公告本

# 發明專利說明書

（填寫本書件時請先行詳閱申請書後之申請須知，作※記號部分請勿填寫）

※ 申請案號：92101470 ※IPC分類：G06F 13/16

※ 申請日期：92 1 23

## 壹、發明名稱

（中文）資料轉換機制

（英文）DATA TRANSFER MECHANISM

## 貳、發明人（共 4 人）

發明人 1 （如發明人超過一人，請填說明書發明人續頁）

姓名：（中文）吉伯 霍瑞曲

（英文）GILBERT WOLRICH

住居所地址：（中文）美國麻州法明漢市吉德明爾道 4 號

（英文）4 CIDER MILL ROAD, FRAMINGHAM, MASSACHUSETTS 01701, U.S.A.

國籍：（中文）美國 （英文）U.S.A.

## 參、申請人（共 1 人）

申請人 1 （如申請人超過一人，請填說明書申請人續頁）

姓名或名稱：（中文）美商英特爾公司

（英文）INTEL CORPORATION

住居所或營業所地址：（中文）美國加州聖塔卡拉瓦市米遜大學路 2200 號

（英文）2200 MISSION COLLEGE BOULEVARD, SANTA CLARA, CALIFORNIA 95052, U.S.A.

國籍：（中文）美國 （英文）U.S.A.

代表人：（中文）大衛 賽門

（英文）DAVID SIMON

發明人 2

姓名：（中文）馬克 B. 羅絲布曲

（英文）MARK B. ROSENBLUTH

住居所地址：（中文）美國麻州奧斯橋市癸斯耀道 4 號

（英文）4 CRESTVIEW DRIVE, UXBRIDGE, MASSACHUSETTS 01569, U.S.A.

國籍：（中文）美國 （英文）U.S.A.

發明人 3

姓名：（中文）戴伯拉 勃斯坦

（英文）DEBRA BERNSTEIN

住居所地址：（中文）美國麻州蘇貝瑞市比克漢道 443 號

（英文）443 PEAKHAM ROAD, SUDBURY, MASSACHUSETTS 01776, U.S.A.

國籍：（中文）美國 （英文）U.S.A.

發明人 4

姓名：（中文）馬修 J. 艾迪蘭達

（英文）MATTHEW J. ADILETTA

住居所地址：（中文）美國麻州活斯特市蒙坦羅道 20 號

（英文）20 MONTICELLO DRIVE, WORCESTER, MASSACHUSETTS 01603, U.S.A.

國籍：（中文）美國 （英文）U.S.A.

## 捌、聲明事項

☐ 本案係符合專利法第二十條第一項☐第一款但書或☐第二款但書規定之期間，其日期為：＿＿＿＿＿＿＿＿

☑ 本案已向下列國家（地區）申請專利，申請日期及案號資料如下：

【格式請依：申請國家（地區）；申請日期；申請案號　順序註記】

1.美國；2002 年 01 月 25 日；10/057,738

2.

3.

☑ 主張專利法第二十四條第一項優先權：

【格式請依：受理國家（地區）；日期；案號　順序註記】

1.美國；2002 年 01 月 25 日；10/057,738

2.

3.

4.

5.

6.

7.

8.

9.

10.

☐ 主張專利法第二十五條之一第一項優先權：

【格式請依：申請日；申請案號　順序註記】

1.

2.

3.

☐ 主張專利法第二十六條微生物：

☐國內微生物　【格式請依：寄存機構；日期；號碼　順序註記】

1.

2.

3.

☐國外微生物　【格式請依：寄存國名；機構；日期；號碼　順序註記】

1.

2.

3.

☐熟習該項技術者易於獲得，不須寄存。

# 玖、發明說明

(發明說明應敘明：發明所屬之技術領域、先前技術、內容、實施方式及圖式簡單說明)

## 技術領域

本發明係關於一種程式設計代理與記憶體資源間轉換資料之方法。

## 先前技術

典型電腦處理系統具有多個匯流排使各組件能夠相互通訊。這些組件間的匯流排通訊通常透過一資料路徑允許資料轉換。通常，該資料路徑內連接一處理代理例如一中央處理器(central processing unit；CPU)或處理器到其它組件如硬碟驅動機、裝置配接器等等。

## 發明內容

本發明揭示一種程式設計代理與記憶體資源間轉換資料之方法。該方法包括在一處理代理與一記憶體資源間轉換資料，指定該記憶體資源經由一推送匯流排將資料推入該處理代理，推送匯流排具複數個資料來源仲裁其使用，並指定該記憶體資源經由一拉取匯流排自該處理代理接收資料，拉取匯流排具複數個目的地仲裁其使用。

## 實施方式

結構：

參考圖1，一電腦處理系統10包括一平行硬體式多線(multithreaded)網路處理器12。該硬體式多線處理器12耦合一記憶體系統或記憶體資源14。記憶體系統14包括動態隨機存取記憶體(dynamic random access memory；DRAM) 14a及靜態隨機存取記憶體14b(static random access

memory；SRAM)。該處理系統10特別適用於能夠分割成數項平行子工作或功能的工作中。明確地說，該硬體式多線處理器12適用於頻寬導向而非等待時間(latency)導向的工作中。該硬體式多線處理器12有多個微引擎或程式設計引擎16，各具有多個硬體受控執行緒(thread)，其對一具體工作可同時啟動並獨立作業。

程式設計引擎16各維護一硬體程式計數器並保持與其相聯之狀態。相應的內容組或執行緒能有效地同時在各程式設計引擎16上啟動，但在任何時間中能實際工作的僅有一個。

在該範例中，八個程式設計引擎16在圖1內予以說明。每一程式設計引擎16皆有能力處理八項硬體執行緒或資料內容。該八個程式設計引擎16操作包括記憶體資源14和匯流排介面的共用資源。硬體式多線處理器12包括一動態隨機存取記憶體(dynamic random access memory；DRAM) 控制器18a及一靜態隨機存取記憶體(static random access memory；SRAM)控制器18b。DRAM記憶體14a及DRAM控制器18a通常係用於處理大量的資料，例如處理網路封包中的網路酬載。SRAM記憶體14b及SRAM控制器18b通常係使用於低等待時間、快速存取工作(例如存取核心處理器20 的查找表、記憶體)等的網路環境中。

推送匯流排26a-26b及拉取匯流排28a-28b係用於在該程式設計引擎16和該DRAM記憶體14a及SRAM記憶體14b間轉換資料。特定言之，該推送匯流排26a-26b係單向匯流排，把資料由記憶體資源14移動至程式設計引擎16

，而拉取匯流排28a-28b係把資料由程式設計引擎16移動至記憶體資源14。

八個程式設計引擎16根據資料特徵存取DRAM記憶體14a或SRAM記憶體14b。因此，低等待時間、低頻寬資料係儲存於SRAM記憶體14b之中，並且從SRAM記憶體14b中擷取，而等待時間較不重要的高頻寬資料則係儲存於DRAM 14a之中，並且從DRAM 14a中擷取。程式設計引擎16可對 DRAM控制器18a或SRAM控制器18b中任一方執行記憶體參考指令。

硬體式多線處理器12亦包含一核心處理器20，用以載入該硬體式多線處理器12其它資源的微碼控制。該範例中，核心處理器20係一基於XScale™之結構。

核心處理器20執行一般用途電腦型功能例如處理協定、異常及封裝處理的特殊支援，其中程式設計引擎16為更詳細之處理如邊界條件而終止該封裝。處理器核心20有一作業系統(未顯示)。透過該作業系統(operating system；OS)，處理器核心20能夠呼叫在程式設計引擎16上操作之功能。該核心處理器20能使用任何支援的OS，特定言之，一即時OS。對於以XScale™結構實施的核心處理器20而言，可以使用的作業系統係例如Microsoft NT即時、VXWorks和μCOS或網際網路上可用的免費OS。

藉由SRAM或DRAM記憶體存取方式便能夠解釋硬體多線設計的優點。例如，一從程式設計引擎16其中之一發出之資料內容(如Thread_0)SRAM存取請求會引起SRAM控

制器18b啟動一SRAM記憶體14b存取。SRAM控制器18b存取SRAM記憶體14b，從SRAM記憶體14b中擷取資料並把資料傳回一提出請求之程式設計引擎16。

在SRAM存取期間，如果程式設計引擎16之一僅有單條執行緒能夠運作的話，該微引擎便可能會停止運作，直到資料從該SRAM記憶體14b回傳為止。

藉由使用每個程式設計引擎16內的硬體式資料內容調換，該硬體式資料內容調換便能夠在同一個微引擎中執行其它的具有唯一程式計數器的資料內容。因此，當第一條執行緒(例如Thread_0)正在等待所讀取的資料進行回傳的時候，另一條執行緒(例如Thread_1)便能夠繼續作業。在執行期間，Thread_1可能會存取DRAM記憶體14a。當Thread_1在該DRAM單元上運作並且Thread_0在SRAM單元上運作時，一新執行緒如Thread_2現在則能在程式設計引擎16內運作。Thread_2可以運作一定時間，直到其需要存取記憶體或執行一些其它長等待時間作業，例如對一匯流排介面進行存取。所以，處理器12能夠利用一程式設計引擎16同時完成或進行匯流排運作、SRAM運作及DRAM運作，而且還有另一條執行緒可用以處理更多的工作。

硬體式資料內容調換亦可同時完成各項工作。舉例來說，可能會有兩條執行緒存取同一個共用資源，如SRAM記憶體14b。當完成其中一個微引擎執行緒或資料內容所請求的工作之後，每個分離的功能性單元(例如，SRAM控制器18b及DRAM控制器18a)便會回報一完成作業的旗標信

號。當程式設計引擎16收到該旗標，其能決定該接通哪一條執行緒。

硬體式的多線處理器12的其中一種應用實例為網路處理器。以網路處理器來說，硬體式的多線處理器12會介接各種如媒體存取控制器(Media Access Controller；MAC)裝置(例如10/100 BaseT八進位媒體存取控制器13a或Gigabit乙太裝置13b)(未顯示)般的網路裝置。通常，以網路處理器來說，硬體式的多線處理器12能夠介接任何類型收發大量資料的通訊裝置或介面。在網路應用中作業的電腦處理系統10能夠接收網路封包並以平行方式處理那些封包。

微引擎資料內容：

如上所述，每一程式設計引擎16支援八項資料內容的多線執行。這允許一執行緒只在另一執行緒發出一記憶體參考後開始執行，並且必須等待其完成後再做更多工作。多線執行對維持程式設計引擎16的有效硬體執行很關鍵，因為記憶體等待時間極其重要。多線執行允許程式設計引擎16藉由越過數幾個執行緒執行有用的獨立工作來隱藏記憶體等待時間。

程式設計引擎16八項資料內容的每一個，允許有效的資料內容調換，具有其暫存器集、程式計數器和資料內容具體局部暫存器。每一資料內容具有一副本，其消除為每一資料內容調換而移動資料內容具體資訊至共用記憶體及微引擎暫存器，並將其從共用記憶體及微引擎暫存器取出

之需要。快速資料內容調換允許一資料內容執行計算同時其它資料內容等待輸入-輸出(I/O)完成，通常係完成外部記憶體存取，或等待來自另一資料內容或硬體單元之信號。

例如，程式設計引擎16之八項資料內容藉維持八個程式計數器及八個相應資料內容暫存器集來執行。許多不同型號的資料內容對應暫存器，如一般用途暫存器(general purpose registers；GPR)、程式設計內代理暫存器、靜態隨機存取記憶體(SRAM)輸入轉換暫存器、動態隨機存取記憶體(DRAM)輸入轉換暫存器、SRAM輸出轉換暫存器、DRAM輸出轉換暫存器。亦可使用局部記憶體暫存器。

例如，GPR用於一般程式設計用途。GPR專門在程式控制下進行讀取和寫入。當該GPR在一指令中作為一資料來源時，其對一執行資料路徑(未顯示)提供運算元。當該GPR在一指令中作為一目的文件時，其對執行盒資料路徑之結果進行寫入。程式設計引擎16亦包括上面討論過的IO轉換暫存器。該IO轉換暫存器用以在程式設計引擎16及程式設計引擎16之外部區域(如DRAM記憶體14a和SRAM記憶體14b等)轉換資料。

匯流排結構：

參考圖2，硬體式多線處理器12將予以更詳細地說明。DRAM記憶體14a和SRAM記憶體14b各自連接DRAM記憶體控制器18a和SRAM記憶體控制器18b。DRAM控制器18a連接一拉取匯流排仲裁器30a和一推送匯流排仲裁器32a，其皆耦合一程式設計引擎16a。SRAM控制器18b連接一

拉取匯流排仲裁器30b和一推送匯流排仲裁器32b，其皆耦合一程式設計引擎16b。匯流排26a-26b和28a-28b組成主匯流排，用於在程式設計引擎16a-16b和DRAM記憶體14a及SRAM記憶體14b間轉換資料。從任一程式設計引擎16a-16b中發出之任一執行緒皆可存取DRAM控制器18a和SRAM控制器18b。

特定言之，推送匯流排26a-26b有多個記憶體源例如記憶體控制器通道及內部讀取暫存器(未顯示)，其藉由推送仲裁器32a-32b仲裁使用推送匯流排26a-26b。任一推送資料轉換之目的地(例如程式設計引擎16)當資料「推入」進來時藉由對Push_ID解碼來辯識，Push_ID和推送資料一起驅動或傳送。拉取匯流排28a-28b亦有多個目的地(例如寫入資料到不同記憶體控制器通道或可寫內部暫存器)來仲裁使用拉取匯流排28a-28b。拉取匯流排28a-28b有一Pull_ID，其例如早於拉取資料兩周期驅動或傳送。

資料功能分佈於程式設計引擎16之中。DRAM記憶體14a和SRAM記憶體14b的連接經由命令請求來執行。一命令請求可以係一記憶體請求。舉例來說，一命令請求可將資料從一位於程式設計引擎16a中的暫存器移動至一共用資源，如DRAM記憶體14a和SRAM記憶體14b。該些命令或請求會被外送至每個功能性單元及共用資源。命令例如I/O命令(如SRAM讀取、SRAM寫入、DRAM讀取、DRAM寫入、從一接收記憶體緩衝器載入資料、移動資料至傳送記憶體緩衝器)指定資料內容對應來源或目的暫存器至程

式設計引擎16中。

通常，資料在微引擎和記憶體資源間轉換，指定該記憶體資源藉該推送匯流排將資料推入一處理代理，推送匯流排具複數個資料來源仲裁其使用，並指定該記憶體資源經由一拉取匯流排自該處理代理接收資料，拉取匯流排具複數個目的地仲裁其使用。

讀取過程：

參考圖3，推送匯流排26a-26b在程式設計引擎16一讀取階段過程中執行一資料讀取過程50。作為讀取處理50之部分該微引擎執行(52)一資料內容。該程式設計引擎16發出(54)一讀取命令到記憶體控制器18a-18b，記憶體控制器18a-18b為記憶體資源(DRAM記憶體14a或SRAM記憶體14b)之一處理(56)該請求。對於讀取命令，在讀取命令發出(54)後，若需要讀取資料則程式設計引擎16核對(58)並繼續程式資料內容。若需要讀取資料繼續程式資料內容或執行緒，則將資料內容調換出去(60)。程式設計引擎16核對(62)確保記憶體控制器18a-18b已完成該請求。當記憶體控制器已完成該請求，資料內容就調換回來(64)。

若不需要該請求繼續資料內容執行，則程式設計引擎16核對(68)記憶體控制器18a-18b是否已完成該命令。若記憶體控制器18a-18b還未完成該請求，則發生一回路返回並且進一步核對(58)。若記憶體控制器18a-18b已完成該請求，當已由記憶體資源處獲得該讀取資料時，則記憶體控制器18a-18b推送(70)該資料進入讀取命令指定之資料

內容相應輸入轉換暫存器。記憶體控制器設置一信號在程式設計引擎16，其可啟動發出讀取的資料內容。程式設計引擎16在輸入轉換暫存器中讀取(72)請求資料並繼續(74)資料內容執行。

寫入處理：

參考圖4，推送匯流排28a-26b在程式設計引擎16一寫入階段過程中執行一資料寫入處理80。在寫入處理80中，微引擎執行(82)一資料內容。程式設計引擎16載入(84)資料到輸出轉換暫存器並發出(86)一寫入命令或請求到記憶體控制器18a-18b。該輸出轉換暫存器設置(88)為一唯讀狀態。對於程式設計引擎16的寫入命令，在該輸出轉換暫存器設置(88)為一唯讀狀態後，程式設計引擎16核對(90)是否需要該請求繼續程式資料內容或執行緒。若是，則將該資料內容調換出去(92)。

若不需要該請求繼續程式資料內容或執行緒，則記憶體控制器18a-18b從輸出轉換暫存器擷取或拉取(94)該資料並發信號(96)至程式設計引擎16以解鎖該輸出轉換暫存器。程式設計引擎16然後核對(98)該資料內容是否調換出去。若是，則將該資料調換回來(100)，若不是則程式設計引擎16繼續(102)資料內容的執行。這樣發信號的資料內容可以重新使用輸出轉換暫存器。該信號亦可能用於啟動資料內容若其已依寫入命令調換出(100)。

資料推送運作：

參考圖5，顯示在不同處理周期中，如周期0至周期5中

，電腦處理系統10之推送匯流排26a-26b內發生一資料推送運作110的情形。每個目標如DRAM記憶體14a或SRAM記憶體14b，傳送或驅動(112)一Target_#_Push_ID至推送仲裁器，其中#指示不同資料內容序號例如資料內容#0至資料內容#7。該Target_#_Push_ID來自讀取命令及一用以推入推送仲裁器32a-32b之資訊資料錯誤位元(如緊接目標的序號代表Push_ID之源位址增量)。對於Push_ID，每個字母指示一具體目的地之推送運作。一「空」Push_ID目的地指示該Push_ID無效。該目標亦將Target_#_Push_Data傳送至推送仲裁器。

Push_ID及Push_Data在推送仲裁器32a-32b之先進先出(first-in, first-outs；FIFO)內暫存(114)並進入佇列(116)，除非Target_#_Push_Q_Full信號已判定。該信號指示該具體目標的Push_ID及Push_Data FIFO幾乎充滿推送仲裁器32a-32b。此情形下，推送仲裁器32a-32b還未暫存一Push_ID或Push_Data，目標不會改變之。通道藉由推送仲裁器32a-32b將Push_ID及Push_Data變為下一個字轉換的Push_ID及Push_Data，若無其它有效轉換則變為無效。由於該Push_Q_Full信號內的等待時間，推送仲裁器32a-32b應該可應付每個目標執行中Push_ID及Push_Data最壞情形數量。

推送仲裁器32a-32b將在全部有效Push_ID間仲裁(118)每個周期，並傳送中間Push_ID。仲裁規則可以係迴圈式、優先配置甚或係可程式編輯。由推送仲裁器32a-32b到

目的地的資料多重推送不保障係在連序周期內。推送仲裁器32a-32b傳送(12)中間Push_Data並且發送(120) Push_ID到目的地。直到目標更新每個Push_ID的目的地位址才發送出希望推送資料的每個字。該Push_Data發送(122)到目的地。在目的地，從目的地得到Push_ID至目的地得到Push_Data的時間以一處理周期來固定。

資料拉取運作：

參考圖6，顯示在不同處理周期(如周期0至周期7)中，電腦處理系統10之拉取匯流排28a-28b內發生一資料拉取運作130的情形。每個目標如DRAM記憶體14a或SRAM記憶體14b，傳送或驅動(132)全Target_#_Pull_ID(即緊接目標的序號代表Pull_ID之源位址增量)及將要拉取資訊之長度(來自寫入命令)至目標。對於Pull_IDs，每個字母指示一來自一具體來源之拉取運作，例如記憶體資源14。一「空」Pull_ID目的地指示該Pull_ID無效。目標在判定其Pull_ID時必須有該拉取資料可用的緩衝空間。

Pull_ID在拉取仲裁器30a-30b之先進先出(first-in, first-outs；FIFO)內暫存(134)並進入佇列(136)，除非Target_#_Pull_Q_Full信號已判定。該信號指示該具體目標的Pull_ID佇列幾乎充滿拉取仲裁器30a-30b。此情形下，拉取仲裁器30a-30b還未暫存該Pull_ID且不會被目標改變。目標將拉取仲裁器30a-30b接收的一Pull_ID變為下一叢集轉換的那個，若無其它有效Pull_ID則變為無效。由於該Pull_Q_Full信號內的等待時間，拉取仲裁器30a-30b

應該可應付每個目標執行中Pull_ID的最壞情形數量。

拉取仲裁器30a-30b在全部當前有效Pull_ID中仲裁(138)每個周期。仲裁規則可以係迴圈式、優先配置甚或係可程式編輯。

拉取仲裁器30a-30b發送(140)所選Pull_ID至資料來源。從拉取仲裁器30a-30b傳送Pull_ID至資料來源提供資料的時間固定為三個處理周期。拉取仲裁器30a-30b為每一個新資料項更新Pull_ID的「源位址」欄。Pull_Data從資料來源拉出(142)並傳送至目標。

拉取仲裁器30a-30b亦判定(146)一Target_#_Take_Data至所選目標。該信號為每一周期判定，周期係一資料有效字傳送至該目標。不過，該判定不保障係在連序處理周期內。拉取仲裁器30a-30b每次最多只判定一項Target_#_Take_Data信號。

為在不同匯流排寬度下於目標與主控程式(master)間轉換，需要拉取仲裁器30a-30b做調整。例如，DRAM控制器18b可能每處理周期接受八位元資料而程式設計引擎16每周期只傳送四位元。此情形下，可用拉取仲裁器30a-30b每處理周期接受四位元，將其合併封裝為八位元，並傳送該資料至DRAM控制器18a。

其它具體實施例：

應瞭解，雖然上述範例已藉由其詳細說明進行了闡述，但以上說明係用以解說本發明範疇而非限制本發明範疇，其由隨附的申請專利範圍之範疇定義。其它觀點、優點

及修改都屬於下列申請專利範圍的範疇內。

## 圖式簡單說明

圖1係一處理系統的方塊圖。

圖2係圖1之處理系統的詳細方塊圖。

圖3係圖1之處理系統內一讀取過程的流程圖。

圖4係圖1之處理系統內一寫入處理的流程圖。

圖5係圖1之處理系統內一推送過程的流程圖。

圖6係圖1之處理系統內一拉取過程的流程圖。

## 圖式代表符號說明

| | |
|---|---|
| 0 | 週期 |
| 5 | 週期 |
| 7 | 週期 |
| 10 | 電腦處理系統 |
| 12 | 多線網路處理器 |
| 14 | 記憶體資源 |
| 16 | 微引擎或程式設計引擎 |
| 20 | 核心處理器 |
| 50 | 資料讀取處理過程 |
| 52 | 執行 |
| 54 | 發出 |
| 56 | 處理 |
| 58 | 核對 |
| 60 | 調換出去 |
| 62 | 核對 |

64 調換回來

68 核對

70 推送

72 讀取

74 繼續

80 資料寫入處理

82 執行

84 載入

86 發出

88 設置

90 核對

92 調換出去

94 擷取或拉取

96 發信號

98 核對

100 調換回來

102 繼續

110 資料推送運作

112 傳送或驅動

120 發送

122 發送

130 資料拉取運作

132 傳送或驅動

134 暫存

| | |
|---|---|
| 136 | 進入佇列 |
| 138 | 發送 |
| 142 | 拉出 |
| 146 | 判定 |
| 13a | 八進位媒體存取控制器 |
| 13b | 乙太裝置 |
| 14a | 動態隨機存取記憶體 |
| 14b | 靜態隨機存取記憶體 |
| 16a | 程式設計引擎 |
| 16b | 程式設計引擎 |
| 18a | 動態隨機存取記憶體控制器 |
| 18b | 靜態隨機存取記憶體控制器 |
| 26a-26b | 推送匯流排 |
| 28a-28b | 拉取匯流排 |
| 30a | 拉取匯流排仲裁器 |
| 30b | 拉取匯流排仲裁器 |
| 32a | 推送匯流排仲裁器 |
| 32b | 推送匯流排仲裁器 |

## 肆、中文發明摘要

本發明揭示一種程式設計代理與記憶體資源間轉換資料之方法。該方法包括在一處理代理與一記憶體資源間轉換資料，指定該記憶體資源經由一推送匯流排將資料推入該處理代理，推送匯流排具複數個資料來源仲裁其使用，並指定該記憶體資源經由一拉取匯流排自該處理代理接收資料，拉取匯流排具複數個目的地仲裁其使用。

## 伍、英文發明摘要

A method for transferring data between programming agents and memory resources. The method includes transferring data between a processing agent and a memory resource, designating the memory resource for pushing the data to the processing agent via a push bus having a plurality of sources that arbitrate use of the push bus, and designating the memory resource for receiving the data from the processing agent via a pull bus having a plurality of destinations that arbitrate use of the pull bus.

陸、(一)、本案指定代表圖為：第 2 圖

(二)、本代表圖之元件代表符號簡單說明：

| | |
|---|---|
| 10 | 電腦處理系統 |
| 12 | 多線網路處理器 |
| 14 | 記憶體資源 |
| 14a | 動態隨機存取記憶體 |
| 14b | 靜態隨機存取記憶體 |
| 16a | 程式設計引擎 |
| 16b | 程式設計引擎 |
| 18a | 動態隨機存取記憶體控制器 |
| 18b | 靜態隨機存取記憶體控制器 |
| 26a-26b | 推送匯流排 |
| 28a-28b | 拉取匯流排 |
| 30a | 拉取匯流排仲裁器 |
| 30b | 拉取匯流排仲裁器 |
| 32a | 推送匯流排仲裁器 |
| 32b | 推送匯流排仲裁器 |

柒、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

# 拾壹、圖式

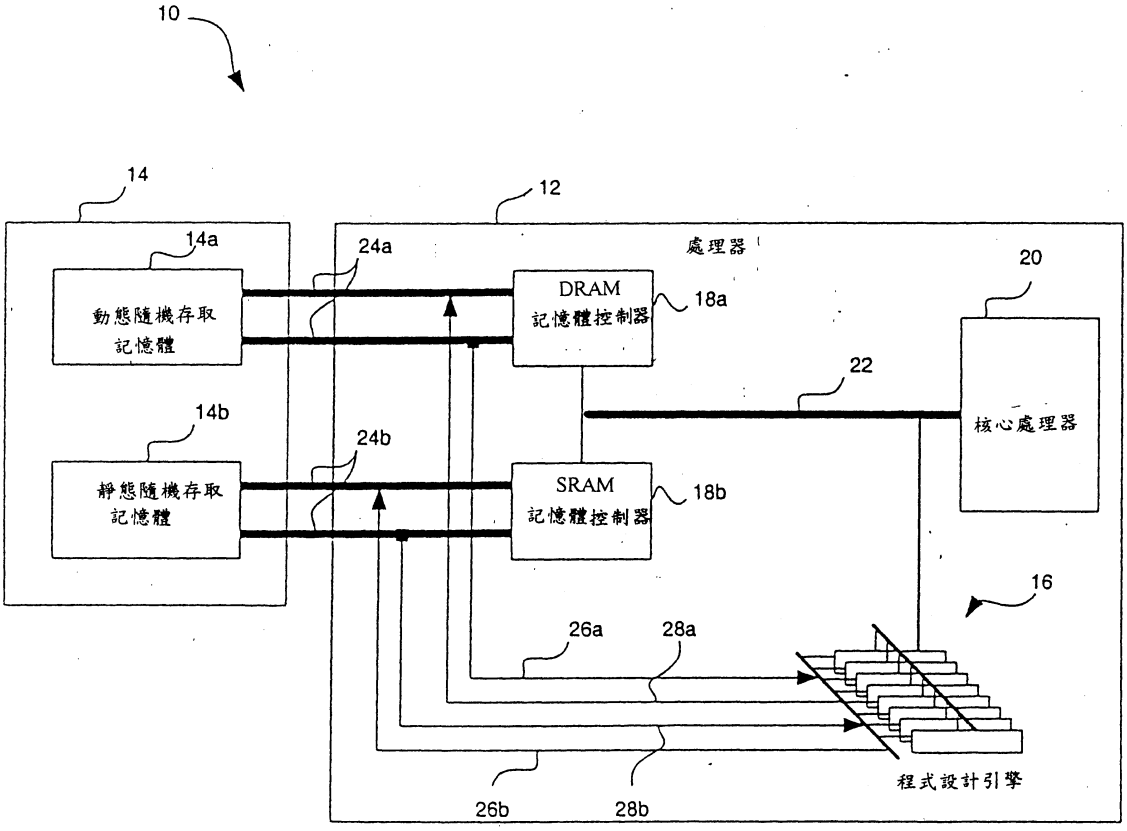

10
14
12
14a
24a
處理器
動態隨機存取
記憶體
DRAM
記憶體控制器
18a
20
22
核心處理器
14b
24b
靜態隨機存取
記憶體
SRAM
記憶體控制器
18b
16
26a
28a
程式設計引擎
26b
28b

圖 1

圖式簡單

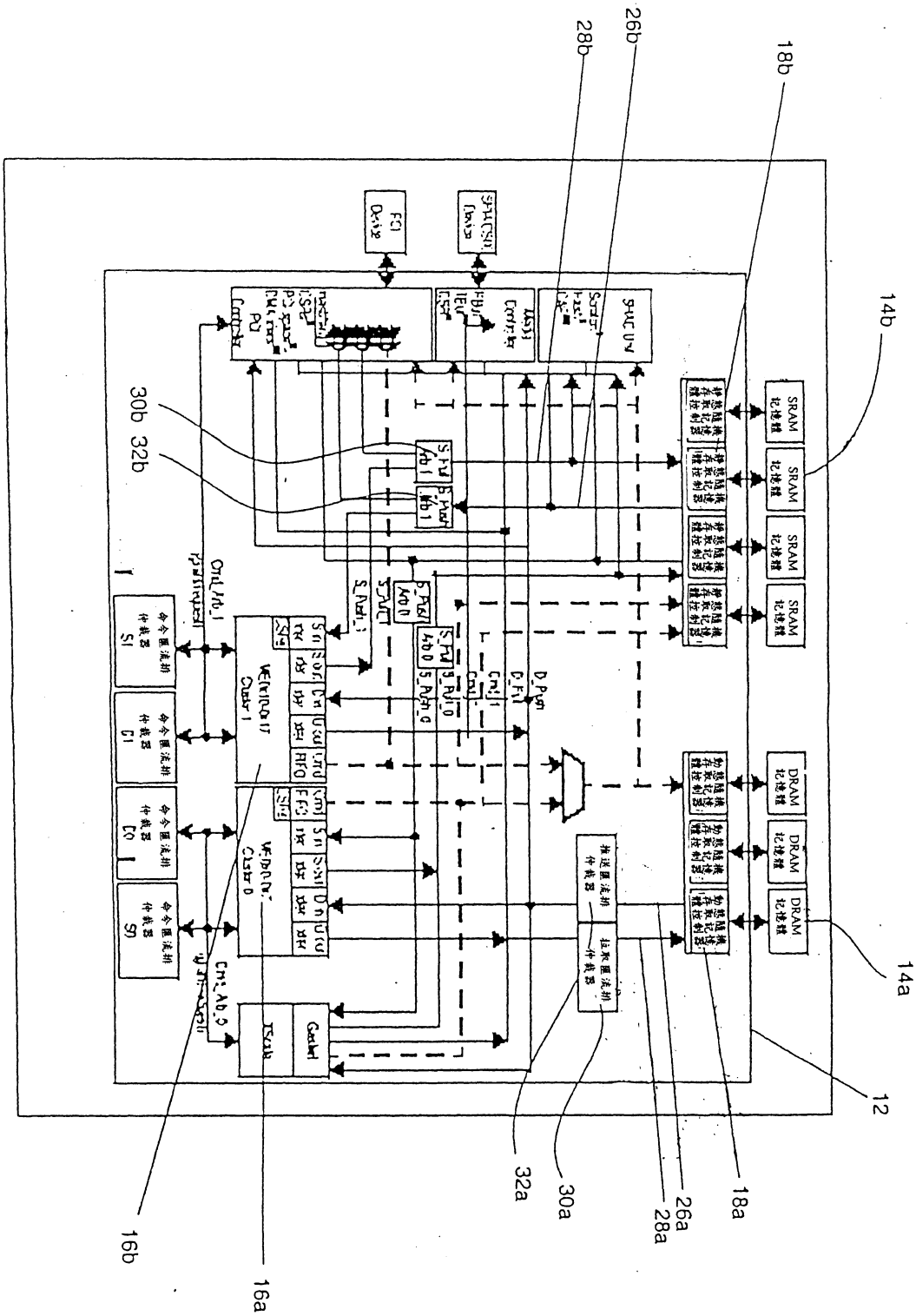

10
12
14a
14b
16a
16b
18a
18b
26a
26b
28a
28b
30a
30b
32a
32b
SRAM
DRAM

圖 2

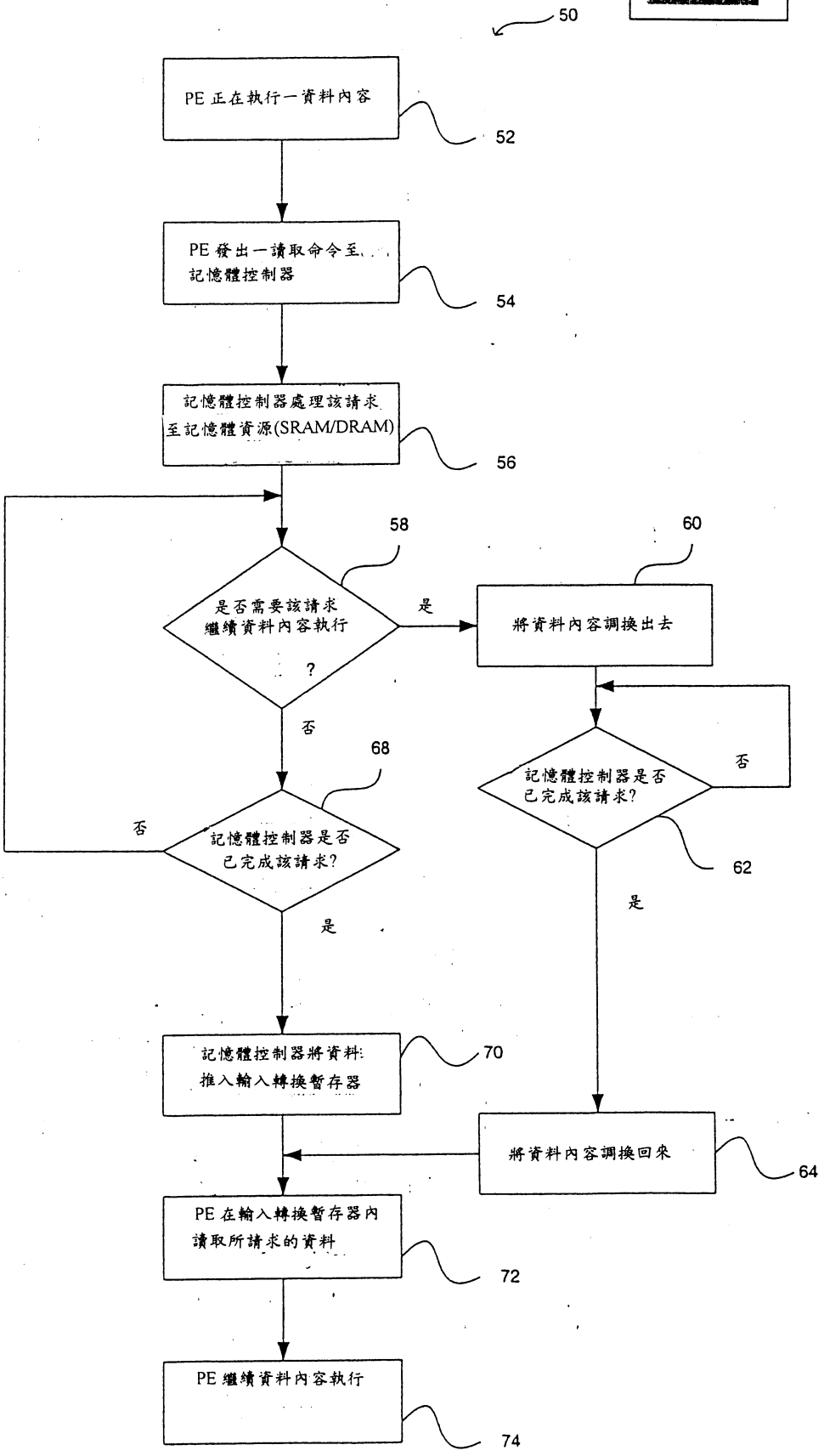

50
PE 正在執行一資料內容
52
PE 發出一讀取命令至記憶體控制器
54
記憶體控制器處理該請求至記憶體資源(SRAM/DRAM)
56
58
是否需要該請求繼續資料內容執行?
是
60
將資料內容調換出去
否
記憶體控制器是否已完成該請求?
否
62
是
68
記憶體控制器是否已完成該請求?
否
是
記憶體控制器將資料推入輸入轉換暫存器
70
將資料內容調換回來
64
PE 在輸入轉換暫存器內讀取所請求的資料
72
PE 繼續資料內容執行
74

圖 3

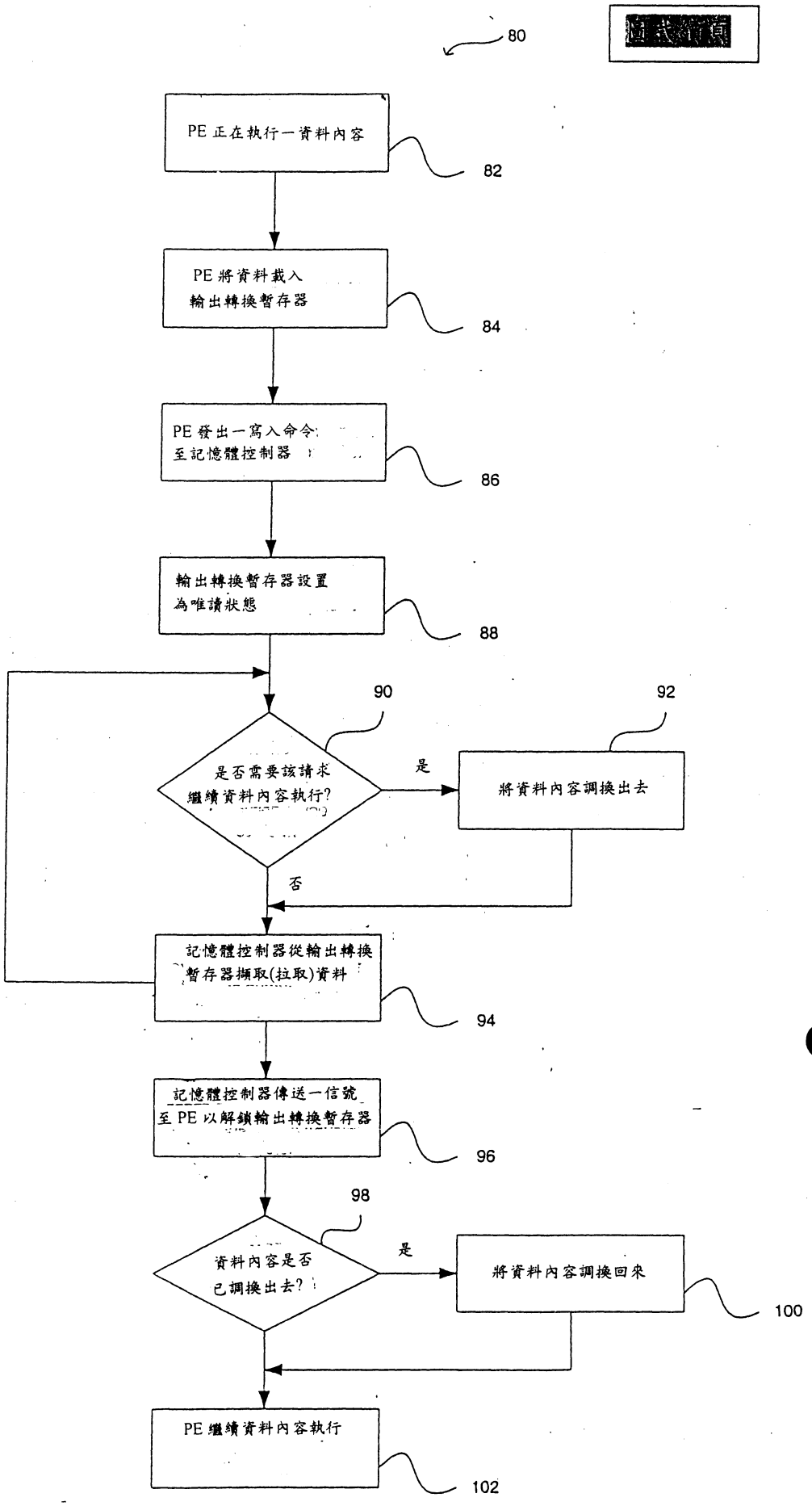

80
PE 正在執行一資料內容
82
PE 將資料載入
輸出轉換暫存器
84
PE 發出一寫入命令
至記憶體控制器
86
輸出轉換暫存器設置
為唯讀狀態
88
90
是否需要該請求
繼續資料內容執行?
是
92
將資料內容調換出去
否
記憶體控制器從輸出轉換
暫存器擷取(拉取)資料
94
記憶體控制器傳送一信號
至 PE 以解鎖輸出轉換暫存器
96
98
資料內容是否
已調換出去?
是
將資料內容調換回來
100
PE 繼續資料內容執行
102

圖 4

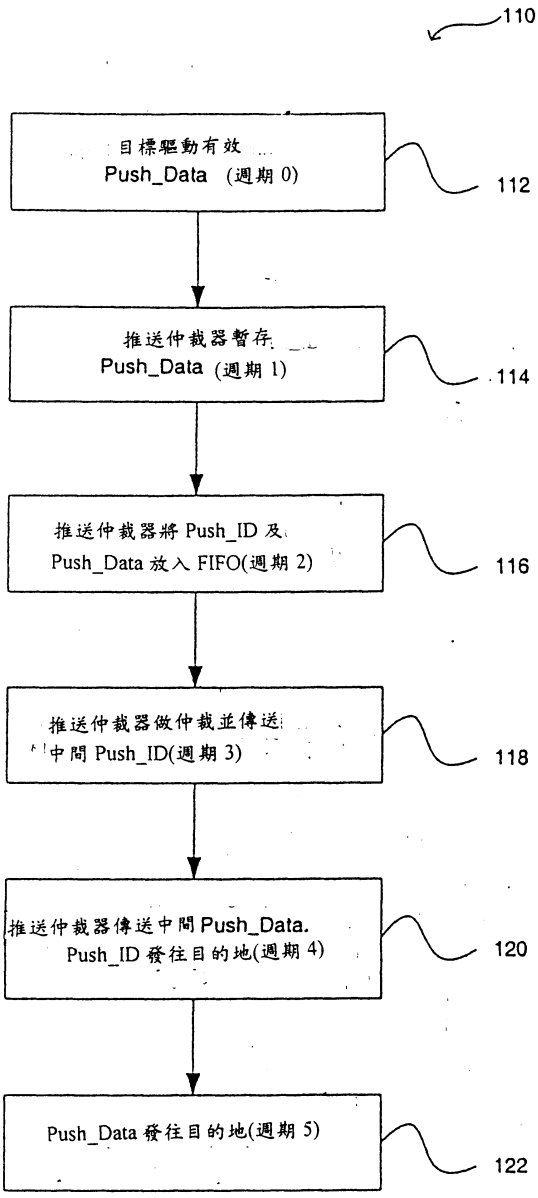

110
目標驅動有效
Push_Data (週期 0)
112
推送仲裁器暫存
Push_Data (週期 1)
114
推送仲裁器將 Push_ID 及
Push_Data 放入 FIFO(週期 2)
116
推送仲裁器做仲裁並傳送
中間 Push_ID(週期 3)
118
推送仲裁器傳送中間 Push_Data.
Push_ID 發往目的地(週期 4)
120
Push_Data 發往目的地(週期 5)
122

圖 5

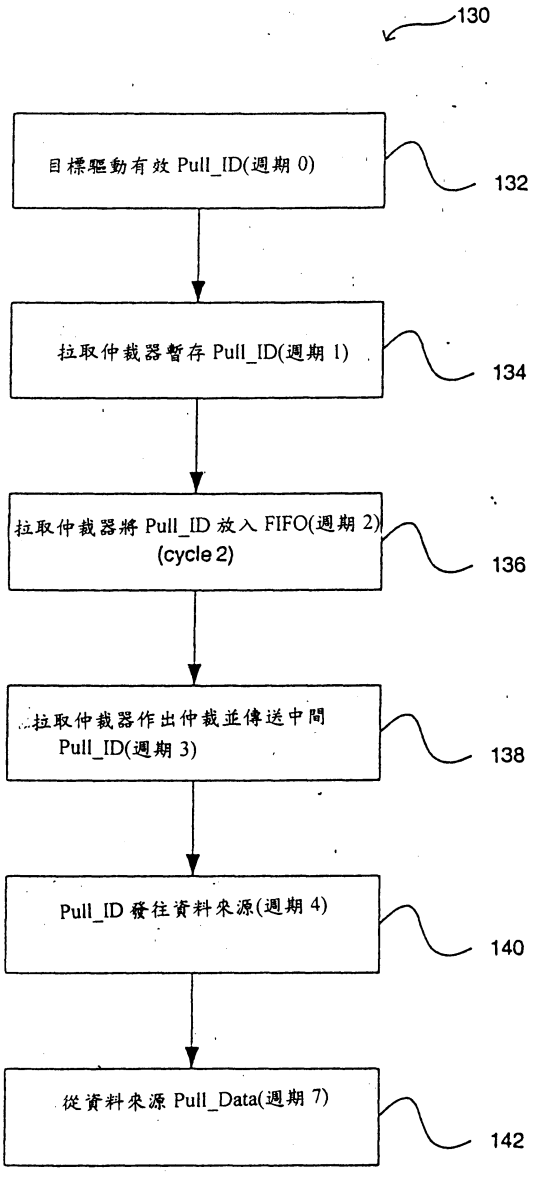

130
目標驅動有效 Pull_ID(週期 0)
132
拉取仲裁器暫存 Pull_ID(週期 1)
134
拉取仲裁器將 Pull_ID 放入 FIFO(週期 2)
(cycle 2)
136
拉取仲裁器作出仲裁並傳送中間
Pull_ID(週期 3)
138
Pull_ID 發往資料來源(週期 4)
140
從資料來源 Pull_Data(週期 7)
142

圖 6

94.3.8
修正本

## 拾、申請專利範圍

1. 一種在一處理代理與一記憶體資源間轉換資料之方法，其包括：

使用一推送匯流排指定該記憶體資源以將資料推入該處理代理，該推送匯流排具有複數個來源仲裁其使用；及

使用一拉取匯流排指定該記憶體資源以自該處理代理接收資料，該拉取匯流排具複數個目的地仲裁其使用。

2. 如申請專利範圍第1項之方法，其中轉換包含：在該程式設計代理上建立複數個資料內容，並且維持程式計數器及資料內容相應暫存器。

3. 如申請專利範圍第2項之方法，其中該程式設計代理在一讀取階段執行一資料內容並發出一讀取命令至一記憶體控制器。

4. 如申請專利範圍第3項之方法，其中該記憶體控制器處理該讀取命令使其傳送至該記憶體資源。

5. 如申請專利範圍第4項之方法，其中若需要該讀取資料繼續資料內容的執行，則將該資料內容調換出去。

6. 如申請專利範圍第5項之方法，其中該記憶體控制器已完成該讀取命令處理後，該記憶體控制器將資料推入該程式設計代理的一輸入轉換暫存器中。

7. 如申請專利範圍第6項之方法，其中在已推送該資料後

，該程式設計代理在該輸入轉換暫存器中讀取該資料並繼續其內容的執行。

8. 如申請專利範圍第2項之方法，其中該程式設計代理在一寫入階段執行一資料內容並將該資料載入一輸出轉換暫存器。

9. 如申請專利範圍第8項之方法，其中該程式設計代理發出一寫入命令至一記憶體控制器，且該輸出轉換暫存器設置成為一唯讀狀態。

10. 如申請專利範圍第9項之方法，其中若需要該寫入命令繼續該資料內容的執行，則將該資料內容調換出去。

11. 如申請專利範圍第10項之方法，其中該記憶體控制器從該輸出轉換暫存器推送該資料，該記憶體控制器並發送一信號至該程式設計代理以解鎖該輸出轉換暫存器。

12. 如申請專利範圍第11項之方法，其中若在該輸出轉換暫存器已解鎖後該資料內容已調換出去，則將該資料內容調換回來並且該程式設計代理繼續該資料內容的執行。

13. 一種資料處理系統，該系統包括：

一記憶體資源；

一處理代理，其配置以在該處理代理與記憶體資源間轉換資料；

一推送匯流排，用以將資料推入該處理代理，且該處理代理具有複數個資料來源仲裁其使用；

一拉取匯流排，用以自該處理代理接收資料，且該處理代理具有複數個目的地仲裁其使用；及

複數個微引擎，其執行多個資料內容並尋找該處理代理之資源。

14.如申請專利範圍第13項之系統，進一步包括一從該記憶體資源到處理代理的資料轉換讀取階段，其中該轉換係單向性。

15.如申請專利範圍第13項之系統，進一步包括一從該處理代理到記憶體資源的資料轉換寫入階段，其中該轉換係單向性。

16.如申請專利範圍第13項之系統，進一步包括複數個程式計數器及複數個資料內容相應暫存器。

17.如申請專利範圍第16項之系統，其中該資料內容相應暫存器係由一組中挑出，包含一般用途暫存器、內部程式設計代理暫存器、靜態隨機存取記憶體(static random access memory；SRAM)輸入轉換暫存器、動態隨機存取記憶體(dynamic random access memory；DRAM)輸入轉換暫存器、SRAM輸出轉換暫存器、DRAM輸出轉換暫存器及局部記憶體暫存器。

18.如申請專利範圍第17項之系統，其中該程式設計代理係配置以執行一資料內容並發出一讀取命令至一記憶體控制器。

19.如申請專利範圍第18項之系統，其中該記憶體控制器係配置以處理該讀取命令使其傳送至該記憶體資源。

20.如申請專利範圍第19項之系統，其中該程式設計代理係配置以在需要該讀取命令繼續資料內容執行時，將該資料內容調換出去。

21.如申請專利範圍第20項之系統，其中該讀取命令處理後，該記憶體控制器係配置以推送該資料至該程式設計代理的一輸入轉換暫存器，並且該程式設計代理係配置以讀取該輸入轉換暫存器內的資料並繼續該資料內容的執行。

22.如申請專利範圍第15項之系統，其中該程式設計代理係配置以執行一資料內容並將該資料載入該程式設計代理一輸出轉換暫存器。

23.如申請專利範圍第22項之系統，其中該程式設計代理係配置以發出一寫入命令至一記憶體控制器，其中該輸出轉換暫存器設置為一唯讀狀態。

24.如申請專利範圍第23項之系統，其中該程式設計代理係配置以在需要該寫入命令繼續資料內容執行時，將該資料內容調換出去。

25.如申請專利範圍第24項之系統，其中該記憶體控制器係配置以從該輸出轉換暫存器推送該資料，並發送一信號至該程式設計代理以解鎖該輸出轉換暫存器。

26.一種基於一電腦可讀取媒體之電腦程式產品，用於使一平行處理器執行一功能，其包括可使該處理器執行以下功能之指令：

使用一推送匯流排指定該記憶體資源以推入資料

至該處理代理，該推送匯流排具複數個資料來源仲裁其使用；及

使用一拉取匯流排指定該記憶體資源以自該處理代理接收資料，該拉取匯流排具複數個目的地仲裁其使用。

27.如申請專利範圍第26項之電腦程式產品，進一步包括使該處理器在該程式設計代理上建立複數個資料內容，並且維持程式計數器及資料內容相應暫存器之指令。

28.如申請專利範圍第26項之電腦程式產品，其中該程式設計代理在一讀取階段執行一資料內容並發出一讀取命令至一記憶體控制器。

29.如申請專利範圍第26項之電腦程式產品，其中該記憶體控制器處理該讀取命令使其傳送至該記憶體資源，若需要該讀取資料繼續資料內容執行則將該資料內容調換出去。