520 的一結果符號 Rs。輸出區段 522 亦接收來自運算元準備區塊 514 的特別數字信號(SPC)。基於此資訊，輸出區段 522 係格式化最後結果(OUT)，以傳遞至輸出路徑 510，並在輸出路徑 512 產生一條件碼

(COND)。有利地包括少於結果位元的條件碼係運送關於結果本質的一般資訊。例如，條件碼可包括位元，指出結果是否為正、負、零、NaN、INF、反向規格化等。如技術中已知者，在一條件碼具有一結果的情況中，該結果的隨後使用者有時可使用條件碼，而不是在其處理中的結果本身。在一些具體實施例中，條件碼可用來指示運算執行期間發生的一異常或其他事件。在其他具體實施例中，條件碼可完全省略。

應該瞭解，雖然例如「尾數路徑」和「指數路徑」的名稱可建議係在特定運算(例如，DFMA)期間，藉由每一路徑的各種電路區塊執行的功能，但是電路區塊連同任何內部資料路徑能夠以一運算相關的方式而於多種使用予以有效利用。範例是在下面描述。

除了資料路徑之外，DFMA 單元 320 亦提供如第五圖的一控制區塊 530 所表示之一控制路徑。控制區段 530 係接收運算碼，並產生各種運算碼有關的控制信號，通常在此是以「OPCTL」表示，其可與透過管線的資料傳遞同步而傳遞至每一電路區塊。(連接 OPCTL 信號至各種電路區塊未在第五圖顯示。)如下述，OPCTL 信號係回應運算碼以啟用、停用、或控制 DFMA 單元 320 的各種電路區塊操作，以便可使用相同管線元件執行不同運算。在此所示的各種 OPCTL 信號可包括運算碼本身或從運算碼取得的一些其他信號，例如藉由在控制區塊 530 中實施的組合邏輯。在一些具體實施例中，在數個管線級中，可使用多個電路區塊實施控制區塊 530。應該瞭解，一給定操作期間，提供給不同區塊的 OPCTL 信號可為相同信號或不同信號。鑒於本發明，熟諳此項技術人士將可建構適當的 OPCTL 信號。

應該注意，一給定電路級之電路區塊可能需要不同

量的處理時間，且在一特定電路級所需的時間可能隨著不同操作而變化。因此，DFMA 單元 320 亦可包括各種時序與同步電路(未在第五圖顯示)，以控制不同管線級的不同路徑上傳遞資料。可使用任何適當時序電路(例如，閂鎖、傳輸閘等)。

A. 運算元準備

第六圖為根據本發明之一具體實施例的運算元準備區塊 514 之區塊圖。運算元準備區塊 514 係接收輸入運算元 A、B 和 C，並提供尾數部分(Am、Bm、Cm)至尾數路徑 516，提供指數部分(Ae、Be、Ce)至指數路徑 518，及提供符號位元(As、Bs、Cs)至符號路徑 520。

運算元 A、B 和 C 是在個別 NaN 偵測區塊 612、614、616 與個別絕對值/負區塊 618、620、622 接收。每一 NaN 偵測區塊 612、614、616 係決定接收的運算元是否為 NaN(在指數位元中全是 1，且在有效數字位元中是一非零數字)，並產生一對應的控制信號。

絕對值/負區塊 618、620、622 可回應 OPCTL 信號(未在圖明確顯示)將運算元的符號位元顛倒。例如，第四圖中列出的任一運算可指定是否使用一運算元的負、或一運算元的絕對值。區塊 618、620、622 可將符號位元顛倒，將一運算元加負號或強迫使符號位元成為非負狀態(真為 IEEE 754 格式為零)。若一輸入運算元是 NaN，適當的絕對值/負區塊 618、620、622 亦使 NaN 未明(例如，藉由將有效數字的前導位元設定成 1)，維持符號位元。絕對值/負區塊 618、620、622 提供其個別輸出至運算元選擇多工器 632、634、636。

對於倍精確算術運算而言，可直接使用絕對值/負區塊 618 產生的運算元 A、B 和 C。對於比較運算而言，

A/B 比較電路 624 係比較運算元 A 和 B。在一具體實施例中，絕對值/負電路 620 係將運算元 B 加負號，且 A/B 比較電路 624 係計算 A 和-B 的加總，當若 A 和-B 是固定點數字。若結果是正，則 A 大於 B；若結果是負，則 A 是小於 B；若結果是零，則 A 等於 B。A/B 比較電路 624 亦可接收來自 NaN 偵測電路 612 和 614 的 NaN 資訊(這些路徑未明確在第六圖顯示)。若 A 或 B 之任一者(或兩者)是 NaN，則 A 和 B 是「無順序」。結果資訊係提供給控制邏輯 630。控制邏輯 630 係將結果資訊當作一信號 R_TEST 提供給輸出區段 522，且亦提供控制信號至運算元選擇多工器 632、634、636。

對於格式轉換運算元而言，輸入可能不是 fp64 格式。一 fp32 擷取電路 626 在 F2D 運算期間是使用中。Fp32 擷取電路 626 係接收運算元 A，並執行非正常 fp32 輸入的所有測試。fp32 擷取電路 626 亦將接收的運算元之有效數字欄位從 23 位元擴充至 52 位元(例如，藉由增加尾零)。fp32 擷取電路 626 係將 8 位元 fp32 指數擴充至 11 位元，並將指數偏移從 127 增加至 1023(例如，藉由將 896 加至 fp32 指數)。

一無符號/有符號(U/S, “Unsigned/Signed”)擷取電路 628 在 I2D 運算期間是使用中。U/S 擷取電路 628 係接收任一 u32、s32、u64 或 s64 格式的固定點運算元 C，並準備將該運算元轉換成 fp64。U/S 擷取電路 628 是將固定點運算元從 1 的補數(或 2 的補數)形式轉換成符號數值(Sign-Magnitude)形式，並預先考慮或附加零，以對齊有效數字欄位中的運算元。U/S 擷取電路 628 係提供其輸出至運算元選擇多工器 636，並亦提供至指數路徑 518 作為一 I2D 輸入信號。

運算元選擇多工器 632、634、636 係回應來自控制

邏輯 630 的信號，以選擇運算元 A、B 和 C。運算元選擇多工器 632 是在來自絕對值/負電路 618 的運算元 A 與常數值 0.0 和 1.0(以 fp64 格式表示)之間選擇。對於 DMUL 和 DFMA 運算而言，可選擇運算元 A。對於 DMIN(DMAX)運算而言，若 A<B(A>B)，則選擇運算元 A；否則可選擇 1.0。對於 DADD 和 I2D 運算而言，可選擇 0.0。

運算元選擇多工器 634 是在來自絕對值/負電路 620 的運算元 B 與常數值 0.0 和 1.0(以 fp64 格式表示)之間選擇。對於 DMUL 和 DFMA 運算而言，可選擇運算元 B。對於 DMIN(DMAX)運算而言，若 B<A(B>A)，可選擇運算元 B；否則選擇 1.0。對於 DADD 和 I2D 運算而言，可選擇 0.0。

運算元選擇多工器 636 是在來自絕對值/負電路 622 的運算元 C、來自 fp32 擷取電路 626 的一擷取 fp32 值、來自 U/S 擷取電路 628 的一擷取無符號或有符號整數值、及一常數值 0.0(以 fp64 格式表示)之中選擇。對於 DADD 和 DFMA 運算而言，可選擇運算元 C。對於 DMUL 和比較運算而言，可選擇常數 0.0。對於 F2D 運算而言，可選擇來自 fp32 擷取電路 626 的擷取 fp32 值，及對於 I2D 運算而言，可選擇來自 U/S 擷取電路 628 的擷取 U/S 值。

選擇多工器 632、634、636 選擇的運算元 A、B 和 C 係提供給特別數字偵測電路 638、640、642。對於 fp64 運算元而言，特別數字偵測電路 638、640、642 係偵測所有特別數字條件，包括反向規格化、NaN、INF 和零。對於 F2D 運算而言，特別數字偵測電路 642 係經由一路徑 644，從 fp32 擷取電路 626 接收 fp32 特別數字資訊。每一特別數字偵測電路 638、640、642 係產生一特別數

字信號(SPC)，以指出運算元是否為一特別數字，且若確認，是何種類型。特別數字信號 SPC 是在信號路徑 524 上提供給輸出區段 522，如第五圖所示。通常習知設計的特別數字偵測邏輯可使用。在一替代具體實施例中，NaN 偵測(藉由電路 612、614 和 616 執行)不在電路 638、640、642 中重複；相反地係，每一特別數字偵測電路 638、640、642 係從 NaN 偵測電路 612、614 和 616 之對應一者接收一 NaN 信號，並使用該信號決定運算元是否為 NaN。

不管是否偵測到任何特別數字，特別數字偵測電路 638、640 和 642 將運算元分尾數、指數與符號位元。特別數字偵測電路 638 提供運算元 A(Am)的尾數部分至尾數路徑 516(第五圖)，提供運算元 A(Ae)的指數部分至指數路徑 518，及提供符號位元(As)至符號路徑 520。特別數字偵測電路 640 提供運算元 B(Bm)的尾數部分至尾數路徑 516，提供運算元 B(Be)的指數部分至指數路徑 518，及提供符號位元(Bs)至符號路徑 520。特別數字偵測電路 642 係提供運算元 C 的尾數部分(Cm)與指數部分(Ce)至指數路徑 518，及提供符號位元(Cs)至符號路徑 520。在一些具體實施例中，特別數字偵測電路 638、640、642 係將前導 1 附加至尾數 Am、Bm、Cm(除了數字是反向規格化之外)。

B. 指數路徑

第七圖為根據本發明之一具體實施例的指數路徑 518 之區塊圖。

一指數計算電路 702 係從運算元準備區塊 514(第五圖)接收指數位元 Ae、Be、和 Ce，並計算 DFMA 結果 A*B+C 的區塊指數。習知的指數計算電路可使用。在一

具體實施例中，若所有運算元是正常數字，指數計算電路加算 Ae 和 Be，並減去 fp64 指數偏移(1023)，以決定乘積 A*B 的指數，然後將乘積指數與指數 Ce 之較大一者選擇作為 DFMA 結果的區塊指數(BLE)。此區塊指數 BLE 係提供給向下資料流最後指數計算電路 704。若一或多個運算元是非正規化(如特別數字信號 SPC 所示)，適當的邏輯可用來決定區塊指數 BLE。在其他具體實施例中，涉及特別數字的運算之指數決定是在如下述的輸出區段 522 中處理。

此外，指數計算區塊 702 係決定運算元 C 的尾數可有效左移或右移以使 Cm 與乘積 Am*Bm 對齊的量。此量是以一 Sh_C 控制信號提供至一移位電路 706。此控制信號係有利地說明 Cm 的額外填補，以便藉由右移 Cm 始終達成一有效的左移或右移。

若在 C 與乘積 A*B 之間存有一相對的減符號，尾數 Cm 提供給一負電路 708，其會有條件將 Cm 加上負號(例如，使用 1 的補數負號)。相對的減符號可在符號路徑 520 中偵測，如下述，且符號控制信號 SignCTL 指示一相對減符號是否出現。負電路 708 的輸出(或 Cm 或 ~Cm)係提供給移位電路 706。

在一具體實施例中，移位電路 706 是一 217 位元桶式移位器(Barrel Shifter)，其能以多達 157 位元將 54 位元尾數 Cm 右移；Sh_C 信號可決定 Cm 右移量。尾數 Cm 可有利地使用一對準以輸入移位器，以致於 Cm 可依需要儘可能右移。217 位元大小可選擇，以便有可供 53 位元尾數 Cm(加一保護位元與一捨入位元)的充足空間整個向左排列，或整個向 106 位元乘積 A*B(加上一保護位元與一捨入位元作為乘積)的右邊排列，其會是 217 位元欄位的 MSB 右邊的排列 55 位元。桶式移位器的右

移移出的任何位元會被丟棄。在其他具體實施例中，一旗標位元可使來保持追蹤從桶式移位器右移移出的所有位元是否為「I」，且此資訊可用在下述的捨入運算。

在一替代具體實施例中，習知交換多工器可用來在乘積 Am*Bm 及 Cm 之間選擇較大的運算元，然後將較小的運算元右移。

對於 D2D 運算而言，尾數 Cm 亦提供給一 D2D 邏輯電路 710。D2D 邏輯電路 710 係接收尾數 Cm、指數 Ce 與符號 Cs，並應用整數捨入規則。在一具體實施例中，D2D 邏輯電路 710 係基於指數 Ce 以決定二進制小數點的位置，然後基於一 OPCTL 信號(未在圖明確顯示)應用選定的捨入規則。可使用實施捨入模式的習知邏輯，並可支援捨入模式的任何組合，包括(但未限於)截掉、捨入至正無限大(ceil)、捨入至負無限大(floor)、和最近數值。

選擇多工器 712 係從 U/S 擷取電路 628(第六圖)接收移位的尾數 C_Shift、D2D 邏輯電路輸出、與 I2D 輸入，並基於一 OPCTL 信號，選擇這些輸入之一者作為供應給尾數路徑 516 的一對齊尾數 C_align。對於倍精確算術運算與比較運算而言，可選擇運算元 C_Shift。對於格式轉換 D2D 或 I2D 而言，可選擇適當的替代輸入。

下限溢位邏輯 713 係組態成以 fp64 和 fp32 結果偵測潛在下限溢位。針對除了 D2F 運算以外之其他運算，下限溢位邏輯 713 可決定 11 位元 fp64 區塊指數 BLE 是否為零或相當接近零，此時，一反向規格化結果是可能的。基於區塊指數，下限溢位邏輯 713 可決定在指數到達零之前，可將尾數左移的最大位元數目。此數目可以 8 位元下限溢位信號 U_fp64 提供給尾數路徑 516(請參見第八圖)。對於 D2F 運算而言，指數可視為一 8 位元 fp32

指數，且下限溢位邏輯 713 可決定允許的最大左移。此數目可以 8 位元下限溢位信號 U_fp32 提供給尾數路徑 516。

指數路徑 518 亦包括最後指數計算邏輯電路 704。區塊指數 BLE 係提供給一減算電路 720。而且提供給減算電路 720 是來自尾數路徑 516 的一區塊移位(BL_Sh)信號。如下述，當乘積 Am*Bm 加至運算元 C_align 時，BL_Sh 信號係反映取消 MSB 的影響。減算電路 720 是從 BLE 減去 BL_Sh，以決定一差 EDIF。下限溢位/上限溢位電路 722 係偵測減算結果 EDIF 中的下限溢位或上限溢位。加 1 電路 724 係將 1 加至結果 EDIF，且基於該下限溢位/上限溢位條件，多工器 720 是將在 EDIF 與 EDIF+1 信號之間選擇為結果指數 Re。結果 Re 與下限溢位/上限溢位信號(U/O)係提供給輸出區段 522。

C. 尾數路徑

第八圖為根據本發明之一具體實施例的尾數路徑 516 之區塊圖。尾數路徑 516 係執行運算元 A、B 和 C 尾數的乘積及加總運算。

53x53 乘法器 802 係從運算元準備區塊 514 接收尾數 Am 和 Bm(如上述)，並計算一 106 位元乘積 Am*Bm。該乘積係提供給一 168 位元加法器 804，該加法器亦接收對齊的尾數 C_align。桶式移位器 706 使用的 217 位元欄位的結尾位元可丟棄，或指出結尾位元是否為非零或全部是 1 的旗標位元可保持。加法器 804 係產生 Sum 和 ~Sum(加總的 2 補數)輸出。多工器 806 係基於該加總的 MSB(符號位元)，以在 Sum 和~Sum 之間選擇。選定的加總(S)係提供至一零偵測電路 814 及一左移電路 816。零偵測電路 814 可決定選定的加總 S 是否為零，並提供

一對應的 R_ZERO 信號至符號路徑 520。

尾數路徑 516 亦使加總 S 正規化。前導零偵測(LZD, "Leading Zero Detection")係使用 LZD 電路 808、810 在 Sum 和~Sum 上同時執行。每一 LZD 電路 808、810 係產生一 LZD 信號(Z1、Z2)，表示在其輸入中的前導零數字。一 LZD 多工器 812 係基於加總的 MSB(符號位元)以選擇相關的 LZD 信號(Z1 或 Z2)。若多工器 806 選擇 Sum，則選擇 Z2，且若多工器 806 選擇~Sum，則選擇 Z1。選定的 LZD 信號係以區塊移位信號 BL_Sh 提供至指數路徑 518，其中該信號是用來調整如上所述的結果指數。

正規化邏輯 818 係選擇一左移量 Lshift，其可決定加總 S 的一正規化移位。對於正常數字結果而言，有利地係，左移量可大到足以將前導 1 移出尾數欄位，留下 52 位元有效數字(加上保護與捨入位元)。然而，在一些實例中，結果是一下限溢位，其應該是以 fp64 或 fp32 反向規格化表示。在一具體實施例中，對於除了 D2F 的運算之外，除非 BL_Sh 係大於下限溢位信號 U_fp64 之外，否則正規化邏輯 818 係從 LZD 多工器 812 選擇輸出 BL_Sh，在此情況，正規化邏輯 818 將 U_fp64 選用為左移量。對於 D2F 運算而言，正規化邏輯 818 係使用 fp32 下限溢位信號 U_fp32 以限制左移量 Lshift。

左移電路 816 係依 Lshift 量將加總 S 予以左移。結果 Sn 係提供給捨入邏輯 820、加 1 加法器 822 與一尾數選擇多工器 824。捨入邏輯 820 可有利地實施為 IEEE 標準算術運算而定義的四個捨入模式(最近值、捨入至負無限大(floor)、捨入至正無限大(ceil)與截掉)，藉由不同模式可選擇不同結果。OPCTL 信號或另一控制信號(未在圖顯示)可用來指定任一捨入模式。基於捨入模式與正

規化加總 Sn，捨入邏輯 820 可決定是否選擇由加 1 加法器 822 所計算的結果 Sn 或 Sn+1。藉由選擇適當的結果(Sn 或 Sn+1)，選擇多工器 824 可回應來自捨入邏輯 820 的一控制信號。

多工器 824 選擇的結果係傳遞至一格式化區塊 826。對於具有一浮點輸出的運算而言，格式化區塊 826 提供尾數 Rm 至輸出區段 522。有利地係，加總 S 是至少 64 位元寬度(支援整數運算)，且格式化區塊 826 亦可移除無關重要的位元。對於 D2I 運算(具有整數輸出)而言，格式化區塊 826 係將結果分成一 52 位元 int_L 欄位(包含 LSB)、與一 11 位元 int_M 欄位(包含 MSB)。Rm、int_L 和 int_M 係傳遞至輸出區段 522。

D. 信號路徑

第九圖為根據本發明之一具體實施例的符號路徑 520 之區塊圖。符號路徑 520 係從運算元準備區塊 514(第五圖)接收運算元 As、Bs、和 Cs 的符號。符號路徑 520 亦接收來自尾數路徑 516 的零結果信號 R_ZERO、與一 OPCTL 信號，以指出運算進行的類型、以及來自運算元準備區塊 514 的特別數字信號 SPC。基於此資訊，符號路徑 520 係決定結果的符號及產生一符號位元 Rs。

更明確地係，符號路徑 520 包括一乘積/加總電路 902 與一最後符號電路 904。乘積/加總電路 902 係接收來自運算元準備區塊 514 的運算元 A、B 和 C 之符號位元 As、Bs 和 Cs。乘積/加總電路 902 係使用符號位元 As 和 Bs、與習知符號邏輯規則以決定乘積 A*B 的符號(Sp)，然後將乘積的符號與符號位元 Cs 相比較，以決定乘積與運算元 C 是否具有相同符號或相反符號。基於該決定，乘積/加總電路 904 可確證或取消確證 SignCTL

信號，此信號係傳遞至指數路徑 518(第七圖)中的最後符號電路 904 與負區塊 708。此外，若乘積與運算元 C 具有相同符號，最後結果亦將具有該符號；若乘積與運算元 C 具有相反符號，則結果將取決於較大一者。

最後符號電路 904 係接收決定最後符號所需的所有資訊。明確地係，最後符號電路 904 係接收符號資訊，包括乘積的符號 Sp、與來自乘積/加總電路 902 的 SignCTL 信號、以及符號位元 As、Bs 和 Cs。最後符號電路 904 亦接收來自尾數路徑 516 的零偵測信號 R_ZERO、及來自運算元準備區塊 514 的特別數字信號 SPC。最後符號電路 904 亦在尾數路徑 516 上接收來自加法器 804 的加總的 MSB，其可指示該加總是否為正或負。

基於此資訊，最後符號電路 904 可將習知的符號邏輯用來決定結果的一符號位元 Rs。例如，對於 DFMA 運算而言，若符號位元 Sp 和 Cs 是相同，結果亦將具有該符號。若 Sp 和 Cs 是相反的符號，則尾數路徑 516 中的加法器 804 計算(Am*Bm)-C_align。若 Am*Bm 大於 C_align，則加法器 804 將計算一正結果 Sum，且應該選擇乘積符號 Sp；若 Am*Bm 是小於 C_align，則加法器 804 將計算一負結果 Sum，且應該選擇符號 Cs。加法器 804 的 Sum 輸出的 MSB 係表示結果的符號，並可用來驅動此項選擇。若結果 Sum 是零，則可確定 R_ZERO 信號，且如適用，最後符號電路 904 亦可選擇任一符號(零可為 fp64 格式的正或負)。對於其他運算而言，最後符號電路 904 可使一或另一運算元的符號通過，做為最終的符號。

E. 輸出區段

第十圖為根據本發明之一具體實施例的 DFMA 單元 320 的輸出區段 522 之區塊圖。

輸出多工器控制邏輯 1002 係接收來自指數路徑 518(第七圖)的下限溢位/上限溢位(Underflow/Overflow，U/O)、來自運算元準備區塊 514(第六圖)的 R_test 和 SPC 信號、及一 OPCTL 信號，以表示運算進行的類型。基於此資訊，輸出多工器控制邏輯 1002 產生選擇控制信號，以用於一有效數字選擇多工器 1004 與一指數選擇多工器 1006。輸出多工器控制邏輯 1002 亦產生條件碼信號 COND，其可表示例如上限溢位或下限溢位、NaN 或其他條件。在一些具體實施例中，在 DSET 運算期間，條件碼亦用來發信通知布林結果。

有效數字選擇多工器 1004 係從尾數路徑 516 接收結果有效數字 Rm，且多達 52 位元的一整數輸出(在 D2I 運算期間使用)、以及許多特別值。在一具體實施例中，特別值包括：一些 1 的 52 位元欄位(用來表示 D2I 運算的最大 64 位元整數)；一些零的 52 位元欄位(在結果為 0.0 或 1.0 時使用)；一 52 位元欄位 0x0_0000_8000_0000(用來表示 D2I 運算的最小 32 位元整數)；一具前導 1 的 52 位元欄位(用來表示內部產生的未明 NaN)；一 max_int32 值，例如，0x7fff_ffff_ffff_ffff(用來表示 D2I 運算的最大 32 位元整數)；一未明的 NaN 值(用來使一輸入運算元通過，此運算元係來自運算元準備區塊 514 的 NaN)；及一 min_denorm 值，例如，在最後位元位置的 1(用於下限溢位)。任一輸入可選擇，此係取決於運算及任何一運算元或結果是否為一特別數字。

指數選擇多工器 1006 係從指數路徑 518 接收結果指數 Re，且多達 11 個整數位元(MSB 代表整數格式輸出)、以及許多特別值。在一具體實施例中，特別值包括：

0x3ff(指數代表 fp64 格式的 1.0)、0x 000(指數代表反向規格化和 0.0)、0x7fe(最大 fp64 指數代表正常數字)及 0x7ff.(代表 fp64 NaN 或 INF 結果)。任一輸入可選擇，此係取決於運算及任一運算元或結果是否為一特別數字。

串接區塊 1008 係接收符號位元 Rs、有效數字選擇多工器 1004 選擇的有效數字位元、與指數多工器 1006 選擇的指數位元。串接區塊 1008 係格式化結果(例如，按照 IEEE 754 標準的符號、指數、有效數字順序)，並提供 64 位元輸出信號 OUT。

F. 運算元旁路或通過路徑

在一些具體實施例中，DFMA 單元 320 提供旁路或通過路徑，以讓運算元將不修改者傳遞通過電路區塊。例如，在一些運算期間，乘法器 802 係將一輸入(例如 Am)乘以 1.0，有效地通過輸入 Am。可提供乘法器 802 附近的輸入 Am 之一旁路，而不是將 Am 乘以 1.0。旁路可有利地消耗與乘法器 802 相同的時脈週期數目，致使 Am 可在正確時間抵達加法器 804 的輸入。但是當旁路過乘法器 802 時，該乘法器可設定成一非作用或低電力狀態，藉此減少與微小增加電路面積交換的功率消耗。同樣地，在一些運算期間，加法器 804 是用來將零加至一輸入(例如，C_align)，有效地通過輸入 C_align。不是將零加至 C_align，而是可提供加法器 804 附近的輸入 C_align 之旁路，特別是對於預先已知多工器 806 所要選擇加法器 804 的 Sum 和~Sum 輸出之一者的運算；輸入 C_align 可旁路至 Sum 和~Sum 路徑之正確一者。再次，旁路可有利地消耗與加法器 804 相同的時脈週期數目，以便不影響時序，但是因為加法器 804 可於旁路過加法

器本身的運算之時，使本身置於一非作用或低電力狀態，所以可減少功率消耗。

因此，在第 3 節(下面)的操作描述是指旁路或通過至一特定電路區塊的各種運算元；應該瞭解，此可藉由控制任何介入的電路區塊執行不影響運算元(例如，加零或乘以 1.0)的運算予以完成，以便區塊的輸入可通過予以輸出或藉由使用一旁路。此外，遵循一些電路區塊附近之一旁路或通過路徑不必然需要在隨後電路區塊上持續遵循該旁路。此外，一電路區塊中修改的一數值可遵循一隨後電路區塊附近的旁路。在一運算期間所旁路之一特定電路區塊情況中，該電路區塊可設定成一非作用狀態以減少功率消耗或允許與其要忽略的輸出正常操作，例如，透過使用選擇多工器或其他電路元件。

應該明白，在此描述的 DFMA 單元只是說明，且各種變化及修改是可能的。在此描述的許多電路區塊提供習知的功能，並可使用在技術中熟知的技術加以實施；因此，省略了這些區塊的詳細描述。操作電路的區塊細分可修改，且不同區塊可組合或改變。此外，管線級數目與特定電路區塊的分配、或特定管線級的操作亦可修改或改變。對於一特定實施的電路區塊選擇與配置將取決於所要支援的運算集，且熟諳此項技術人士應該明白，不是所有在此描述的區塊需用於每一可能運算的組合。

3. DFMA 單元運算

DFMA 單元 320 係以面積效率方式，有利地利用上述電路區塊支援在第四圖中列出的所有運算。因此，DFMA 單元 320 的運算係取決於執行運算的至少一些方面。下列部分描述將 DFMA 單元 320 用來執行在第四圖

中列出的每一運算。

應該注意，浮點異常(包括，例如上限溢位或下限溢位條件)是在 DFMA 單元 320 中處理，不需要額外處理週期。例如，一輸入運算元是 NaN 或其他特別數字之運算是在第五圖的運算元準備區塊 514 中偵測，且一適當特別數字輸出是在輸出區段 522 中選擇。在一 NaN、下限溢位、上限溢位或其他特別數字發生在運算過程的情況，條件係可偵測，且一適當特別數字輸出可在輸出區段 522 中選擇。

A. 融合乘加(DFMA)

對於 DFMA 運算而言，DFMA 單元 320 係接收 fp64 格式的運算元 A0、B0 和 C0；及一運算碼，以指示是否執行 DFMA 運算。NaN 電路 612、614、616 係決定選定運算元之任一者是否為 NaN。如適用，絕對值/負電路 618、620、622 可針對每一運算元加負符號位元(或不加負號)。運算元選擇多工器 632、634、和 636 係選擇絕對值/負電路 618、620、622 的個別輸出，並提供這些輸出至特別數字偵測電路 638、640、642。特別數字偵測電路 638、640 和 642 係決定每一運算元是否為一特別數字，並在路徑 524 上產生適當的特別數字 SPC 信號。特別數字偵測電路 638、640 和 642 係提供尾數 Am、Bm 和 Cm(具前導 1 附加於正常數字，且前導零附加於反向規格化)至尾數路徑 516；提供指數 Ae、Be 和 Ce 至指數路徑 518；及提供符號位元 As、Bs 和 Cs 至符號路徑 520。

A/B 比較電路 624、fp32 擷取電路 626 和 U/S 整數擷取電路 628 係不用於 DFMA 運算，且若需要，這些電路可設定成一非作用或低電力狀態。

在符號路徑 520 中，乘積/加總電路 902 是從符號位

元 As 和 Bs 決定乘積 A*B 是否為正或負，並比較乘積 Sp 的符號與符號位元 Cs。若乘積與 Cs 具有相反的符號，可確定 SignCTL 信號係表示相反符號；若乘積與 Cs 具有相同符號，則取消確證 SignCTL 信號。

在指數路徑 518(第七圖)中，指數計算區塊 702 係接收指數 Ae、Be 和 Ce。指數計算區塊 702 係加入指數 Ae 和 Be，以決定乘積 A*B 的區塊指數，然後將乘積區塊指數與指數 Ce 之較大者選為結果區塊指數 BLE。指數計算區塊 702 亦從兩者之較大者減去乘積區塊指數與指數 Ce 之較小者，並產生一對應的移位控制信號 Sh_C。下限溢位邏輯 713 係偵測區塊指數 BLE 是否對應至一下限溢位或潛在下限溢位，並產生下限溢位信號 U_fp64(在 DFMA 運算期間不使用 U_fp32 信號)。

負區塊 708 係從運算元準備區塊 514 接收尾數 Cm，並從符號路徑 520 接收 SignCTL 信號。若確定 SignCTL 信號，負區塊 708 將尾數 Cm 予以反轉表示相對減符號，並提供反轉的 Cm 至移位電路 706。否則，負區塊 708 提供沒有修改的 Cm 至移位電路 706。

移位電路 706 係以對應至移位控制信號 Sh_C 的量，將負區塊 708 提供的尾數 Cm 予以右移，並提供移位的 C_Shift 尾數至選擇多工器 712。選擇多工器 712 係選擇移位的尾數 C_Shift，並如運算元 C_align，將該移位的尾數提供至尾數路徑 516。

在尾數路徑 516(第八圖)中，乘法器 802 係計算 106 位元乘積 Am*Bm，並提供該乘積至 168 位元加法器 804。乘法器 802 的運算可與指數計算區塊 702 的運算同時發生。

加法器 804 係從指數路徑 518 的選擇多工器 712 接收運算元 C_align，並加入輸入 Am*Bm 和 C_align，以

決定 Sum 和~Sum。基於 Sum 的 MSB，多工器 806 係將輸出之一者選擇為最後的總數。若 Sum 是正(MSB 是零)，則選擇 Sum，而若 Sum 是負(MSB 是 1)，則選擇~Sum。LZD 電路 808 和 810 係分別決定~Sum 和 Sum 的前導零數字；多工器 812 係將 LZD 輸出之一者選擇為前導零的數目，並提供前導零信號 BL_Sh 至指數路徑 518 與正規化邏輯 818。

多工器 806 選擇的最後總數 S 亦提供至零偵測電路 814。若最後總數是零，零偵測電路 814 斷定符號路徑 520 為 R_ZERO 信號；否則，不是 R_ZERO 信號。

除非 U_fp64 信號指示一下限溢位，否則正規化邏輯 818 將前導零信號選擇為正規化信號 Lshift，在此情況，尾數只移位至對應至 1 的指數之點，致使結果會以非正規化形式表達。移位電路 816 係回應 Lshift 信號以將選定的加總 S 左移，以產生一正規化的總數 Sn。加 1 加法器 822 是將 1 加至正規化總數 Sn。捨入邏輯 820 係使用捨入模式(藉由一 OPCTL 信號指定)與正規化總數 Sn(在路徑 821 上)的 LSB，以決定正規化總數是否應該捨入。若確證，則捨入邏輯 820 控制選擇多工器 824，以從加法器 822 選擇輸出 Sn+1；否則；選擇多工器 824 選擇正規化總數 Sn。選擇多工器 824 提供選定的結果 Rm 至輸出區段 522。在一些具體實施例中，選擇多工器 824 係從結果尾數丟棄前導位元(1 係表示正常數)。

在與捨入運算的同時，指數路徑 518(第七圖)係計算結果指數 Re。明確地係，減算區塊 720 係從指數計算區塊 702 接收區塊指數 BLE、及從尾數路徑 516 接收區塊移位信號 BL_Sh。減算區塊 720 係從其減去兩輸入，並提供結果 EDIF 至下限溢位/上限溢位邏輯 722、加 1 加法器 724、及選擇多工器 726。下限溢位/上限溢位邏輯

722係使用結果的MSB以決定一下限溢位或上限溢位是否發生，並產生U/O信號，以反映是否出現下限溢位或上限溢位。基於U/O信號，選擇多工器726是在減算結果EDIF與加1加法器724的輸出之間選擇。選定的值係以結果指數Re、連同U/O信號同時提供給輸出區段522。

在與捨入運算的同時，符號路徑520(第九圖)中的最後符號電路904可決定最後符號Rs，其決定係基於乘積/加總電路902所決定之符號；R_ZERO信號及從尾數路徑516接收的加總之MSB；及從運算元準備區塊514接收的特別數字SPC信號。

輸出區段522(第十圖)係從尾數路徑516接收結果尾數Rm，從指數路徑518接收結果指數Re，及從符號路徑520接收結果信號Rs，以及從運算元準備區塊514接收特別數字SPC信號，及從指數路徑518接收U/O信號。基於SPC和U/O信號，輸出多工器控制邏輯1002係產生用於有效數字多工器1004的一控制信號，及用於指數多工器1006的一控制信號。輸出多工器控制邏輯1002亦產生不同條件碼COND，例如指示結果是否為一上限溢位、下限溢位或NaN。

有效數字多工器1004係選擇正常數字與反向規格化的有效數字Rm。對於下限溢位而言，可選擇零或min_denorm有效數字，此係取決於捨入模式。對於上限溢位(INF)而言，選擇有效數字0x0_0000_0000_0000。在任何輸入運算元是NaN的情況，可選擇未明的NaN有效數字。若在運算期間產生NaN，可選擇內部(未明)NaN尾數0x8_0000_0000。

指數多工器1006選擇正常數字的結果指數Re。對於反向規格化與下限溢位而言，選擇指數0x000。對於

INF 或 NaN 而言，選擇最大指數 0x7ff。

串接區塊 1008 係接收選定的有效數字與指數及符號 Rs，並產生最終的 fp64 結果 OUT。條件碼可依需要加以設定。

應該注意，DFMA 單元 320 是在相同週期數目內完成所有 DFMA 運算，不管是上限溢位或下限溢位。根據 IEEE 754 標準，DFMA 單元 320 亦實施浮點算術運算的預期之預設上限溢位/下限溢位行為：傳回一適當結果 OUT，且一狀態旗標(在條件碼 COND 中)係設定成指示上限溢位/下限溢位條件。在一些具體實施例中，使用者定義的陷阱(Trap)可實施以處理這些條件；條件碼 COND 可用來決定一陷阱是否應該發生。

B. 乘算

乘算(DMUL)可類似上述運算元 C 設定成零的 DFMA 運算加以實施；DFMA 單元 320 然後計算 A*B+0.0。在一具體實施例中，當運算碼指示 DMUL 運算時，選擇多工器 636(第六圖)可用來將 fp64 零值取代輸入運算元 C。

C. 加算

加算(DADD)可類似上述運算元 B 設定成 1.0 的 DFMA 運算加以實施；DFMA 單元 320 然後計算 A*1.0+C。在一具體實施例中，當運算碼指示 DADD 運算時，選擇多工器 634(第六圖)可用來將 fp64 1.0 值取代輸入運算元 B。

D. DMAX 和 DMIN

對於 DMAX 或 DMIN 運算而言，運算元準備區塊

514(第六圖)係接收運算元 A 和 B。NaN 電路 612 和 614 係決定所選定運算元之一者或二者是否為 NaN。如適當，絕對值/負電路 618、620 可加負符號位元(或不加負號)。

A/B 比較電路 624 是從絕對值/負電路 618、620 接收運算元 A 和 B，並例如藉由從 A 減去 B 執行比較，倘若運算元為整數。基於該減算，A/B 比較電路 624 可產生 COMP 信號，指出 A 是否大於、小於、或等於 B。COMP 信號係提供至控制邏輯 630，此產生對應的 R_Test 信號，並亦產生關於選擇多工器 632、634 和 636 的選擇信號。

明確地係，對於 DMAX 運算，若 A 大於 B，運算元 A 多工器 632 係選擇運算元 A，且若 A 小於 B，則選擇運算元 1.0；當若 B 大於 A 時，運算元 B 多工器 634 選擇運算元 B，若 B 小於 A，則選擇運算元 1.0。對於 DMIN 運算而言，若 A 小於 B，運算元 A 多工器 632 選擇運算元 A，且若 A 大於 B，則選擇運算元 1.0；當若 B 小於 A 時，運算元 B 多工器 634 選擇運算元 B，且若 B 大於 A，則選擇運算元 1.0。對於 DMAX 和 DMIN 二者而言，A=B 的特別情況可藉由控制多工器 632 加以處理，以選擇運算元 A，而多工器 634 係選擇運算元 1.0，或藉由控制多工器 632 以選擇運算元 1.0，而多工器 634 選擇運算元 B。無論如何，運算元 C 多工器 636 可有利地操作以選擇運算元 0.0。

特別數字偵測電路 638、640 和 642 係決定運算元是否為特別數字，並在路徑 524 上產生適當的特別數字 SPC 信號。特別數字偵測電路 638、640 和 642 係提供尾數 Am、Bm 和 Cm(具有前導 1 附加於正常數字，且前導零附加於反向規格化)至尾數路徑 516；提供指數 Ae、

Be 和 Ce 至指數路徑 518、及提供符號位元 As、Bs 和 Cs 至符號路徑 520。

fp32 擷取電路 626 與無符號/有符號整數擷取電路 628 是不用於 DMAX 或 DMIN 運算，且若需要，這些電路可設定成一非使用或低電力狀態。

尾數路徑 516、指數路徑 518 與符號路徑 520 的操作係如上述關於 DFMA 運算。對於 DMAX 運算而言，尾數路徑 516、指數路徑 518 與符號路徑 520 係計算 max(A, B)*1.0+0.0；對於 DMIN 運算而言，尾數路徑 516、指數路徑 518 與符號路徑 520 係計算 min(A, B)*1.0+0.0。因此，對於正常數字而言，Rm、Re 和 Rs 係對應至想要結果的尾數、指數與符號。

輸出區段 522(第十圖)係處理特別數字。特別地係，DMAX 和 DMIN 運算的結果並未針對 NaN 運算元加以定義，且該結果可設定成一 NaN 值。輸出多工器控制邏輯 1002 係使用特別數字 SPC 信號以決定結果是否應該為 NaN；若確證，有效數字多工器 1004 係選擇未明的 NaN 輸入，而指數多工器係選擇 0x7ff。否則，選擇結果 Rm 和 Re。條件碼可依需要予以設定。

在一替代具體實施例中，有尾數路徑 516、指數路徑 518 和符號路徑 520 的一些或所有組件可省略；省略的組件可置於一低電力狀態。旁路可包括各種延遲電路(閂鎖等)，以便路徑占用與尾數路徑 516、指數路徑 518 和符號路徑 520 的最長的相同管線級數目。此可確保 DFMA 單元 320 中所有運算需要完成的相同週期數目，此簡化指令發送邏輯。

E. DSET

類似 DMAX 和 DMIN，DSET 運算係使用運算元準

備區塊 514(第六圖)中的 A/B 比較電路 624。不像 DMAX 和 DMIN，DSET 不會傳回輸入運算元之一者，而是傳回一布林值，以指出是否滿足測試的條件。

對於 DSET 運算而言，運算元準備區塊 514(第六圖)係接收運算元 A 和 B。NaN 電路 612 和 614 係決定所選定的運算元之任一者或二者是否為 NaN。如適用，絕對值/負電路 618、620 可加負符號位元。

A/B 比較電路 624 是從絕對值/負電路 618、620 接收運算元 A 和 B，及例如藉由從 A 減去 B 執行比較，倘若運算元是整數及考慮其個別符號位元。基於該減算，A/B 比較電路 624 可產生 COMP 信號，指出 A 是否大於、小於或等於 B。COMP 信號係提供至控制邏輯 630，以產生對應 R_Test 信號，並亦產生 A 多工器 632、B 多工器 634、和 C 多工器 636 的選擇信號。由於 DSET 運算的結果是布林，所以在一具體實施例中，所有三個多工器 632、634 和 636 都選擇零運算元。在另一具體實施例中，多工器 632 和 634 選擇運算元 A 和 B；特別數字偵測電路 638 和 640 決定運算元是否為特別數字，並在路徑 524 上產生適當的特別數字 SPC 信號。

fp32 擷取電路 626 與無符號/有符號整數擷取電路 628 是不用於 DSET 運算，且若需要，這些電路可設定成一非使用或或低電力狀態。

尾數路徑 516、指數路徑 518 與符號路徑 520 的操作係如上述關於 DFMA 運算，或可侷部或全部省略。任何省略的組件可置於一低電力狀態。如上述，旁路路徑可包括各種延遲電路(閂鎖等)，以便路徑占用與尾數路徑 516、指數路徑 518 與符號路徑 520 的最長相同的管線級數目。此可確保 DFMA 單元 320 的所有運算需要完成的相同週期數目，此簡化指令發送邏輯。

輸出區段 522(第十圖)係處理特別數字。特別地係，根據 IEEE 754 標準，若 A 或 B(或二者)是 NaN，A 和 B 是無順序。輸出多工器控制邏輯 1002 係接收 R_Test 信號，以指示 A 是否大於、小於、或等於 B；特別數字 SPC 信號指出 A 或 B 是否為 NaN；及一 OPCTL 信號指出是否要求特定測試操作。輸出多工器控制邏輯 1002 係使用 R_Test 和 SPC 信號以決定是否滿足要求的測試。在一具體實施例中，一 DSET 運算的結果係以一條件碼加以提供，並忽略結果 OUT；在此情況中，輸出多工器控制邏輯 1002 係設定條件碼 COND 以指示結果，並可任意選擇輸出 OUT 的有效數字與指數。在另一具體實施例中，輸出 OUT 可設定，以反映測試結果，在此情況中，輸出多工器控制邏輯 1002 係操作有效數字多工器 1004 與指數多工器 1006，以若滿足此測試，選擇對應至邏輯 TRUE(真)的 64 位元值；若不滿足此測試，則選擇對應至邏輯 FALSE(假)的 64 位元值。

F. 格式轉換

在一些具體實施例中，DFMA 單元 320 亦支援來回於倍精確度格式的格式轉換操作。現將描述下列一些範例。

1.fp32 至 fp64 (F2D)

對於 F2D 運算而言，一 fp32 輸入運算元 A 係轉換成一對應的 fp64 數字。特別數字輸入係適當地處理；例如，fp32 INF 或 NaN 係轉換成 fp64 INF 或 NaN。所有 fp32 反向規格化可轉換成 fp64 正常數字。

運算元準備區塊 514(第六圖)係接收 fp32 運算元 A。絕對值/負電路 618 可使無需修改的運算元 A 通過至

fp32 擷取區塊 626。fp32 擷取區塊 626 係執行運算元 A 至 fp64 格式的初始向上轉換。明確地係，fp32 擷取區塊 626 係擷取 8 位元指數，並加入 1023-127=896，以產生具有 fp64 格式的正確偏移之 11 位元指數。23 位元尾數係使用尾零填補。fp32 擷取區塊 626 亦決定運算元 A 是否為一 fp32 特別數字(例如，INF、NaN、零或反向規格化)，並經由路徑 644 提供該資訊至特別數字偵測電路 642。fp32 擷取區塊 626 亦可否定或應用絕對值至運算元。

運算元 C 多工器 636 係選擇由 fp32 擷取區塊 626 提供的向上轉換運算元；運算元 A 多工器 632 與運算元 B 多工器 634 係選擇零運算元。除非運算元是一 fp32 反向規格化，否則特別數字偵測電路 642 可預先考慮尾數的一前導 1。除了 fp32 反向規格化係視為正常數字之外，特別數字偵測電路 642 亦使用由 fp32 擷取區塊 626 提供的特別數字資訊，當作是其特別數字 SPC 信號(因為所有 fp32 反向規格化能夠以 fp64 的正常數字予以表示)。

尾數路徑 516 與指數路徑 518 的操作係如上述以 fp64 格式計算 0.0*0.0+C 的 DFMA 運算。尾數路徑 516 與指數路徑 518 中的正規化元件係使向上轉換的 fp64 運算元正規化。在一替代具體實施例中，請即參考第八圖，來自指數路徑 518 的對齊尾數 C_align 可在尾數路徑 516 的加法器 804 附近旁路至多工器 806 的 Sum 輸入；乘法器 802 與加法器 804 可置於一低電力狀態。符號路徑 520 可有利地使符號位元 Cs 通過。

在輸出區段 522(第十圖)中，除非特別數字 SPC 信號指示輸入運算元是 fp32 INF、NaN 或零，否則選擇正規化的 fp64 結果(Rm、Rs、Re)。若輸入運算元是 fp32

INF，輸出多工器控制邏輯 1002 操作有效數字多工器 1004 選擇 fp64 INF 有效數字(0x0_0000_0000_0000)，及操作指數多工器 1006 選擇 fp64 INF 指數(0x7ff)。若輸入運算元是 fp32NaN，輸出多工器控制邏輯 1002 操作有效數字多工器 1004 以選擇 fp64 未明 NaN 有效數字，及操作指數多工器 1006 以選擇 fp64NaN 指數(Ox7ff)。若輸入運算元是 fp32 零，輸出多工器控制邏輯 1002 操作有效數字多工器 1004 選擇 fp64 零有效數字(0x0_0000_0000_0000)，及操作指數多工器 1006 以選擇 fp64 零指數(0x000)。條件碼可依需要予以設定。

2. 整數至 fp64 (I2D)

對於 I2D 運算而言，一整數(u64、s64、u32 或 s32 格式)係轉換成 fp64 格式。運算元準備區塊 514(第六圖)係接收 64 位元整數運算元 C。對於 32 位元整數格式而言，32 個前導零可預先考慮。絕對值/負電路 622 可使無需修改的運算元 C 通過至 U/S 擷取區塊 628。U/S 擷取區塊 628 係執行運算元 C 至 fp64 格式的初始轉換。明確地係，擷取區塊 628 係決定運算元 C(例如，使用一優先權編碼器)中前導 1 的位置。一 11 位元指數係藉由將一指數欄位初始化成 1086(對應至 2^{63})而決定。對於 32 位元輸入格式而言，丟棄前導 1，並使用尾零填補尾數，以產生 52 位元有效數字。對於 64 位元輸入格式而言，若需要 53 位元，將尾數截掉，並丟棄前導 1。若需要，可保留一保護位元與捨入位元。

U/S 擷取區塊 628 亦決定輸入運算元是否為零，並為特別數字偵測電路 642 產生一對應的控制信號。其他特別數字(反向規格化、INF 和 NaN)不會在 I2D 運算期間發生，且不需要偵測。

運算元C多工器636係選擇藉由U/S擷取區塊628提供的向上轉換之運算元；運算元A多工器632與運算元B多工器634之每一者係選擇零運算元。特別數字偵測電路642係使用藉由U/S擷取區塊628提供的零資訊產生一特別數字SPC信號，以指出輸入運算元是否為零。

尾數路徑516與指數路徑518的操作係如同上述關於計算0.0*0.0+C的DFMA運算。尾數路徑516與指數路徑518中的正規化元件係使向上轉換的fp64運算元予以正規化。在一替代具體實施例中，來自指數路徑518的對齊尾數C_align可在尾數路徑516的加法器804附近旁路至多工器806的Sum輸入；乘法器802與加法器804可置於一低電力狀態。符號路徑520可有利地使符號位元Cs通過。

在輸出區段522(第十圖)中，除非特別數字SPC信號指示輸入運算元是整數零，否則選擇正規化的fp64結果(Rm、Rs、Re)。若輸入運算元是整數零，輸出多工器控制邏輯1002則操作有效數字多工器1004以選擇fp64零有效數字(0x0_0000_0000_0000)，及操作指數多工器1006以選擇fp64零指數(0x000)。條件碼可依需要加以設定。

3.fp64至fp32 (D2F)

由於fp64比fp32涵蓋更大範圍的浮點值，所以從fp64轉換至fp32(D2F)需要偵測在fp32值中的上限溢位與下限溢位。

對於D2F運算而言，運算元C係提供給運算元準備區塊514(第六圖)。絕對值/負電路622可依需要執行絕對值或運算元負，並將運算元C傳遞至運算元C多工器

636，選擇運算元 C 以提供給特別數字偵測電路 642。特別數字偵測電路 642 可偵測 fp64 反向規格化、零、INF 或 NaN，並提供對應的 SPC 信號至輸出區段 522。選擇多工器 632 和 634 可選擇 0.0 運算元。

在指數路徑 518(第七圖)中，指數計算區塊 702 是以 897 將 fp64 指數向下偏移，以決定一對應的 fp32 指數。若 fp32 指數係下限溢位，指數計算區塊 702 產生一 Sh_C 信號，此將使 C 的尾數向右移位以除去下限溢位(若需要 217 位元以上的移位，C 的尾數將變成零)。移位電路 706 將根據 Sh_C 信號使 C 的尾數向右移位。結果將由多工器 712 選擇，並以對齊的尾數 C_align 提供給尾數路徑 516。下限溢位邏輯 713 可偵測一 fp32 下限溢位及產生 U_fp32 信號。

在尾數路徑 516(第八圖)中，乘法器 802 係計算乘積 0.0*0.0(或旁路過)。乘積(零)是由加法器 804 加至尾數 C_align。加總結果是由多工器 806 選擇(由於輸入是符號/數值形狀)。電路 814 可偵測零結果；非零結果的正規化是如上面 DFMA 運算情況的描述。捨入邏輯 820 可用來決定是否要捨入；應該注意，加 1 的加法器 822 需要將 1 加至第 24 位元位置(而不是第 53 位元位置)，結果會是一 23 位元 fp32 尾數。

輸出區段 522(第十圖)係組合該結果。23 位元 fp32 有效數字是在 52 位元欄位 Rm 中提供。除非結果不是一 fp32 正常數字，否則輸出多工器控制邏輯 1002 可控制有效數字多工器 1004 以選擇 Rm。對於 fp32 零或 INF 而言，選擇零的尾數 0x0_0000_0000_0000；對於 fp32 NaN 而言，選擇未明 fp32NaN 尾數。對於 fp32 反向規格化而言，可使用 Rm。

8 位元 fp32 指數是在一 11 位元指數欄位中提供。

除非結果不是 fp32 正常數字，否則輸出多工器控制邏輯 1002 可控制指數多工器 1004 以選擇 Re。對於 fp32 反向規格化或零而言，選擇零指數 0x 000。對於 fp32 INF 或 NaN 而言，則選擇最大 fp32 指數 0x7ff。

串接區塊 1008 係將 Rm 和 Re 裝入 64 位元輸出欄位的 31 位元，並附加符號位元 Rs。11 位元指數中的 3 MSB 會被丟棄，如同在 52 位元有效數字中的 29 LSB。fp32 結果可依需要例如在 64 位元欄位的 MSB 或 LSB 中對齊。條件碼可依需要予以設定。

4.fp64 至整數(D2I)

對於 D2I 運算而言，可偵測上限溢位與下限溢位。上限溢位係設定成最大整數值，且下限溢位係設定成零。

待轉換的運算元被提供為 fp64 格式的運算元 C。絕對值/負電路 622 可依需要執行絕對值或運算元負，並將運算元 C 傳遞給運算元 C 多工器 636，選擇運算元 C 以提供給特別數字偵測電路 642。特別數字偵測電路 642 可偵測 fp64 反向規格化、零、INF 或 NaN，並提供對應的 SPC 信號至輸出區段 522。選擇多工器 632 和 634 選擇 0.0 運算元。

在指數路徑 518(第七圖)中，指數計算區塊 702 係使用指數 Ce 以決定 Cm 的移位量，以對齊在整數位置的二進位點，並產生一對應的 Sh_C 信號。在一具體實施例中，指數計算區塊 702 係移除指數偏差，並說明有效數字的寬度、要使用的整數格式、與 32 位元格式的結果如何在 64 位元欄位中表示(例如，使用 32 MSB 或 32 LSB)。指數 Ce 亦用來決定結果是否為目標整數格式的上限溢位或下限溢位；若確認，一對應的上限溢位或下

限溢位信號(未在圖明確顯示)將有利傳送至輸出區段522(第十圖)中的輸出多工器控制邏輯1002。

移位電路706可依C_Shift量將Cm予以移位，且C_Shift信號係由多工器712選擇為C_align信號。

在尾數路徑516(第八圖)中，乘法器802提供0.0結果至加法器804。加法器804將0.0加入C_align，並選擇Sum或~Sum，此係取決於C是否為正或負值。移位器816可有利地不將結果移位。整數格式區塊826可將該結果分成一11位元MSB欄位int_M、與一53位元LSB欄位int_L。

在輸出區段522(第十圖)中，除了在一上限溢位、下限溢位或特別數字運算元的情況之外，輸出多工器控制邏輯1002可控制有效數字多工器1004與指數多工器1006，以分別選擇int_L和int_M結果。針對上限溢位，可選擇輸出格式(u32、s32、u64或s64)中的最大整數；針對下限溢位，可選擇零。條件碼可視需要加以設定。

4. 進一步具體實施例

雖然本發明已描述關於特定具體實施例，但是熟諳此技術人士應明白許多修改是可能的。例如，一DFMA單元可實施，以支援更多、較少或不同的功能組合，並支援任何格式或更多格式組合的運算元與結果。

在此描述的各種旁路與通過路徑亦可改變。大體上，在描述任何電路區塊周圍的一旁路情況，該路徑可由該區塊中的一恆等操作(即是，對其運算元沒有影響的操作，例如加入零)所取代。在一給定操作期間省略的一電路區塊可置於一閒置狀態(例如，一減少電量狀態)或通常係以向下資料流區塊所忽略的結果進行操作，例如，透過選擇多工器或其他電路的操作。

DFMA 管線可分成任何數量的階段，且每一階段的組件組合可視需要而予以變化。認定為在此特定電路區塊的功能亦可在管線級上分開；例如，一乘法器樹狀可占用多個管線級。不同區塊的功能亦可修改。在一些具體實施例，例如可使用不同加法器電路或乘法器電路。

此外，DFMA 單元已促進瞭解的電路區塊描述；熟諳此項技術人士可確認，該等區塊可使用各種電路組件與佈局實施，且在此描述的區塊並未限於一組特定組件或實體佈局。可視需要將區塊實際組合或分開。

一處理器可包括一執行核心中的一或多個 DFMA 單元。例如，在超純量指令發送(即是，每週期發送一個以上指令)或 SIMD 指令發送想要的情況，可實施多個 DFMA 單元，且不同 DFMA 單元可支援功能的不同組合。一處理器亦可包括多個執行核心，且每一核心可擁有自己的 DFMA 單元。

在執行核心支援 SIMD 指令發送的一些具體實施例中，單一 DFMA 單元可與適當的輸入序列與輸出收集邏輯結合使用，使多個資料集可在單一 DFMA 管線中循序處理。

第十一圖為根據本發明之一具體實施例的包括一 DFMA 功能單元 1102 的執行核心 1100 之區塊圖。DFMA 單元 1102 可類似或相似於上述的 DFMA 單元 320。核心 1100 係發送 SIMD 指令，意謂具 P 不同組單精確度運算元的相同指令可同時發送至一組 P 單精確度 SIMD 單元 1104。每一 SIMD 單元 1104 係接收相同運算碼與一不同組的運算元；P 個 SIMD 單元 1104 係同時操作以產生 P 個結果。P 通路 SIMD 指令是以一連串的 P 單指令、單資料(SISD, “Single-Instruction, Single Data”)指令發送至 DFMA 單元 1102。

一輸入管理器 1106(其可為一指令發送單元的一部分)係收集 SIMD 指令的運算元，且當已收集 SIMD 指令的所有 P 組運算元時，傳遞運算元與適用的運算碼至 P SIMD 單元 1104 或 DFMA 單元 1102 任一。一輸出收集器 1008 可從 SIMD 單元 1104 或 DFMA 單元 1102 收集結果，並經由結果匯流排 1110，將結果傳遞至一暫存器檔案(未在第十一圖中明確顯示)。在一些具體實施例中，結果匯流排 1110 亦提供一旁路至輸入管理器 1106，以便結果傳遞至輸入管理器 1106，供與一隨後的指令一起使用，同時傳遞至暫存器檔案。為了要提供使用一 DFMA 單元 1102 的 SIMD 行為之外觀，輸入管理器 1106 可有利地將指令序列發送至 DFMA 單元 1102，例如，藉由在 P 連續時脈週期之每一者上，發送相同運算碼與不同組的運算元。

第十二圖為根據本發明之一具體實施例顯示 DFMA 單元 1102 的序列指令發送之區塊圖。一輸入運算元收集單元 1202(可包括在第十一圖的輸入管理器 1106)包括兩收集器 1204、1206。每一收集器 1204、1206 是 32 位元暫存器的配置，可提供 P 個三個一組單精確度運算元 A、B 和 C 的足夠空間；換句話說，每一收集器 1204、1206 可儲存單一 SIMD 指令的所有運算元。輸入運算元收集單元 1202 係例如從第三圖的暫存器檔案 324、及/或從第十一圖的結果匯流排 1110 獲得運算元；標籤或其他習知技術可用來決定哪些運算元可被收集用於一給定的指令。足夠的收集器 1206 可提供，以在發送指令之前，允許在數個時脈週期內收集一給定指令的運算元。

對於單精確度指令而言，一收集器(例如，收集器 1204)載入 P SIMD 單元 1104 需要的所有運算元以執行

一指令。當指令發送給 P SIMD 單元 1104 時，整個收集器 1204 可有利地同時讀取，且不同 A、B、C 運算元三個一組係傳遞至 SIMD 單元 1104 之每一者。

對於 DFMA 單元 1102 的指令而言，運算元是倍精確度(例如，64 位元)。可使用兩收集器 1204、1206 中的對應暫存器儲存每一運算元；例如，收集器 1204 中的暫存器 1208 可儲存運算元 A 的一實例的 32MSB(例如，符號位元、11 個指數位元、與有效數字的 20MSB)，而收集器 1206 中的暫存器 1210 可儲存相同運算元的 32LSB(例如，有效數字的剩餘 32 位元)。如此可使用兩單精確度收集器 1204、1206 收集 P 通道倍精確度 SIMD 指令所需的所有運算元三個一組 A、B、C。

核心 1100 只有一 DFMA 單元 1102，且 P 運算元集係有利地使用輸出多工器(mux)1212、1214 循序傳遞，其兩者是受到一計數器 1216 的控制。多工器 1212 和 1214 係回應計數器 1216 以從個別的收集器 1204 和 1206 選擇運算元三個一組的 MSB 和 LSB。例如，在顯示的資料路徑中，多工器 1212 可從收集器 1204 中的暫存器 1208 選擇運算元 A 的 32 MSB，而多工器 1214 是從收集器 1206 中的暫存器 1210 選擇相同運算元 A 的 32 LSB。64 位元是在一個倍精確度寬路徑上傳遞至 DFMA 單元 1102。同樣地，使用相同計數器 1216 控制的對應多工器(未在圖明確顯示)，運算元 B(從暫存器 1220 和 1222)和 C(從暫存器 1224 和 1226)可傳遞至 DFMA 單元 1102。在下一時脈週期上，在傳遞所有 P 組運算元之前，來自收集器 1204 和 1206 中的下一組暫存器的運算元 A、B、和 C 可傳遞至 DFMA 單元 1102 等。

多工器 1212 和 1214、連同收集器 1204 和 1206 係一起提供 DFMA 單元 1102 的 SIMD 執行的外觀，雖然

此減少了傳送量。因此，核心 1100 的編程模型可假設 P 通道 SIMD 執行可用於所有指令，包括倍精確度指令。

應該明白，在此描述的運算元集合及序列邏輯只是說明，且不同變化及修改是可能的。在一 SIMD 能力核心中，任何數量的 DFMA 單元可被提供，且指令可同時發送至任何數量的 DFMA 單元。在一些具體實施例中，有關單精確度運算的倍精確度運算之傳送量是與 DFMA 單元的數目成比例。例如，若存在 P 個 SIMD 單元與 N 個 DFMA 單元，倍精確度傳送量將會是單精確度傳送量的 N/P。在一些具體實施例中，N 最好是等於 P；在其他具體實施例中，其他因數(例如，在暫存器檔案與功能單元之間的內部資料路徑的寬度)可將倍精確度傳送量限制在少於單精確度傳送量，不管提供的 DFMA 單元的數目。在此情況中，N 最好是不大於允許的其他限制因素。

亦應該注意，由於 DFMA 單元是與單精確度功能單元分開，所以當不使用時，此單元可關閉電源，例如，當繪圖處理器或核心專門用於繪圖處理器或不需要倍精確度的其他計算。此外，DFMA 單元可從積體電路設計移除，不會影響其他電路組件的操作。此可促進不同晶片提供不同程度倍精確度運算支援的乘積系列設計。例如，一 GPU 系列可包括一具許多核心的高端 GPU，該等核心之每一者包括至少一 DFMA 單元；及一不具有硬體式倍精確度支援及沒有 DFMA 單元的低端 GPU。

此外，雖然本發明已描述關於一繪圖處理器，但是熟諳此技術人士應該明白，本發明的態樣亦可使用在其他處理器，例如算術共處理器、向量處理器或一般目的處理器。

因此，雖然本發明已描述關於特殊具體實施例，但是應可明白本發明意欲涵蓋下列申請專利範圍的範疇內之所有修改及等效物。

【圖式簡單說明】

第一圖為根據本發明之一具體實施例的一電腦系統之區塊圖；

第二圖為根據本發明之一具體實施例，可在一繪圖處理單元中實施的一繪圖管線之區塊圖；

第三圖為根據本發明之一具體實施例的一執行核心之區塊圖；

第四圖列出根據本發明之一具體實施例，一個倍精確度功能單元可執行的倍精確度算術、比較運算與格式轉換操作；

第五圖為根據本發明之一具體實施例的一個倍精確度功能單元之簡化區塊圖；

第六圖為第五圖的倍精確度功能單元的一運算元預備區塊之區塊圖；

第七圖為第五圖的倍精確度功能單元的一指數路徑之區塊圖；

第八圖為第五圖的倍精確度功能單元的一尾數路徑之區塊圖；

第九圖為第五圖的倍精確度功能單元的一信號路徑之區塊圖；

第十圖為第五圖的倍精確度功能單元的一輸出區段之區塊圖；

第十一圖為根據本發明之一具體實施例的一執行核心之區塊圖；及

第十二圖為顯示根據本發明之一具體實施例的一

倍精確度功能單元的運算元序列之區塊圖。

【主要元件符號說明】

100 電腦系統
102 中央處理單元
104 系統記憶體
105 記憶體橋接
106 匯流排或通訊路徑
107 輸入/輸出橋接
108 使用者輸入裝置
110 顯示裝置
112 繪圖子系統
113 匯流排或通訊路徑
114 系統磁碟
116 開關
118 網路轉接器
120 附加介面卡
121 附加介面卡
122 繪圖處理單元(GPU)
124 繪圖記憶體
200 繪圖管線
202 多執行緒核心陣列
204 前端
206 資料組合器
208 設定模組
210 光柵化器
212 彩色組合模組
214 光柵操作模組(ROP)
218 幾何模組
224 像素模組
226 訊框緩衝器
300 執行核心
302 拾取與分發單元
304 發送單元
320 倍精確度融合乘加(DFMA)單元
322 功能單元
324 暫存器檔案
326 資料傳輸路徑
502 運算元輸入路徑
504 運算元輸入路徑
506 運算元輸入路徑
508 運算碼路徑
510 信號路徑
512 信號路徑
514 運算元準備區塊
516 尾數路徑
518 指數路徑
520 符號路徑
522 輸出區段
524 信號路徑
530 控制區塊
612 NaN 偵測區塊
614 NaN 偵測區塊
616 NaN 偵測區塊

618 絕對值/負區塊
620 絕對值/負區塊
622 絕對值/負區塊
624 比較電路
626 擷取電路
628 無符號/有符號(U/S)擷取電路
630 控制邏輯
632 運算元選擇多工器
634 運算元選擇多工器
636 運算元選擇多工器
638 特別數字偵測電路
640 特別數字偵測電路
642 特別數字偵測電路
702 指數計算電路
704 向下資料流最後指數計算電路
706 移位電路
708 負電路
710 邏輯電路
712 選擇多工器
713 下限溢位邏輯
720 減算電路
722 下限溢位/上限溢位電路
724 加 1 電路
726 選擇多工器
802 乘法器
804 加法器
806 多工器
808 LZD 電路
810 LZD 電路
812 LZD 多工器
814 零偵測電路
816 移位電路
818 正規化邏輯
820 捨入邏輯
821 路徑
822 加 1 加法器
824 尾數選擇多工器
826 整數格式區塊
902 乘積/加總電路
904 最後符號電路
1002 輸出多工器控制邏輯
1004 有效數字選擇多工器
1006 指數選擇多工器
1008 串接區塊
1100 執行核心
1102 DFMA 功能單元
1104 單精確度 SIMD 單元
1106 輸入管理器
1108 輸出收集器
1110 結果匯流排
1202 輸入運算元收集單元
1204 收集器
1206 收集器
1208 暫存器
1210 暫存器

1212 輸出多工器
1214 輸出多工器
1216 計數器
1220 暫存器
1222 暫存器
1224 暫存器
1226 暫存器

101年8月29日修正替換頁
61～64頁

七、申請專利範圍：

1. 一種繪圖處理器，包含：

一繪圖管線，其調適成產生影像資料，該繪圖管線包括一處理核心，其調適成可執行複數個同時發生的執行緒，其中該繪圖管線是在單精確度運算元上運算；

該處理核心更包括一多用途倍精確度功能單元，其調適成選擇性執行一組倍精確度輸入運算元的複數個倍精確度運算之一，該多用途倍精確度功能單元包括至少一算術邏輯電路，該些倍精確度運算包括一將兩個倍精確度運算元相加的加法運算、一將兩個倍精確度運算元相乘的乘法運算、和一融合乘加法運算，該融合乘加法運算計算一第一倍精確度運算元和一第二倍精確度運算元的乘積，再將一第三倍精確度運算元加上該乘積，其中該多用途倍精確度融合乘加功能單元有足夠寬度以在一單一通過中執行該些倍精確度運算的每一個，使得該些倍精確度運算的每一個在相同時脈週期數目中完成；

其中該倍精確度功能單元的所有算術邏輯電路有足夠寬度以倍精確度運算。

2. 如申請專利範圍第1項之繪圖處理器，其中該倍精確度功能單元係進一步調適，使得該複數個倍精確度運算之每一者可在相同時脈週期數目中完成，不管上限溢位或下限溢位條件是否發生。

3. 如申請專利範圍第2項之繪圖處理器，其中該倍精確度功能單元係進一步調適成於上限溢位或下限溢位條件確實發生情況中，產生一上限溢位或下限溢位結果，其遵循一浮點算術標準；及設定一輸出狀態旗標，表示上限溢位或下限溢位條件是否發生。

4. 如申請專利範圍第1項之繪圖處理器，其中該倍精確

度功能單元係進一步調適，使得完成該複數個倍精確度運算之任一者所需的時間不受到一浮點異常的影響。

5. 如申請專利範圍第 1 項之繪圖處理器，其中該複數個倍精確度運算更包括一倍精確度比較(DSET)操作，其執行一第一運算元與一第二運算元的比較測試，及產生布林結果，表示是否滿足該比較測試。

6. 如申請專利範圍第 1 項之繪圖處理器，其中該複數個倍精確度運算更包括：

一個倍精確度最大(DMAX)操作，其傳回兩個倍精確度輸入運算元之較大一者；及

一個倍精確度最小(DMIN)操作，其傳回兩個倍精確度輸入運算元之較小一者。

7. 如申請專利範圍第 1 項之繪圖處理器，其中該複數個倍精確度運算更包括至少一格式轉換操作，將一運算元從一個倍精確度格式轉換成一非倍精確度格式。

8. 如申請專利範圍第 1 項之繪圖處理器，其中該複數個倍精確度運算更包括至少一格式轉換操作，將一運算元從一非倍精確度格式轉換成一倍精確度格式。

9. 一種繪圖處理器，包含：

一繪圖管線，其調適成產生影像資料，該繪圖管線包括一處理核心，其調適成執行複數個同時發生的執行緒；

該處理核心包括一單精確度功能單元，其調適成執行一或多個單精確度運算元的算術運算；

該處理核心更包括一個倍精確度融合乘加(DFMA)功能單元，其調適成執行一組倍精確度輸入運算元的融合乘加運算，並提供一個倍精確度結果，並對一對的倍精確度輸入運算元執行一加法運算且提供一倍精確度結果，並對一對的倍精確度輸入運算元執行一乘法運算且提供一倍精確度結果；

其中該 DFMA 功能單元包括一 DFMA 管線，其具有足夠資料路徑寬度，透過該 DFMA 管線，在單一通過中執行該融合乘加運算、該加法運算或該乘法運算，使得該融合乘加運算、該加法運算和該乘法運算之每一個均在相同時脈週期數目中完成。

10. 如申請專利範圍第 9 項之繪圖處理器，其中該 DFMA 功能單元包括：

一乘法器，其調適成在單循環中計算兩個倍精確度尾數的乘積；及

一加法器，其調適成在單循環中計算兩個倍精確度尾數的加總。

11. 如申請專利範圍第 9 項之繪圖處理器，其中：

該融合乘加運算、該加法運算、與該乘法運算均在相同時脈週期數目內完成，不管一上限溢位或下限溢位條件是否發生。

12. 如申請專利範圍第 11 項之繪圖處理器，其中該 DFMA 功能單元進一步組態成於上限溢位或下限溢位條件發生情況中，產生一上限溢位或下限溢位結果，其遵循一浮點算術標準；及設定一輸出狀態旗標，表示上限溢位或下限溢位條件是否發生。

13. 如申請專利範圍第 9 項之繪圖處理器，其中該處理核心包括一單精確度功能單元的副本數目(P)，其調適成平行操作；及 DFMA 功能單元的副本數目(N)。

14. 如申請專利範圍第 13 項之繪圖處理器，其中該數目 P 係大於數目 N。

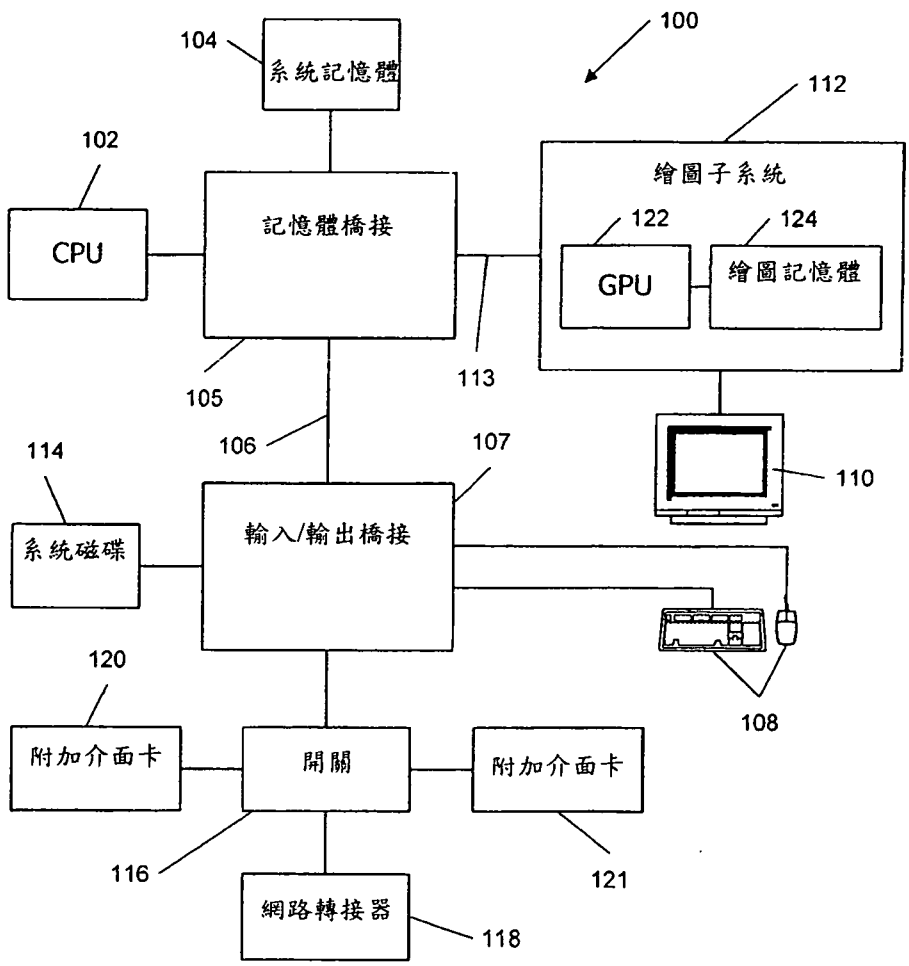
15. 如申請專利範圍第 14 項之繪圖處理器，其中該數目 N 是 1。

16. 如申請專利範圍第 15 項之繪圖處理器，其中該處理核心更包括一輸入管理器電路，其調適成收集該 DFMA 功能單元的 N 組倍精確度輸入運算元，並在不同時脈週

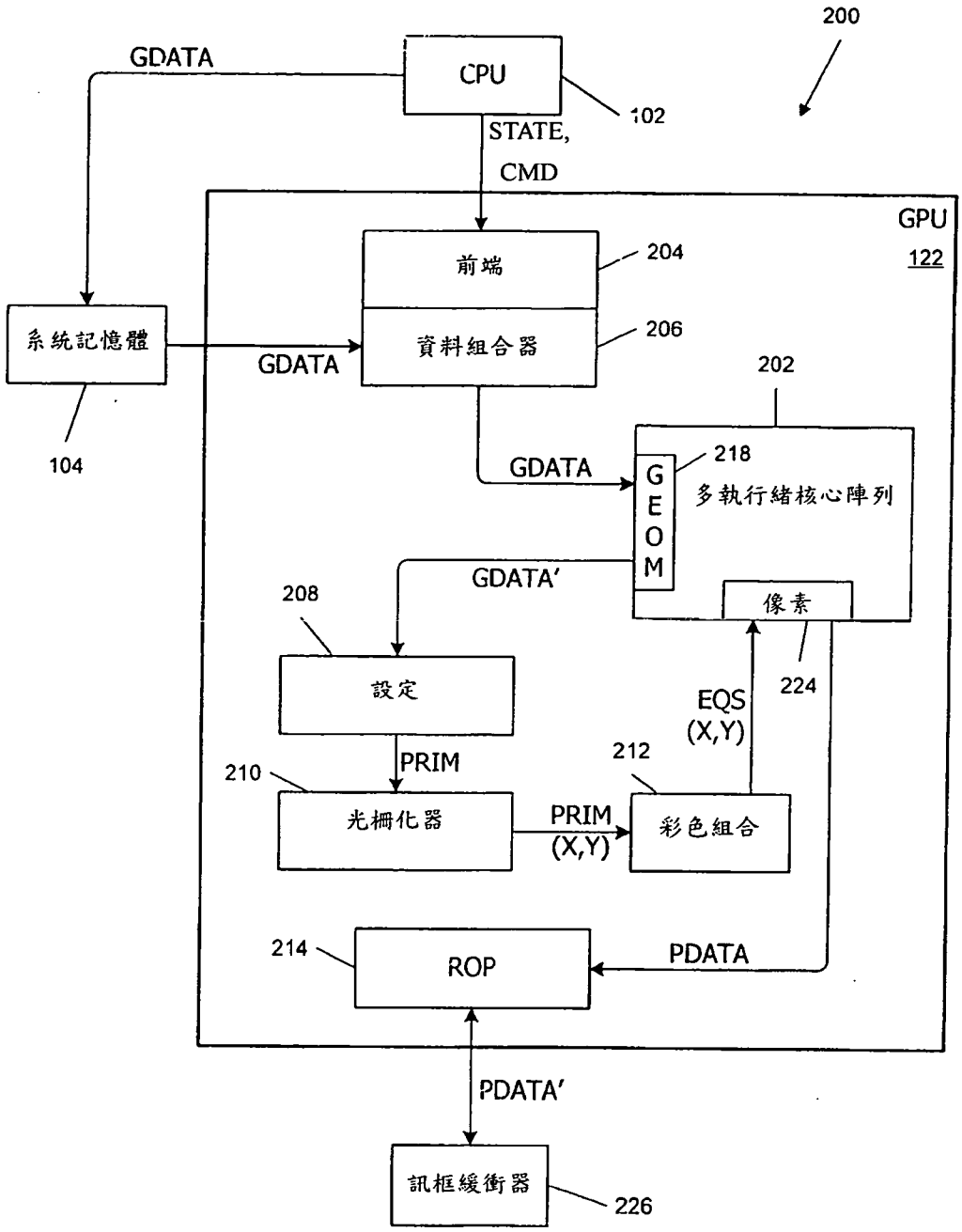
期上，傳遞該 *N* 組倍精確度運算元之不同一些者至 DFMA 功能單元。

17. 如申請專利範圍第 16 項之繪圖處理器，其中該輸入管理器電路進一步調適成收集該單精確度功能單元的 P 組單精確度輸入運算元，並平行傳遞該 P 組單精確度運算元之不同一者至該單精確度功能單元的 P 副本之每一者。

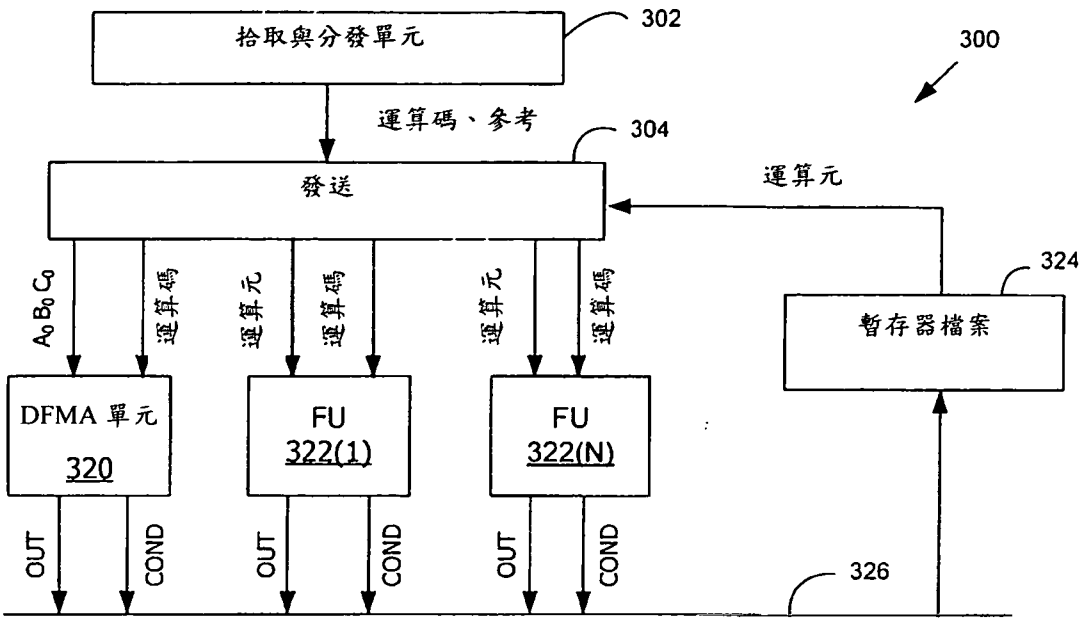
八、圖式：


104
系統記憶體
100
112
102
繪圖子系統
CPU
記憶體橋接
122
124
GPU
繪圖記憶體
105
113
106
107
110
114
系統磁碟
輸入/輸出橋接
108
120
附加介面卡
開關
附加介面卡
116
121
網路轉接器
118

第一圖


200
GDATA
CPU
102
STATE,
CMD
GPU
122
前端
204
系統記憶體
資料組合器
206
GDATA
202
104
GDATA
218
G
E
O
M
多執行緒核心陣列
GDATA'
208
像素
224
設定
EQS
(X,Y)
PRIM
212
210
光柵化器
PRIM
(X,Y)
彩色組合
214
ROP
PDATA
PDATA'
訊框緩衝器
226

第二圖


拾取與分發單元
302
300
運算碼、參考
304
發送
運算元
324
暫存器檔案
$A_0 B_0 C_0$
運算碼
運算元
運算碼
運算元
運算碼
DFMA 單元
320
FU
322(1)
FU
322(N)
OUT
COND
OUT
COND
OUT
COND
326

第三圖

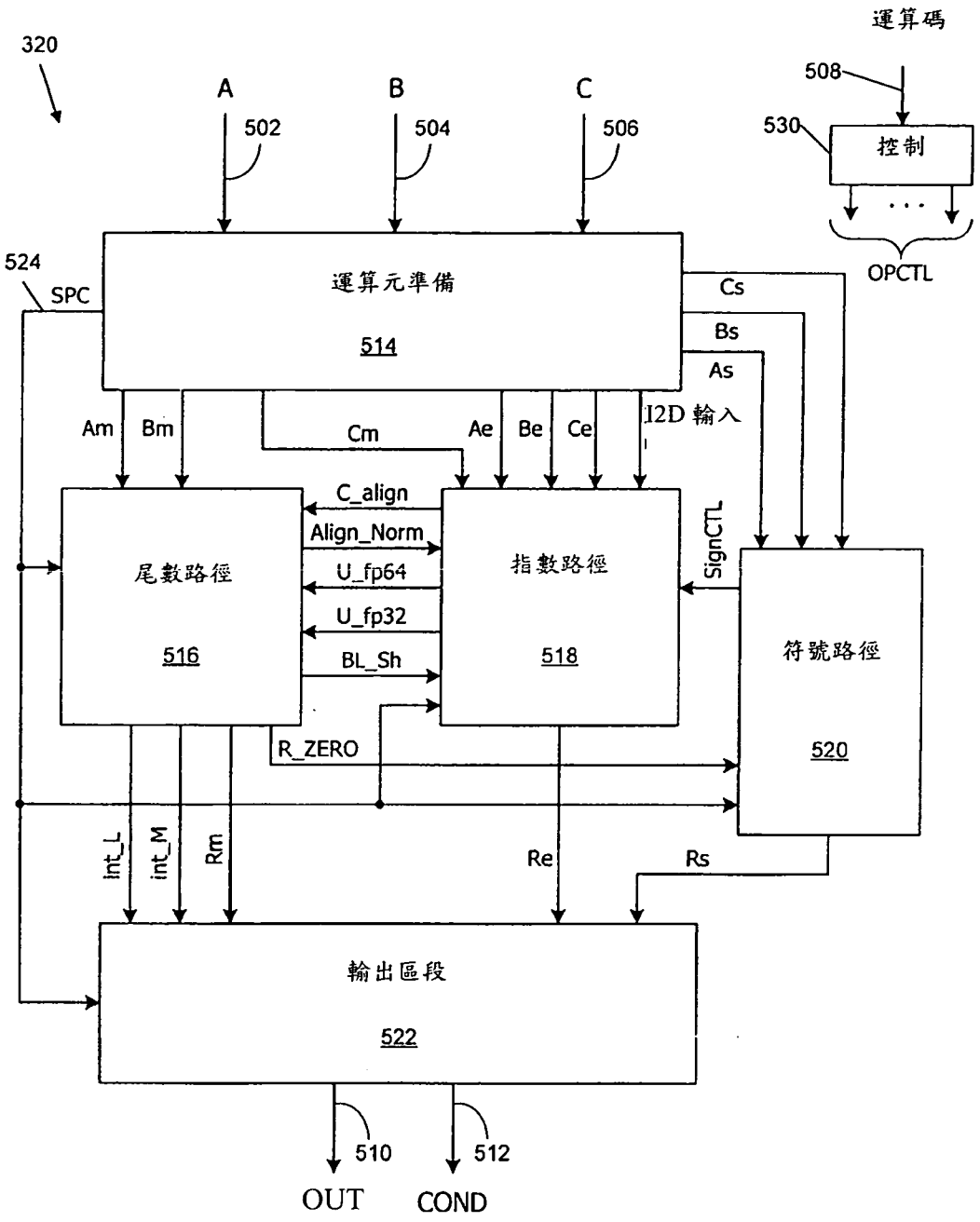
400

倍精確度算術運算 402		
名稱	輸入	結果
DADD	A,C: fp64	A+C
DMUL	A,B: fp64	A*B
DFMA	A,B,C: fp64	A*B+C: fp64

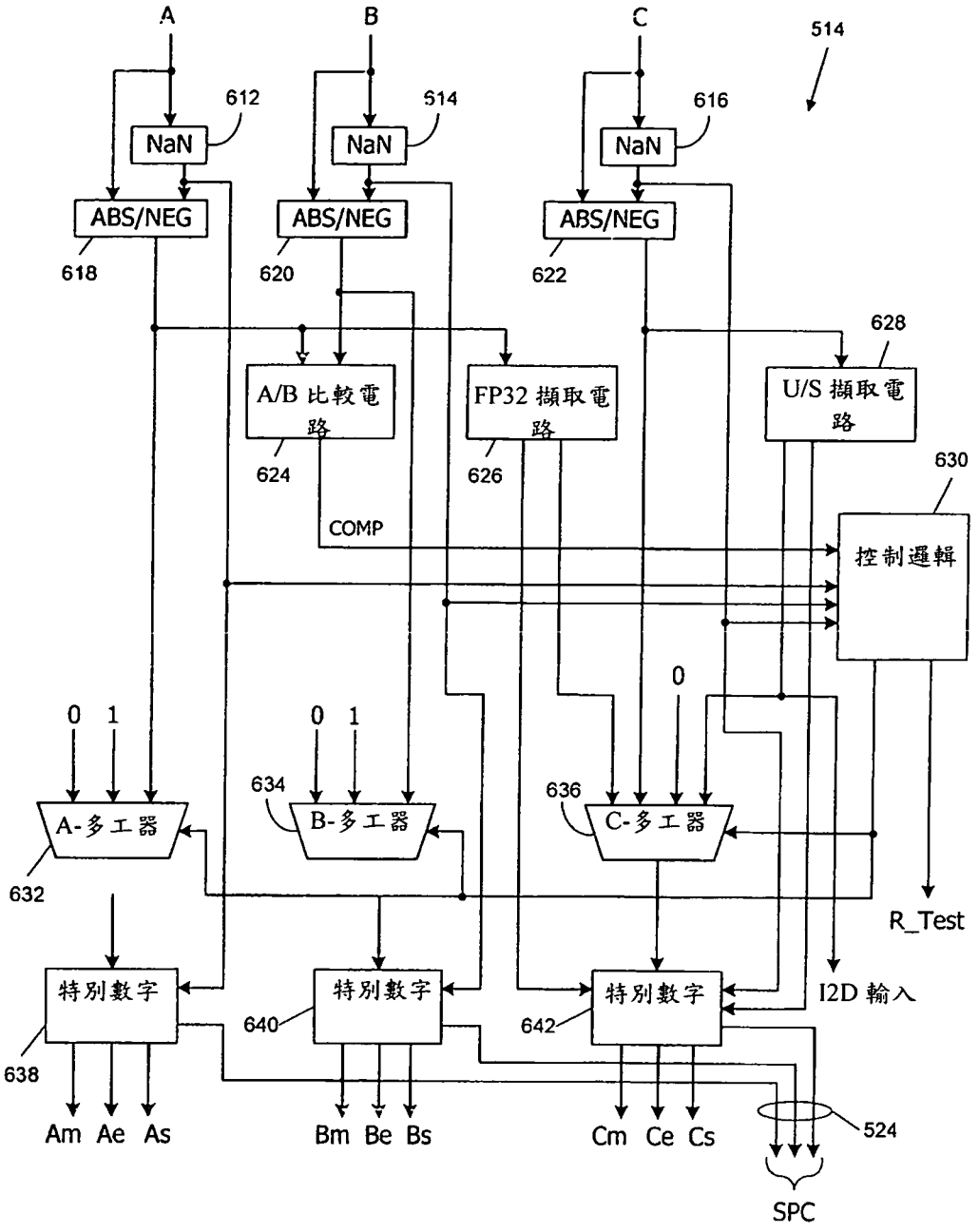
倍精確度比較 404		
名稱	輸入	結果
DMAX	A,B: fp64	max(A,B)
DMIN	A,B: fp64	min(A,B)
DSET	A,B: fp64	R: boolean

格式轉換與捨入 406		
名稱	輸入	結果
D2F	A: fp64	A': fp32
F2D	A: fp32	A': fp64
D2I	A: fp64	A': u/s64 or u/s32
I2D	C: u/s64 or u/s32	C': fp64
D2D	A: fp64	A': fp64

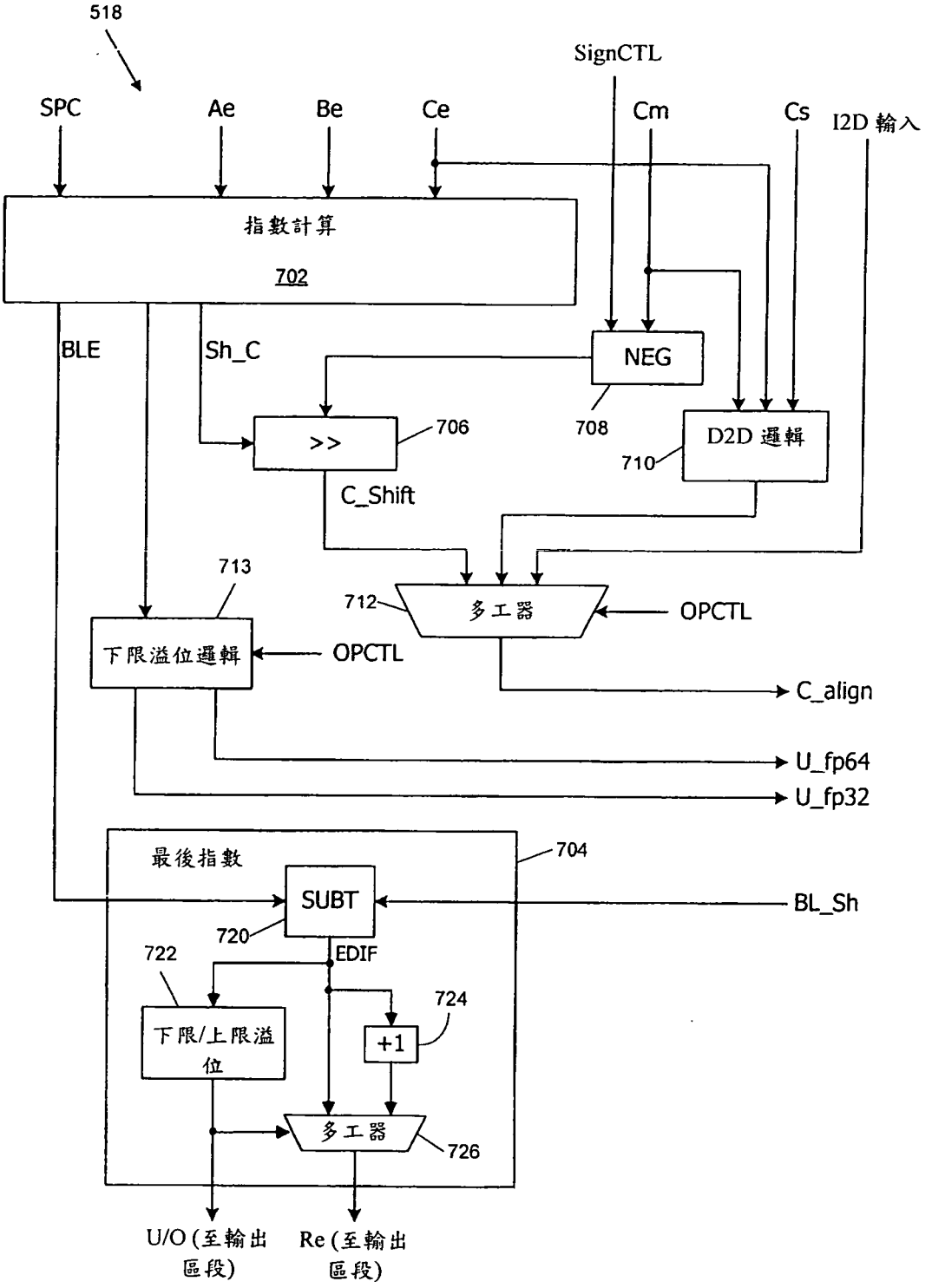
第四圖


320
A
B
C
502
504
506
運算碼
508
530
控制
OPCTL
524
SPC
運算元準備
514
Cs
Bs
As
I2D 輸入
Am
Bm
Cm
Ae
Be
Ce
C_align
Align_Norm
U_fp64
U_fp32
BL_Sh
尾數路徑
516
指數路徑
518
SignCTL
符號路徑
520
R_ZERO
int_L
int_M
Rm
Re
Rs
輸出區段
522
510
512
OUT
COND

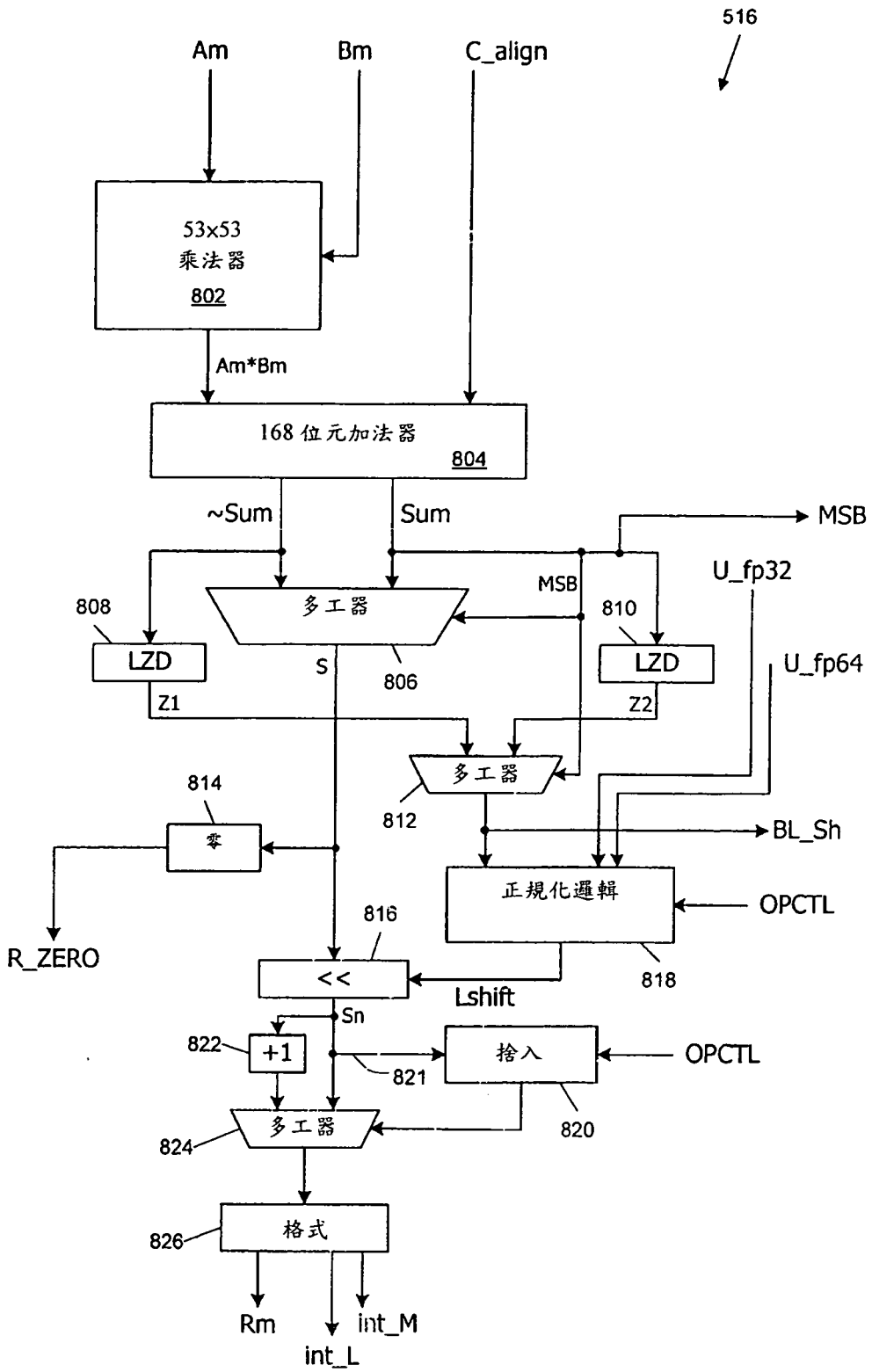
第五圖


A
B
C
514
612
614
616
NaN
NaN
NaN
ABS/NEG
ABS/NEG
ABS/NEG
618
620
622
628
A/B 比較電路
FP32 擷取電路
U/S 擷取電路
624
626
630
COMP
控制邏輯
0 1
0 1
0
634
636
A-多工器
B-多工器
C-多工器
632
R_Test
特別數字
特別數字
特別數字
I2D 輸入
640
642
638
Am Ae As
Bm Be Bs
Cm Ce Cs
524
SPC

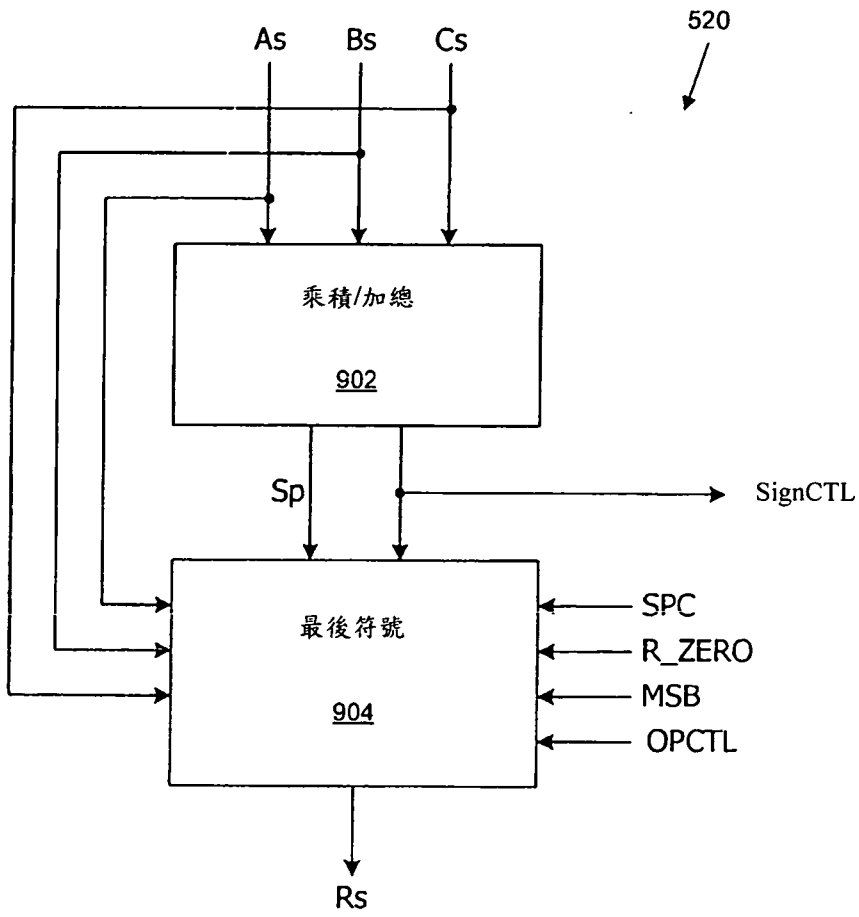
第六圖


518
SPC
Ae
Be
Ce
SignCTL
Cm
Cs
I2D 輸入
指數計算
702
BLE
Sh_C
NEG
708
>>
706
C_Shift
D2D 邏輯
710
713
712
多工器
OPCTL
下限溢位邏輯
OPCTL
C_align
U_fp64
U_fp32
最後指數
704
SUBT
BL_Sh
720
EDIF
722
724
下限/上限溢位
+1
多工器
726
U/O (至輸出區段)
Re (至輸出區段)

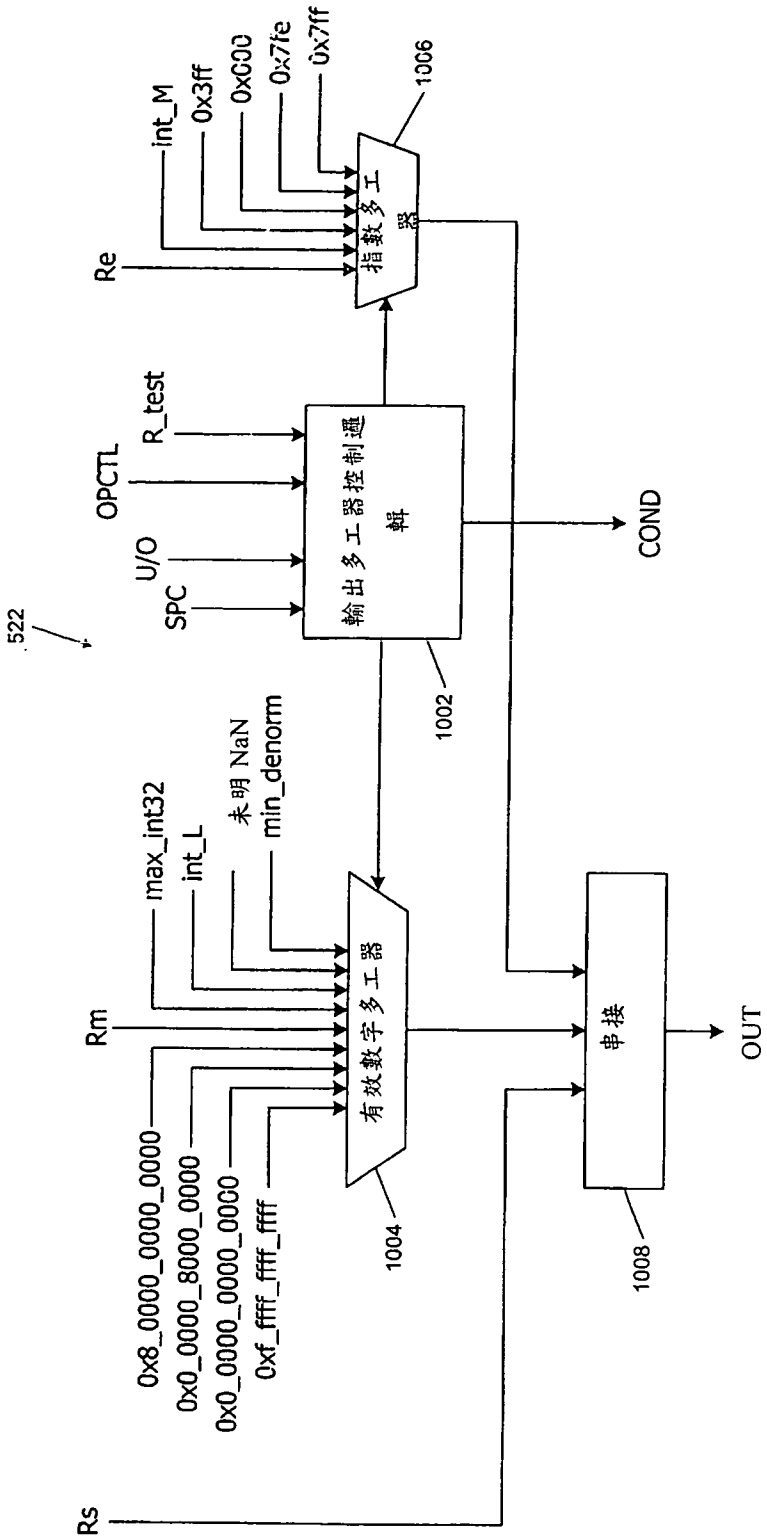
第七圖


516
Am
Bm
C_align
53x53
乘法器
802
Am*Bm
168 位元加法器
804
~Sum
Sum
MSB
MSB
U_fp32
808
多工器
810
LZD
S
806
LZD
U_fp64
Z1
Z2
多工器
814
812
BL_Sh
零
正規化邏輯
OPCTL
816
R_ZERO
<<
818
Lshift
Sn
822
+1
捨入
OPCTL
821
多工器
820
824
格式
826
Rm
int_M
int_L

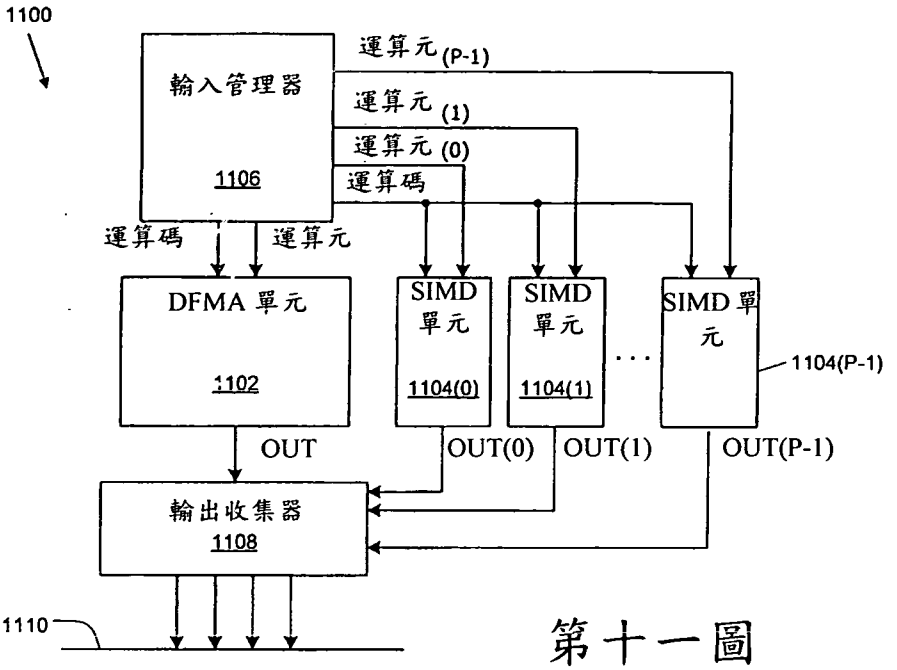
第八圖


520
As
Bs
Cs
乘積/加總
902
Sp
SignCTL
最後符號
904
SPC
R_ZERO
MSB
OPCTL
Rs

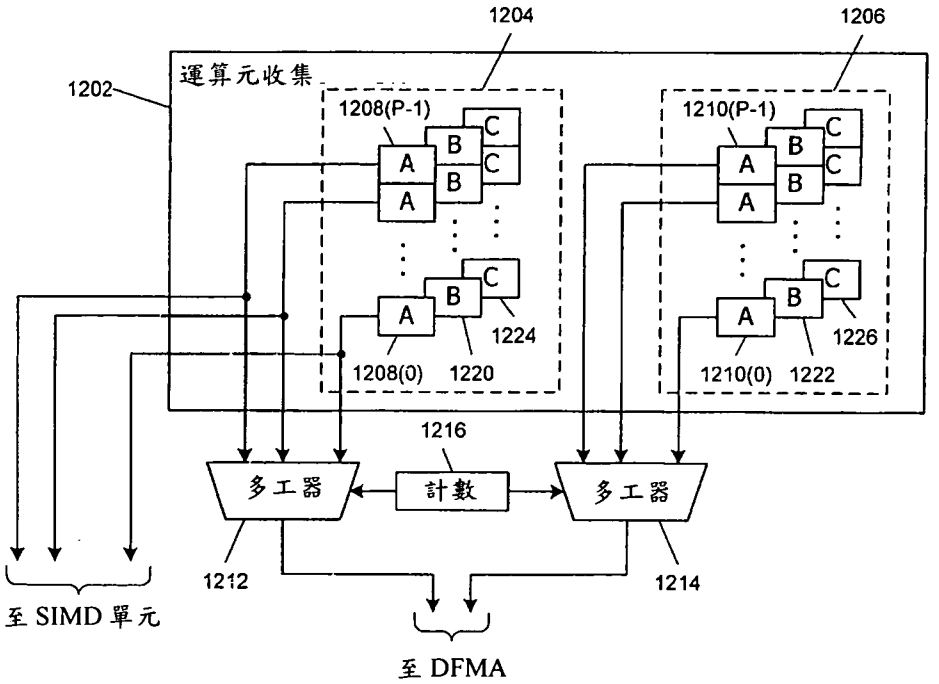
第九圖


522
Re
int_M
0x3ff
0xC00
0x7fe
0x7ff
1006
指數多工器
OPCTL
R_test
U/O
SPC
輸出多工器控制邏輯
COND
1002
Rm
max_int32
int_L
未明 NaN
min_denorm
0x8_0000_0000_0000
0x0_0000_8000_0000
0x0_0000_0000_00C0
0xf_ffff_ffff_ffff
有效數字多工器
1004
串接
OUT
1008
Rs

第十圖


1100
輸入管理器
1106
運算元 (P-1)
運算元 (1)
運算元 (0)
運算碼
運算碼
運算元
DFMA 單元
1102
SIMD 單元
1104(0)
SIMD 單元
1104(1)
SIMD 單元
1104(P-1)
OUT
OUT(0)
OUT(1)
OUT(P-1)
輸出收集器
1108
1110

第十一圖


1204
1206
1202
運算元收集
1208(P-1)
1210(P-1)
A
B
C
1224
1226
1208(0)
1220
1210(0)
1222
1216
多工器
計數
多工器
1212
1214
至 SIMD 單元
至 DFMA

第十二圖