



(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 I516915 B

(45)公告日：中華民國 105 (2016) 年 01 月 11 日

(21)申請案號：102146392

(22)申請日：中華民國 102 (2013) 年 12 月 16 日

(51)Int. Cl. : G06F11/14 (2006.01)

G06F13/38 (2006.01)

(30)優先權：2012/12/28 美國

61/746,972

2013/05/13 美國

13/892,894

(71)申請人：英特爾股份有限公司 (美國) INTEL CORPORATION (US)
美國(72)發明人：傑亞拉卡許 普拉雷達查 JAYAPRAKASH, PRAHLADACHAR BHARADWAJ
(US)；布朗 亞歷山大 BROWN, ALEXANDER (US)；達斯沙瑪 迪班德拉 DAS
SHARMA, DEBENDRA (US)；泰利依爾 尤奈德 THALIYIL, JUNAID (IN)

(74)代理人：林志剛

(56)參考文獻：

TW 200813707A

TW 200834316A

US 2009/0106636A1

US 2011/0209027A1

US 2012/0079312A1

PCIE Express Base Specification Revision 3.0 2010/11/10

審查人員：李國福

申請專利範圍項數：32 項 圖式數：13 共 69 頁

(54)名稱

即時錯誤復原

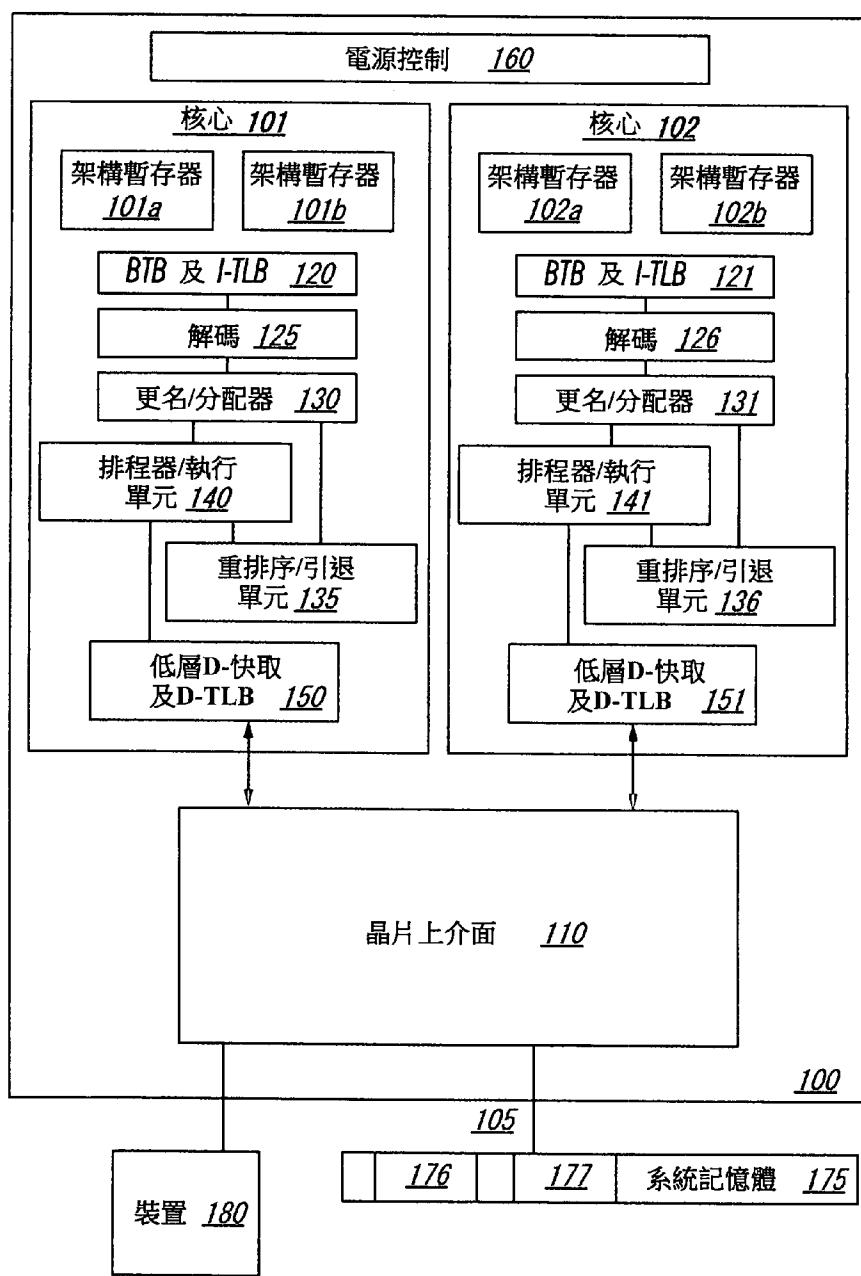
LIVE ERROR RECOVERY

(57)摘要

在一序列資料連結之一埠口識別出一封包，且判定封包與一錯誤相關。基於判定封包與錯誤相關來啟動進入一錯誤復原模式中。進入錯誤復原模式中會使序列資料連結被強壓下。在一種態樣中，將資料連結強壓下使得在錯誤復原模式期間之所有隨後的入站封包被停止且所有未決的出站請求和完成被中止。

A packet is identified at a port of a serial data link, and it is determined that the packet is associated with an error. Entry into an error recovery mode is initiated based on the determination that the packet is associated with the error. Entry into the error recovery mode can cause the serial data link to be forced down. In one aspect, forcing the data link down causes all subsequent inbound packets to be dropped and all pending outbound requests and completions to be aborted during the error recovery mode.

指定代表圖：



第 1 圖

符號簡單說明：

- 160 . . . 電源控制
- 101 . . . 核心
- 101a . . . 架構暫存器
- 101b . . . 架構暫存器
- 120 . . . BTB 及 I-TLB
- 125 . . . 解碼
- 130 . . . 更名/分配器
- 140 . . . 排程器/執行單元
- 135 . . . 重排序/引退單元
- 150 . . . 低層 D-快取及 D-TLB
- 102 . . . 核心
- 102a . . . 架構暫存器
- 102b . . . 架構暫存器
- 121 . . . BTB 及 I-TLB
- 126 . . . 解碼
- 131 . . . 更名/分配器
- 141 . . . 排程器/執行單元
- 136 . . . 重排序/引退單元
- 151 . . . 低層 D-快取及 D-TLB
- 110 . . . 晶片上介面
- 180 . . . 裝置
- 105 . . . 匯流排
- 175 . . . 系統記憶體
- 176 . . . 應用程式碼

I516915

TW I516915 B

177 · · · 碼

100 · · · 處理器

公告本

發明摘要

※申請案號：102146392

※申請日：102 年 12 月 16 日

※IPC 分類：G06F 11/14 (2006.01)

【發明名稱】(中文/英文)

13/38 (2006.01)

即時錯誤復原

Live error recovery

【中文】

在一序列資料連結之一埠口識別出一封包，且判定封包與一錯誤相關。基於判定封包與錯誤相關來啟動進入一錯誤復原模式中。進入錯誤復原模式中會使序列資料連結被強壓下。在一種態樣中，將資料連結強壓下使得在錯誤復原模式期間之所有隨後的入站封包被停止且所有未決的出站請求和完成被中止。

【英文】

A packet is identified at a port of a serial data link, and it is determined that the packet is associated with an error. Entry into an error recovery mode is initiated based on the determination that the packet is associated with the error. Entry into the error recovery mode can cause the serial data link to be forced down. In one aspect, forcing the data link down causes all subsequent inbound packets to be dropped and all pending outbound requests and completions to be aborted during the error recovery mode.

【代表圖】

【本案指定代表圖】：第(1)圖。

【本代表圖之符號簡單說明】：

160：電源控制

101：核心

101a：架構暫存器

101b：架構暫存器

120：BTB 及 I-TLB

125：解碼

130：更名/分配器

140：排程器/執行單元

135：重排序/引退單元

150：低層 D-快取及 D-TLB

102：核心

102a：架構暫存器

102b：架構暫存器

121：BTB 及 I-TLB

126：解碼

131：更名/分配器

141：排程器/執行單元

136：重排序/引退單元

151 低層 D-快取及 D-TLB

110：晶片上介面

180：裝置

105：匯流排

175：系統記憶體

176：應用程式碼

177：碼

100：處理器

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：無

發明專利說明書

(本說明書格式、順序，請勿任意更動)

【發明名稱】(中文/英文)

即時錯誤復原

Live error recovery

【技術領域】

[0001] 本揭露關於計算系統，且尤其是（但不完全是）連結錯誤抑制。

【先前技術】

[0002] 半導體處理和邏輯設計的進展已允許增加可存在於積體電路裝置上的邏輯數量。因此推論，電腦系統配置已從系統中的單一或多個積體電路演進至存在於個別積體電路上的多個核心、多個硬體執行緒、和多個邏輯處理器、以及整合於這類處理器內的其他介面。處理器或積體電路通常包含單一實體處理器晶粒，其中處理器晶粒可包括許多核心、硬體執行緒、邏輯處理器、介面、記憶體、控制器集線器、等等。

[0003] 由於更大的能力使更多處理能力裝在更小的封裝中，因此更小的計算裝置之普及性已提高。智慧型手機、平板電腦、輕薄筆記型電腦、及其他使用者設備已呈指數成長。然而，這些更小的裝置依賴用於資料儲存器和超過形狀因數之複雜處理兩者的伺服器。因此，對高效能

計算市場（即，伺服器空間）的需求也已增加。例如，在現代的伺服器中，通常不只有具有多個核心的單一處理器，還有多個實體處理器（也稱為多個插座）來增加計算能力。但由於處理能力隨著計算系統中的裝置數量成長，因此插座與其他裝置之間的通訊變得更為關鍵。

[0004] 事實上，互連已從主要處置電子通訊之較傳統的多點匯流排成長至促進快速通訊之成熟的互連架構。不幸的是，由於需要未來處理器以更高的速率消耗，因此對現有的互連架構之能力寄予相應的需求。

【圖式簡單說明】

[0005] 第 1 圖繪示用於包括多核心處理器之計算系統的方塊圖之實施例。

[0006] 第 2 圖繪示包括互連架構之計算系統的實施例。

[0007] 第 3 圖繪示包括分層堆疊之互連架構的實施例。

[0008] 第 4 圖繪示在互連架構中被產生或接收之請求或封包的實施例。

[0009] 第 5 圖繪示用於互連架構之發射器和接收器對的實施例。

[0010] 第 6 圖繪示示範能力結構的圖示。

[0011] 第 7 圖繪示示範控制狀態暫存器的圖示。

[0012] 第 8 圖繪示依照一個實施例之示範錯誤邏輯

的示意圖。

[0013] 第 9 圖繪示依照一個實施例之示範錯誤邏輯的另一示意圖。

[0014] 第 10 圖繪示示範進入錯誤復原模式中的流程圖。

[0015] 第 11 圖繪示示範進入錯誤復原模式中的另一流程圖。

[0016] 第 12 圖繪示包括多個處理器插座之計算系統の方塊之實施例。

[0017] 第 13 圖繪示關於計算系統之方塊圖的另一實施例。

【發明內容及實施方式】

[0018] 在下面的說明中，提出了許多具體細節，如特定類型的處理器和系統配置、特定硬體結構、特定架構和微架構細節、特定暫存器配置、特定指令類型、特定系統元件、特定測量/高度、特定處理器管線階段和操作等之實例，以便提供本發明的全面性了解。然而，本領域之技藝者將清楚明白無需採用這些具體細節來實現本發明。在其他情況下，未詳細說明熟知的元件或方法，如特定和其他處理器架構、用於所述之演算法的特定邏輯電路/碼、特定韌體碼、特定互連操作、特定邏輯配置、特定製造技術和材料、特定編譯器實作、碼中演算法的特定表達式、電腦系統的特定斷電和閘控技術/邏輯及其他具體操

作細節，以避免不必要的模糊本發明。

[0019] 雖然可針對在專用積體電路（如在計算平台或微處理器）中的能量保存和能量效率來說明下面的實施例，但其他實施例可適用於其他類型的積體電路和邏輯裝置。類似的技術和本文所述之實施例的教導可適用於也可得益於更好的能量效率和能量保存之其他類型的電路或半導體裝置。例如，所揭露之實施例並不限於桌上型電腦系統或輕薄筆記型電腦TM。而且，也可使用於其他裝置中，如手持裝置、平板電腦、其他薄筆記型電腦、系統晶片（SOC）裝置、及嵌入式應用。手持裝置的一些實例包括蜂巢式電話、網際網路協定裝置、數位相機、個人數位助理（PDA）、及手持PC。嵌入式應用通常包括微處理器、數位信號處理器（DSP）、系統晶片、網路電腦（NetPC）、機上盒、網路集線器、廣域網路（WAN）交換器、或能進行下面所教之功能和操作的任何其他系統。此外，本文所述之設備、方法、和系統並不限於實體計算裝置，而是也可有關用於能量保存和效率的軟體最佳化。如在下面的說明中將變得顯而易見，本文所述之方法、設備、和系統的實施例（無論是否有關硬體、韌體、軟體、或以上之組合）對於與效能考量取得平衡的「綠色技術」未來而言係重要的。

[0020] 隨著計算系統正在進展，其中的元件變得愈來愈複雜。因此，用以在元件之間耦接和傳遞的互連架構也增加複雜度以確保頻寬需求符合用於最佳元件操作。再

者，不同的市場區段要求不同態樣的互連架構以配合市場的需要。例如，伺服器需要更高的效能，而行動生態系統有時能夠為了省電而犧牲整體效能。然而，大多數結構的單一目的在於以最大省電來提供最高可能的效能。於下，討論一些互連，其將可能得益於本文所述的發明之態樣。

[0021] 參考第 1 圖，描繪出關於包括多核心處理器之計算系統的方塊圖之實施例。處理器 100 包括任何處理器或處理裝置，如微處理器、嵌入式處理器、數位信號處理器（DSP）、網路處理器、手持處理器、應用處理器、協處理器、系統晶片（SOC）、或用以執行碼的其他裝置。在一實施例中，處理器 100 包括至少兩個核心，核心 101 和 102，其可包括不對稱核心或對稱核心（所示之實施例）。然而，處理器 100 可包括可以是對稱或不對稱的任何數量之處理元件。

[0022] 在一實施例中，處理元件意指用以支援軟體執行緒的硬體或邏輯。硬體處理元件之實例包括：執行緒單元、執行緒槽、執行緒、程序單元、內文、內文單元、邏輯處理器、硬體執行緒、核心、及/或能夠保持處理器之狀態（如執行狀態或架構狀態）的任何其他元件。換言之，在一實施例中，處理元件意指能夠無關於碼（如軟體執行緒、作業系統、應用程式、或其他碼）的任何硬體。實體處理器（或處理器插座）通常意指積體電路，其可能包括任何數量的其他處理元件，如核心或硬體執行緒。

[0023] 核心通常意指位於能夠維持獨立架構狀態之

積體電路上的邏輯，其中每個獨立維持的架構狀態與至少一些專用執行資源相關。對照於核心，硬體執行緒通常意指位於能夠維持獨立架構狀態之積體電路上的任何邏輯，其中獨立維持的架構狀態共享存取執行資源。如所能看到的，當某些資源被共享且其他資源專用於架構狀態時，硬體執行緒與核心的命名之間的界線重疊。又，核心和硬體執行緒通常被作業系統視為個別邏輯處理器，其中作業系統能夠對每個邏輯處理器個別地排程操作。

[0024] 如第 1 圖所示，實體處理器 100 包括兩個核心，核心 101 和 102。在此，考量核心 101 和 102 是對稱核心，即，具有相同配置、功能單元、及 / 或邏輯的核心。在另一實施例中，核心 101 包括失序處理器核心，而核心 102 包括有序處理器核心。然而，核心 101 和 102 可個別地選自任何類型的核心，如本質核心、軟體管理核心、適用於執行本有指令集架構 (ISA) 的核心、適用於執行轉譯指令集架構 (ISA) 的核心、共同設計的核心、或其他已知核心。在異質核心環境（即，不對稱核心），某種形式的轉譯（如二進制轉譯）可用來在一或兩個核心上排程或執行碼。又，為了進一步討論，於下進一步詳細說明核心 101 所示之功能單元，在所示實施例中如核心 102 中的單元以類似方式操作。

[0025] 如圖所示，核心 101 包括兩個硬體執行緒 101a 和 101b，其也可稱為硬體執行緒槽 101a 和 101b。因此，在一實施例中，如作業系統的軟體實體可能將處理器

100 視為四個單獨的處理器，即，能夠同時執行四個軟體執行緒的四個邏輯處理器或處理元件。如上面所提及，第一執行緒與架構狀態暫存器 101a 相關，第二執行緒與架構狀態暫存器 101b 相關，第三執行緒可與架構狀態暫存器 102a 相關，且第四執行緒可與架構狀態暫存器 102b 相關。在此，如上所述，架構狀態暫存器（101a、101b、102a、和 102b）之各者可稱為處理元件、執行緒槽、或執行緒單元。如圖所示，架構狀態暫存器 101a 被複製在架構狀態暫存器 101b 中，所以能夠對邏輯處理器 101a 和邏輯處理器 101b 儲存個別架構狀態/內文。在核心 101 中，也可對執行緒 101a 和 101b 複製其他較小資源，如分配器和更名器方塊 130 中的指令指標和更名邏輯。一些資源，如重排序/引退單元 135 中的重排序緩衝器、ILTB 120、載入/儲存緩衝器、及佇列可透過分區被共享。其他資源，如通用內部暫存器、分頁表基址暫存器、低層資料快取和資料-TLB 151、執行單元 140、及失序單元 135 的部分可能被完全地共享。

[0026] 處理器 100 通常包括可被處理元件完全地共享、透過分區被處理元件共享、或專用於處理元件的其他資源。在第 1 圖中，繪示具有處理器之說明性邏輯單元/資源之純粹示範處理器的實施例。請注意處理器可包括、或省略這些功能單元之任一者，以及包括未示出之任何其他已知的功能單元、邏輯、或韌體。如圖所示，核心 101 包括簡化的代表性失序（000）處理器核心。但在不同實

施例中可利用有序處理器。OOO 核心包括用以預測將被執行/採用之分支的分支目標緩衝器 120、及用以對指令儲存位址轉譯項目的指令轉譯緩衝器（I-TLB）120。

[0027] 核心 101 更包括耦接至提取單元 120 的解碼模組 125，用以解碼提取之元件。在一實施例中，提取邏輯分別包括與執行緒槽 101a、101b 相關的個別序列器。核心 101 通常與定義/指定可執行於處理器 100 上之指令的第一 ISA 相關。為第一 ISA 之一部分的機器碼指令通常包括引用/指定將被進行的指令或操作的指令部分（稱為運算碼）。解碼邏輯 125 包括從其運算碼識別出這些指令且在管線中傳送解碼指令以供處理（如第一 ISA 所定義）的電路。例如，如於下更詳細所論述，在一實施例中，解碼器 125 包括設計或適用於識別出特定指令（如執行指令）的邏輯。由於被解碼器 125 識別出，因此架構或核心 101 採取特定、預定動作來進行與適當指令相關的工作。重要的是請注意可回應於單一或多個指令（其中一些指令可為新或舊指令）而進行本文所述之工作、方塊、操作、及方法之任一者。在一實施例中，請注意解碼器 126 識別出相同的 ISA（或其子集）。另外，在異質核心環境中，解碼器 126 識別出第二 ISA（第一 ISA 的子集或不同 ISA）。

[0028] 在一實例中，分配器和更名器方塊 130 包括分配器以保留資源，如用以儲存指令處理結果的暫存器檔案。然而，執行緒 101a 和 101b 可能能夠進行失序執行，

其中分配器和更名器方塊 130 也保留其他資源，如用以追蹤指令結果的重排序緩衝器。單元 130 也可包括暫存器更名器，用以將程式/指令參考暫存器更名成處理器 100 內部的其他暫存器。重排序/引退單元 135 包括用以支援失序執行且之後有序引退失序執行之指令的元件，如上述重排序緩衝器、載入緩衝器、和儲存緩衝器。

[0029] 在一實施例中，排程器和執行單元方塊 140 包括排程器單元，用以排程執行單元上的指令/操作。例如，在具有可用浮點數執行單元之執行單元的埠口排程浮點數指令。也包括與執行單元相關的暫存器檔案以儲存資訊指令處理結果。示範執行單元包括浮點數執行單元、整數執行單元、跳躍執行單元、載入執行單元、儲存執行單元、及其他已知的執行單元。

[0030] 較低層資料快取和資料轉譯緩衝器（D-TLB）150 係耦接至執行單元 140。資料快取係用以儲存最近使用/操作的元件（如資料運算元），其可能保持記憶體一致性狀態。D-TLB 係用以將最近虛擬/線性儲存至實體位址轉換。作為特定實例，處理器可包括用以將實體記憶體分成複數個虛擬分頁的分頁表結構。

[0031] 在此，核心 101 和 102 共享存取較高層或更外部快取，如與晶片上介面 110 相關的第二層快取。請注意較高層或更外部快取意指從執行單元增加或更遠離的快取層。在一實施例中，較高層快取是最後一層資料快取（處理器 100 上之記憶體階層中的最後快取），如第二或

第三層資料快取。然而，較高層快取並不受限於此，因為其可與指令快取相關或包括指令快取。反而可在解碼器 125 之後耦接追蹤快取（一種指令快取）以儲存最近解碼的蹤跡。在此，指令可能意指巨集指令（即，解碼器所識別的通用指令），其可解碼成一些微指令（微操作）。

[0032] 在所示之配置中，處理器 100 也包括晶片上介面模組 110。在歷史上，（於下更詳細所述之）記憶體控制器已經被包括在處理器 100 外部的計算系統中。在這種情況下，晶片上介面 11 係用以與處理器 100 外部的裝置（如系統記憶體 175、晶片組（通常包括用以連接至記憶體 175 的記憶體控制器集線器及用以連接周邊裝置的 I/O 控制器集線器）、記憶體控制器集線器、北橋、或其他積體電路）通訊。而且，在這種情況下，匯流排 105 可包括任何已知的互連，如多點匯流排、點對點互連、序列互連、平行匯流排、一致性（例如，快取一致性）匯流排、分層協定架構、差動匯流排、及 GTL 瀏流排。

[0033] 記憶體 175 可專用於處理器 100 或與系統中的其他裝置共享。記憶體 175 類型的常見實例包括 DRAM、SRAM、非揮發性記憶體（NV 記憶體）、及其他已知的儲存裝置。請注意裝置 180 可包括圖形加速器、耦接至記憶體控制器集線器的處理器或卡、耦接至 I/O 控制器集線器的資料儲存器、無線收發器、快閃裝置、音頻控制器、網路控制器、或其他已知的裝置。

[0034] 然而，最近隨著更多邏輯和裝置整合於單一

晶粒（如 SOC）上，這些裝置之各者可合併於處理器 100 上。例如，在一實施例中，記憶體控制器集線器在相同封裝及/或具有處理器 100 的晶粒上。在此，核心的部分（核心上部分） 110 包括一或更多控制器，用於與如記憶體 175 或圖形裝置 180 的其他裝置連接。包括用於與上述裝置連接之互連和控制器的配置通常稱為核心上（或非核心配置）。作為一實例，晶片上介面 110 包括用於晶片上通訊的環形互連及用於晶片外通訊的高速序列點對點連結 105。又，在 SOC 環境中，甚至更多裝置（如網路介面、協處理器、記憶體 175、圖形處理器 180、及任何其他已知的電腦裝置/介面）可整合於單一晶粒或積體電路上以提供具有高功能性和低功率消耗的小形成因數。

[0035] 在一實施例中，處理器 100 能夠執行編譯器、最佳化、及/或轉譯碼 177 以編譯、轉譯、及/或最佳化應用程式碼 176 來支援本文所述之設備和方法或與其連接。編譯器通常包括用以將原始文/碼轉譯成目標文/碼的程式或程式組。通常，以編譯器編譯程式/應用程式碼是在多個階段和經過中完成以將高階程式語言碼轉換成低階機器或組合語言碼。又，單道編譯器仍可用於簡單的編譯。編譯器可利用任何已知的編譯技術且進行任何已知的編譯器操作，如詞法分析、預處理、剖析、語意分析、碼產生、碼轉換、及碼最佳化。

[0036] 較大的編譯器時常包括多個階段，但在兩個一般階段內最常包括這些階段：（1）前端，即，通常這

裡可能發生語法處理、語意處理、及一些轉換/最佳化、及（2）後端，即，通常這裡發生分析、轉換、最佳化、及碼產生。一些編譯器有關繪示編譯器的前端與後端之間之模糊輪廓的中間。因此，可在任何上述階段或經過、以及編譯器之任何其他已知的階段或經過中發生涉及編譯器之插入、關聯、產生、或其他操作。作為說明性實例，編譯器可能在一或更多編譯階段中插入操作、呼叫、函式等，如在編譯的前端階段中插入呼叫/操作，且接著在轉換階段期間將呼叫/操作轉換成較低階碼。請注意在動態編譯期間，編譯程式碼或動態最佳化碼可插入上述操作/呼叫，以及最佳化碼以在運行時間期間執行。作為特定說明性實例，可在運行時間期間動態地最佳化二進制碼（已編譯碼）。在此，程式碼可包括動態最佳化碼、二進制碼、或以上之組合。

[0037] 類似於編譯器，如二進制轉譯器的轉譯器靜態地或動態地轉譯碼以最佳化及/或轉譯碼。因此，提到執行碼、應用程式碼、程式碼、或其他軟體環境可有關：（1）動態地或靜態地執行編譯程式、最佳化碼最佳化器、或轉譯器以編譯程式碼、維持軟體結構、進行其他操作、最佳化碼、或轉譯碼；（2）執行包括操作/呼叫的主程式碼，如已被最佳化/編譯的應用程式碼；（3）執行與主程式碼關聯的其他程式碼（如程式庫）以維持軟體結構、進行其他軟體相關操作、或最佳化碼；或（4）以上之組合。

[0038] 一個互連結構架構包括周邊元件互連（PCI）快捷（PCIe）架構。PCIe 的主要目標係用以使來自不同廠商的元件和裝置能在開放架構中相互操作、跨多個市場區段；客戶端（桌上型電腦和行動裝置）、伺服器（標準和企業）、及嵌入式和通訊裝置。PCI 快捷是被定義用於各種各樣未來計算和通訊平台的高效能通用 I/O 互連。已透過 PCI 屬性的修訂來維護一些 PCI 屬性，如其使用模型、載入儲存架構、及軟體介面，而之前的平行匯流排實作已替換成高可適性、完全的序列介面。PCI 快捷的較新版本利用點對點互連、交換為基的技術、及封包化協定的進展以傳送新等級的效能和特徵。電源管理、服務品質（QoS）、熱插拔/熱抽換支援、資料完整性、及錯誤處置為 PCI 快捷所支援的一些進階特徵。

[0039] 參考第 2 圖，繪示出由互連一組元件之點對點連結組成的結構之實施例。系統 200 包括耦接至控制器集線器 215 的處理器 205 和系統記憶體 210。處理器 205 包括任何處理元件，如微處理器、主機處理器、嵌入式處理器、協處理器、或其他處理器。處理器 205 係透過前端匯流排（FSB）206 來耦接至控制器集線器 215。在一實施例中，FSB 206 是序列點對點互連，如下所述。在另一實施例中，連結 206 包括符合不同互連標準的序列、差動互連架構。

[0040] 系統記憶體 210 包括任何記憶體裝置，如隨機存取記憶體（RAM）、非揮發性（NV）記憶體、或可

被系統 200 中的裝置存取之其他記憶體。系統記憶體 210 係透過記憶體介面 216 來耦接至控制器集線器 215。記憶體介面之實例包括雙資料速率（DDR）記憶體介面、雙通道 DDR 記憶體介面、及動態 RAM（DRAM）記憶體介面。

[0041] 在一實施例中，控制器集線器 215 是周邊元件互連快捷（PCIe 或 PCIE）互連階層中的根集線器、根聯合體、或根控制器。控制器集線器 215 之實例包括晶片組、記憶體控制器集線器（MCH）、北橋、互連控制器集線器（ICH）、南橋、及根控制器/集線器。晶片組之詞通常意指兩個實體上分離的控制器集線器，即，耦接至互連控制器集線器（ICH）的記憶體控制器集線器（MCH）。請注意目前系統通常包括與處理器 205 整合的 MCH，而控制器 215 係用以類似方式與 I/O 裝置通訊，如下所述。在一些實施例中，透過根聯合體 215 來可選地支援點對點路由。

[0042] 在此，控制器集線器 215 係透過序列連結 219 來耦接至交換器/橋接器 220。輸入/輸出模組 217 和 221（其也可稱為介面/埠口 217 和 221）包括/實作分層協定堆疊以提供控制器集線器 215 與交換器 220 之間的通訊。在一實施例中，多個裝置能夠耦接至交換器 220。

[0043] 交換器/橋接器 220 將封包/信息從裝置 225 上流（即，階層往根聯合體上升）路由至控制器集線器 215 和從處理器 205 或系統記憶體 210 下流（即，階層向下遠

離根控制器) 路由至裝置 225。在一實施例中，交換器 220 稱為多個虛擬 PCI 至 PCI 橋接器裝置的邏輯組件。裝置 225 包括將耦接至電子系統的任何內部或外部裝置或元件，如 I/O 裝置、網路介面控制器 (NIC)、附加卡、音頻處理器、網路處理器、硬碟機、儲存裝置、CD/DVD ROM、螢幕、印表機、滑鼠、鍵盤、路由器、可攜式儲存裝置、韌體裝置、通用序列匯流排 (USB) 裝置、掃描器、及其他輸出/輸出裝置。通常在如裝置之 PCIe 專用語中係稱為端點。雖然未具體顯示出，但裝置 225 可包括用以支援傳統或其他型式 PCI 裝置的 PCIe 至 PCI/PCI-X 橋接器。PCIe 中的端點裝置被分類為傳統的 PCIe、或根聯合體整合端點。

[0044] 圖形加速器 230 也透過序列連結 232 來耦接至控制器集線器 215。在一實施例中，圖形加速器 230 係耦接至 MCH，該 MCH 耦接至 ICH。交換器 220、及相應之 I/O 裝置 225 接著耦接至 ICH。I/O 模組 231 和 218 也用以實作分層協定堆疊以在圖形加速器 230 與控制器集線器 215 之間進行通訊。類似於上述 MCH，圖形控制器或圖形加速器 230 本身可整合於處理器 205 中。

[0045] 轉到第 3 圖，繪示出分層協定堆疊的實施例。分層協定堆疊 300 包括任何形式的分層通訊堆疊，如快速路徑互連 (QPI) 堆疊、PCIe 堆疊、下一代高效能計算互連堆疊、或其他分層堆疊。雖然緊接於下涉及第 2-5 圖之說明係有關 PCIe 堆疊，但相同的概念可適用於其他

互連堆疊。在一實施例中，協定堆疊 300 是包括交易層 305、連結層 310、及實體層 320 的 PCIe 協定堆疊。介面（如第 2 圖中的介面 217、218、221、222、226、和 231）可表示為通訊協定堆疊 300。表示為通訊協定堆疊也可稱為實作/包括協定堆疊的模組或介面。

[0046] PCI 快捷使用封包以在元件之間傳遞資訊。在交易層 305 和資料連結層 310 中形成封包以將資訊從傳輸元件傳送至接收元件。由於傳輸封包流過其他層，因此它們擴展了在那些層處置封包的必要額外資訊。在接收端上，逆程序發生且封包從其實體層 320 表示轉換成資料連結層 310 表示，且最後（針對傳輸層封包）轉換成能被接收裝置之交易層 305 處理的形式。

交易層

[0047] 在一實施例中，交易層 305 紣用以提供裝置的處理核心與互連架構（如資料連結層 310 和實體層 320）之間的介面。在此方面，交易層 305 的主要職責是組裝和拆卸封包（即，交易層封包、或 TLPs）。交易層 305 通常對 TLPs 管理授權式流量控制。PCIe 實作分裂的交易，即，隨時間所分開之請求和回應的交易，當目標裝置為回應收集資料時允許連結傳送其他流量。

[0048] 另外，PCIe 利用授權式流量控制。在此架構中，裝置在交易層 305 中對每個接收緩衝器通知初始的信用量。在連結之相對端的外部裝置（如第 2 圖中的控制器

集線器 215) 計數每個 TLP 所消耗的信用量。若交易不超過信用限制，則可傳送交易。當接收回應時，恢復信用量。信用架構的優點是倘若未碰到信用限制，則信用回傳的延遲時間不會影響效能。

[0049] 在一實施例中，四個交易位址空間包括配置位址空間、記憶體位址空間、輸入/輸出位址空間、及信息位址空間。記憶體空間交易包括用以將資料傳輸至記憶體映射位置及從中傳輸資料的一或多個讀取請求及寫入請求。在一實施例中，記憶體空間交易能夠使用兩個不同的位址格式，例如，短位址格式（如 32 位元位址）、或長位址格式（如 64 位元位址）。配置空間交易係用以存取 PCIe 裝置的配置空間。對配置空間的交易包括讀取請求和寫入請求。信息空間交易（或簡稱為信息）被定義為支援 PCIe 代理器之間的帶內通訊。

[0050] 因此，在一實施例中，交易層 305 組裝封包標頭/承載資料 306。可在位於 PCIe 規格網站的 PCIe 規格中找到用於目前封包標頭/承載資料的格式。

[0051] 快速地參考第 4 圖，繪示出 PCIe 交易描述符的實施例。在一實施例中，交易描述符 400 是用於傳送傳輸資訊的機制。在此方面，交易描述符（transaction descriptor）400 支援識別出系統中的交易。其他可能的用途包括追蹤預設交易排序的修改及傳輸與通道之關聯。

[0052] 交易描述符 400 包括全域識別符欄位 402、屬性欄位 404 及通道識別符欄位 406。在所示之實例中，描

繪全域識別符欄位 402 包含本地交易識別符欄位 408 和來源識別符欄位 410。在一實施例中，全域交易識別符 402 對於所有未處理請求而言是特有的。

[0053] 根據一實作，本地交易識別符欄位 408 是請求代理器所產生的欄位，且其對於對此請求代理器要求完成的所有未處理請求而言是特有的。再者，在本實例中，來源識別符 410 特有地識別出 PCIe 階層內的請求代理器。藉此，連同來源 ID 410，本地交易識別符 408 欄位提供階層域內之交易的全域識別。

[0054] 屬性欄位 404 指定交易的特性和關係。在此方面，屬性欄位 404 可能用以提供允許修改交易之預設處置的額外資訊。在一實施例中，屬性欄位 404 包括優先序欄位 412、保留欄位 414、排序欄位 416、及無窺探（no-snoop）欄位 418。在此，優先序子欄位 412 可被啟動器修改以對交易分派優先序。保留屬性欄位 414 被保留用於未來、或供應商定義的使用。可使用保留屬性欄位來實作使用優先序或安全屬性之可能的使用模型。

[0055] 在本實例中，排序屬性欄位 416 係用以供應傳達可修改預設排序規則之排序類型的可選資訊。根據一示範實作，「0」之排序屬性表示施用預設排序規則，其中「1」之排序屬性表示隨意排序，其中寫入能通過在相同方向上的寫入，且讀取完成能通過在相同方向上的寫入。窺探屬性欄位 418 係用以判斷交易是否被窺探。如圖所示，通道 ID 欄位 406 識別出與交易關聯的通道。

連結層

[0056] 連結層 310（也稱為資料連結層 310）當作交易層 305 與實體層 320 之間的中間階段。在一實施例中，資料連結層 310 的職責是提供用於在兩個元件之間交換交易層封包（TLP）的可靠機制。資料連結層 310 的一側接受交易層 305 所組裝的 TLP，施用封包序列識別符 311（即，識別號碼或封包號碼），計算並施用錯誤偵測碼（即，CRC 312），及將修改的 TLP 提交至實體層 320 以跨實體層而傳輸至外部裝置。

實體層

[0057] 在一實施例中，實體層 320 包括用以將封包實體地傳輸至外部裝置的邏輯子方塊 321 和電子子方塊 322。在此，邏輯子方塊 321 負責實體層 320 的「數位」功能。在此方面，邏輯子方塊包括用以準備輸出資訊以藉由實體子方塊 322 傳輸的傳輸部、及用以在將收到之資訊傳送至連結層 310 之前識別出且準備收到之資訊的接收部。

[0058] 實體方塊 322 包括發射器和接收器。發射器被邏輯子方塊 321 供應發射器序列化，且傳輸至外部裝置上的符元。接收器從外部裝置被供應序列化符元，且將收到之信號轉換成位元流。位元流被解序列化且供應至邏輯子方塊 321。在一實施例中，採用 8b/10b 傳輸碼，其中傳

輸/接收 10 位元符元。在此，使用特殊符元來框架出具有訊框 323 的封包。另外，在一實例中，接收器也提供從輸入序列流恢復的符元時脈。

[0059] 如上所述，雖然參考 PCIe 協定堆疊的具體實施例來說明交易層 305、連結層 310、和實體層 320，但分層協定堆疊並不受限於此。事實上，可包括/實作任何分層協定。作為一實例，表示為分層協定的埠口/介面包括：（1）用以組裝封包的第一層，即，交易層；用以序列封包的第二層，即，連結層；及用以傳輸封包的第三層，即，實體層。作為一具體實例，利用共同標準介面（CSI）分層協定。

[0060] 接下來參考第 5 圖，繪示出 PCIe 序列點對點結構的實施例。雖然繪示出 PCIe 序列點對點連結的實施例，但序列點對點連結並不不受限於此，因為它包括用於傳輸序列資料的任何傳輸路徑。在所示之實施例中，基本 PCIe 連結包括兩個低電壓、差動驅動的信號對：傳輸對 506/511 與接收對 512/507。藉此，裝置 505 包括用以將資料傳輸至裝置 510 的傳輸邏輯 506 及用以從裝置 510 接收資料的接收邏輯 507。換言之，兩個傳輸路徑（即，路徑 516 和 517）、及兩個接收路徑（即，路徑 518 和 519）係包括在 PCIe 連結中。

[0061] 傳輸路徑意指用於傳輸資料的任何路徑，如傳輸線、銅線、光線、無線通訊通道、紅外線通訊連結、及其他通訊路徑。在兩個裝置（如裝置 505 與裝置 510）

之間的連接稱為連結，如連結 415。連結可支援一個通道，每個通道代表一組差動信號對（一對用於傳輸、一對用於接收）。為了擴展頻寬，連結可聚合由 xN 表示的多個通道，其中 N 是任何支援的連結寬度，如 1、2、4、8、12、16、32、64、或更寬。

[0062] 差動對意指用以傳輸差動信號的兩個傳輸路徑，如線 416 和 417。作為一實例，當線 416 從低電壓準位切換至高電壓準位時（即，上升邊緣），線 417 從高邏輯準位驅動至低邏輯準位（即，下降邊緣）。差動信號可能表現出較佳的電特性，如較佳的信號完整性，即，交叉耦合、電壓過衝/下衝、振鈴等等。這允許較佳的時序視窗，該較佳的時序視窗使傳輸頻率能更快。

[0063] 錯誤處置和抑制可以是互連平台（如 PCIe）以及其他互連平台（包括行動產業處理器介面（MIPI）、Intel® 快速路徑互連（QPI）、及其他互連平台和架構）的重要元素。例如，在多插座環境中，單一交易所造成的錯誤會導致已停止資料的競爭情況、全系統重設、及其他影響。在一些傳統架構中，已允許與錯誤相關的入站封包繼續在系統內傳播，直到呼叫錯誤處置器且採取適當的動作為止。在一些情況下，等待錯誤處置器動作的時間已導致大視窗，而允許在其他影響中，保持不抑制錯誤、和隨後的資料。在上述情況中，在其他問題中，系統（主機）記憶體可能留在未保護狀態，因為對系統記憶體之隨後的寫入交易或與資料完成交易繼續傳播，直到錯誤處置器採

取動作以補救偵測到的錯誤為止。

[0064] 在一些實作中，能在互連架構中的埠口（如 PCIe 相容埠口）上支援即時錯誤復原（LER）特徵。LER 能用以擷取位於偵測到錯誤之根埠口的錯誤且防止在埠口之外傳播錯誤。藉由使迫使連結進入使所有出站請求被中止且在觸發 LER 條件的封包之後的所有入站封包也被停止的連結壓下狀態中，錯誤之偵測能觸發即時錯誤復原。這會導致錯誤將在偵測到錯誤的特定埠口中被抑制。能將錯誤報告給全域錯誤偵測和處置模組，包括識別出偵測到錯誤的埠口和連結。在一些情況下，軟體為基的錯誤評估工具（如錯誤處置器）能接著處理並清除所報告的錯誤而無需全系統重設。當確認壓下連結之埠口的入站和出站佇列已被排出時，能啟動連結之復原以將連結帶回至傳輸狀態（例如，下面的抑制和清除錯誤），又，無需全系統重設。

[0065] 在一示範實作中，即時錯誤復原（LER）模式能包括偵測特別定義之錯誤（如進階錯誤報告（AER）、其他 PCIe 錯誤）、或 PCIe 相容埠口之入站或出站路徑上的專用定義錯誤。錯誤的錯誤遮罩能被識別出且用以判斷是否已定義觸發（LER）模式的錯誤類型。在觸發 LER 模式事件的錯誤上，埠口的關聯連結被迫使至連結壓下狀態且所有出站請求和入站交易被停止。此外，完成封包也被停止，雖然在一些情況下，介面邏輯能合成將被發送至發出請求之核心（或其他 PCIe 埠口）的主要中止完成來代

替被停止的完成。軟體為基的控制器能檢查 LER 模式「併列空」狀態指示器以識別出所有入站（和出站）併列是空的，且控制器能觸發從 LER 模式離開以使訓練於壓下連結的連結開始復原連結。在一實例中，能透過 PCIe 擴展能力結構、或用於容易識別 LER 相關條件以及未來彈性的其他暫存器或結構來容納與 LER 模式相關的功能。

[0066] 轉到第 6 圖，顯示出示範 LER 能力結構 605 的簡化圖。能在示範能力結構內提供各種欄位或部分。例如，在第 6 圖之實例中，在可能的其他欄位中，能力欄位能包括 LER 能力（LER_CAP）欄位、LER 標頭（LER_HDR）欄位、LER 控制狀態（LER_CTRLSTS）欄位、LER 不可校正錯誤遮罩（LER_UNCERRMSK）欄位、LER 不可校正專用錯誤遮罩（LER_XPUNCERRMSK）欄位、LER 根埠口錯誤遮罩（LER_RPERRMSK）欄位。LER_CAP 能力欄位能識別 LER 能力結構以及連接至下一個能力結構的點。LER_HDR 欄位可以是至少部分供應商定義的且識別特定裝置所採用之 LER 功能的型式和特徵。另外，LER_CAP 和 LER_HDR 能識別出裝置是否（及到多大程度）支援 LER。在其他實例中，透過識別特定裝置（及對應埠口）所支援之 LER 的型式，軟體控制器能判定特定裝置所支援的 LER 特徵。

[0067] 在 LER_CTRLSTS 能力欄位的情況下，能提供各種態樣和控制位元用於提供 LER 模式。例如，在其

他實例中，LER_CTRLSTS 欄位能以各種控制位元來具體化 LER 控制和狀態暫存器以致能 LER 和 LER 功能的各種狀態位元，包括佇列空條件、LER 嚴重性映射、主動停止模式、中斷啟動、嚴重性重映射。

[0068] 能透過 LER 能力結構來提供各種錯誤遮罩以用於識別出哪些錯誤碼會觸發 LER 條件。PCIe 及其他平台能支援各種錯誤條件碼及對應遮罩。在其他實例中，LER 能提供能與標準錯誤遮罩一起處理的額外遮罩以使 LER 被致能。在一實例中，能提供 LER 遮罩映射至透過埠口的架構所提供之對應錯誤狀態遮罩。例如，PCIe 平台能包括不可校正錯誤遮罩暫存器、根埠口錯誤遮罩暫存器、一或更多專用錯誤遮罩暫存器（例如，以擴展已超出平台之正式規範所定義外會被遮罩的錯誤組）且在可能的其他實例中，LER 能力結構能包括對應錯誤遮罩，如 LER 不可校正錯誤遮罩（LER_UNCERRMSK）和 LER 不可校正專用錯誤遮罩（LER_XPUNCERRMSK）、及 LER 根埠口錯誤遮罩（LER_RPERRMSK）。在一實作中，LER_UNCERRMSK 位元能服務來自 LER 模式的遮罩錯誤事件。換言之，基於 LER 錯誤遮罩的值，能判斷是否已指定特定錯誤類型（例如，也在平台之錯誤遮罩暫存器中被識別或辨識出）以當偵測時觸發 LER。使用者（如管理員）能定義 LER 遮罩暫存器的哪些遮罩位元被設定以選擇哪些錯誤觸發 LER。在其他可能的實例中，作為一實例，若對應於變形交易層封包（TLP）的位元具有在平台

(例如，PCIe 相容平台) 之不可校正錯誤狀態遮罩暫存器中設定的值以指出上述 TLP 錯誤被記錄為不可校正(或其他)錯誤，且 LER_UNCERRMSK 暫存器的對應位元指出相同錯誤會觸發 LER，當偵測到變形 TLP 錯誤(例如，使用交易層邏輯)時，能觸發 LER 模式。

[0069] 暫時轉到第 7 圖，繪示出示範 LER 控制和狀態暫存器(或 LER_CTRLSTS 能力欄位) 705 的圖示。在第 7 圖之特定實例中，在其他可能的實例中，能提供各種欄位和控制位元，如 LER_Status 位元、LER_Port_Quiesced 、 LER_Enable 位元、 LER_Severity_Enable 位元、 LER_Drop_TXN 位元、及 LER_INTEN 位元。在一實例中， LER_Status (或狀態) 位元能表示偵測到錯誤，其使 PCIe 埠口進入即時錯誤復原(LER)模式中。如上所述，在 LER 模式中，連接會立即地被迫使進入連結壓下去能狀態且當設定此位元時，所有出站交易被中止(包括可能導致錯誤的封包)。又，在其他條件和實例中，在設定狀態位元之後，其可能僅在清除所有相關無遮罩狀態位元、或設定對應 LER 遮罩位元之後被清除。當清除無遮罩錯誤條件之後，接著可藉由軟體將值(例如，「1」)寫入至欄位來清除此位元。在 LER 事件之後清除狀態位元會使連結自動地開始重新訓練為傳輸狀態，結束中止出站交易及阻止入站交易。在一些實作中，能配置迫使連結至壓下狀態，使得 LER 事件不觸發「驚奇 LinkDown (surprise LinkDown)」錯誤。此

外，在其他實例中，在一些實作中，一些裝置（如某些 PCIe 卡）能配置以當它們接收指出「Disabled」狀態的訓練序列時進入內部重設中。

[0070] 繼續進行第 7 圖之實例，狀態暫存器 705 能包括埠口靜止位元（例如，LER_Port_Quiesced），指出在埠口已進入 LER 模式之後，埠口沒有更多未決的入站或出站封包。埠口靜止位元能用以確認封包已成功地在 LER 事件之後從被迫使進入連結壓下狀態中的連接排出。此外，軟體控制器（如錯誤管理工具或錯誤處置器）能參考埠口靜止位元來識別出何時清除 LER 狀態位元是安全的且允許埠口被帶出 LER 模式。

[0071] 能利用其他位元來啟動透過 LER 模式所提供的各種功能。例如，LER 中斷致能位元（例如，LER_INTEN）能指出當觸發 LER 事件（例如，透過設定 LER 狀態位元）時，是否將產生中斷（如信息發信中斷（MSI））。另外，能使用 LER 交易停止位元（例如，LER_Drop_Txn）以識別出在進入 LER 之後，之後的交易是否當埠口配置允許時就會被停止。此外，能提出 LER 嚴重性變更位元（例如，LER_Severity_En），當其被設定時會迫使觸發 LER 模式的錯誤被發信作為可校正錯誤，而不是不可校正（或類似）錯誤。例如，致能 LER_Severity_En 會導致通常將發信作為不可校正非致命嚴重性 1 或不可校正致命嚴重性 2，而不是被發信作為可校正嚴重性 0 的錯誤。例如，在其他情況中，能設定上述

位元以使得提高裝置中的錯誤處置能力。此外，能提供額外位元（例如，LER_Enable）以選擇性地致能或去能 LER 功能，有效地打開或關閉 LER 功能。

[0072] 轉到第 8 圖，顯示出能用以實作至少一部分 LER 功能之示範邏輯的示意方塊圖 800。例如，能接收對應於在一埠口偵測到之特定錯誤類型的錯誤信號 805。能藉由標準錯誤狀態遮罩 810 來遮罩信號 805 的錯誤條件以判斷（例如，在 815 中）錯誤條件的嚴重性是否高於「嚴重性 0」（例如，可校正錯誤）。若是如此，則信號 815 能前進（例如，至 AND 閘 825）而與 LER 錯誤遮罩 820 比較以判斷錯誤條件是否會觸發 LER 模式。若 LER 功能被致能（例如，在 LER 控制狀態暫存器 705 的 835 中）且 LER 施用於所偵測之錯誤（在 805 中），則能設定 LER 狀態位元（例如，在 830 中）以呼叫 LER 模式且迫使連結進入去能狀態 840 中。例如，能傳送交握、去能訓練序列、或其他信號以將連結送進壓下狀態中。在去能或壓下狀態中，對所有出站請求施加主要中止（例如，藉由以所有「1」來建構完成，所有出站完成被停止，且所有入站封包將被停止。此外，若設定嚴重性變更位元 845，則邏輯（在 850 中）能迫使來自較高嚴重性等級（例如，不可校正嚴重性 1 或嚴重性 2）之錯誤條件的嚴重性指出較低可校正嚴重性等級。這能確保回應於（透過信號 805 所傳遞之）錯誤條件而不進行更侵略性的步驟（例如，全系統重設）。

[0073] 在 LER 模式下，未處理的交易及入站和出站封包被停止並從連結排出。當硬體（或軟體）邏輯停止未決的交易時，能設定埠口靜止位元 855。另一控制器（如至少部分軟體為基的錯誤處置器）能嘗試解決和分析錯誤條件。錯誤處置器（或其他工具）能進一步清除錯誤暫存器（例如，當解析錯誤條件時）。此外，當清除錯誤暫存器時，錯誤處置器能藉由埠口靜止位元 855 的值來識別出在連結（或埠口）上沒有任何未決的交易，及清除 LER 狀態位元以使連結離開 LER 模式。在一實例中，從 LER 離開包括重新訓練連結且帶回至主動傳輸狀態。

[0074] 如上所述，LER 控制狀態暫存器（例如，705）能包括控制位元，用來改變 LER 映射的嚴重性以改變觸發 LER 之錯誤碼的嚴重性。例如，如在由第 9 圖之示意圖 900 繪示的實例所示，能致能 LER 控制狀態暫存器中的 LER 嚴重性變更（或映射）位元（例如，在 845 中）。致能 LER 控制狀態暫存器會導致通常已被視為不可校正錯誤（例如，沒有致能 LER），而不是被發信作為可校正嚴重性 0（而不是嚴重性 1 或 2）錯誤的錯誤條件（例如，905）。因此，在此特定實例中，能設定對應嚴重性 0 位元，能產生嚴重性 0 MSI，及能確立嚴重性 0 ERR[0]接腳而不是嚴重性 1 或 2。若沒有嚴重性重映射，則即使錯誤在埠口被 LER 模式抑制，仍能觸發較高嚴重性中斷，如系統管理中斷（SMI）、不可遮罩中斷（NMI）、機器檢查架構（MCA）錯誤、或其他中斷或錯

誤事件。示範嚴重性重映射能確保藉由將嚴重性重映射至較低嚴重性（可校正錯誤）來正確地抑制由於 LER 所觸發的錯誤，使得不會從全域錯誤產生邏輯（例如，910）觸發 SMI、NMI 等。反而，全域錯誤產生邏輯 910 能將錯誤處理為嚴重性 0 錯誤。

[0075] 如上所介紹，在 LER 模式下，能在觸發 LER 模式之後立即地處置不同封包，以便抑制位於特定埠口的錯誤。例如，能在錯誤被記錄於錯誤暫存器中的相同時脈週期上觸發 LER 以啟動終止受影響連結的程序。表格 1 顯示當觸發 LER 模式時將如何處置各種封包類型：

表格 1

封包類型	入站路徑	出站路徑	註解
非一次投遞式 TLP	停止	停止	當線重新訓練時重新初始化信用
一次投遞式 TLP	停止	停止	當線重新訓練時重新初始化信用
完成 TLP	停止；發送新的主要中止合成完成	停止	核心將在沒有 CMP 下擋置，所以在 LER 之後立即地合成主要中止完成且當連結重新訓練時重新初始化完成信用

[0076] 如上所述，在一些情況下，入站完成封包當在 LER 模式下停止時會被攔截和合成以便將對應主要中止完成傳遞至負責發送施用完成之對應請求的核心。在一些情況下，核心能等待已停止的完成且合成主要中止完成能防止核心擋置。

[0077] 能對 LER 致能不同種類的錯誤。在其他實例中，錯誤種類能包括例如與輸入（例如，從另一裝置或關聯遠端代理器）封包（如 TLP）相關的錯誤、與輸出（例如，出站至另一裝置或關聯遠端代理器）封包相關的錯誤、以及非封包錯誤（如缺陷埠口錯誤）。

[0078] 轉到第 10 圖，顯示出繪示關於與連結上之一或多輸入 TLP 相關的錯誤之示範 LER 事件的流程圖。端點 1005 能在 PCIe 相容連結之埠口上與主機（例如，1025）通訊，PCIe 相容連結包括 PCIe 實體層 1010、資料連結層 1015、及交易層 1020。在傳輸 L0 狀態下，能在 PCIe 實體層 1010 上發送多個寫入（WR）和讀取（RD）請求封包。在其他實例中，一些封包在被傳送至系統或主記憶體 1025 之前會被併列或緩衝於 TLP 併列 1030 中。

[0079] 在第 10 圖之實例中，能結合寫入請求封包 WR3 來判定錯誤（如變形 TLP 錯誤）。埠口錯誤偵測邏輯能偵測錯誤且進一步判定為會觸發 LER 狀態之類型的錯誤。藉此，偵測邏輯能立即地（例如，在相同時脈週期上）確立 LER 狀態信號而由此觸發將相關連結轉成壓下或去能狀態 1035。在一些情況下，導致錯誤的封包能根據基於錯誤之平台的標準協定（例如，符合 PCIe 正式規範）而被停止。此外，當連結轉成去能狀態以在埠口提供錯誤抑制時，確立的 LER 狀態信號也能用以停止來自端點 1005 之每個隨後的入站封包（例如，RD 3、WR 4、RD 4、WR 5 等）。以此方式，能防止記憶體（例如，1025）

在為錯誤條件之原因的 TLP（例如，WR 3）之後從 TLP 毀損。端點 1005 當識別出進入去能狀態中時會停止發送額外的 TLP。另外，出站封包（例如，RD_CMP1、RD_CMP2 等）也會在進入且在 LER 模式內期間被停止。然而，在一些實作中，若當啟動 LER 模式時傳輸出站 TLP，則可允許出站 TLP 之傳輸完成不中斷。

[0080] 第 11 圖繪示示範轉成 LER 模式的另一圖示。變形或其他錯誤的 TLP（例如，「壞」封包 1105）能被端點 1005 的發射器處理且釋放於被（例如，埠口之）積體 I/O 邏輯（IIO）1110 處理的 PCIe 連結上之端點 1005 的發射器（Tx（ex））上。IIO 能解碼壞 TLP 1105 且（例如，從 LER 錯誤遮罩）判定 TLP 1105 會觸發 LER 事件且壞 TLP 1005 能被停止。藉此，IIO 1110 能藉由進入復原狀態中，且接著從復原狀態進入去能狀態來啟動強壓下連結。此外，IIO 能開始發送去能訓練序列（例如，1115）。又，端點 1005 的接收器（Rx（ex））能接收訓練序列且識別出連結被終止且停止發送 TLP 及本身開始重覆訓練序列（例如，在 1120）。在其他可能的實例中，在一實例中，訓練序列能包括 16 位元組的 TS1 訓練序列。訓練序列（例如，1115）能被發送以引導連結進入復原狀態。訓練序列能與去能位元組一起發送，使得這兩個連結進入去能狀態。

[0081] 依照確立 LER 信號且進入 LER 狀態中，能開始清除（例如，1125）IIO 管線。TLP 能被緩衝於 IIO 切

換併列中。在 LER 中，清除能包括排出併列、在端點 1005 結束發送 TLP 之前都解碼從端點 1005 接收的每個 TLP、及停止來自重試緩衝器的封包。交易層也能丟棄任何未處理交易。例如，進入去能狀態會使資料連結層轉成 DL_Inactive 狀態，這會使資料連結層重試緩衝器中的所有封包被丟棄，及交易層丟棄任何未處理傳輸。此外，如上所述，在其他實例中，LER 模式能進一步包括合成一或更多停止的完成封包、改變 LER 條件的嚴重性（例如，變成可校正的）。當已排出所有入站和出站併列時，已清除所有暫存器，且清除 LER 狀態位元，連結能被帶出 LER 模式且被重新訓練。

[0082] 能類似於由入站封包產生的錯誤來處置涉及輸出封包的錯誤。在一實例中，若輸入的非一次投遞式請求導致完成被回傳非支援請求或完成符中止狀態，則這些完成可被停止且觸發 LER 模式。在另一實例中，若出站同位錯誤或損壞的 TLP 觸發 LER，則上述損壞的 TLP 能被停止。在於埠口出站偵測到非致命損壞錯誤的情況下，錯誤能被記錄於入站路徑上的標準錯誤記錄暫存器中且損壞的封包會被停止。另外，與入站錯誤一樣，致能 LER 模式會導致產生可校正錯誤中斷而不是非致命或致命的不可校正中斷（例如，若致能 LER 嚴重性重映射特徵）。又，對出站錯誤觸發的 LER 模式能進一步造成 PCI 快捷埠口連結被強壓下，具有對此中止之連結和停止之出站完成的所有出站請求（包括導致錯誤的請求或完成）。如在

先前實例中連結接著能進入復原。

[0083] 類似的原理也能適用於非封包錯誤。例如，如上所述，LER 能用於對缺陷埠口隔離錯誤。例如，LER 能去能偵測到被視為致命的且會導致系統關機之錯誤的埠口。在其他實例中，這類致命的錯誤能包括例如流量控制協定錯誤、驚奇壓下錯誤、及資料連結層協定錯誤。

[0084] 請注意上述設備、方法、及系統可實作在如上述之任何電子裝置或系統中。作為具體圖示，下面的圖提出用於利用如本文所述之本發明的示範系統。當更詳細說明下面的系統時，從上面的討論揭露、說明、及回顧一些不同的互連。而且如顯而易見地，上述的進展可適用於那些互連、結構、或架構之任一者。

[0085] 現在參考第 12 圖，所顯示的係依照本發明之實施例之第二系統 1200 的方塊圖。如第 12 圖所示，多處理器系統 1200 是點對點互連系統，且包括經由點對點互連 1250 耦接的第一處理器 1270 和第二處理器 1280。處理器 1270 和 1280 之各者可以是一些型式的處理器。在一實施例中，1252 和 1254 是一序列點對點同調互連結構（如 Intel 的快速路徑互連（QPI）架構）的一部分。於是，可在 QPI 架構內實作本發明。

[0086] 儘管只有顯示兩個處理器 1270、1280，但將了解本發明之範圍並不以此為限。在其他實施例中，一或多額外的處理器可存在於特定處理器中。

[0087] 顯示處理器 1270 和 1280 分別包括積體記憶

體控制器單元 1272 和 1282。處理器 1270 也包括作為其匯流排控制器單元點對點（P-P）介面 1276 和 1278 的部分；同樣地，第二處理器 1280 包括 P-P 介面 1286 和 1288。處理器 1270、1280 可使用 P-P 介面電路 1278、1288 經由點對點（P-P）介面 1250 來交換資訊。如第 12 圖所示，IMC 1272 和 1282 將處理器耦接至各別記憶體（即記憶體 1232 和 1234），其可以是本地附接於各別處理器之主記憶體的一部分。

[0088] 處理器 1270、1280 各使用點對點介面電路 1276、1294、1286、1298 經由各別 P-P 介面 1252、1254 來與晶片組 1290 交換資訊。晶片組 1290 也經由介面電路 1292 沿著高效能圖形互連 1239 與高效能圖形電路 1238 交換資訊。

[0089] 共享快取（未示出）可包括在任一處理器中或兩者處理器之外；還可經由 P-P 互連與處理器連接，使得若將處理器置於低功率模式中，則任一或兩者處理器的本地快取資訊可儲存於共享快取中。

[0090] 晶片組 1290 可經由介面 1296 來耦接至第一匯流排 1216。在一實施例中，第一匯流排 1216 可以是周邊元件互連（PCI）匯流排、或如 PCI 快捷匯流排或另一第三代 I/O 互連匯流排的匯流排，雖然本發明之範圍並不以此為限。

[0091] 如第 12 圖所示，各種 I/O 裝置 1214 係與匯流排橋接器 1218 一起耦接至第一匯流排 1216，其中匯流

排橋接器 1218 將第一匯流排 1216 耦接至第二匯流排 1220。在一實施例中，第二匯流排 1220 包括低接腳數 (LPC) 匯流排。在一實施例中，各種裝置係耦接至第二匯流排 1220，包括例如鍵盤及/或滑鼠 1222、通訊裝置 1227 及如磁碟機或通常包括指令/碼和資料 1230 之其他大容量儲存裝置的儲存單元 1228。此外，音頻 I/O 1224 被顯示為耦接至第二匯流排 1220。請注意其他架構係可能的，其中所包括的元件和互連架構有所不同。例如，系統可實作多點匯流排或其他這類架構，來取代第 12 圖之點對點架構。

[0092] 接下來轉到第 13 圖，描繪出依照本發明之系統晶片 (SOC) 設計的實施例。作為特定說明性實例，SOC 1300 係包括在使用者設備 (UE) 中。在一實施例中，UE 意指被終端使用者用來通訊的任何裝置，如手持電話、智慧型手機、平板電腦、輕薄筆記型電腦、具有寬頻適配器的筆記型電腦、或任何其他類似的通訊裝置。UE 通常連接至基地台或節點，實際上這可能對應於 GSM 網路中的行動站 (MS)。

[0093] 在此，SOC 1300 包括 2 個核心 1306 和 1307。類似於上面的討論，核心 1306 和 1307 可符合指令集架構，如 Intel® 架構核心™ 為基的處理器、先進微裝置公司 (AMG) 處理器、MIPS 為基的處理器、ARM 為基的處理器設計、或其客戶、以及其被授權者或採用者。核心 1306 和 1307 係耦接至與匯流排介面單元 1309 和 L2 快

取 1312 關聯的快取控制 1308 以與系統 1300 的其他部分通訊。互連 1310 包括晶片上互連，如 IOSF、AMBA、或其他上述互連，這可能實作所述發明的一或更多態樣。

[0094] 介面 1310 將通訊通道提供至其他元件，如用以與 SIM 卡連接的用戶身分模組（SIM）1330、用以保存用於藉由核心 1306 和 1307 執行來初始化和啟動 SOC 1300 之啟動碼的啟動唯讀記憶體 1335、用以與外部記憶體（例如，DRAM 1360）連接的 SDRAM 控制器 1340、用以與非揮發性記憶體（例如，快閃 1365）連接的快閃控制器 1345、用以與周邊裝置連接的周邊控制 1350（例如，序列周邊介面）、用以顯示和接收輸入（例如，觸碰致能輸入）的視頻編解碼器 1320 和視頻介面 1325、用以進行圖形相關計算的 GPU 1315、等等。這些介面之任一者可結合本文所述之發明的態樣。

[0095] 另外，系統繪示用於通訊的周邊裝置，如藍芽模組 1370、3G 數據機 1375、GPS 1380、及 WiFi 1385。請注意如上所述，UE 包括用於通訊的無線電。因此，這些周邊通訊模組並非所有都是必須的。然而，在 UE 中，會包括形成用於外部通訊之無線電的一些周邊通訊模組。

[0096] 儘管已針對有限數量的實施例來說明本發明，但本領域之那些技藝者將了解據此的許多修改和變化。預期所附之申請專利範圍涵蓋如落在本發明之真正精神和範圍內的所有這樣的修改和變化。

[0097] 設計可經歷各種不同階段，從建立到模擬到製造。表現設計的資料可以一些方式來表現設計。首先，如在模擬中有用的，可使用硬體描述語言或另一功能描述語言來表現硬體。此外，可在設計程序的一些階段中產生具有邏輯及/或電晶體閘的電路級模型。再者，在一些階段中，大多數設計達到表現硬體模型中的各種裝置之實體佈置的資料級。在使用傳統半導體製造技術的情況下，表現硬體模型的資料可以是對用以製造積體電路的遮罩指明不同遮罩層上存在或不存在各種特徵的資料。在設計的任何表現中，資料可儲存於任何形式的機器可讀媒體中。如磁碟的記憶體或磁或光儲存器可以是機器可讀媒體，用以儲存經由經調變或以其他方式產生來傳輸資訊的光或電波所傳輸之資訊。當指出或傳送碼或設計的電子載波被傳輸到進行複製、緩衝、或重新傳輸電子信號的磁碟區時，便產生了新副本。因此，通訊提供者或網路提供者可在有形的機器可讀媒體上至少暫時地儲存具體化本發明之實施例之技術的物件，如編碼成載波的資訊。

[0098] 如本文所使用之模組有關硬體、軟體、及/或韌體之任何組合。作為一實例，模組包括與用以儲存適用於被微控制器執行的碼之非暫態媒體相關的硬體（如微控制器）。因此，在一實施例中，提到模組係意指專門配置以識別及/或執行被保存於非暫態媒體上之碼的硬體。再者，在另一實施例中，使用模組係意指包括碼的非暫態媒體，其專門適用於被微控制器執行來進行預定操作。而

且，在又一實施例中，能推論模組（在本實例中）之詞可意指微控制器與非暫態媒體之組合。往往被繪示成分開的模組邊界通常會改變且可能重疊。例如，第一和第二模組可共享硬體、軟體、韌體、或以上之組合，同時可能保留一些單獨的硬體、軟體、或韌體。在一實施例中，使用邏輯之詞包括如電晶體、暫存器的硬體、或如可程式化邏輯裝置的其他硬體之硬體。

[0099] 在一實施例中，使用「用以」或「配置以」之用語意指佈置、放置在一起、製造、提供以銷售、進口及/或設計設備、硬體、邏輯、或元件以進行指定或決定工作。在本實例中，未正在操作的設備或其元件若被設計、耦接、及/或互連以進行指定工作，則仍「配置以」進行指定所述工作。僅作為說明性實例，邏輯閘在操作期間可提供 0 或 1。但「配置以」將致能信號提供至時脈的邏輯閘不包括可提供 1 或 0 之每個可能的邏輯閘。反而，邏輯閘是以在操作期間，1 或 0 輸出係用以致能時脈之一些方式耦接的一個邏輯閘。再次請注意使用「配置以」之詞不需要操作，而是將重點放在設備、硬體、及/或元件的潛在狀態，其中在潛在狀態下，設備、硬體、及/或元件被設計為當設備、硬體、及/或元件正在操作時進行特定工作。

[0100] 再者，在一實施例中，使用「能夠/能夠用以」、及/或「可操作以」之用語有關以能夠以指定方式來使用設備、邏輯、硬體、及/或元件之這樣的方式所設

計的一些設備、邏輯、硬體、及/或元件。在一實施例中，注意如上之用以、能夠用以、或可操作以意指設備、邏輯、硬體、及/或元件的潛在狀態，其中設備、邏輯、硬體、及/或元件並非正在操作，而是以能夠以指定方式來使用設備之這樣的方式來設計。

[0101] 如本文所使用，值包括數字、狀態、邏輯狀態、或二進制邏輯狀態的任何已知表示。通常，使用邏輯準位、邏輯值、或邏輯的值也稱為「1」和「0」，其簡單地表示二進制邏輯狀態。例如，1 級指高邏輯準位且 0 級指低邏輯準位。在一實施例中，儲存胞元（如電晶體或快閃胞元）也許能夠保存單一邏輯值或多個邏輯值。然而，已使用電腦系統中之值的其他表示。例如，十進制數的十也可表示為 1010 的二進制值和十六進制字母 A。由此，值包括能夠保存於電腦系統中之資訊的任何表示。

[0102] 此外，狀態可以值或一部分值來表示。作為一實例，第一值（如邏輯 1）可表示預設或初始狀態，而第二值（如邏輯 0）可表示非預設狀態。另外，在一實施例中，重設和設定之詞分別意指預設和更新值或狀態。例如，預設值可能包括高邏輯值（即，重設），而更新值可能包括低邏輯值（即，設定）。請注意值的任何組合可用來表示一些狀態。

[0103] 上面提出的方法、硬體、軟體、韌體或碼之實施例可經由儲存於可被處理元件執行之機器可存取、機器可讀、電腦可存取、或電腦可讀媒體上的指令或碼來實

作。非暫態機器可存取/可讀媒體包括提供（即，儲存及/或傳輸）為機器（如電腦或電子系統）可讀的形式之資訊的任何機制。例如，非暫態機器可存取媒體包括隨機存取記憶體（RAM），如靜態RAM（SRAM）或動態RAM（DRAM）、ROM、磁或光儲存媒體、快閃記憶體裝置、電子儲存裝置、光儲存裝置、聲儲存裝置、用於保存從暫態（傳播）信號（例如，載波、紅外線信號、數位信號）接收的資訊之其他形式的儲存裝置、等等，這些都與可從中接收資訊的非暫態媒體不同。

[0104] 用以編程邏輯來進行本發明之實施例的指令可儲存於系統中的記憶體（如DRAM、快取、快閃記憶體、或其他儲存器）內。再者，能經由網路或藉由其他電腦可讀媒體來分佈指令。因此，機器可讀媒體可包括用於儲存或傳輸為機器（例如，電腦）可讀的形式之資訊的任何機制，但並不限於軟碟、光碟、唯讀光碟機（CD-ROM）、及磁光碟機、唯讀記憶體（ROM）、隨機存取記憶體（RAM）、可抹除可程式化唯讀記憶體（EPROM）、電子可抹除可程式化唯讀記憶體（EEPROM）、磁或光卡、快閃記憶體、或用於經由電子、光、聲或其他形式之傳播信號（例如，載波、紅外線信號、數位信號等）在網際網路上傳輸資訊之有形的機器可讀儲存器。因此，電腦可讀媒體包括適用於儲存或傳輸為機器（例如，電腦）可讀的形式之電子指令或資訊的任何類型之有形的機器可讀媒體。

[0105] 下面的實例涉及依照本說明書的實施例。一或多更多實施例可提出一種設備、一種系統、一種機器可讀儲存器、一種機器可讀媒體、及一種方法，用以在一序列資料連結之一埠口識別出一封包，識別位於一序列資料連結之一埠口的一封包，判定封包與一錯誤相關，及基於判定封包與錯誤相關來啟動進入一錯誤復原模式中。進入錯誤復原模式中會使序列資料連結被強壓下。

[0106] 在至少一實例中，將資料連結強壓下使所有隨後的入站封包被停止。

[0107] 在至少一實例中，將資料連結強壓下更使所有未決的出站請求和完成被中止。

[0108] 在至少一實例中，錯誤邏輯係用以對被中止的完成之其一者產生並發送一主要中止完成。

[0109] 在至少一實例中，錯誤邏輯更用以停止封包。

[0110] 在至少一實例中，錯誤是一交易層封包錯誤、一流量控制協定錯誤、及一資料連結層協定錯誤之至少一者。

[0111] 在至少一實例中，錯誤邏輯更用以判定錯誤是用以觸發錯誤復原模式的一組錯誤之其一者。

[0112] 在至少一實例中，這組錯誤可從對應於錯誤復原模式的一錯誤遮罩暫存器識別出。

[0113] 在至少一實例中，這組錯誤更包括在被另一錯誤遮罩識別為不可校正錯誤的錯誤中。

[0114] 在至少一實例中，進入錯誤復原模式包括進入一去能狀態。

[0115] 在至少一實例中，錯誤與一不可校正嚴重性相關，且錯誤邏輯更用以迫使錯誤與一可校正嚴重性相關。

[0116] 在至少一實例中，錯誤邏輯更用以使錯誤被校正。

[0117] 在至少一實例中，錯誤邏輯更用以識別出資料連結上的所有交易已被停止，及識別出用於資料連結的所有錯誤暫存器已被清除。

[0118] 在至少一實例中，錯誤邏輯係用以基於一暫存器的一靜止控制位元值來判定交易已被停止。

[0119] 在至少一實例中，錯誤邏輯係基於識別出資料連結上的所有交易已被停止且識別出用於資料連結的所有錯誤暫存器已被清除來清除對應於資料連結之一狀態控制暫存器的一錯誤復原位元。

[0120] 在至少一實例中，錯誤邏輯更用以基於清除錯誤復原位元來重新訓練連結。錯誤復原位元能指出是否錯誤復原模式係有效的。

[0121] 在至少一實例中，進入錯誤復原模式係在與偵測錯誤之相同的時脈週期上被啟動。

[0122] 在至少一實例中，將資料連結強壓下包括進入一復原狀態，及從復原狀態轉成一去能狀態。

[0123] 在至少一實例中，強壓資料連結包括發送一

串訓練序列，且訓練序列包括指出嘗試進入去能狀態的位元。

[0124] 在至少一實例中，錯誤被指定為一不可校正錯誤，但錯誤被指定為一可校正錯誤，其中錯誤反而使一可校正錯誤中斷。

[0125] 在至少一實例中，封包是透過資料連結被發送至另一裝置的一出站封包。

[0126] 在至少一實例中，封包是透過資料連結從另一裝置接收的一入站封包。

[0127] 在至少一實例中，封包是透過一互連，且在一些情況下，是一周邊元件互連快捷（PCIe）相容互連來發送。

[0128] 在至少一實例中，封包係傳遞於一第一和第二微處理器之間。

[0129] 一或多個實施例可提出一種設備、一種系統、一種機器可讀儲存器、一種機器可讀媒體、及一種方法，用以與一序列資料連結連接，接收使用資料連結所傳遞的一封包，判定封包與一錯誤相關，及基於判定封包與錯誤相關來啟動進入一錯誤復原模式中。進入錯誤復原模式中會使一入站封包被停止和一出站請求被中止。

[0130] 在至少一實例中，介面邏輯係用以在錯誤復原模式期間中止所有未決的出站請求和完成。

[0131] 在至少一實例中，介面邏輯係用以在錯誤復原模式期間停止所有入站封包。

[0132] 一或更多實施例可提出一種設備、一種系統、一種機器可讀儲存器、一種機器可讀媒體、及一種方法，用以提供錯誤邏輯，具有交易層邏輯、資料連結層邏輯、及實體層邏輯，用以偵測與一特定封包相關的一特定錯誤，基於一第一暫存器值來判定特定錯誤被指定為一不可校正錯誤，基於一第二暫存器值來判定特定錯誤會觸發一錯誤復原模式，其中錯誤復原模式會使特定錯誤之後的入站和出站封包被停止，從一第三暫存器值判斷錯誤復原模式是否被致能，及基於至少部分之第一、第二、及第三暫存器值來服務特定錯誤。

[0133] 在至少一實例中，錯誤邏輯更用以基於判定第三暫存器值指出錯誤復原模式被致能來啟動錯誤復原模式，且啟動錯誤復原模式包括確立一暫存器的一錯誤復原狀態控制位元。

[0134] 在至少一實例中，錯誤邏輯更用以識別出錯誤復原狀態控制位元被清除，及基於清除錯誤復原狀態控制位元來重新啟動資料連結。

[0135] 在整篇本說明書中提到的「一個實施例」或「一實施例」係表示結合實施例所述之特定特徵、結構、或特性係包括在本發明之至少一個實施例中。因此，在整篇本說明書內各種地方出現之「在一個實施例中」或「在一實施例中」的說法不一定都指相同的實施例。再者，在一或更多實施例中，可以任何適當方式來結合特定特徵、結構、或特性。

[0136] 在上述說明書中，已針對特定示範實施例來提出詳細說明。然而，將清楚明白可在不脫離如在所附之申請專利範圍中所提出的發明之更廣精神和範圍下對其進行各種修改和變化。因此，本說明書及圖被視為說明性意義而不是限制性意義。再者，上述實施例及其他示範語言之使用不一定係指相同的實施例或相同的實例，而係可指不同和相異的實施例、以及可能相同的實施例。

【符號說明】

[0137]

100：處理器

101：核心

102：核心

101a：架構狀態暫存器

101b：架構狀態暫存器

102a：架構狀態暫存器

102b：架構狀態暫存器

130：分配器和更名器方塊

135：重排序/引退單元

120：指令轉譯緩衝器

151：低層資料快取和資料-TLB

140：執行單元

125：解碼模組

126：解碼器

110：晶片上介面

175：系統記憶體

105：匯流排

180：裝置

176：應用程式碼

177：碼

200：系統

205：處理器

210：系統記憶體

215：控制器集線器

206：前端匯流排

216：記憶體介面

219：序列連結

220：交換器/橋接器

217：埠口

221：埠口

225：I/O 裝置

230：圖形加速器

232：序列連結

231：I/O 模組

218：I/O 模組

300：分層協定堆疊

305：交易層

310：連結層

- 320：實體層
- 321：邏輯子方塊
- 322：電子子方塊
- 306：封包標頭/承載資料
- 311：封包序列識別符
- 312：CRC
- 323：訊框
- 400：交易描述符
- 402：全域識別符欄位
- 404：屬性欄位
- 406：通道識別符欄位
- 408：本地交易識別符欄位
- 410：來源識別符欄位
- 412：優先序欄位
- 414：保留欄位
- 416：排序欄位
- 418：無窺探欄位
- 505：裝置
- 510：裝置
- 506：傳輸邏輯
- 511：傳輸邏輯
- 507：接收邏輯
- 512：接收邏輯
- 516：路徑

- 517 : 路徑
518 : 路徑
519 : 路徑
605 : 能力結構
705 : 狀態暫存器
805 : 錯誤信號
810 : 錯誤狀態遮罩
815 : 信號
825 : AND 閘
820 : LER 錯誤遮罩
840 : 去能狀態
855 : 埠口靜止位元
910 : 全域錯誤產生邏輯
1005 : 端點
1010 : 實體層
1015 : 資料連結層
1020 : 交易層
1030 : 佇列
1105 : 壓封包
1110 : 積體 I/O 邏輯
1200 : 系統
1250 : 點對點互連
1270 : 第一處理器
1280 : 第二處理器

- 1272：積體記憶體控制器單元
- 1282：積體記憶體控制器單元
- 1276：P-P 介面
- 1278：P-P 介面
- 1286：P-P 介面
- 1288：P-P 介面
- 1232：記憶體
- 1234：記憶體
- 1294：對點介面電路
- 1298：對點介面電路
- 1252：P-P 介面
- 1254：P-P 介面
- 1290：晶片組
- 1292：介面電路
- 1239：高效能圖形互連
- 1238：高效能圖形電路
- 1296：介面
- 1216：第一匯流排
- 1214：I/O 裝置
- 1218：匯流排橋接器
- 1220：第二匯流排
- 1222：鍵盤/滑鼠
- 1227：通訊裝置
- 1230：碼和資料

1228：儲存單元

1224：音頻 I/O

1300：系統晶片

1306：核心

1307：核心

1309：匯流排介面單元

1312：L2 快取

1308：快取控制

1310：互連

1360：DRAM

1340：SDRAM 控制器

1365：快閃

1345：快閃控制器

1350：周邊控制

1320：視頻編解碼器

1325：視頻介面

1315：GPU

1370：藍芽模組

1375：3G 數據機

1380：GPS

1385：WiFi

申請專利範圍

1. 一種設備，包含：

錯誤邏輯，實施至少部分的硬體電路，用以

識別位於一序列資料連結之一埠口的一封包；

接收對應於該封包之一錯誤碼；

從該錯誤碼判定該封包與一錯誤之特定嚴重層級的關聯，其中該特定嚴重層級係使用復原模式來處理；

○ 判定該錯誤之該嚴重層級要從該特定嚴重層級改變為一較低嚴重層級，其中錯誤之該較低嚴重層級被校正代替進入該錯誤復原模式；以及

啟動進入用於其它錯誤之該特定嚴重層級之該錯誤復原模式中，其中進入該錯誤復原模式中會使相應的該序列資料連結被強壓下。

2. 如申請專利範圍第 1 項所述之設備，其中將該資料連結強壓下使所有隨後的入站封包被停止。

○ 3. 如申請專利範圍第 2 項所述之設備，其中將該資料連結強壓下更使所有未決的出站請求和完成被中止。

4. 如申請專利範圍第 3 項所述之設備，其中該錯誤邏輯更用以對該被中止的完成之其一者產生並發送一主要中止完成。

5. 如申請專利範圍第 1 項所述之設備，其中該錯誤邏輯更用以停止相應於該其它錯誤之該封包。

6. 如申請專利範圍第 1 項所述之設備，其中該錯誤是包含一交易層封包錯誤、一流量控制協定錯誤、及一資料

連結層協定錯誤之群組的至少一者。

7.如申請專利範圍第 1 項所述之設備，其中該錯誤邏輯更用以判定該錯誤是用以觸發該錯誤復原模式的一組錯誤之其一者。

8.如申請專利範圍第 7 項所述之設備，其中包括在該組錯誤之該錯誤包含比較該錯誤碼與一錯誤位元遮罩。

9.如申請專利範圍第 1 項所述之設備，其中判定該錯誤之該嚴重層級要從該特定嚴重層級改變為該較低嚴重層級係基於如果該錯誤碼具有相應於該錯誤復原模式之錯誤位元遮罩的比較。

10.如申請專利範圍第 1 項所述之設備，其中進入該錯誤復原模式包括進入一去能狀態。

11.如申請專利範圍第 1 項所述之設備，其中該特定嚴重層級錯誤與一不可校正嚴重性相關，且該錯誤邏輯更用以迫使該錯誤與一可校正嚴重性相關。

12.如申請專利範圍第 11 項所述之設備，其中該錯誤邏輯更用以使該錯誤被校正。

13.如申請專利範圍第 1 項所述之設備，其中該錯誤邏輯更用以識別出該資料連結上的所有交易已被停止，及識別出用於該資料連結的所有錯誤暫存器已被清除。

14.如申請專利範圍第 13 項所述之設備，其中該錯誤邏輯係用以基於一暫存器的一靜止控制位元值來判定該些交易已被停止。

15.如申請專利範圍第 13 項所述之設備，其中該錯誤

邏輯係基於識別出該強壓的資料連結上的所有交易已被停止且識別出用於該強壓的資料連結的所有錯誤暫存器已被清除來清除對應於該強壓的資料連結之一狀態控制暫存器的一錯誤復原位元。

16.如申請專利範圍第 15 項所述之設備，其中該錯誤邏輯更用以基於清除該錯誤復原位元來重新訓練 (initiate retraining) 該強壓的連結，其中該錯誤復原位元指出是否錯誤復原模式係有效的。

17.如申請專利範圍第 1 項所述之設備，其中進入該錯誤復原模式係在與偵測該錯誤之相同的一時脈週期上被啟動。

18.一種設備，包含：

邏輯，實施至少部分的硬體電路，用以

於一序列資料連結上接收一封包；

識別關聯於該封包之一錯誤碼；

從該錯誤碼判定該封包與一錯誤之特定嚴重層級的關聯，其中該特定嚴重層級係使用復原模式來處理；

判定該錯誤之該嚴重層級要從該特定嚴重層級改變為一較低嚴重層級，其中錯誤之該較低嚴重層級被校正代替進入該錯誤復原模式；以及

啟動進入用於其它錯誤之該特定嚴重層級之該錯誤復原模式中，其中進入該錯誤復原模式中會使該介面邏輯停止一入站封包和中止一出站請求在偵測該其它錯誤之後。

19.如申請專利範圍第 18 項所述之設備，其中該邏輯係用以在錯誤復原模式期間中止所有未決的出站請求和完成。

20.如申請專利範圍第 18 項所述之設備，其中該邏輯係用以在錯誤復原模式期間停止所有入站封包。

21.一種設備，包含：

錯誤邏輯，實施至少部分的硬體電路且包含交易層邏輯、資料連結層邏輯、及實體層邏輯，其中該錯誤邏輯係用以：

偵測與一特定封包相關的一特定錯誤；

基於一第一暫存器值來判定該特定錯誤被指定為一不可校正錯誤；

基於一第二暫存器值來判定該特定錯誤會觸發一錯誤復原模式，其中該錯誤復原模式會使該特定錯誤之後的入站和出站封包被停止；

基於判定該特定錯誤係用以觸發該錯誤復原模式，從一第三暫存器值判斷該錯誤復原模式是否被致能；及

基於至少部分之該第一、該第二、及該第三暫存器值來服務該特定錯誤。

22.如申請專利範圍第 21 項所述之設備，其中該錯誤邏輯更用以基於判定該第三暫存器值指出該錯誤復原模式被致能來啟動該錯誤復原模式，且啟動該錯誤復原模式包括確立一暫存器的一錯誤復原狀態控制位元。

23.如申請專利範圍第 22 項所述之設備，其中該錯誤邏輯更用以：

識別出該錯誤復原狀態控制位元被清除；及
基於清除該錯誤復原狀態控制位元來重新啟動該資料連結。

24.一種方法，包含：

識別位於一序列資料連結之一埠口的一封包；
接收對應於該封包之一錯誤碼；
從該錯誤碼判定該封包與一錯誤之特定嚴重層級的關聯，其中該特定嚴重層級係使用復原模式來處理；
判定該錯誤之該嚴重層級要從該特定嚴重層級改變為一較低嚴重層級，其中錯誤之該較低嚴重層級被校正代替進入該錯誤復原模式；以及

啟動進入用於其它錯誤之該特定嚴重層級之該錯誤復原模式中，其中進入該錯誤復原模式中會使相應的該序列資料連結被強壓下。

25.如申請專利範圍第 24 項所述之方法，更包含將相應於該其它之該序列資料連結強壓下，其中將該其它資料連結強壓下包括：

進入一復原狀態；及
從該復原狀態轉成一去能狀態。

26.如申請專利範圍第 25 項所述之方法，其中強壓該其它資料連結包括發送一串訓練序列，且該訓練序列包括指出嘗試進入該去能狀態的一位元。

27.如申請專利範圍第 24 項所述之方法，其中該封包是透過該資料連結被發送至另一裝置的一出站封包。

28.如申請專利範圍第 24 項所述之方法，其中該封包是透過該資料連結從另一裝置接收的一入站封包。

29.一種系統，包含：

一第一裝置；

一第二裝置，透過一互連來通訊地耦接至該第一裝置，該第二裝置包含錯誤邏輯，可執行以：

識別位於該互連的一序列資料連結之一埠口的一封包；

接收對應於該封包之一錯誤碼；

從該錯誤碼判定該封包與一錯誤之特定嚴重層級的關聯，其中該特定嚴重層級係使用復原模式來處理；

判定該錯誤之該嚴重層級要從該特定嚴重層級改變為一較低嚴重層級，其中錯誤之該較低嚴重層級被校正代替進入該錯誤復原模式；以及

在該埠口上啟動用於其它錯誤之該特定嚴重層級之一錯誤復原模式，其中該錯誤復原模式係用以控制該錯誤觸發該錯誤復原模式至該對應埠口。

30.如申請專利範圍第 29 項所述之系統，更包含一錯誤處理器。

31.如申請專利範圍第 29 項所述之系統，其中該互連包含一周邊元件互連快捷（PCIe）相容互連。

32.如申請專利範圍第 29 項所述之系統，其中該第一

I516915

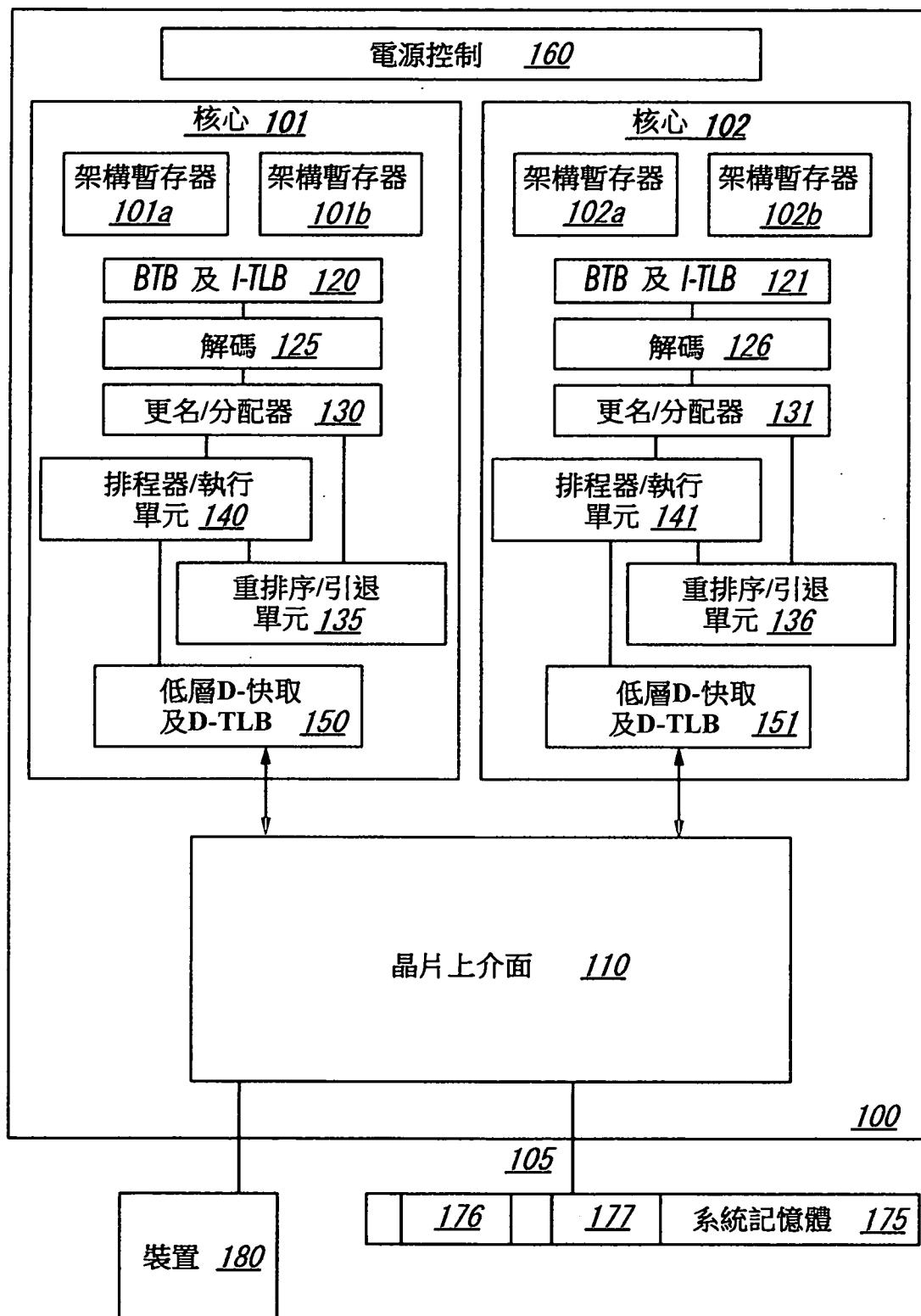
第 102146392 號

104.9.9.

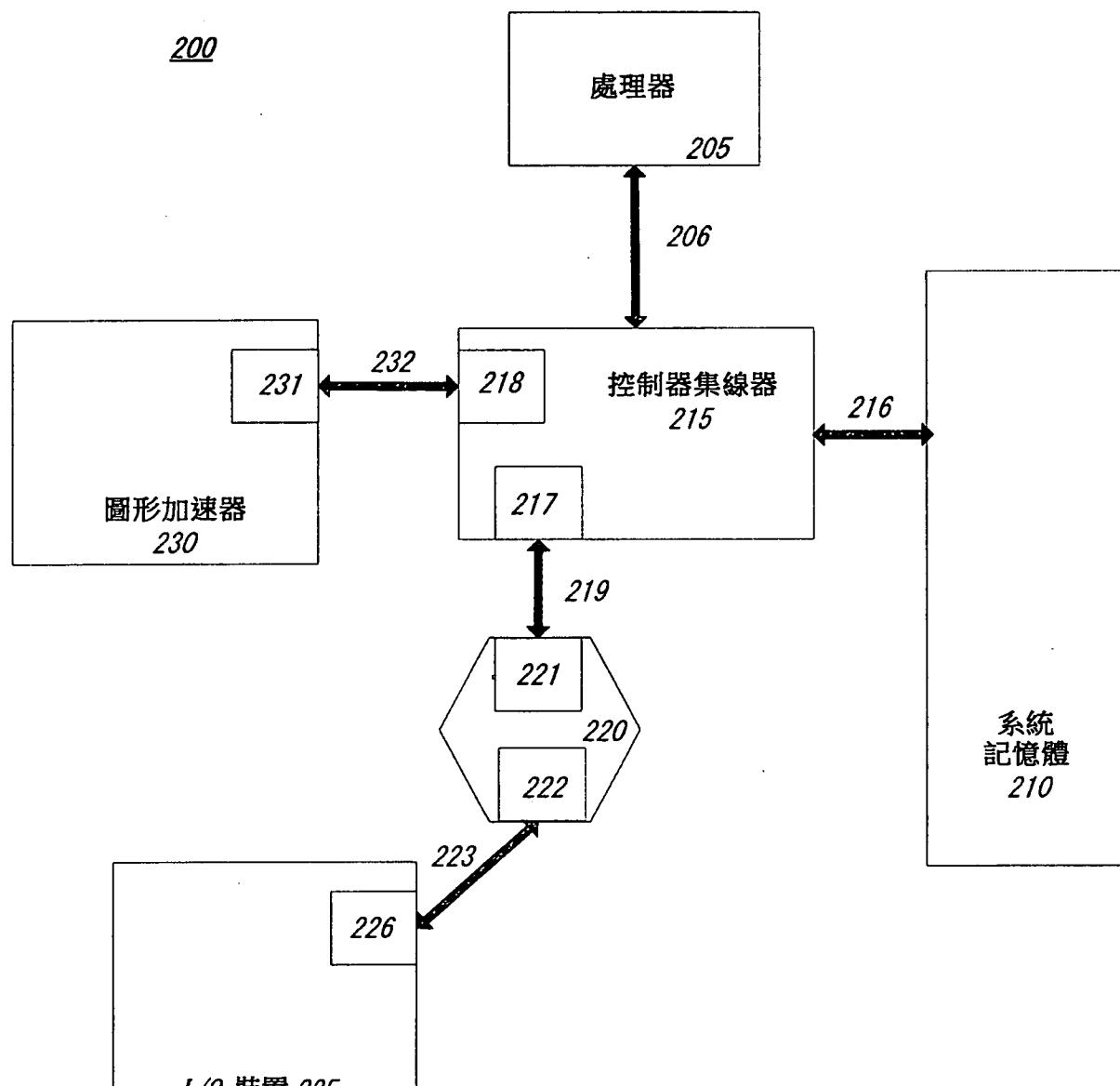
民國 104 年 9 月 9 日修正

裝置包含一第一微處理器且該第二裝置包含一第二微處理器。

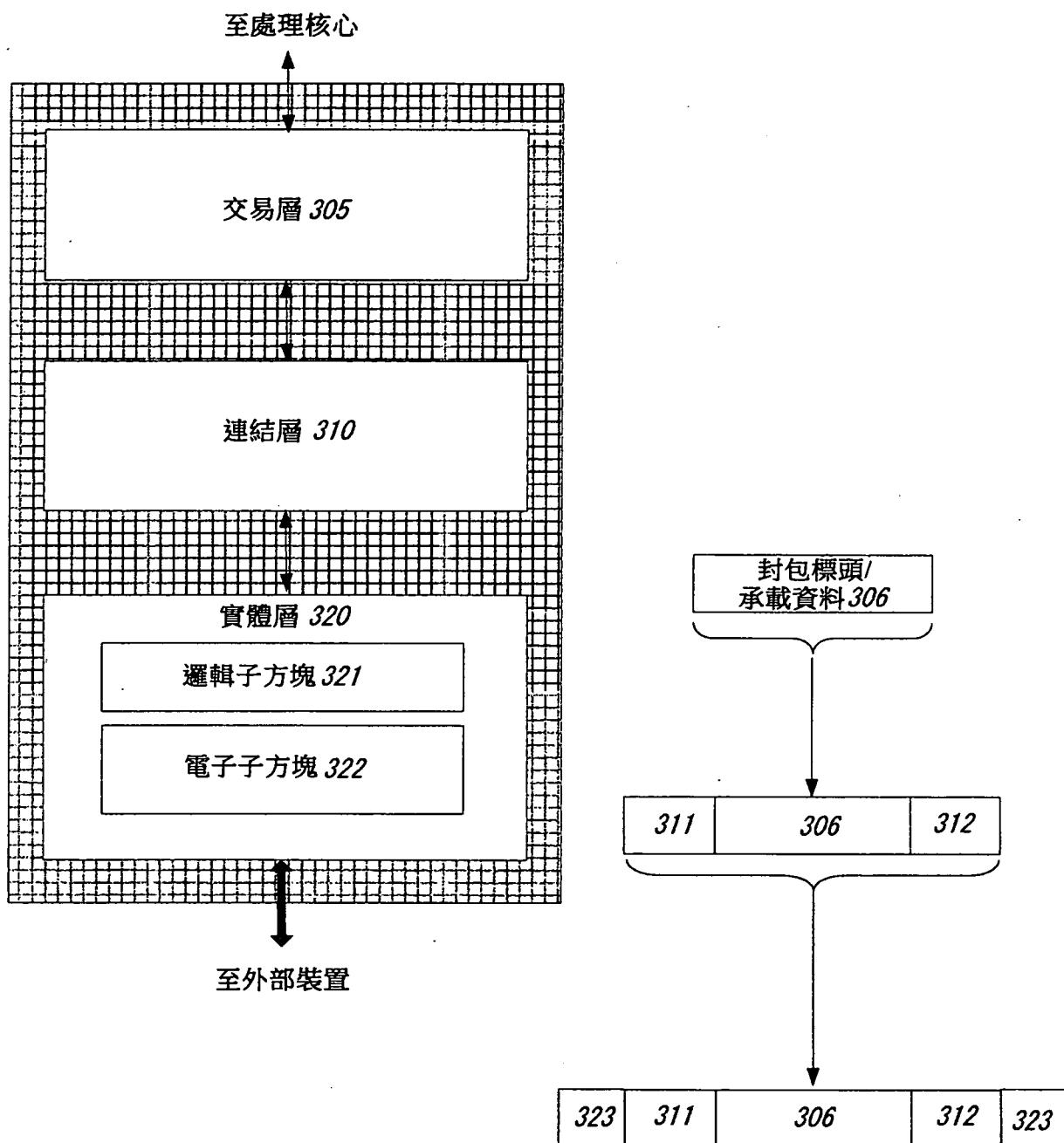
圖 式



第 1 圖



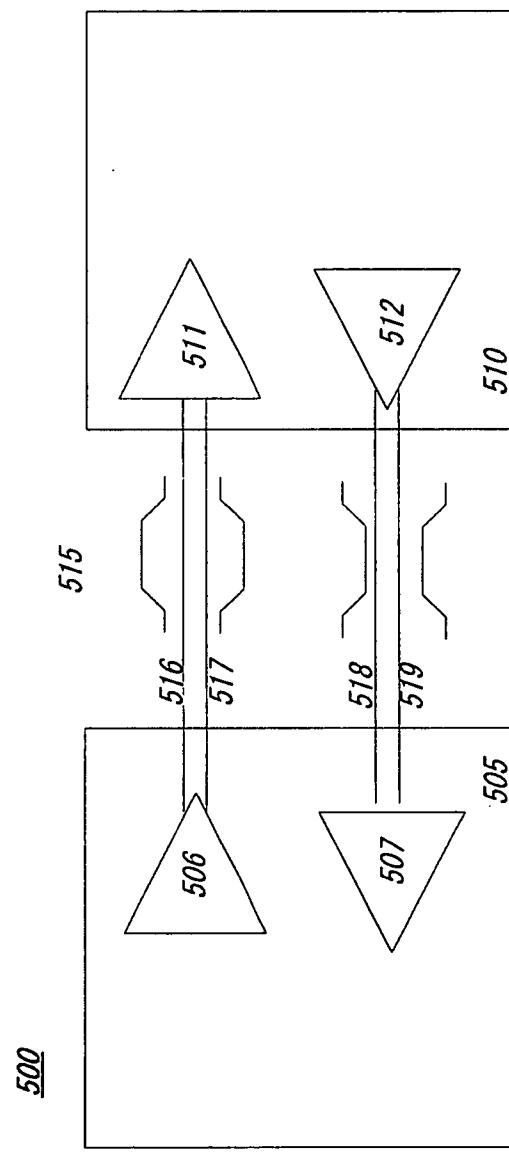
第 2 圖

300

第 3 圖

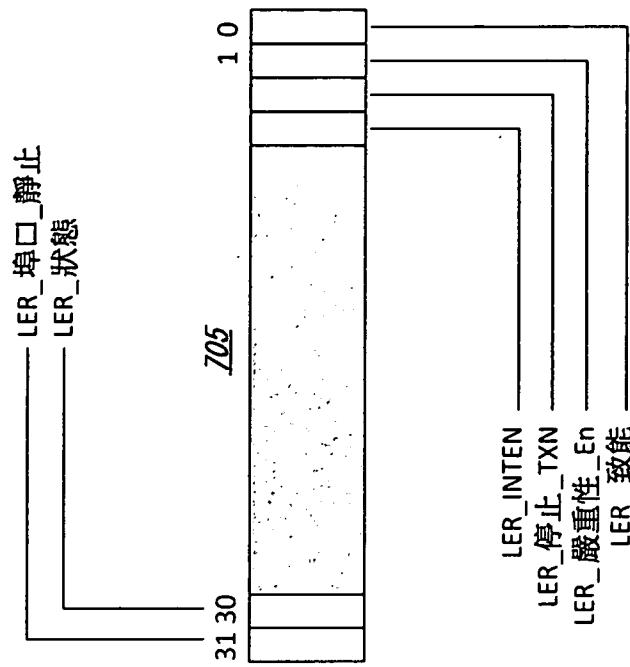
全域 ID 402		屬性位元 404			
本地交易 ID 408		來源 ID 410	優先序 412	保留的 414	排序 416
				無親探 418	通道 ID 406

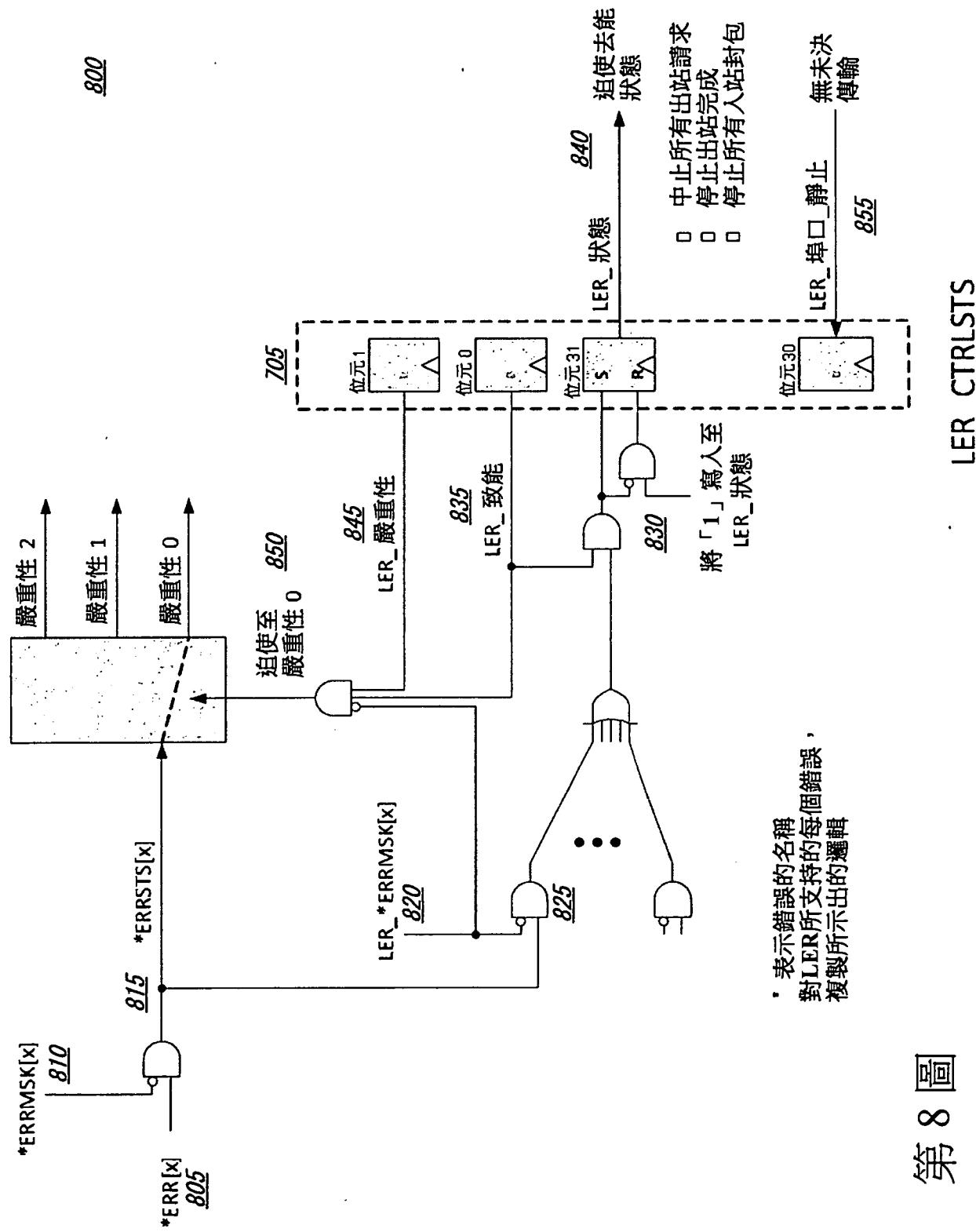
第 4 圖

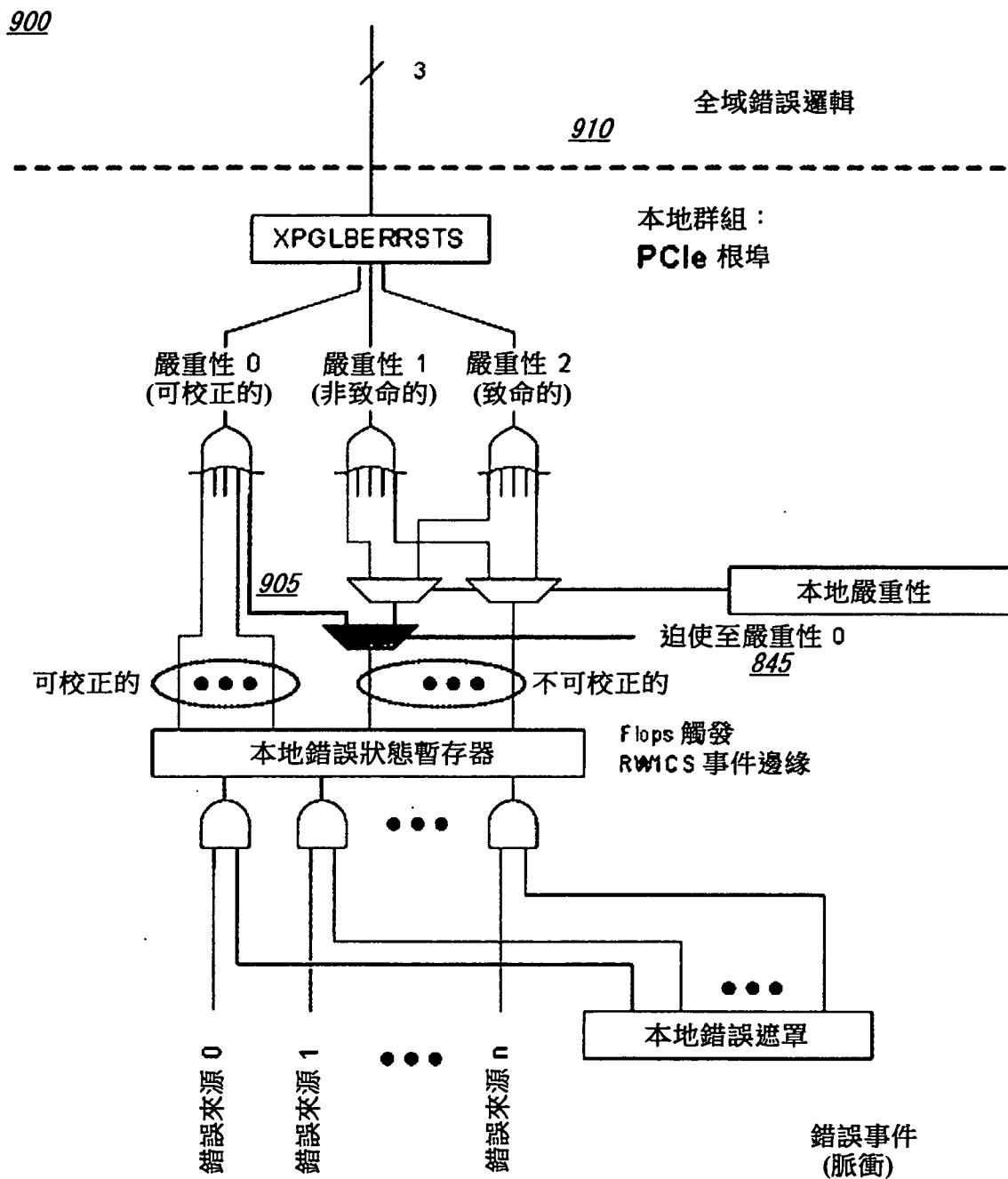


第 5 圖

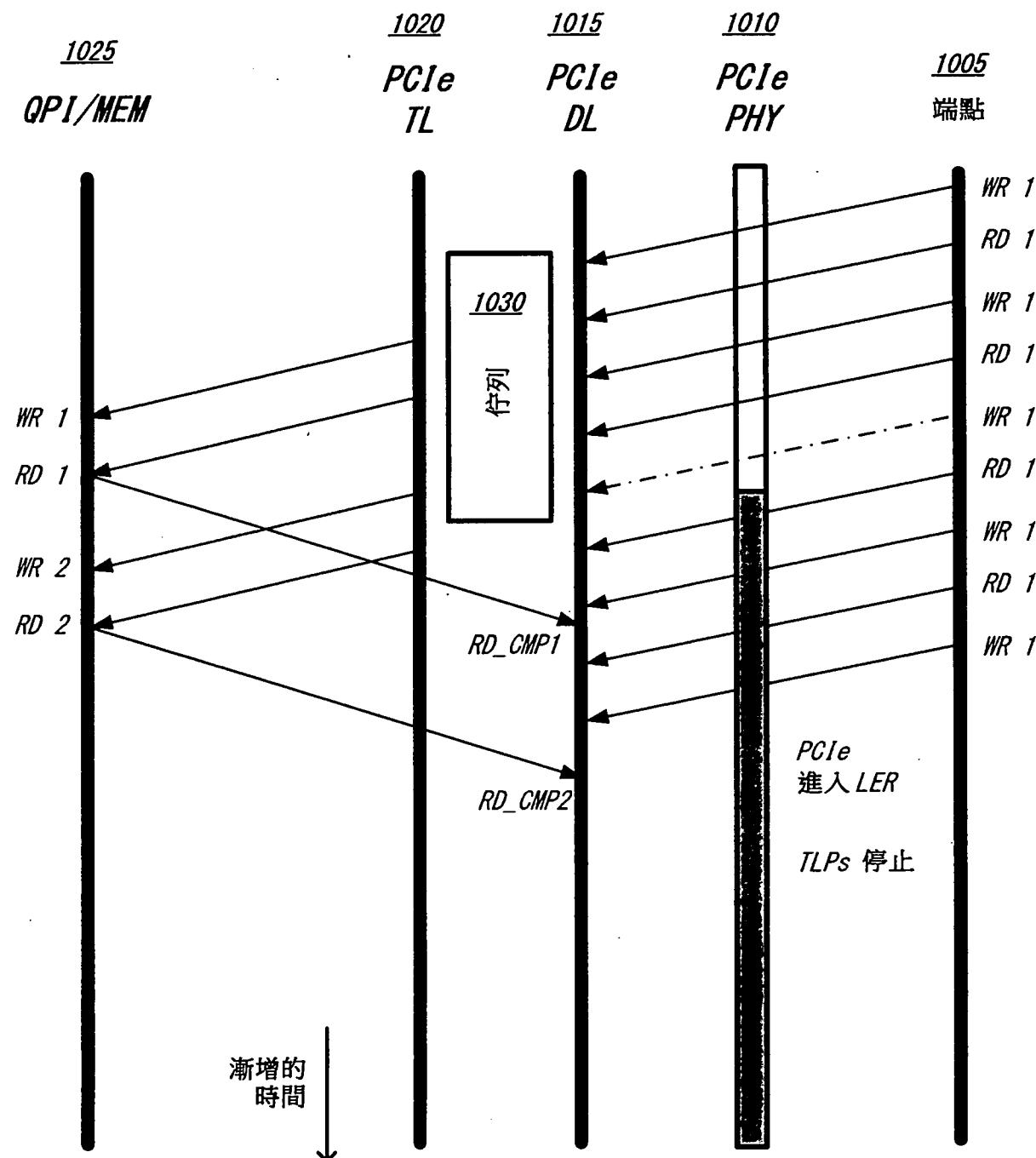
1516915



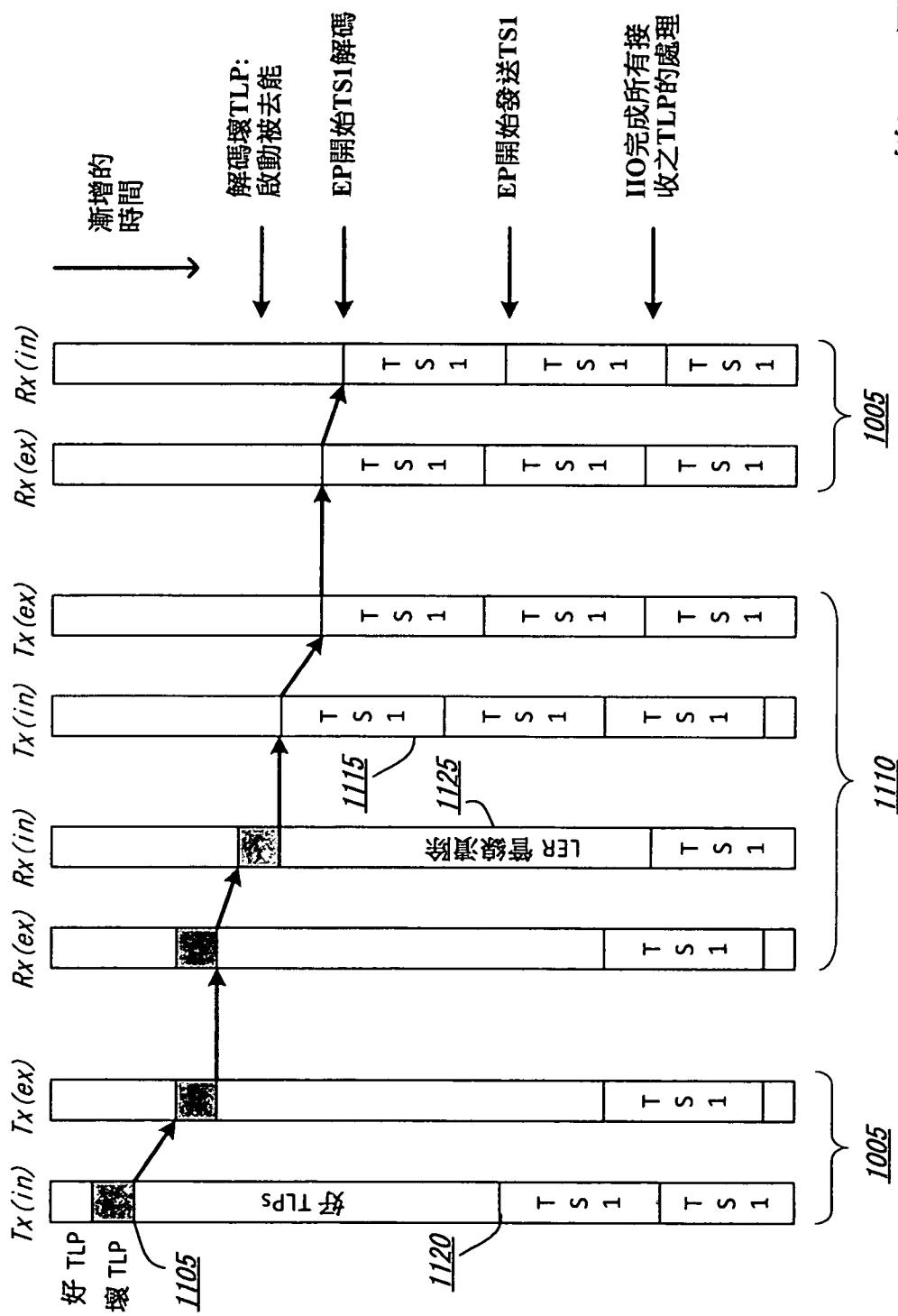




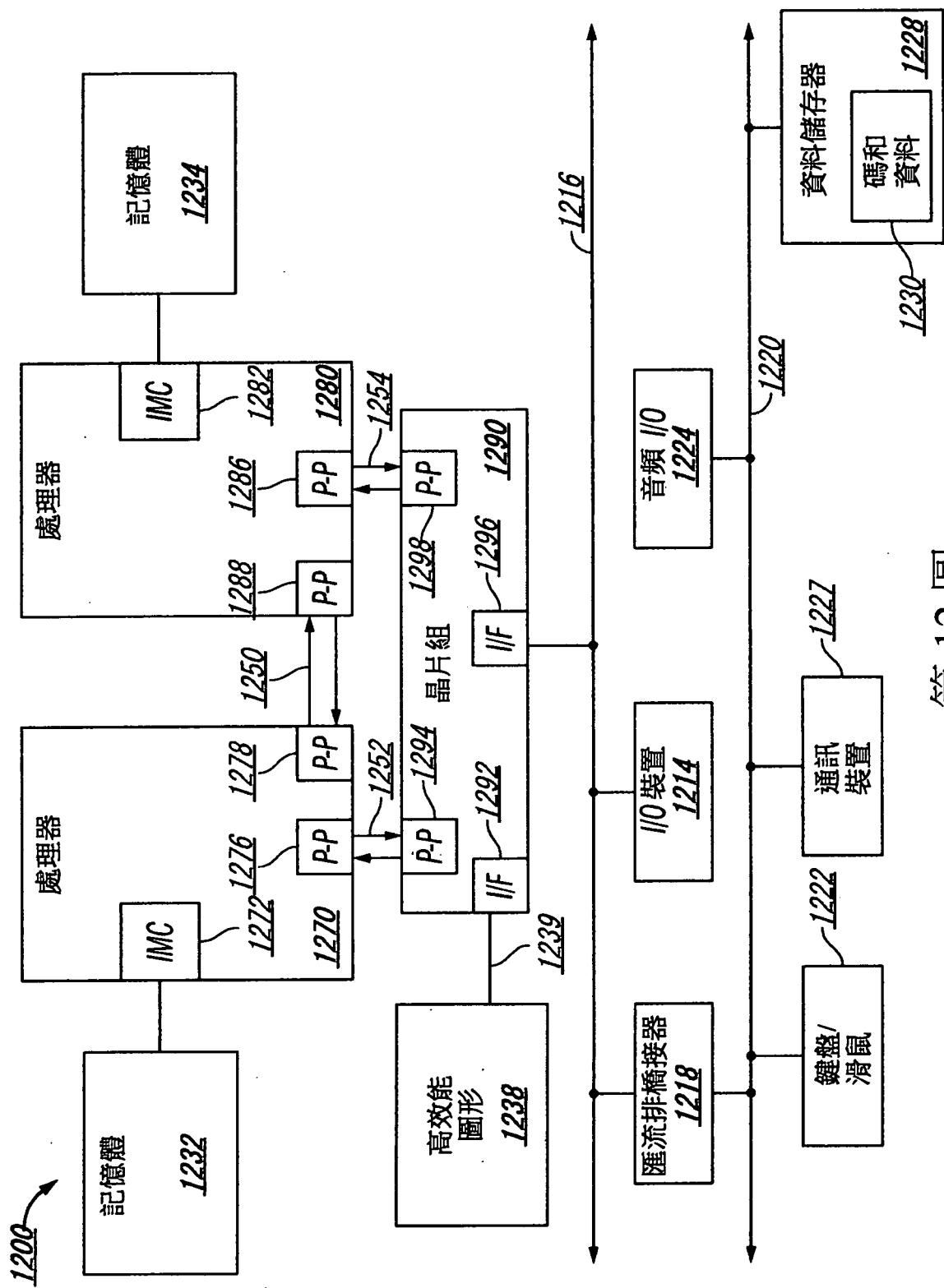
第 9 圖



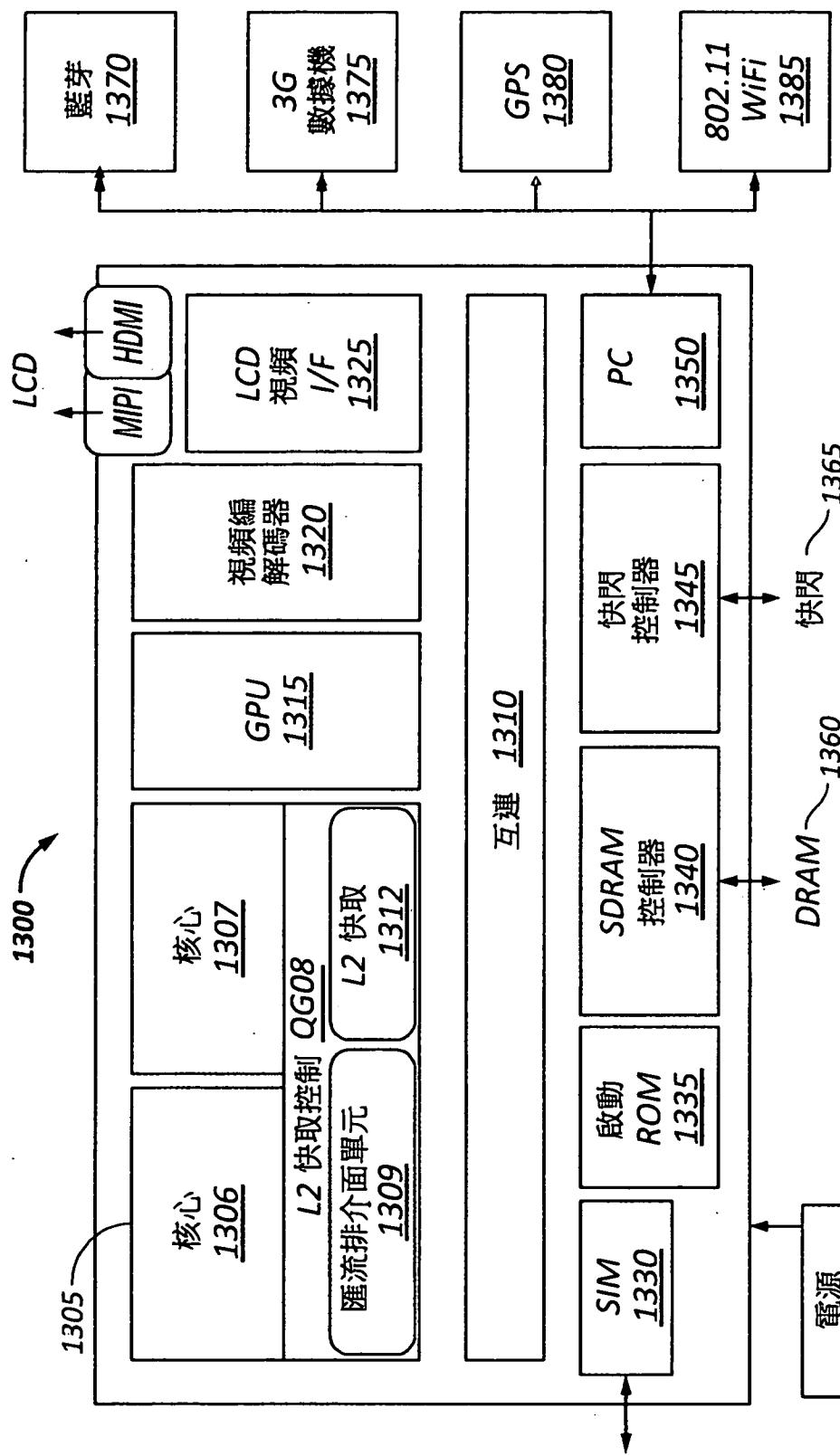
第 10 圖



第 11 圖



第 12 圖



第 13 圖