

告 本

請日期：

87.12.16

案號：

81120942

別：

6106F 9/22

以上各欄由本局填註)

發明專利說明書

425528

一、
發明名稱

中 文

於一完成無序多線執行處理器中排序載入及儲存指令之記憶系統

英 文

MEMORY SYSTEM FOR ORDERING LOAD AND STORE INSTRUCTIONS IN A PROCESSOR THAT PERFORMS OUT-OF-ORDER MULTITHREAD EXECUTION

二、
發明人

姓 名
(中文)

1. 艾卡利，海森

姓 名
(英文)

1. AKKARY, HAITHAM

國 籍

1. 美國

住、居所

1. 美國奧立岡州波特蘭市西北艾內思塔夏大道17840號

三、
申請人

姓 名
(名稱)
(中文)

1. 美商英特爾公司

姓 名
(名稱)
(英文)

1. INTEL CORPORATION

國 籍

1. 美國

住、居所
(事務所)

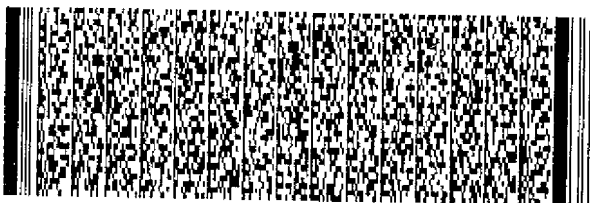
1. 美國加州聖塔卡拉瓦市米遜大學路2200號

代表人
姓 名
(中文)

1. F. 湯姆士. 當烈二世

代表人
姓 名
(英文)

1. F. THOMAS DUNLAP, JR.



本案已向

國(地區)申請專利

申請日期

案號

主張優先權

美國 US

1997/12/16 08/991,734

有

有關微生物已寄存於

寄存日期

寄存號碼

無



五、發明說明 (1)

發明背景

發明範圍

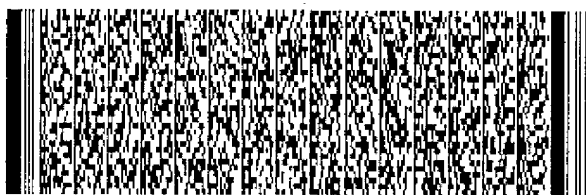
本發明與處理器有關，特別是具有記憶體排序緩衝器的處理器。

背景技術

目前的超級定標(SUPERSCALER)處理器，例如微處理器，進行如分支預測與無序執行等技術以增進執行效率。具有無序執行管線的處理器，以跟指令被抓取與解碼之順序不同的順序執行某些指令。沒有附屬性的指令可以不依順序地執行。無序執行藉由避免執行單元僅因為程式指令的順序就被閒置，來增進處理器的執行效率。指令結果在執行後被重新排序。

處理資料附屬性的工作透過限制指令解碼成循序而簡化。然後處理器可以確認資料如何經過暫存器由一指令流通到後續的指令。為確保程式的正確性，暫存器被重新配置且指令在保留站等待直到其輸入參數產生為止，此時它們才被發配給適當的功能單元執行。暫存器重新配置器、保留站與相關的機制將有相關性的指令鏈結在一起，以使附屬指令不會在其所依附指令之前執行。因此，該等處理器為循序抓取與解碼所限制。

當來自指令快取記憶體的指令漏失掉或失誤預測了一分支時，處理器必須等到指令區塊由較高層快取記憶體或記憶體中被抓取，或者直到失誤預測的分支被解決後，才會重置錯誤路徑的執行。此種運作的結果是，指令快取記憶



五、發明說明 (2)

體漏失與失誤預測分支之前與後的獨立指令不能平行地執行，雖然如此做可能是正確的。

記憶體排序緩衝器已被用來排序載入與儲存。處理器中需要一改良的機制，以讓處理器由猜測錯誤中回復。

發明概要

在本發明一實施例中，處理器包括含有載入緩衝器與儲存緩衝器的記憶體排序緩衝器(MOB)，其中該MOB將載入與儲存指令排序，以便維持在不同線中的載入與儲存指令間之資料一致性，其中至少一線依附於至少另一線。在本發明另一實施例中，處理器包括一執行管線以同時地至少執行部份的線，其中至少一線依附於至少另一線，該執行管線包括一記憶體排序緩衝器將載入與儲存指令排序。該處理器也包括偵測電路以偵測與載入緩衝器中之載入指令相關的猜測錯誤。

圖示概述

本發明由以下的詳細說明與隨附之本發明實施例的圖示可以更徹底瞭解，但不應將本發明限定於所描述的特定實施例，其僅係為幫助說明與了解。

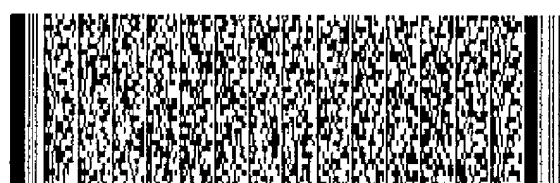
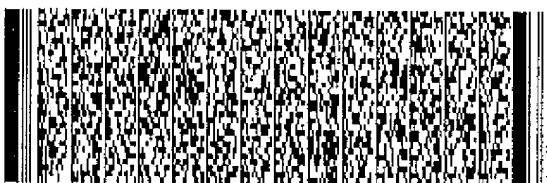
圖1是一處理器之一實施例中某些元件的高階方塊圖示。

圖2是根據本發明一實施例之處理器的方塊圖。

圖3是根據本發明另一實施例之處理器的方塊圖。

圖4是一兩線的範例之流程圖。

圖5是一兩線的另一範例之流程圖。



五、發明說明 (3)

圖6是一四線的範例之流程圖。

圖7是顯示圖6之線的重疊執行的圖形顯示。

圖8是說明本發明一實施例之個別追蹤緩衝器的方塊圖。

圖9顯示指出在兩個時點上程式與退役順序的一個陣列。

圖10是圖8中追蹤緩衝器一實施例之某些元件的方塊圖表示。

圖11是圖8中追蹤緩衝器另一實施例之某些元件的方塊圖表示。

圖12是圖10的追蹤緩衝器一指令佇列陣列之一實施例的部份圖形表示。

圖13是圖10的追蹤緩衝器一資料與附屬性陣列之一實施例的部份圖形表示。

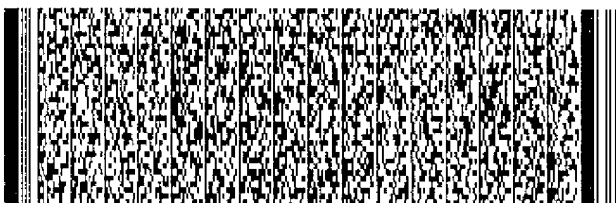
圖14舉例說明用來產生圖10的陣列之附屬性欄位的修飾器暫存器之一實施例與一被修正過的暫存器。

圖15是一用來產生圖13的陣列之附屬性欄位的邏輯或閘。

圖16說明一實施例產生圖13的陣列之附屬性欄位的運作之一流程圖。

圖17是根據本發明之一實施例在一追蹤緩衝器中一特定暫存器和位置有附屬性的圖解表示法。

圖18是圖10的追蹤緩衝器的一輸出暫存器檔案一實施例之部分圖解表示法。



五、發明說明 (4)

圖19是圖10的追蹤緩衝器的一輸入暫存器檔案一實施例之部分圖解表示法。

圖20是根據本發明一實施例用以連接圖18之輸出暫存器檔案與圖19之輸入暫存器檔案的比較器與重演觸發邏輯之方塊圖。

圖21是說明輸出暫存器檔案的內容可被使用時點之流程圖。

圖22是說明根據本發明一實施例於圖2之MOB中，個別記憶體排序緩衝器(MOB)的方塊圖。

圖23是圖22的MOB中之一個的儲存緩衝器一實施例的部分圖式表示。

圖24是圖22的MOB中之一個的載入緩衝器一實施例的部分圖式表示。

圖25舉例說明一比較器比較載入與儲存指令的位址。

圖26舉例說明一比較器比較儲存與載入指令的位址。

圖27是根據本發明之一實施例的MOB控制電路與儲存緩衝器的方塊圖表示。

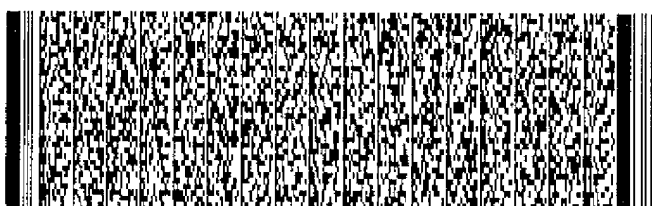
圖28是根據本發明之一實施例的MOB控制電路與載入緩衝器的方塊圖表示。

圖29 是六條線的例子之流程圖。

圖30是說明在一時點 t_1 圖29的線之間的關係之樹。

圖31是說明假設在線T1退役之前，線T4被重新設定之時點 t_2 ，圖29的線間的關係之樹。

圖32是說明假設線T4被重新設定之前線T1退役之時點



五、發明說明 (5)

t2，圖29的線間的關係之樹。

圖33是說明在時點t3圖29的線間的關係之樹。

圖34是一流程圖，說明五條線的例子。

圖35是說明在時點t1圖34的線間之關係的樹。

圖36是說明在時點t2圖34的線間之關係的樹。

圖37是根據圖2以外之另一實施例之處理器的方塊圖。

圖38是包含圖2的處理器之一電腦系統。

較佳實施例詳細描述

A. 線的產生與管線108的概觀

B. 關於追蹤緩衝器114的細節

1. 追蹤緩衝器114A

a. 指令佇列陣列 202A

b. DAD陣列206A和附屬性產生電路212A

c. 輸出暫存器檔案210A和輸入暫存器檔案208A

2. 追蹤緩衝器114'

C. 重演順序演算法

D. 第二層或最終退役

E. 記憶系統

1. 儲存緩衝器與載入緩衝器

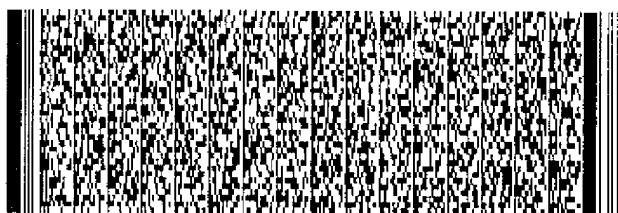
2. 載入與儲存位址的比較

a. 載入指令的執行

b. 儲存指令的執行

c. 重設

3. 儲存指令的重演



五、發明說明 (6)

4. 多個載入指令的重演

5. 載入與儲存指令的最終退役

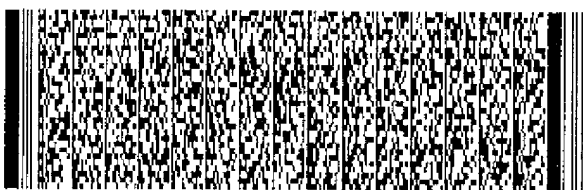
F. 關於線管理邏輯與最終退役邏輯的額外資訊

G. 沒有多線的實施例

H. 額外的資訊與實施例

圖1說明一處理器10的一些元件。處理器10包括一執行管線12和一追蹤緩衝器14，它位於執行管線12之外。執行管線12可包括一記憶體排序緩衝器。導線18上的指令被提供到執行管線12以便執行。該等指令也藉由導線22提供給追蹤緩衝器14。該等指令可在執行管線12中被猜測性地執行。猜測的例子包括資料猜測和附屬性猜測。各種廣泛的猜測都可能被包括。處理器10包括，包含在追蹤緩衝器14中，的機制以偵測失誤猜測(misspeculations)並由其中恢復。

當偵測到失誤猜測的時候，失誤猜測的指令透過導線24從追蹤緩衝器14被提供到執行管線12，並在執行管線12被重演。如果一指令被"重演"，該指令和所有依賴其上之指令均再被執行，雖不必然一定要同時地。若一指令被"完全重演"，該指令與依程式順序跟隨其後之所有指令會再被執行。程式順序是在一在循序處理器中指令將被執行的順序。指令可完全地依照程式順序，或依程式順序以外之某種次序通過導線18。處理器10可以是一循序或無序處理器。附屬指令的再執行可能造成依賴於該附屬指令的指令被重演。再執行指令的數目可藉由控制引發重演之事件而



五、發明說明 (7)

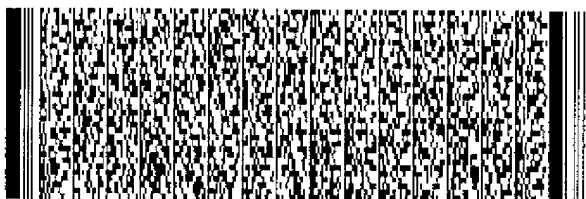
加以控制。通常，執行一詞可包括原始執行與再執行。指令的結果至少有一部份經由導線26被提供給追蹤緩衝器。最終退役邏輯34會在確保指令原始執行或再執行中被正確地被執行後將追蹤緩衝器14裡的指令除役。

執行管線12可以是各種執行管線中之任何一種，而且可以是更大管線中之一個區段。執行管線12可以與各種處理器連結使用。在圖2中提供範例說明一有執行管線108的處理器50之元件，而在圖3說明一有執行管線308的處理器100。在圖2本發明之一實施例中，執行管線108包括暫存器重新配置。在其他的實施例中，執行管線不包括暫存器重新配置。處理器可以同時地處理多條線(如在圖2中處理器50的情形)，或非同時地處理多條線(如在圖3中處理器100的情形)。首先討論處理器50。

說明書中參考的"一個實施例"或"實施例"，表示包含在本發明至少一實施例中，與該實施例合併敘述的特定特徵，架構，或特性。說明書中各處出現的"在一實施例中"之片語，未必都參照相同的實施例。

A. 線的產生與管線108之概觀

指令經由導線102被提供給一指令快取(I-cache)104。解碼器106如所說明接收來自I-cache104的指令，但也可以在指令到達I-cache104之前將其解碼。依迴路境與實施時之選擇而定，"指令"一詞可包括巨集操作(macro-op)，微操作(uops)，或一些其他形式的指令。各種指令集中之任何一種都可能被使用，包括，但並非僅限於精減指令集



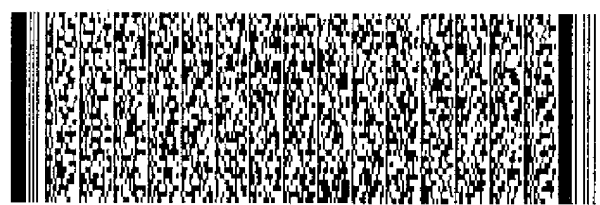
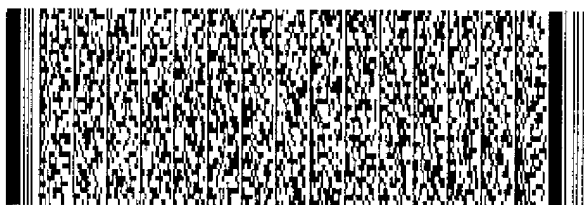
五、發明說明 (8)

運算(RISC)或複雜指令集運算(CISC)指令。再者，解碼器106可將CISC指令解碼為RISC指令。來自I-cache104的指令經由MUX 110被提供到管線108，並經由導線118被提供到追蹤緩衝器114。

追蹤是一組指令。線包括追蹤與有關的訊號，如暫存器值與程式計數器值。

線管理邏輯124藉由將起始計數經導線130提供給程式計數器112A, 112B, ..., 112X, (其中X代表程式計數器的數目)使I-cache104中一程式或程序產生不同的線。例如，X可為4或更大，更小。線管理邏輯124也藉由停止關聯的程式計數器來結束線。線管理邏輯124可使程式計數器開始另一條線。不同線的一部分同時從I-cache104被讀取。

為了決定要在程式或程序的何處產生線，線管理邏輯124可經導線128由解碼器106讀取指令。線可包括程式設計師或編譯者所插入清楚地界定線的開始與結束之指令。另外，線管理邏輯124可分析程式或程序的指令，以將供應到I-cache104的程式或程序打散到不同的線之中。舉例來說，分支，迴路，反向分支，返回，跳越，程序呼叫，和函數呼叫可能是分離線的好處所。線管理邏輯124在考慮由何處開始線時，可考慮線的潛在長度，涉及變數之多寡，後續的線間有多少個共同的變數，及其他因素。線管理邏輯124在決定線的邊界時可考慮程式順序。程式順序是在一循序處理器上的線當中，線與指令將被執行之順序。那些線中之指令可被無序執行(與程式順序相對)。線



五、發明說明 (9)

基本上可被管線108獨立地處理。線管理邏輯124可包括一含有歷史表列的預測機制，以避免做出最佳選擇以下之決定。舉例來說，線管理邏輯124可產生一線，之後再判定該線實際上不是程式順序中的一部份。在此情況中，若再一次遇到相同的程式碼，預測機制可用來判斷是否要再次產生相同的線。線管理邏輯124可利用線的動態產生與來自編譯者或程式設計者的快速指令提示之結合，以決定從指令中的何處產生線。

動態地產生線是由一非特別為多線而編寫或編譯之程式來產生線，其中至少一線依賴於另一線上。程式可由一包括執行管線108與線管理邏輯124的晶片開始。動態地產生線，執行線，並在執行中偵測與改正猜測錯誤稱為動態的多線。

圖4說明一包括條件反向分支指令的線T1。依照程式順序，線T2接在條件分支指令之後執行。依照時間順序，線T2被預測在線T1第一次到達條件分支指令時開始執行。因此，線T1與T2的一部分同時地被執行。如果線T2涉及失誤猜測，線T2已完成的指令會被重演。

線管理邏輯124可透過導線130檢測程式計數器的計數。檢測計數的一個目的是決定應在何時結束一線。例如，當條件分支的情況不符合時，若允許線T1的程式計數器繼續，它將會超前於線T2的第一個指令。因此，當情況不符合的時候，線管理邏輯124停止線T1的程式計數器。

圖5說明包含一函數呼叫指令的線T1。依照程式順序，

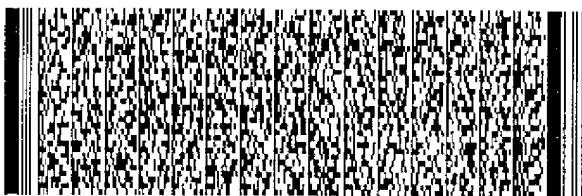


五、發明說明 (10)

當到達呼叫指令的時候，程式計數器跳到該函數的位置並且執行至一返回指令，此時程式計數器返回該呼叫之後的指令。依照程式順序，線T2由接續在返回之後的指令開始。依照時間順序，線T2被預測在線T1第一次到達該呼叫時開始執行。如果線T2涉及失誤猜測，線T2已完成的指令會被重演。線T1在其程式計數器答達到線T2的第一個指令時結束。圖5中的載入MX與儲存MX指令將會在以下討論。

圖6說明程式一個區段部份的線T1，T2，T3，和T4。不同的程式計數器產生線T1，T2，T3，和T4。線T1包括到點A(函數呼叫指令)的指令，而後從點B，到點C(條件反向分支指令)，到點D並再次到點C(迴路可能重複數次)。線T2由程式順序中，立即接在點A所呼叫函數的返回指令後面之指令開始。線T3由程式順序中，立即接在點C的條件反向分支後面之指令開始，並且繼續到點E，到點F，到點G，到點H，和點I；它是一前往緊接在點A後面線T2開始處的指令之返回指令。線T4由程式順序中，立即接在點E的條件反向分支後面之指令開始。

如圖7中所示，線T1，T2，T3，和T4有一部分被同時地讀取，解碼與執行。這些線因未遵從程式順序而被無序地讀取，解碼與執行。依照時間順序，線T2，T3，和T4的執行分別緊接在點A，C和E的指令後面開始。垂直的破折線顯示一父子關係。線T2，T3和T4在暫存器中及/或記憶體位置中的資料被確認為正確之前，依據其中的資料預測性地執行。處理器100具有機制以偵測失誤猜測，並使被失

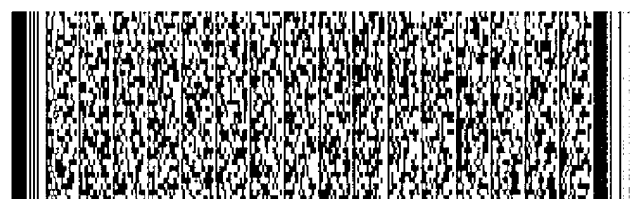
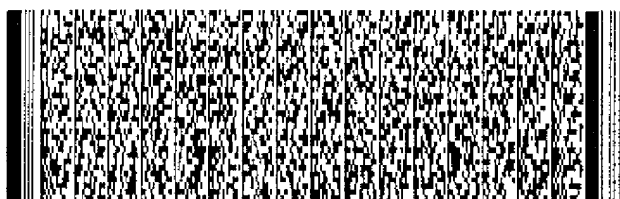


五、發明說明 (11)

誤猜測的指令重演。它把線T4變成不在程式順序的一部份內。線T4可能直到線管理邏輯124判定線T4不是程式順序的一部份之後才被執行。此時，線T4可能被重新設定，而在處理器100中進行或處理T4的資源可能被解除指派，然後指派給另外一條線。依照程式順序，線T1，T2，和T3將被執行如下：首先線T1，然後線T3，然後線T2。

參照圖2，來自MUX110的指令被重新配置/指派單元150接收，它提供暫存器檔案152中被重新配置的實體暫存器之一實體暫存器識別碼(PRID)。該PRID透過旁通導線126提供給追蹤緩衝器114。指派包括分配暫存器給指令與分配排程/發送單元156的保留站中的項目。一旦保留站中一特定指令的參數準備就緒，該指令就被發送到執行單元158中的一個執行單元(例如，整數，浮點數)，或包含位址產生單元(AGU)172，記憶體排序緩衝器(MOB)178與資料快取176的一記憶體執行管線中。依指令而定，參數可由暫存器檔案152經導線168提供。在本發明的一實施例之下，一線中的附屬指令可如此連結以便它們不會被無序地執行。然而，來自不同線的附屬指令可能同時無序地被讀取，解碼，與執行。線的某些執行可能是猜測的。

為了高執行效率，保留站與相關的機制被設計成有低延遲及高頻寬指令發送。延遲及頻寬需求對可在保留站中等候的指令之數目設立限制。藉由將追蹤緩衝器114置於管線108之外，很多的指令可被執行/重演而不用大大地減低管線108的生產。在執行管線108和追蹤緩衝器114之間延

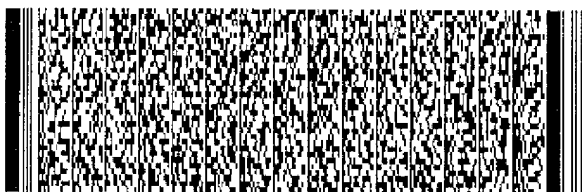


五、發明說明 (12)

遲的影響可透過管線而減少。

執行的結果與相關資訊經由導線122(在暫存器的情形中)及經由MUX192與導線196由寫回單元162寫到追蹤緩衝器114中。執行的結果與相關資訊也可被寫到暫存器檔案152與和有關的重排序緩衝器(ROB)164。一旦指令的結果與和資訊被寫到暫存器檔案152與ROB164,指令就管線108相關之順序退役。此退役稱為第一層或初始退役。在第一層退役或之前,在排程/發送單元156中給退役指令的資源包括保留站,暫存器檔案152與ROB164被解除指派。然而,所有與該指令有關的細節被維持在追蹤緩衝器114與MOB 178中直到最終退役,描述如下。

附屬性在依照程式順序中較遲的線所使用的資料產生於較早的線中時,存在於較遲的線與較早的線之間。這些資料可能是透過記憶體中較早的線所產生或非記憶體指令所產生。舉例來說,若在較遲的線中之載入指令與較早的線中之儲存指令有相同的位址,該較遲的線可能依賴於該較早的線。較遲的線也可能依賴於較早的線,若較遲的線中之指令涉及較早的線中所修改之暫存器。同樣地,當依照程式順序中較遲的線使用較早的線中所產生之資料時,較遲的線附屬於較早的線。"附屬性"一詞也用在片語"附屬性猜測"中。附屬性猜測的一個例子,是猜測在一載入指令與一較早的儲存指令之間沒有附屬性。位址匹配是檢驗附屬性猜測錯誤之技術的一個例子。資料猜測的一個例子是猜測在暫存器中的資料是正確的資料。暫存器匹配是檢



五、發明說明 (13)

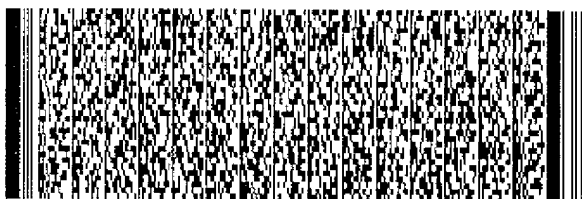
驗資料猜測錯誤之技術的一個例子。

B. 有關追蹤緩衝器114的細節

參照圖8，追蹤緩衝器114包括追蹤緩衝器114A, 114B, 114C, ..., 114Y，其中Y表示追蹤緩衝器的數目。舉例來說，如果 $Y=4$ （也就是， $Y=D$ ），就有4個追蹤緩衝器。如果Y比3更少，追蹤緩衝器114就不包括在圖8中顯示的所有追蹤緩衝器。Y可能與X（程式計數器的數目）相同或不同。追蹤緩衝器114可以是被區分為個別的追蹤緩衝器的單一記憶，或實體上分開的追蹤緩衝器，或此二者的某些組合。

參照圖9，在一實施例中，線管理邏輯124包括一指定線ID的程式順序（也是退役順序）的陣列198。在該範例中，每個追蹤緩衝器有一獨特的線ID，或一對一映射到一線ID。舉例來說，追蹤緩衝器114A被分配給線ID 1，追蹤緩衝器114B被分配給線ID 2等。線ID可以是固定的或規劃的。在一實施例中，每個程式計數器與一特定的線ID及追蹤緩衝器有關。（或者，沒有如此的限制關係）

圖9顯示在時點 t_1 與時點 t_2 線的退役順序之一個範例。在該範例中，只有四個追蹤緩衝器和四個線IDs。相關的線號碼顯示在括弧中。因其實施而定，括弧中的線號碼實際上並不包含在陣列198中。在時點 t_1 ，程式與退役順序是線T1，T3，T2和T4，如圖6中的例子。在時點 t_1 和時點 t_2 之間，將會判定線T4並不在程式順序中。因此，線T4被重新設定，以製造空間給追蹤緩衝器114D中的線T5（未顯示在圖5中）。線T5與線ID 4有關。線T1退役，製造空間給



五、發明說明 (14)

追蹤緩衝器114A中的線T6。線T6與線ID 1有關。在時點t2，程式與退役順序是線T3，T2，T5和T6。(如果線T1在線T4被重新設定之前退役，則線T5與T6將會有不同的線ID，但程式與退役順序將不變)。依所使用演算法而定，可能一開始在陣列198中線T2在線T3之前，但程式與退役順序會被更正，如在時點t1的陣列198。

如所述及，線的程式順序是在一循序處理器上線將會被執行的順序。指令的程式順序是在一循序處理器上指令將會被執行的順序。線管理邏輯124不一定要在一開始決定線的真正程式順序。然而，線管理邏輯124最後還是決定真正的程式順序。

參照圖8，追蹤緩衝器114A，114B，...，114Y透過連接到導線118之導線118A，118B，...118Y接收指令。在導線118A，118B，...，118Y與導線118之間可能有解多工電路。或者，致能訊號可控制哪個追蹤緩衝器被啟動。又或者，可能有足夠平行導線以處理平行的交易。追蹤緩衝器114A，114B，...，114Y經由連接到導線120的導線120A，120B...120Y，提供重演的指令與相關資訊給管線108。要注意的是來自追蹤緩衝器114的多個指令可能同時地通過導線120與MUX 110以再執行。同時，來自解碼器106的多個指令也可能第一次通過MUX 110。線ID和指令ID(instr ID)伴隨著每個指令通過管線。重演計數也可能伴隨著該指令。在載入與儲存指令的情況中，一載入緩衝器ID(LBID)與一儲存緩衝器ID(SBID)也可能伴隨著該指



五、發明說明 (15)

令。在一實施例中，LBID與SBID伴隨著每個指令，雖然LBID與SBID在指令並非載入或儲存的情況中可能是毫無意義的。如下所描述，一PRID或值也可伴隨著一被再執行的指令。

追蹤緩衝器114A, 114B, ..., 114Y透過連接到導線126的旁通導線126A, 126B, ... 126Y接收來自重新配置/指派單元150的PRID, LBID與SBID值。追蹤緩衝器114A, 114B, ..., 114Y透過連接到導線122的導線122A, 122B... 122Y與連接到導線196的導線196A, 196B, ..., 196Y, 接收寫回結果資訊與有關訊號。重演訊號透過連接到導線194的導線194A, 194B, ..., 194Y提供。多工及/或致能電路及/或重要數量的平行導線可能用在導線120, 126, 122, 194與196中。追蹤緩衝器可能完全相同或有些不同。

在圖10中，追蹤緩衝器114A說明追蹤緩衝器的第一個實施例。在圖11中，追蹤緩衝器114A'說明追蹤緩衝器的第二個實施例。追蹤緩衝器的其他實施例可包括追蹤緩衝器114A和114A'的變化或相當不同的架構。

1. 追蹤緩衝器114A

參照圖10，追蹤緩衝器114A包括一指令佇列陣列202A，一資料與附屬性(DAD)陣列206A，一輸入暫存器檔案208A，一輸出暫存器檔案210A，附屬性產生電路212A，與控制電路224A。"陣列"一詞打算包括廣義的多面向，不限於特定形式的資訊。

a. 指令佇列陣列202A



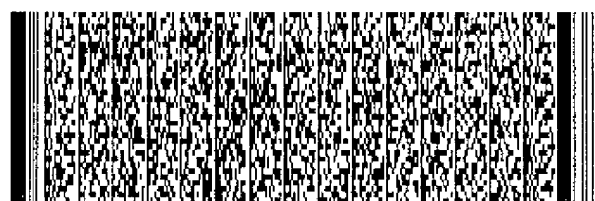
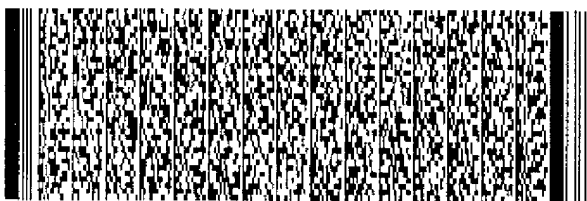
五、發明說明 (16)

參考圖12，以下描述指令佇列陣列202A的架構，及根據本發明的一實施例它和其他元件的交互作用。指令佇列陣列202A接收從I-cache 104所讀取指令，該指令是一特定線之一部份。在線中的指令依照順序被讀取與寫入到指令佇列陣列202A之內。指令為另外一條線之部份者，被寫入不同追蹤緩衝器的一指令佇列之內，或由指令佇列陣列202A在一不同時間寫入。指令佇列陣列202A包括每個指令識別碼(instr ID)的各種資訊的欄位。不同的實施例可包括一些不同的欄位與不同數目的資料列。在指令佇列陣列202A的實施例中，程式計數器的值不列入考慮，但在其他的實施例中可列入考慮。指令佇列陣列202A與圖示中說明的所有其他元件可包括各種未舉例說明之不同的欄位，訊號與架構。這些欄位，訊號與架構因其隨實施而變故未舉例說明，為熟悉該項技藝人士所瞭解，且將會使本說明書複雜而模糊本發明。

指令在追蹤緩衝器114A中等待直到它們最終退役或被捨棄(例如，因為，判定該線不在程式循序執行的一部份中)。如果指令佇列陣列202A在該追蹤內還有指令未執行時被塞滿，則指令不會被追蹤緩衝器114或重新配置/指派單元150所接收到，直到指令佇列陣列202A中一指令最終退役而其中有一列被解除指派。系統100裡各種不同陣列的項目可由首與尾指標的運作來指派與解除指派。

指令佇列陣列202A與下列各行程式碼合併描述：

```
I0: mul R1, R2 → R1
```



五、發明說明 (17)

```

I1: mul R3 , R4 → R2
I2: add R1 , R2 → R1
I3: add 10 , R1 → R4
I4: store    R2 → Mx
I5: store    R1 → My ,

```

這是在一線中的最前六個指令。很明顯在追蹤緩衝器114A之外的一追蹤緩衝器在程式順序中早於追蹤緩衝器114A。

"操作碼"欄位包含與特定指令有關的操作碼。"目標"，"來源1"，和"來源2"欄位識別指令的目的，來源1與來源2。"來源1的索引"欄位，識別追蹤緩衝器114A中包含來源的指令項目。例如，instr ID0的目標被用來當成instr ID2的來源1。因此，0被放在instr ID2的"來源1的索引"欄位內。instr ID2的目標被用來當成instr ID3的來源2。因此，2被放在instr ID3的"來源2的索引"欄位內。X標示為不管。

"有效1"和"有效2"欄位的位元，當instr ID之一個對應的來源參數已在先前由追蹤緩衝器114A中的線以外之指令所產生時被設定成一第一值(例如，邏輯0)，而在instr ID之該來源參數已在先前由線當中之指令所產生時被設定成一第二值(例如，邏輯1)。instr ID0的來源1(R1)由指令佇列陣列202A內之追蹤的外部產生。因此，instr ID0的有效1是一邏輯0。instr ID3的來源2來自instr ID2的目標。因此，instr ID3的有效2是一邏輯1。

指令I3包括將R1加到一常數"10"。該常數可與指令一起



五、發明說明 (18)

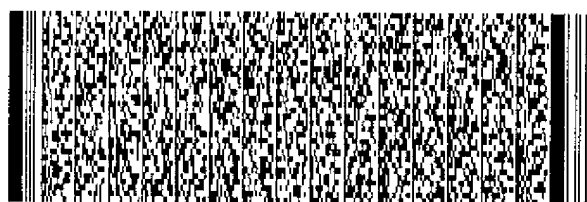
儲存在一特別的暫存器中(未顯示)，在來源1欄位中，或透過一些其他機制。在圖12中，X(不管)出現在instr ID3的來源1欄位中。或者，一些指示器可被放在來源1欄位中。

儲存緩衝器ID(SBID)欄位保存與儲存緩衝器裡一儲存指令有關的SBID，描述如下。載入緩衝器(LBID)欄位保存與載入緩衝器裡一載入指令有關的LBID項目，描述如下。SBID和LBID值由重新配置/指派單元150分配，並透過旁通導線126寫到指令佇列陣列。線ID號碼欄位可包含在指令佇列陣列202A中，但因其已隱含而並不需要。

b. DAD陣列206A與附屬性產生電路212A

參照圖13，DAD陣列206A之一實施例包括以一對一方式與指令佇列陣列202A中的instr ID項目對應之"instr ID"項目(列)。的確，指令佇列陣列202A與DAD陣列206A可能是相同陣列的不同部分。然而，在一些實施例中，指令佇列陣列202A與DAD陣列206A有不同的讀取埠。

DAD陣列206A包括一含有指令所產生的值或暫存器檔案152裡的PRID兩者之一的"值或PRID"欄位。值經由寫回單元162與寫回匯流排122與196從執行單元被寫回到追蹤緩衝器114A。"狀態"欄位，可能是二個位元，指出"值或PRID"欄位到底包含"值"或一"PRID"。在一實施例中，"值或PRID"可能並未保存一有效的"值"或一有效的"PRID"。"重演計數"欄位，獨一地識別一指令派遣，在每次相同instr ID的指令在管線108中重演時被增量。在一實施例



五、發明說明 (19)

之下，一指令可能在管線108中同時被重演一次以上。在此情形中，在一實施例下，只有與最高"重演計數"有關的資訊被寫回到DAD陣列206A。

"附屬性欄位"包括每個邏輯的暫存器之一位元。在圖13中，為了簡化，只有四個邏輯的暫存器(R1, R2, R3, 和 R4)表示出來。然而，數目可遠較此為大。在範例中，附屬性欄位項目被設定成1，以指出在追蹤的輸入值和指令項目之間存在一資料附屬性鍊結，與沒有附屬性時設定成0。附屬性欄位項目識別若收到一輸入值(例如當偵測到數值失誤猜測)時，追蹤內的哪個指令將需要被執行。

當指令被讀取，解碼，和寫入到追蹤緩衝器114A內，附屬性位元循序地被計算，並被寫入DAD陣列206A中。附屬性位元可能在判定是否要重演一指令之前就產生。圖13中的附屬性位元是六個指令I0-I5的，在上面的段落B.1.a中敘述。

附屬性欄位可透過機械方式產生。在描述此一種方式之前，其產生將以較直觀層次解釋。

i. 直觀層次

指令I0的結果只附屬於暫存器R1和R2。因此，1被放在instr ID0(它保存與指令I0有關的資訊)的R1和R2行，而0保持在R3和R4行。

指令I1的結果只附屬於暫存器R3和R4。因此，0被放在instr ID1的R1和R2行，而1在R3和R4行中。

指令I2的結果直接附屬於，分別在指令I0和I1中產生



五、發明說明 (20)

的，暫存器R1和R2。在指令I0中，R1附屬於追蹤一開始時R1與R2的值。在指令I2中，R2附屬於追蹤一開始時R3與R4的值。因此，指令I2間接地附屬於追蹤一開始時R1-R4的值，且1被放到instr ID2的R1-R4行中。

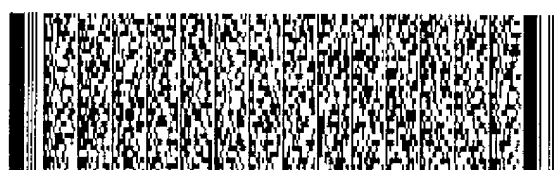
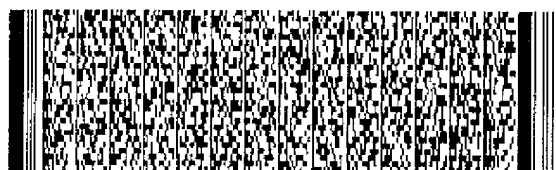
指令I3的結果直接地附屬於指令I2中所產生的暫存器R1。因此，指令I3間接地附屬於追蹤一開始時R1-R4的值，因為指令I2附屬於這些值，且1被放到instr ID3的R1-R4行中。

指令I4的結果直接地附屬於暫存器R2，其係在指令I1中產生。R2附屬於追蹤一開始時暫存器R3和R4的值。因此，0被放到R1和R2行之內，而1被放到instr ID4的R3和R4行之內。

指令I5的結果直接地附屬於暫存器R1，其係在指令I2中產生，指令I2附屬於追蹤一開始時暫存器R1-R4的值。因此，1被放到instr ID5的R1-R4行之內。

ii 一種機械的方式

以下是根據本發明的一個實施例可用來產生附屬性欄位的暫存器和演算法。參照圖14，附屬性產生電路212A包含暫時暫存器230，232，234，和236，每個邏輯暫存器一個，加上一額外的暫時暫存器240。暫時暫存器230，232，234，和236包含邏輯暫存器R1，R2，R3，和R4的修正元。修正過的暫存器240包含一組位元指示哪個邏輯暫存器將被追蹤裡的指令修改。暫存器230，232，234，236，和240每次有一新的指令被寫入到追蹤緩衝器內更



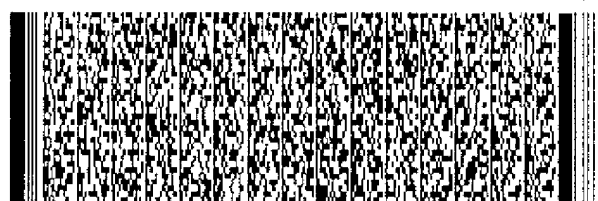
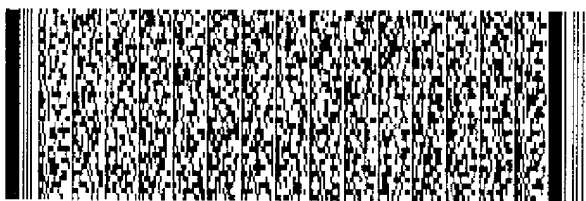
五、發明說明 (21)

新。暫存器之間的界限是任意的。舉例來說，他們可以全部在一聯合暫存器中。

對每個邏輯暫存器，都提供了一追蹤緩衝器位址暫存器，它指到追蹤緩衝器114A中要修改該邏輯暫存器的最後一個指令。修正過的位元，和最後修正元位址用來計算要寫入追蹤緩衝器114A內之下一個指令的附屬性位元。

要注意在此處所使用，修改暫存器僅表示將一值寫入暫存器內。它並不一定表示暫存器的內容與指令的結果不同。舉例來說，如果R1和R2的內容相乘(如其在指令I0之時)，且結果被寫入暫存器R1之內，R1的內容不必然會與指令I0的結果不同。舉例來說，R1的內容在指令之後將不會不同，若R1的內容是"0"，或R2在指令之前是"1"。

在圖16中，流程圖250代表為指令的每個來源參數(舉例來說，來源1和來源2)產生DAD陣列206A的附屬性欄位之演算法。在步驟252中判定在暫存器240中的相關位元是否被設定。如步驟254所描述，如果暫存器240中的位元未被設定，那麼與該暫存器有關的附屬性欄位的位元被設定成邏輯1。如步驟258所陳述，若暫存器240中的位元被設定，則使用由相關暫存器的修正元暫存器(230，232，234，或236)所產生的索引，讀取來源附屬性欄位。接著，如步驟262所述，使用邏輯或運算合併來源附屬性位元與目前指令附屬性位元。此種邏輯或運算由圖15中(其中多個位元以輸入表示)的或閘244說明。在執行圖16的演算法時，所提及之修正過的暫存器和修正元是就在一指令執行前所存



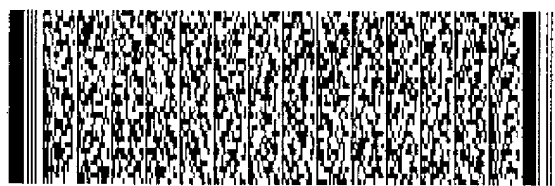
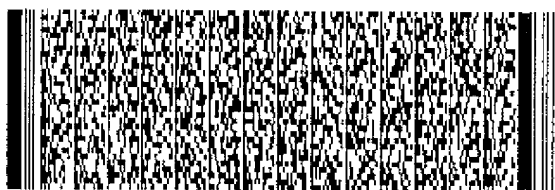
五、發明說明 (22)

在的那些。

有關於I0，在指令I0之前，暫存器240的R1，R2，R3，和R4是邏輯0，而暫存器230，232，234，和236的值是X(不管)。在步驟252之下，暫存器240裡R1和R2的修改過的位元都是0。因此，在步驟254之下，在DAD陣列206A的instr ID0列中，R1和R2的附屬性欄位的位元都被設成1。暫存器R3與R4不牽涉在內，而在instr ID0列中保持為0。指令I0修改了暫存器R1。因此，0被放到暫存器230之內，指出指令I0是最新近修改過暫存器R1的指令。在暫存器232，234，和236中的值維持X(不管)。暫存器240的R1位元被設成1，指出R1已經被追蹤裡面的一個指令修正過。

指令I1的附屬性欄位以指令I0相似的方式產生。修正過的暫存器240之R1邏輯暫存器欄維持設成1。邏輯1被放到修正過的暫存器240之R2欄之內。暫存器232中的1代表指令I1。

關於指令I2，在指令I2之前，在步驟252之下，暫存器240中R1和R2的修正過的位元都是邏輯1(也就是，設定成)。在步驟258之下，R1(230)和R2(232)的修正元暫存器，緊接在指令I2之前，被當成索引使用。暫存器230為指令I0包含一0值。DAD陣列206A的instr ID 0裡指令I0的附屬性欄位是0011。暫存器232為指令I1包含一1值。instr ID1裡指令I1的附屬性欄位是1100。在步驟262之下，0011與1100的邏輯或值是1111。因此，1111被放到DAD陣列 206A在instr ID2的附屬性欄位中。R1被指令I2



五、發明說明 (23)

修改。然而，一暫存器R1的1值已經在暫存器240中。一2值被放在暫存器230中，指出指令I2是最新近修改過指令R1的指令。

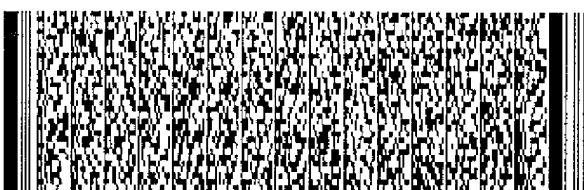
指令I3的附屬性欄位以指令I2相似的方式產生。邏輯1被加入到修改過的暫存器240的R4欄，而一3值表示指令I3放在暫存器236之內。邏輯或運算產生1111。

關於指令I4，在指令I4之前，在步驟252之下，暫存器240中R2的修改過的位元被設定成一1值。在步驟258之下，R2(232)的修正元暫存器，緊接在指令I4之前，被當成索引使用。暫存器232為指令I1包含一1值。在DAD陣列206A的instr ID1中，指令I1的附屬性欄位是1100。在步驟262下，1100(來自instr ID1的來源1)和0000(沒有來源2)的邏輯或運算結果是1100。因此，1100被放到DAD陣列206A的instr ID4列的附屬性欄位之內。

指令I5的附屬性欄位以指令I4類似的方式產生。指令I5和I6修改外部記憶體位置，且不改變暫存器230，232，234，236，或240。

附屬性資訊可被排程/發送單元156使用，或者排程/發送單元156可只求得其自己的附屬性資訊。

在重演中一序列或一串指令有不同的方式由追蹤緩衝器114A發送出來。一種方式是依序讀取追蹤緩衝器，並摘錄附屬性位元被設定之指令，且將其送出重演。然而，零可能會在管線中造成泡沫效應。另外一個方式藉由送出指令以執行/重演之前，將邏輯值封裝以除去泡沫。參照圖



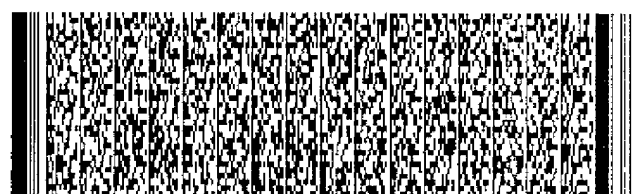
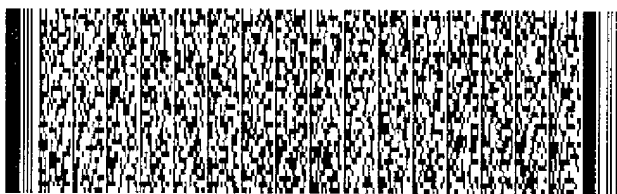
五、發明說明 (24)

17，另外一個方式包括一些額外硬體，包括每個邏輯暫存器的陣列268。陣列268包括附屬於暫存器R1之指令的instr ID值。陣列268中的值當作指令佇列陣列202A中整個instr ID項目的指標。如此允許對指令緩衝器非常快遠的讀取。一次讀取一個指令區塊(也許2或4個指令)。追蹤緩衝器114A可以是多埠的，且有四個解碼器，並將由暫存器陣列所取得之指標的每一個傳遞到該等解碼器中，且指令I0，I2，I3，和I5能在一個周期內被讀取。暫存器R1陣列可在附屬性欄位產生時，在重演開始之前，組成。間接的層次促進高頻寬重演。

c. 輸出暫存器檔案210A和輸入暫存器檔案208A

追蹤緩衝器114包括偵測電路以檢測某些失誤猜測。根據本發明的一實施例，每個追蹤緩衝器有一保存相關線的暫存器內文之輸出暫存器檔案，與一輸入暫存器檔案以接收依照程式順序立刻要進行之線的暫存器之內文。暫存器內文是邏輯暫存器的內容或其狀態。輸出暫存器檔案內容的改變時常被更新，可能是每次暫存器中一有變化。輸入暫存器檔案的內容只在比較之後才被更新，描述如下。

圖18和19說明一輸出暫存器檔案208A(在追蹤緩衝器114A中)和一輸入暫存器檔案208B(在追蹤緩衝器114B中)的實施例，雖然可使用其他實施例。輸出暫存器檔案208A與輸入暫存器檔案210B包括一值或PRID欄位及一狀態欄位。狀態欄位指示在值或PRID欄位中是否保存一有效的值或一有效的PRID。在一實施例中，會有一有效的值或一有



五、發明說明 (25)

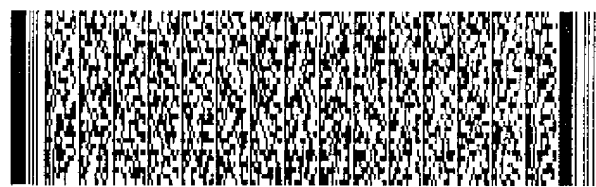
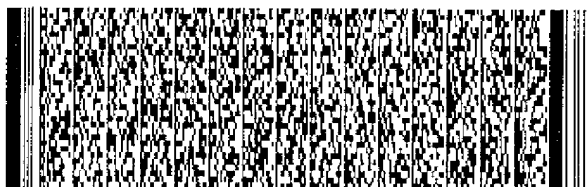
效的PRID兩者之一。在另外一實施例中，可兩者皆無，在該情況中附屬於輸入暫存器檔案的指令可等到有其中之一。

注意在上面所描述的範例中指令I0涉及暫存器R1和R2，兩者都不是先前包括指令I0的線裡面的目標。然而，暫存器R1和R2的值或PRID可以由輸入暫存器檔案208A取得以在指令I0的執行中使用。

參照圖20，比較器280B比較當前線的輸入暫存器檔案208B(在追蹤緩衝器114B中)的內容，與依照程式順序中立刻要進行之線的輸出暫存器檔案210A(在追蹤緩衝器114A中)的內容。此比較可在立刻要進行的線執行結束時做，或在前一線的原來執行期間做。比較也在前面的線之退役結束時做。在一實施例中，比較只在前面的線之退役結束時做。

各種不同的事件可觸發比較器280B的比較。比較被用來偵測失誤猜測。如果在輸入和輸出暫存器檔案之間有差異，則立刻要進行之線的輸出暫存器之一或更多值已經改變。為了因應，輸入暫存器檔案208B被更新，且重演觸發邏輯284B使那些被影響的指令以改變過的暫存器值重演。附屬性欄位可被重演觸發邏輯284B使用。不保證被改變的值是那些最終的正確值(也就是，在一純粹循序處理器中會產生的暫存器值)。那些指令可能需要再一次被重演，也許數次。

在一實施例中，線的偵測電路包括一輸出暫存器檔案，

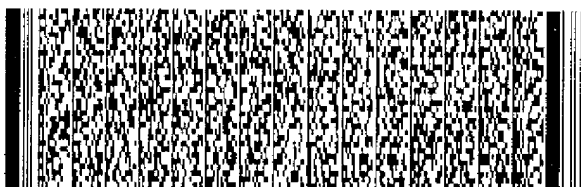


五、發明說明 (26)

一輸入暫存器檔案，一比較器及相關控制電路；以偵測，保存在包含輸入暫存器檔案的追蹤緩衝器中，指令內的某些猜測錯誤。在其他實施例中，偵測電路可包括一些不同的電路。

例如，參照圖21，線T2是目前的線，且與追蹤緩衝器114B有關。線T1是緊接在線T2之前的線，且與追蹤緩衝器114A有關。線T1包括一函數呼叫，該函數，和由函數呼叫的返回。線T2的執行緊接在函數呼叫之後開始。輸出暫存器210A的內容當其在函數呼叫當中存在時，被複製到輸入暫存器檔案208B中。線T2的指令依據輸入暫存器檔案208B中暫存器內文猜測地執行。在返回指令當時，輸入暫存器檔案208B的內容被比較器280B與輸出暫存器檔案210A的內容作比較。如果有差異，輸入暫存器檔案208B被更新，而線T2裡那些被影響的指令被重演。比較也可在一或更多的中間時間做。這可藉由更平均地分配指令的重演而幫助避免瓶頸，但可能引起額外的重演若，舉例來說，輸出暫存器檔案的內容在函數期間改變一次以上。在當中輸出暫存器檔案經常地改變，它可能需要一中間緩衝器接收輸出暫存器檔案210A的內容。然後比較可在中間緩衝器的內容和輸入暫存器檔案208B中間進行。

如圖8和10中所說明，暫存器內文透過導線216在輸出暫存器檔案和輸入暫存器檔案之間傳遞。導線216連接每個輸入暫存器與每個，可保存緊接在前的線之追蹤的，追蹤緩衝器的輸出暫存器檔案。如果能保證程式順序總是會跟



五、發明說明 (27)

隨特定追蹤緩衝器的順序，則導線216的配置可相當單純。輸出和輸入暫存器檔案可由圖10和11中顯示的控制電路224A來控制。

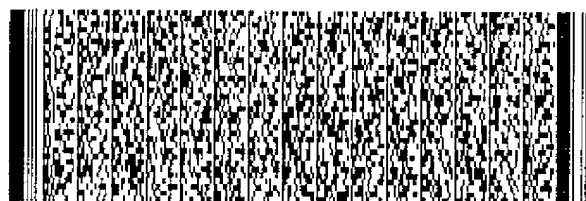
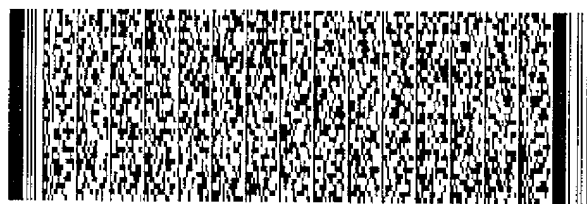
因為輸出和輸入暫存器檔案會提供一值或一PRID，在接收一輸入暫存器中之內容與可以使用來自輸入暫存器檔案之一暫存器當一來源參數執行指令之間可能有非常小的延遲。如果不能取得一值，暫存器檔案152的PRID可用在管線108中的執行。

可預見許多指令將會被重演數次，當正確的來源參數藉著各個線的暫存器檔案運作時。然而，也可預見對許多程式而言，大量指令將完全不需要重演，或相對很小的次數，使每一單位時間中正確地執行的指令實質增加，並使執行一程式所需要的總時間減少。

2. 追蹤緩衝器114'

參照圖11，追蹤緩衝器114A'類似於追蹤緩衝器114(在圖10中)。然而，在追蹤緩衝器114A'中，附屬性欄位是在附屬性產生與解碼電路218A決定一指令要被重演之後產生。若要重演指令的發送和附屬性的判定以管線方式執行，這可能引起重演中一些起始延遲，一旦處理開始可能會有一點點額外的延遲。

在一實施例中，附屬性產生與解碼電路218A只保存一個欄位給附屬性資訊。(在圖13中，有四個欄位。)該相同欄位可被重複使用。舉例來說，在附屬於暫存器R1之指令的重演期間，該欄位可用來列出附屬於暫存器R1的指令。在

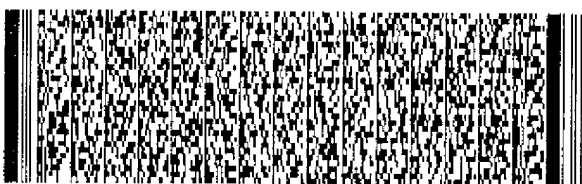


五、發明說明 (28)

附屬於暫存器R2之指令的重演期間，該相同欄位可用來列出附屬於暫存器R2的指令，等等。附屬性產生與解碼電路218A可以只包括一修正元欄位和一修正元暫存器。(在圖14中，有四個)或者，附屬性產生與解碼電路218A可包括多個附屬性欄位與暫存器。附屬性產生與解碼電路218A一次只能判定幾個指令的附屬性。

資料陣列214A包括一值或PRID欄位，一狀態位元欄位，和重演計數欄位給每個instr ID項目(如圖10和13的DAD陣列206A中)。或者，資料陣列214A的內容可放到附屬性產生與解碼電路218A中而使資料陣列214A並非必要。有二個有利的理由維持資料陣列214A與附屬性產生與解碼電路218A分開。首先，它們可能涉及不同的讀取埠。其次，在一實施例中，附屬性產生與解碼電路218A並沒有像指令佇列陣列202A與資料陣列214A一樣多的列數。換句話說，在一實施例中，附屬性產生與解碼電路218A重複使用列，正像它可能重複使用附屬性欄位一樣。當然，有許多可能性。

如以下所將更詳細描述，MOB 178在載入指令將被重演的時候透過導線194發出訊號。一具有附屬性欄位的陣列(像圖13中R1的)可能產生，以列出附屬於將被重演之載入指令上的指令。然而，對一載入指令而言，附屬指令的表列由載入指令開始，而非如暫存器的情況中由追蹤裡的第一個指令開始。載入指令的附屬性欄位可能在附屬性產生與解碼電路218A中(在圖11中)。(當然，其他追蹤的載入



五、發明說明 (29)

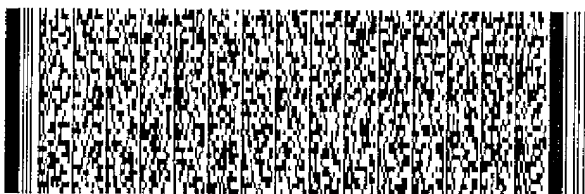
指令將會從其他的追蹤緩衝器重演。)在一實施例中，附屬性產生與解碼電路218A被用來當成載入指令和暫存器兩者的附屬性欄位。相同欄位可用做兩者。在另一實施例中，暫存器的附屬性欄位在DAD陣列206A中，且載入的附屬性欄位在附屬性產生與解碼電路218A中。

還有另一實施例，載入指令被完整重演(也就是，所有跟在載入之後的指令都被再執行)，因此附屬性欄位並不需要。

C. 重演順序演算法

當重演觸發邏輯(如重演觸發邏輯284B)判定一來源參數(或其他的輸入值)被失誤猜測，它便觸發對應的追蹤緩衝器(如追蹤緩衝器114B)發派直接或間接附屬於管線108中要重演之失誤猜測的來源參數之指令。那些直接或間接附屬的指令可由追蹤緩衝器中之DAD陣列的附屬性欄位識別，或透過如圖13中另一陣列來識別。

被識別出的指令，從追蹤緩衝器中以指令存在於追蹤緩衝器內之順序(其為程式順序)被發派執行。舉例來說，instr ID0項目中的指令在instr ID1項目中的指令之前或在相同時間被發派。然而，指令可能在排程/發送單元156的控制之下無序地執行，如在任何無序處理器中一樣。控制位元被附加到從追蹤緩衝器中被發派的指令，以指示重新配置/指派單元150是否要(1)進行暫存器重新配置，(2)略過重新配置/指派單元150中重新配置別名的表列搜尋，而改以使用來自對應追蹤緩衝器的PRID，或(3)完全略過



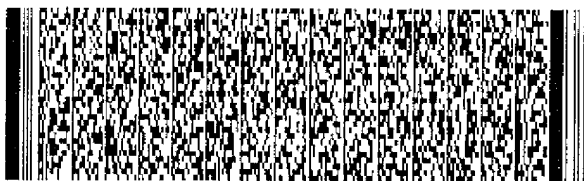
五、發明說明 (30)

重新配置並使用來自DAD陣列的值，就像指令中的常數參數一樣。

如與圖8一起解釋的，當instr ID的一個對應的來源參數已經由一追蹤緩衝器114A中的線以外的指令所產生(舉例來說，目的地)時，"有效1"和"有效2"欄位是設定成第一值(舉例來說，邏輯0)的位元，而在instr ID的來源參數已經由線中之一指令所產生時設成第二值(舉例來說，邏輯1)。從追蹤緩衝器114A發派出來重演的指令可依下列決定其來源參數：

(1) 有效的位元1。如果在指令佇列陣列202A中的有效位元被設成一邏輯1，來源參數被用來讀取DAD陣列206A中對應的值或PRID。如果DAD陣列狀態欄位的值位元與PRID位元都不是有效的，表示來源參數暫存器仍然還沒有被重新配置。在此情況下，指令及具有邏輯零值的值與PRID狀態位元經由導線120和MUX 110一起被發派，讓重新配置/指派單元150執行別名表列搜尋(暫存器重新配置)如其一般所做。如果PRID或值是有效的，它經由導線120和MUX 110連同指令一起傳遞到重新配置/指派單元150，以因應略過重新配置階段。

(2) 有效的位元0。如果來源參數的有效位元被設成一邏輯0，輸入參數來自追蹤之外。來源暫存器名稱用來存取輸入暫存器檔案208A。來自輸入暫存器檔案208A的值或PRID連同指令一起被傳送到重新配置/指派單元150，以因應略過重新配置階段。



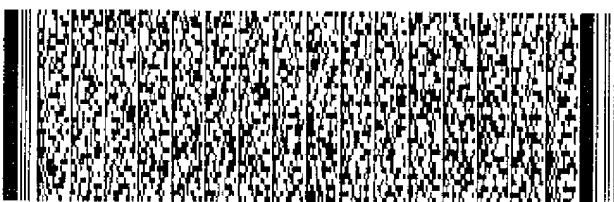
五、發明說明 (31)

有效位元究竟是0或1，對每個被發派的指令而言，在DAD陣列206A中的值與PRID狀態欄位的位元被重設成或維持邏輯0。這達成二個目的。首先，它確保在PRID被複製到來自重新配置階段的項目內之前所發送之較後面的附屬指令，將會被允許由重新配置別名表列中重新配置，以避免使用來自追蹤緩衝器114A之陳舊的PRID。其次，它也確保指令不會退役，直到最後一個執行事件被寫回，從而允許指令只有當所有資料失誤猜測都已被更正的時候才退役。

D. 第二層或最終退役

指令在所有先前的線之所有指令都退役，且屬於指令的所有重演事件都已經被服務過後，才最終地由追蹤緩衝器114中退役。陳述另外一種方法，指令在能確保其已經以正確的來源參數執行過的時候，才最終地退役。線依照順序退役。舉例來說，直到所有早先的線都已經退役(也就是，所有早先線的指令都已經退役)之前，線X中的指令不能退役。在線中的指令依照順序退役，雖然所有預備好退役的指令可同時地退役。

最終退役由最終退役邏輯134所控制。在本發明的一實施例中，最終退役包括(1)委託結果到循序暫存器檔案，(2)服務中斷，例外，及/或分支失誤猜測；(3)追蹤緩衝器與MOB 178資源項目的解除指派；和(4)向MOB發訊號將儲存標註成退役的，並將其發送到記憶體。將項目解除指派可能包括移動一首指標。如下所述，在MOB 178中的儲存指



五、發明說明 (32)

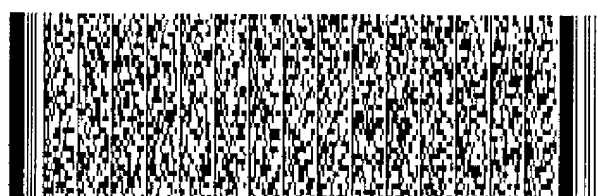
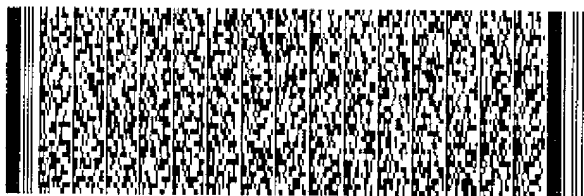
令不會被解除指派，直到它確定相關的資料被複製到資料快取176或其他的記憶體之後。關於MOB 178中載入與儲存指令最終退役的細節描述如下。

E. 記憶體系統

圖22說明圖2的MOB 178之一實施例，包括MOB 178A, 178B, ..., 178Y，其中Y代表MOB的數目，且符合追蹤緩衝器114的數目。MOB 178A, 178B, ..., 178Y分別保存追蹤緩衝器114A, 114B, ..., 114Y內追蹤的載入和儲存指令之附本。載入指令被保存在載入緩衝器182A, 182B, ..., 182Y中。儲存指令被保存在儲存緩衝器184A, 184B, ..., 184Y中。導線292代表攜帶訊號往返MOB 178之各種不同導線。重演導線194將來自MOB 178的訊號提供給追蹤緩衝器114，警告追蹤緩衝器114一載入指令應被重演。控制電路302執行多種控制函數。

1. 儲存緩衝器與載入緩衝器

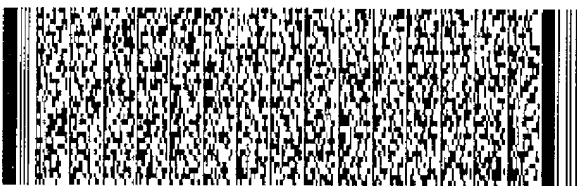
圖23說明儲存緩衝器184A之一實施例，它代表儲存緩衝器184B, ..., 184Y。各種其他實施例都可使用。儲存緩衝器184A包括儲存緩衝器項目的列中之各種不同欄位。每個項目由一儲存緩衝器ID(SBID)識別。重新配置/指派單元150在每個儲存指令第一次被讀取與執行的時候，指派一SBID項目給它，但重演時不會。儲存指令直到最終退役之前有相同的SBID值。舉例來說，在圖23中，項目SBID 0被指派給指令STORE 0。項目SBID 1被指派給指令STORE 1，等等。一LBID欄位保存一"store LBID"值描述如下，在圖



五、發明說明 (33)

23 中說明。在一實施例中，當指令佇列陣列202A(在圖12中)之一個項目保存儲存指令的時候，指令佇列陣列202A的SBID項目保存用以識別儲存緩衝器184A中保存著儲存指令之項目的SBID值，而LBID欄位替該儲存指令保存store LBID值，如果有的話。SBID值和store LBID值透過管線108與儲存指令相隨。在該實施例中，LBID欄位也可能不包含在儲存緩衝器184A中。

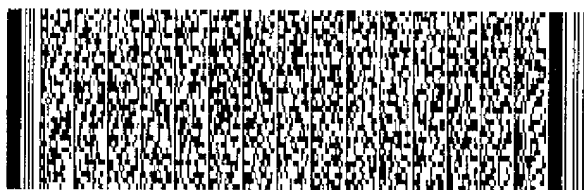
instr. ID欄位保存著指令佇列陣列202A中儲存指令的指令ID。線緩衝器ID是暗指儲存緩衝器184A和追蹤緩衝器114A兩者。操作碼欄位保存著儲存指令的操作碼。儲存位址欄位保存著儲存指令所指向的位址。在舉例說明的實施例中，位址由AGU 172產生。SB位址有效欄位包含一指出儲存位址是否為有效的位元。資料欄位保存將被儲存的資料。資料有效欄位包括一指出資料是否有效的位元。可以使用分開的位址和資料有效位元，因為有效位址可能與有效資料在不同的時間到達。位址和資料兩者在儲存指令被執行之前到達。在一實施例之下，資料包含在指令的一部份中。退役欄位包括一在最終退役邏輯134指示儲存指令應該退役時被設定的位元，而該位元在收到來自記憶體對記憶體的儲存已完成之確認時，被重新設定。載入和儲存的退役在下面討論。重演計數欄位包括一重演計數數字(且類似圖13裡的DAD陣列206A的重演計數欄位)。重演計數欄位並非必要。在一實施例之下，儲存指令每次只能重演一次，且沒有重演計數欄位。



五、發明說明 (34)

圖24舉例說明一載入緩衝器182A的一實施例，它代表載入緩衝器182B, ..., 182Y。各種不同的實施例都能夠使用。載入緩衝器182A包括載入緩衝器項目的列中之各種欄位。每個項目由一載入緩衝器ID(LBID)識別。重新配置/指派單元150在每個載入指令第一次被讀取與執行的時候，指派一LBID項目給它，但重演時不會。載入指令在最終退役之前有相同的LBID值。舉例來說，在圖24中，項目LBID 0被指派給指令load 0。項目LBID 1被指派給指令load 1，等等。(LBID項目數值與SBID欄位可稱為一MOBID)。SBID欄位保存"load SBID"值，描述如下，在圖24中說明。在一實施例中，當指令佇列陣列202A(在圖12中)之一個項目保存載入指令的時候，指令佇列陣列202A的LBID項目保存用以識別載入緩衝器182A中保存著載入指令之項目的LBID值，而SBID欄位替該儲存指令保存load SBID值，如果有的話。LBID值和load SBID值透過管線108與載入指令相隨。在該實施例中，SBID欄位也可能不包含在載入緩衝器182A中。

instr ID欄位保存指令佇列陣列202A中載入指令的指令ID。線緩衝器ID暗指載入緩衝器182A和追蹤緩衝器114A兩者。操作碼欄位保存載入指令的操作碼。載入位址欄位保存載入指令由何處載入的位址。項目有效欄位包括一指出該項目被一有效的載入指令佔用的位元。在說明的實施例中，不包括位址有效欄位，因為位址已經由AGU 172產生。PRID欄位保存一來自重新配置/指派單元152的PRID



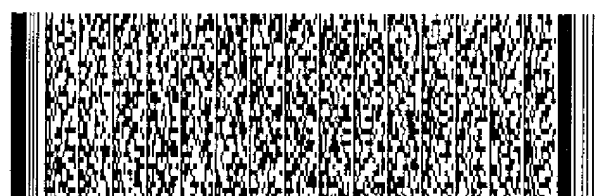
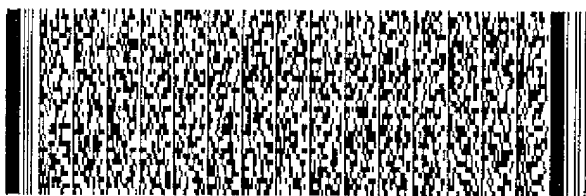
五、發明說明 (35)

值，它指出暫存器檔案152中載入指令的目標。SB Hit，SBID，線ID，與重演計數欄位(如果有)可視為狀態欄位的一部份，並與儲存指令的執行一起在下面描述。

在重新配置/指派單元150第一次接收到儲存和載入指令時，儲存和載入指令的項目被指派在儲存緩衝器184和載入緩衝器182中，而接收所載入值之暫存器的項目被指派在暫存器檔案150與ROB 164中。這些項目並不遵從第一層次退役，而是像追蹤緩衝器114裡的項目，維持指派直到最終退役。因此，項目在重演時不重新指派。如果儲存或載入緩衝器滿了，來自I-cache 104的儲存或載入指令將不會通過重新配置/指派單元150，直到一項目被釋放出來。然而，由追蹤緩衝器中被重新執行的載入或儲存指令將會通過重新配置/指派單元150。

2. 載入和儲存位址的比較

參照圖5，依照程式順序，線T1中的store MX在線T2中的load MX被執行之前執行。然而，因為同時執行，依照時間順序，store MX可能在load MX之前或之後執行。如果store MX依照時間順序在load MX之前執行，那麼load MX的猜測性執行將會依照store MX相關之正確順序。如果依照程式順序所有在store MX之前的指令都已經退役，可以確定load MX將會由記憶體位置MX載入正確的值。正確的值是如果線由一循序處理器執行時將已經被載入的值。如果依照程式順序並非所有在store MX之前的指令都已經退役，那麼store MX的資料總是有可能是不正確的。



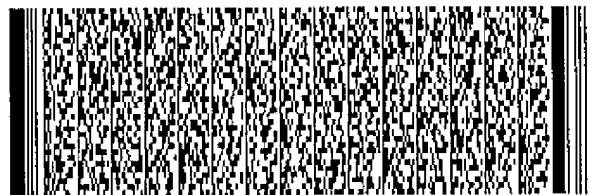
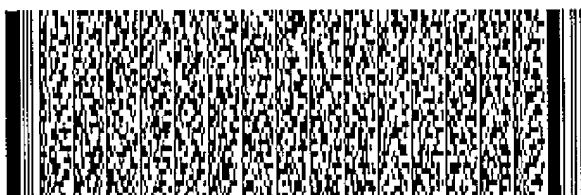
五、發明說明 (36)

相反的，如果依照時間順序store MX在load MX之後執行，那麼load MX的猜測性執行不會依照store MX相關之正確順序，而且不確定load MX將會載入正確的值。正確的值出現在記憶體位置 MX中(或儲存緩衝器項目的資料欄位保存store MX直到store MX最終退役)只會是巧合。為了要確保執行的終極正確性，MOB 178包括各種機制以確使記憶體資料在線之間的一致性。

a. 載入指令的執行

在載入指令執行之前，它的位址與儲存指令的位址一起比較以決定哪一，如果有的話，儲存指令是最接近先前相符儲存指令(CEMSI)。“相符”指的是與載入指令有相同的位址。“先前”指的是CEMSI在程式順序中在載入指令之前。“最接近”指的是在CEMSI和將被執行的載入指令之間沒有其他相符的儲存指令。如果只有一個先前相符儲存指令，它就是CEMSI。

如果有CEMSI，載入指令由CEMSI的資料欄位讀取它的資料。如果沒有CEMSI，載入指令由記憶體取它的資料，例如資料快取176，L2快取，或主記憶體。來自儲存緩衝器184或記憶體的資料經由MUX 192傳送，並寫到追蹤緩衝器114中由線ID和instr ID所指定的項目中。資料也可能被寫到暫存器檔案152中PRID所指定的暫存器中。資料也可能依快取規則(舉例來說，寫回，寫過等)被儲存在資料快取176中。因為能略過記憶體，例如資料快取176，L2快取，或主記憶體，所以MUX 192是一旁路。



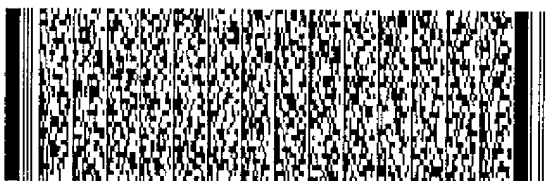
五、發明說明 (37)

在一實施例中，一個不同的比較器與每一個儲存緩衝器184的每個項目聯合，以比較將被執行的載入之位址與儲存指令之位址。圖25裡的比較器320是一範例，並接收載入指令位址和儲存緩衝器184A中項目SBID 1的儲存位址。來自其他比較器的導線322和輸出導線被連接到MOB控制電路302。

載入SBID指向與將被執行的載入指令有關的最接近先前儲存指令(CESI)的LBID。CESI是在儲存緩衝器中與載入指令有相同線ID者。如果有CEMSI，它不是CESI就是在程式順序中比CESI早者。重新配置/指派單元150記錄程式中儲存與載入指令的順序，並提供SBID和LBID值。它們可透過導線126寫到追蹤緩衝器114。在一實施例之下，如果沒有與載入指令有關之CESI，那麼沒有與該指令相關之載入SBID。這在追蹤內的第一個記憶體指令是載入的時候發生。各種技術都可用來處理這種情形，包括重新配置/指派單元150送出某些訊號以指出沒有有效的載入SBID。陣列翻轉位元，描述如下，可如此使用。

考慮下列程式順序中之儲存與載入指令：

```
store 0
store 1
load 0
store 2
load 1
store 3
```



五、發明說明 (38)

store 4

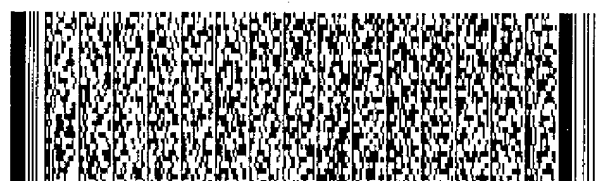
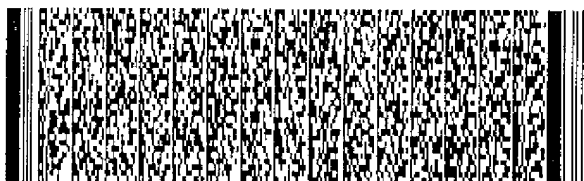
load 2.

在LBID欄位中的store LBID值在儲存緩衝器182A中說明。在SBID欄位中的load SBID在載入緩衝器184A中說明。舉例來說，LBID項目1的SBID欄位中的2指出在儲存緩衝器184A的項目SBID 2之儲存指令保存與LBID項目1中之載入指令相關之CESI。指令store 0，store 1，store 2，及load 0比load 1舊或早。指令store 3，store 4，及load 2比load 1新或晚。

控制電路302有各種方法可決定哪一個，如果有的話，儲存指令是CESI。這些方法的例子與圖27一起討論，其中儲存緩衝器184A，184B，184C，和184D是MOB178中僅有的儲存緩衝器，且分別與線A，B，C，和D有關。假定程式順序是線A，然後線B，然後線C，然後線D。在該範例中，要被執行的載入指令在載入緩衝器182C中。有一CESI，它在儲存緩衝器184C中。

導線342，344，346，和348是各種比較器的輸出導線。導線362，364，366，和368提供使那些比較器能夠執行比較的控制訊號。在不同的實施例中，控制電路302致能(1)對每個儲存緩衝器中的所有項目的比較器，(2)只有在儲存緩衝器中，有跟依照程式順序，與載入指令的線ID相同或更早之線ID相關的那些比較器，或(3)只有在程式順序中比載入指令更早的項目有關的那些比較器。

相符判定邏輯356決定哪一，如果有，儲存指令是

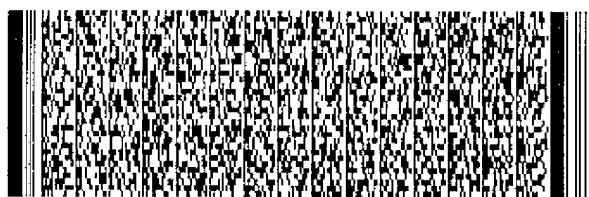


五、發明說明 (39)

CEMSI。在圖27中，在儲存緩衝器184C上面部分的store MX指令是CEMSI。如果那個store MX指令不在儲存緩衝器184C中，那麼CEMSI將會是在儲存緩衝器184B裡的store MX指令。當比較器和相符判定邏輯356在決定是否有CEMSI的時候，可能在資料快取176(和其他記憶體)中搜尋，以備如果沒有CEMSI。相符判定邏輯356包括資料路徑控制邏輯390，它在導線370之上提供訊號以控制MUX 192是否由記憶體或儲存緩衝器傳送資料。

在一種方式中，MOB控制電路302有二個優先順序要決定。一個可能是要決定儲存緩衝器內儲存指令的優先順序。另外一個可能是決定儲存緩衝器的優先順序。可以決定依照任一順序。進位鏈架構可用來決定儲存緩衝器內的優先順序。舉例來說，在一實施例中，除了與載入指令有相同線ID的那一個儲存緩衝器之外，每個儲存緩衝器被判定哪一，如果有，相符儲存指令依照程式順序是最新的。對與載入指令有相同線ID的儲存緩衝器而言，它被判定哪一，如果有，相符指令依照程式順序最接近CESI(包括與其相同)。然後，它被判定儲存緩衝器中哪一個的相符指令在程式順序中與載入指令的線ID最接近。

儲存緩衝器184可以是每個都有頭部和尾部的循環陣列。一開始，帶有較大SBID值的儲存指令比較新。然而，當儲存項目被解除指派並指派，尾部最後會翻轉，所以頭部指向一比尾部所指向的更高之SBID項目。在一實施例中，翻轉位元在尾部由最高SBID值走向最低的值時被變



五、發明說明 (40)

動，並被提供給最接近相符判定邏輯356。

b. 儲存指令的執行

當儲存指令執行的時候，它的位址與載入指令的位址一起比較以決定，如果有，在程式順序中比較後面的哪一載入指令(來自相同或更新的線)有跟儲存指令相同的位址。被store SBID所指出之一最接近較遲載入指令(CLLI)指定可被考慮的最早載入指令。

在一實施例中，一個不同的比較器與每一個載入緩衝器182的每個項目聯合以做比較。其中之一是比較器324，在圖26中說明。僅係做為範例，比較器324與載入緩衝器182A的項目LBID 1有關。比較器324在一個輸入接收儲存指令的位址，而在另一輸入接收載入緩衝器182A裡項目LBID 1的載入位址欄位中之位址。輸出導線326上的訊號象徵該等位址是否相同。導線326以及來自其他比較器的輸出導線被連接MOB控制電路302。比較器(例如比較器324)也可以把儲存指令的狀態位元與載入緩衝器中之狀態位元作比較，如下所述。

圖28與圖27類似。然而，在圖28的載入緩衝器182A-182D中之載入指令位址與將被執行的儲存指令的位址一起比較，且相符判定邏輯356決定是否要重演載入指令。在一實施例中，相符判定邏輯包括重演觸發邏輯394，它在導線194之上提供訊號，以指示追蹤緩衝器那些載入指令要重演。在一實施例中，相符判定邏輯356由CLLI開始考慮載入指令與儲存指令的相符。不同的演算法



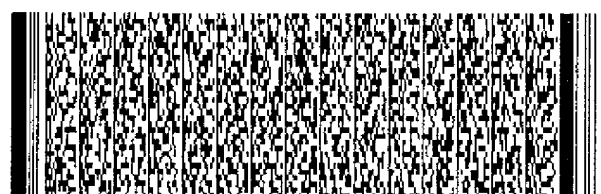
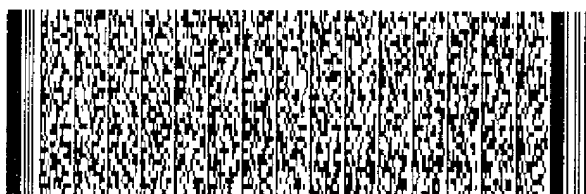
五、發明說明 (41)

都可被使用。線管理邏輯124指出在程式順序中那些線ID比正被執行中之儲存指令的線ID更新者。在一實施例中，所有比較器都被致能。在另一實施例中，只有在載入緩衝器中依照程式順序有線ID跟載入指令的線ID相同或更新的導線被致能。還有另一實施例中，只有在載入緩衝器中與CLLI及更新的指令有關之導線被致能。要考慮的線可以在之前，之後，或在判定期間中決定依照程式順序載入緩衝器裡面的那個載入指令比儲存指令更新。

在一實施例之下，用來偵測在載入指令的執行中的一些猜測錯誤之偵測電路，包括與載入緩衝器相關的比較器，相符判定邏輯356的部分，與相關之控制電路。在其他實施例中，偵測電路可包括一些不同電路。偵測電路對猜測錯誤的偵測並不需要在執行管線中。不同的相符判定邏輯可與資料路徑控制邏輯及重演觸發邏輯連接一起使用。

i. 有位址相符的情形

那些有一位址相符之較新指令的狀態欄位(SB hit, SBID, 線ID, 重演計數(如果用了))，在判定中被考慮是否要重演。狀態欄位指出載入指令是否由記憶體(舉例來說，資料快取176)或儲存緩衝器的資料欄位取得它的資料。SB hit欄位有，舉例來說，一0值如果資料來自記憶體，和一1值若資料來自儲存緩衝器。SBID與線ID欄位保存資料所來自之儲存指令的SBID值和線ID值。儲存指令的線ID不一定是位址相符之載入指令的線ID。載入指令的線ID隱含在載入緩衝器中。重演計數欄位(如果用了)指出



五、發明說明 (42)

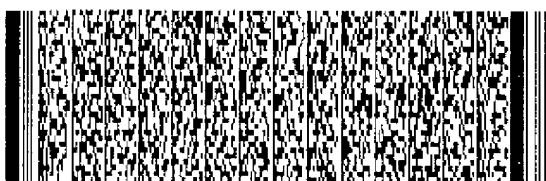
包括了哪個重演。(如果SB hit是0, SBID, 線ID, 和重演計數欄位裡的資料是無意義的)。

如果 SB Hit = 0(來自記憶體中的先前資料), 載入緩衝器經由導線194, 通知載入指令之線ID所識別的追蹤緩衝器有一重演事件, 而該載入指令與所有的附屬指令由追蹤緩衝器中被重演。instr ID和線ID經由導線194傳送以指示哪個指令被重演。

如果 SB Hit = 1(來自儲存緩衝器中的先前資料), SBID欄位, 線ID欄位, 和重演計數欄位(如果過了)中的值控制是否觸發一重演。在第一種情形中, 特定載入指令的狀態欄位之線ID等於儲存指令的線ID, 而特定載入指令的狀態欄位中之SBID符合儲存指令的SBID。在此第一種情形中, 如果載入指令的重演計數比狀態欄位中之重演計數大, 載入指令會被重演。如果沒有重演計數(因為儲存指令每次只能被重演一次), 那麼載入指令會被重演。

在第二種情形中, 狀態欄位中的線ID等於儲存指令的線ID, 但是狀態欄位裡的SBID不符合儲存指令的SBID。在該第二種情形中, 如果狀態欄位中的SBID比儲存指令的SBID小則載入指令會被重演, 但若狀態欄位中的SBID比儲存指令的SBID大則不會被重演。

在第三種情形中, 狀態欄位和儲存指令的線ID不相符。可預見這是稀少的情形。為了單純, 在一實施例之下, 載入指令被重演(即使它可能與程式順序相反)。它可能是錯誤的重演。載入指令在重演的時候將會收到正確的儲存



五、發明說明 (43)

資料。其他的方式能夠被使用，但是它們對此一稀少的情形可能甚為複雜。

ii. 沒有位址相符的情形

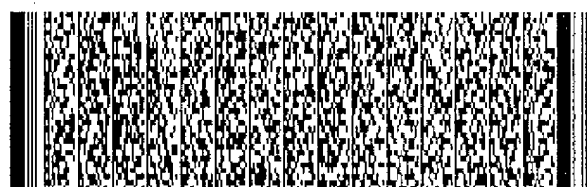
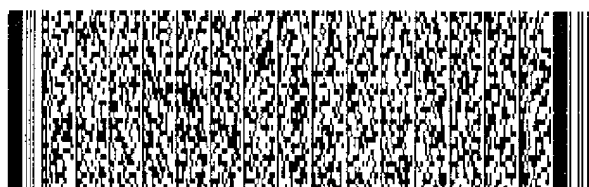
如果位址不相符，除了在下述稀少的之外不會引發重演。如果 $SB\ hit = 1$ ，狀態欄位的線ID與儲存指令的線ID相符，而狀態欄位的SBID與儲存指令的SBID相符。在此情況中，會重演，且被重演的載入指令會由一新的項目或記憶體接收它的資料。

c. 重新設定

一線在該線被判定其不在程式順序中時被重設。然而，來自其他線的載入可能已經由該線中與儲存指令相關之資料欄位內取走資料。線管理邏輯124送出一訊號給控制電路302。在一實施例中，當線被重設的時候，該重設之線的線ID與每一個載入緩衝器中的每一個載入比較（除了對應到重設的線之載入緩衝器）。載入指令的重演被觸發，其中狀態欄位中之線ID與被重設之線的線ID相符。載入指令由相關的追蹤緩衝器中被重演。

3. 儲存指令的重演

如上所述，載入指令對應儲存指令的執行而重演。在一實施例中，儲存指令因追蹤緩衝器中暫存器的比較指出有一暫存器值已經改變而被重演。舉例來說，參照圖12和13，在追蹤緩衝器114A中的instr ID 4和5，是儲存指令，顯示附屬於暫存器R1-R4。

4. 多重載入指令的重演

五、發明說明 (44)

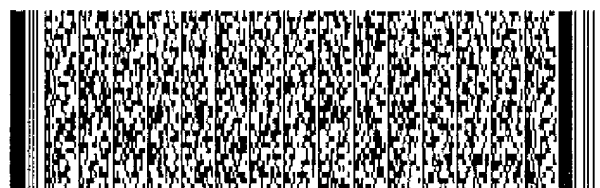
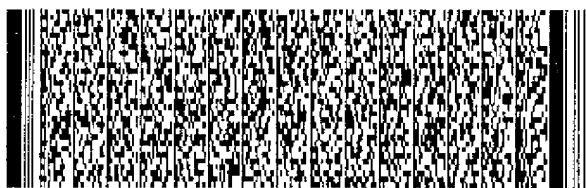
有可能載入緩衝器中一個以上的載入指令有跟儲存指令相符的狀態欄位。為了要避免複雜的邏輯，一個方式是控制電路302偵測何時有多個載入位址相符，並使在追蹤內最早載入之後的所有指令被再執行。

5. 載入與儲存指令的最終退役

當載入或儲存指令要最終退役的時候，最終退役邏輯134提供訊號給追蹤緩衝器114與MOB 184，指出一指令要最終退役。在追蹤緩衝器中的項目(由instr ID和線ID所標識)被解除指派。在載入指令的情況下，載入緩衝器中線ID和LBID所標識的項目被解除指派。在載入指令的情況下，最終退役完成。在儲存指令的情況下，在解除指派之前，資料欄位中的資料必須委託給記憶體。儲存緩衝器中項目的解除指派與跟隨之最終退役，在接收到儲存完成的確認之後才發生。或者，項目可在確認之前最終退役，但是項目的解除指派在接收到確認之前不能發生。導線200上的訊號能指示線管理邏輯124儲存的最終退役何時完成，而下一條線可以開始。

SB退役指出一指令已退役。在最終退役邏輯134指出一指令應該退役時，SB退役欄位裡的一個位元被主張。一旦SB退役欄位被主張，相關的指令依照順序被寫到記憶體中。當MOB 184A知道指令已經被寫到記憶體，SB退役欄位被解除主張，而指令被解除指派。

載入緩衝器182A與儲存緩衝器184A可以是有一頭部和一尾部的佇列。當指令被解除指派時，頭部被移動。在載入



五、發明說明 (45)

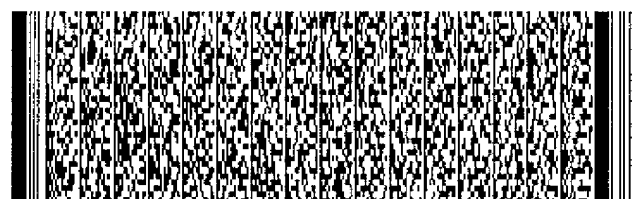
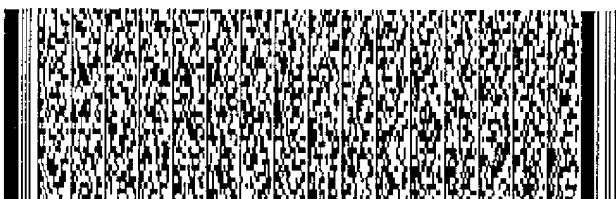
緩衝器184A中，和追蹤緩衝器114中，退役和解除指派可同時地發生。最終退役邏輯134藉著導線136和140提供訊號。Demux 188選擇是否載入緩衝器182或儲存緩衝器184中的一個會收到一退役訊號。Demux 188是選擇性的，且能由載入緩衝器182與儲存緩衝器184中的致能埠取代。

F. 有關線管理邏輯與最終退役邏輯的額外資訊

在一實施例中，線管理邏輯124使用樹結構保存線順序。在此樹結構下，程式順序(也是退役順序)從頂端向底部流動，而在右邊的節點依照程式順序比左邊的節點早。根依照程式順序是第一個。樹是一種抽象概念，然而樹結構是實現此樹的電路。

線由跟隨在反向分支或函數呼叫後面的指令開始。也就是，線由假設反向分支不成立或函數未被呼叫的下一個指令開始(如圖4和5中線T2所說明)。如此做，從線(節點)的觀點來看，線之子節點的程式順序與線開始(產生)的順序相反。舉例來說，在圖6中，照時間順序，線T2的執行在線T3的執行之前開始，但是照程式順序，線T3在線T2之前發生。

在一實施例中，三個事件可能使線從樹中被移除：(1) 當線退役時，在樹的根之線被移除。當在根的線退役時，照程式順序中的下一個線(節點)變成根，而且節點也對應地再指派。(2) 在程式順序中最後的線從樹被移除，以空出來讓程式順序中更高的線加入到樹中。在此，樹像後進先出(LIFO)堆疊班般運作。(3) 當發現父線的程式計數器



五、發明說明 (46)

超出起始計數和結束計數間的範圍之外時，線可被重設而藉此由樹中移除。當子線在反向分支上產生的情況中(舉例來說，圖6和29中的線T4)，起始計數是反向分支的目標，而結束計數是反向分支指令上的程式計數器值。在函數呼叫之後開始的線也可因沒有從函數的返回而被重新設定，雖然這很少發生。處理沒有由一函數返回之可能性的一種方式，是忽略此可能性，並讓系統在它變成程式順序中最低者時，最終地由樹中將線移除，如事件(2)中一樣。當線從樹中被移除時，指派給那條線的資源(例如追蹤緩衝器，儲存緩衝器，和載入緩衝器)被解除指派。

事件(1)和(3)在圖29中說明，其中包括圖6的範例的線，加上線T5和T6。線T5接著點J上的反向分支指令開始，而線T6接著點K上的函數呼叫開始。一般假定只有四個追蹤緩衝器。圖30舉例說明在時點t1的樹結構。線T3被加到樹之前，線T2被加到樹中。線T3被加到樹之後，線T4被加到節點。線T2與T2是線T1的孩子。線T4是線T3的一個孩子。照由上而下，由右而左的規則，程式和退役順序是線T1，T3，T4，和T2。圖31舉例說明在時點t2的樹結構，它假定線T4在線T1退役之前被重設。程式和退役順序是線T1，T3，T2，和T5。圖32舉例說明在時點t2的樹結構，它假定線T1在線T4被重設之前退役。程式和退役順序是線T3，T4，T2，和T5。圖33舉例說明在時點t3的樹結構，它是在線T1退役，且線T4被重設的時間之後。程式和退役順序是T3，T2，T5和T6。



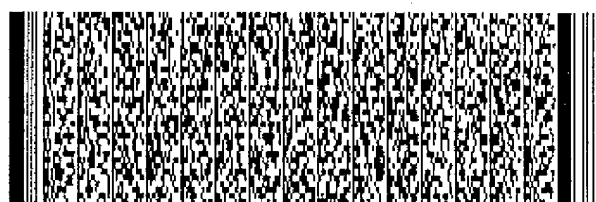
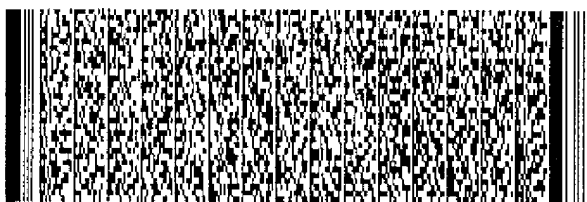
五、發明說明 (47)

事件(2)在圖34中說明，它包括巢狀函數。依照時間順序，線依T1，T2，T3，T4，和T5的順序產生(開始)。然而，依程式順序是T1，T5，T4，T3，和T2。在該範例中，只有四個追蹤緩衝器。因此，並非所有的五條線同時存在。圖35舉例說明在時點t1的樹結構，它是在線T5開始之前。程式和退役順序是T1，T4，T3，和T2。線T5還不是樹結構的一部份。圖36舉例說明在時點t2的樹結構，它是在線T5開始之後。線T2，是在程式順序中最低的，由樹結構中被移除以留出空間給線T5。從樹中被移除的線可在稍後被重新開始。或者，另外一條線可執行從樹中被移除的線之所有或部份的指令。在一實施例，在重設的情況中，線可尋求加入下一個接著的線，而非重設的線。或者，線可繼續直到結束。陣列198的函數可在樹的節點中執行。

子線的線ID適當地被依照程式順序放置到樹結構中(雖然線管理邏輯124所決定的程式順序可能改變)。當一條線加入或符合依照程式順序在樹中下一條線的程式計數時，線被完結。如果線只有一子，則那就是程式順序裡的下一條線。舉例來說，在圖33中，線T2是在樹中依照程式順序的下一條線。

最終退役邏輯134由樹結構取得資訊以組合陣列198，或直接從樹結構的電路。在樹結構和線管理邏輯124之其他邏輯與最終退役邏輯134的邏輯之間，可能有解碼電路。陣列198可能不需要。

簡言之，樹結構至少為下列目的而提供資訊：(1)樹指出



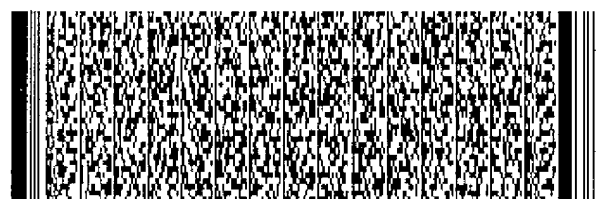
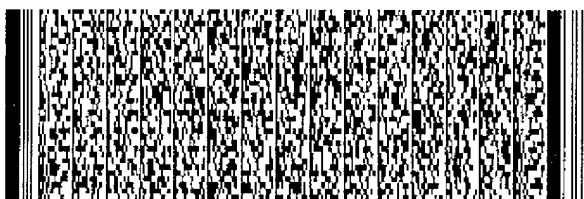
五、發明說明 (48)

退役順序;(2)樹指出程式順序，它被，舉例來說，如上述的MOB 178，使用;(3)樹藉由指出另一線的起始指令來指示一條線的終結點;(4)樹藉由指示哪一個資源可用與哪一個資源被解除指派，而用在線資源指派中。

G. 沒有多重線的實施例

圖3舉例說明包括管線308的處理器100。處理器100類似於處理器50。然而，追蹤緩衝器300是唯一的追蹤緩衝器，而MOB 310是唯一的MOB。處理器50並非設計來處理多重的線。因此，處理器100不需要線管理邏輯。舉例來說，追蹤緩衝器300可能與追蹤緩衝器114A類似，除了不需要多重線的特定元件外。舉例來說，導線216和輸出暫存器檔案210將不需要。各種電路可用來偵測猜測錯誤，包括已廣為人知的電路。舉例來說，MOB 310可能類似於MOB 178A，除了不需要多重線的特定特點外。舉例來說，在載入緩衝器中將不需要線ID欄位。處理器100的其他元件可依其在處理器50中的組態而有些修改，以移除多重線相關之特點。追蹤緩衝器300與MOB 310可與各種猜測以一起使用，並由其中的錯誤復原。追蹤緩衝器讓大量指令為了最終退役前可能的重演而保存在管線之外。

處理器50可與一非多重線程式連接使用。在該情況中，線管理邏輯124可以總是保存著程式順序裡的相同線ID。或者，能夠使線管理邏輯124失效。在非多重線的情形中，只使用了追蹤緩衝器114中的一個與MOB 178中的一個。或者，追蹤緩衝器能夠結合以製造更大的追蹤緩衝



五、發明說明 (49)

器，而MOB能夠結合以製造更大的MOB。

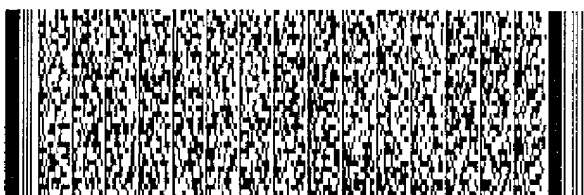
H. 額外的資訊和實施例

參照圖37，處理器400是一包括多管線單元402的多處理器(MP)晶片。多管線單元400不同於圖2的共享資源管線108之處，在於多管線單元402的每一個管線0, 1, ... W(W可能與X相等或更多或更少)都包含了一完整管線(舉例來說，每條管線獨立的重新配置/指派單元)。或是，處理器400可與處理器50基本上相同或非常不同。其他的處理器可能包括多管線單元402的一些特徵和管線108的一些特徵。

在此處提到的每一個處理器，可以包含在各種電腦系統的一部份中。參照圖38，僅係作為範例，處理器50可以是電腦系統430的一部份。系統430也可包括一第二處理器434。在處理器50裡面可包括一內建第二層(L2)快取。處理器50可透過處理器匯流排442與記憶體控制器440溝通。記憶體控制器440可透過匯流排452和454與主記憶體446與週邊設備448溝通。

類似於管線108或308(在圖2和3中)的管線可用在不使用暫存器重新配置的處理器中。在此一情形中，涉及暫存器重新配置的元件(舉例來說，重新配置/指派單元150)可修改以移除與重新配置相關的特徵。

所敘述與舉例說明的電路和細節僅是範例。各種其他電路和細節能夠用在其位置中。再者，各種設計可在大小，延遲等作取捨。舉例來說，最大運作時脈頻率可能必須



五、發明說明 (50)

減少，如果在執行路徑(舉例來說，在保留站，暫存器檔案，ROB)中的緩衝器太大的話。在此處舉例說明的元件可依各種不同的技術而設計與建構。

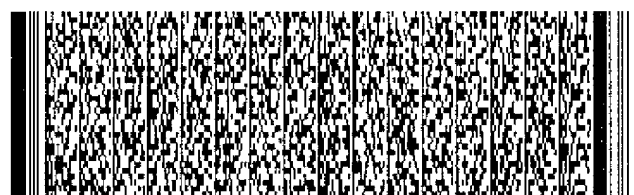
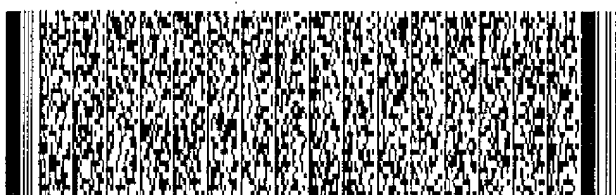
在舉例說明的二個結構間可能有中間結構(例如緩衝器)或訊號。一些導線可能不是像所舉例說明是連續的，而是被中間結構斷開。圖形中盒子的邊界是為了說明的目的。真實的裝置不必用如此定義的邊界來包括元件。被舉例說明元件的相關大小並不是建議為真實的相關大小。在某些實施例中箭頭顯示某些資料的流動，但不是每個訊號，例如資料請求。在上面被描述的邏輯的高訊號，它能夠被替換為邏輯的低訊號，反之亦然。

在處理器中舉例說明的元件，可能全部在同一處理器晶片上。或者，例如，追蹤緩衝器可在與執行管線不同的晶片上。

"連接"，"偶合"等詞，與相關的術語不限制在直接的連接或直接的偶合，而是可包括間接的連接或間接的偶合。術語"回應的"和有關的術語指的是一訊號或事件在某種程度上被另一訊號或事件影響，但不必然是完全地或直接地。如果說明書指出元件"可能"，"可以"，或包括"最好"在內，該特定元件不需要被包括在內。

MOB可使用資料相符而非位址相符來偵測失誤猜測。

熟於該項技藝而由此揭露獲益人士，將會發現來自前面描述的許多其他變化和圖示，可在本發明的範圍中完成。因此，定義本發明範圍的是以下的申請專利範圍及其中的



五、發明說明 (51)

任何修正。

元 件 符 號 說 明

10	處 理 器	12	執 行 管 線
14	追 蹤 緩 衝 器	34	最 終 退 役 邏 輯
50	處 理 器	100	處 理 器
104	指 令 快 取	106	解 碼 器
108	管 線	110	MUX
114	追 蹤 緩 衝 器	112A-112X	程 式 計 數 器
124	線 管 理 邏 輯	134	最 終 退 役 邏 輯
150	重 新 配 置 / 指 派 單 元		
152	暫 存 器 檔 案	156	分 配 排 程 / 發 送 單 元
158	執 行 單 元	162	寫 回 單 元
164	重 排 序 緩 衝 器	172	包 含 位 址 產 生 器
176	資 料 快 取	178	記 憶 體 排 序 緩 衝 器
182	載 入 緩 衝 器	184	儲 存 緩 衝 器
188	DEMUX	192	MUX
202A	指 令 佇 列 陣 列	206A	資 料 與 附 屬 性 陣 列
208A	輸 入 暫 存 器 檔 案	208B	輸 出 暫 存 器 檔 案
210A	輸 出 暫 存 器 檔 案	212A	附 屬 性 產 生 電 路
214A	資 料 陣 列	218A	附 屬 性 產 生 與 解 碼 電 路
224A	控 制 電 路	280B	比 較 器
284B	重 演 觸 發 邏 輯	300	追 蹤 緩 衝 器
302	MOB 控 制 電 路	310	MOB
312	載 入 緩 衝 器	314	儲 存 緩 衝 器



五、發明說明 (52)

320	比較器	324	比較器
356	相符判定邏輯	390	資料路徑控制邏輯
394	重演觸發邏輯	400	多管線單元
402	多管線單元	430	系統
434	處理器	440	記憶體控制器
446	主記憶體	448	週邊設備

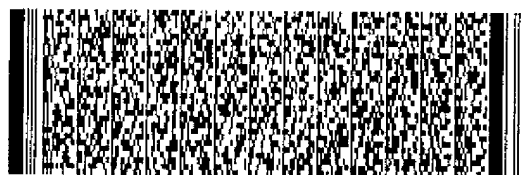
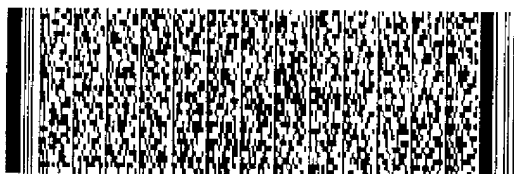


四、中文發明摘要 (發明之名稱：於一完成無序多線執行處理器中排序載入及儲存指令之記憶系統)

在本發明一實施例中，處理器包括含有載入緩衝器與儲存緩衝器的記憶體排序緩衝器(MOB)，其中該MOB將載入與儲存指令排序，以便維持在不同線中的載入與儲存指令間之資料一致性，其中至少一線依附於至少另一線。在本發明另一實施例中，處理器包括一執行管線以同時地至少執行部份的線，其中至少一線依附於至少另一線，該執行管線包括一記憶體排序緩衝器將載入與儲存指令排序。該處理器也包括偵測電路以偵測與載入緩衝器中之載入指令相關的猜測錯誤。

英文發明摘要 (發明之名稱：MEMORY SYSTEM FOR ORDERING LOAD AND STORE INSTRUCTIONS IN A PROCESSOR THAT PERFORMS OUT-OF-ORDER MULTITHREAD EXECUTION)

In one embodiment of the invention, a processor includes a memory order buffer (MOB) including load buffers and store buffers, wherein the MOB orders load and store instructions so as to maintain data coherency between load and store instructions in different threads, wherein at least one of the threads is dependent on at least another one of the threads. In another embodiment of the invention, a processor includes an execution



四、中文發明摘要 (發明之名稱：於一完成無序多線執行處理器中排序載入及儲存指令之記憶系統)

英文發明摘要 (發明之名稱：MEMORY SYSTEM FOR ORDERING LOAD AND STORE INSTRUCTIONS IN A PROCESSOR THAT PERFORMS OUT-OF-ORDER MULTITHREAD EXECUTION)

pipeline to concurrently execute at least portions of threads, wherein at least one of the threads is dependent on at least another one of the threads, the execution pipeline including a memory order buffer that orders load and store instructions. The processor also includes detection circuitry to detect speculation errors associated with load instructions in a load buffer.



六、申請專利範圍

1. 一種處理器包含：

一記憶體排序緩衝器(MOB)包括載入緩衝器和儲存緩衝器，其中MOB將載入和儲存指令排序，以便維持不同的線中之載入和儲存指令間之資料一致性，其中至少一線附屬於至少另外一線。

2. 如申請專利範圍第1項中之處理器，進一步包含線管理邏輯以控制線的動態產生。

3. 如申請專利範圍第1項中之處理器，進一步包含線管理邏輯以提供程式順序和退役順序的指示。

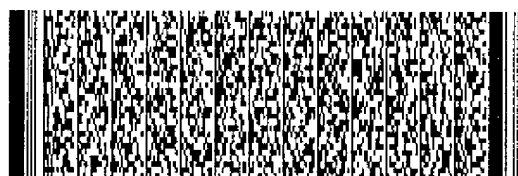
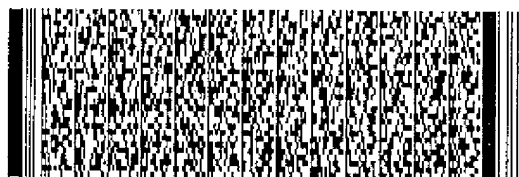
4. 如申請專利範圍第3項中之處理器，其中線在最終退役中依照退役順序由MOB中退役。

5. 如申請專利範圍第3項中之處理器，其中線在確保所有的指令都已經被沒有猜測錯誤地執行，或已經成為重設的線之一部份後，在最終退役中依照退役順序由MOB中退役。

6. 如申請專利範圍第1項中之處理器，其中儲存緩衝器包括一欄位以保存要儲存到記憶體的資料。

7. 如申請專利範圍第1項中之處理器，其中載入緩衝器包括狀態欄位，內含有載入指令是否有先前所接收來自記憶體的資料或資料在儲存緩衝器中之指示。

8. 如申請專利範圍第1項中之處理器，其中載入緩衝器包括狀態欄位，其含有對儲存緩衝器項目的一儲存緩衝器識別號碼(SBID)的指示，載入指令先前由該儲存緩衝器接收資料。



六、申請專利範圍

9. 如申請專利範圍第1項中之處理器，其中載入緩衝器包括狀態欄位，其含有對儲存指令的一線識別號碼的指示，載入指令先前由該儲存指令接收資料。

10. 一種處理器，包含：

一執行管線以同時地執行線的至少一部分，其中至少一線附屬於至少另外一線，執行管線包括一記憶體排序緩衝器，以將載入和儲存指令排序；和

偵測電路，以偵測與載入緩衝器中之載入指令有關的猜測錯誤。

11. 如申請專利範圍第10項中之處理器，進一步包括重演觸發電路。

12. 如申請專利範圍第10項中之處理器，進一步包括一在執行管線以外的追蹤緩衝器，保存載入與儲存指令直到最終退役。

13. 如申請專利範圍第10項中之處理器，其中偵測電路包含在執行管線之中。

14. 一種處理器，包含：

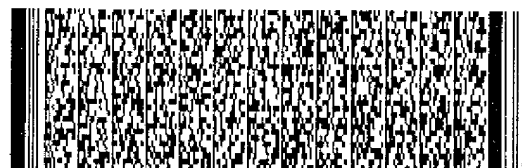
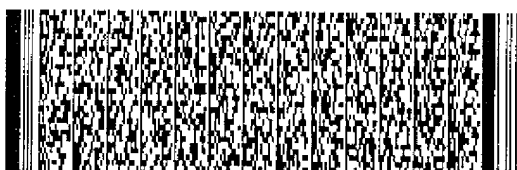
一記憶體排序緩衝器包括：

載入緩衝器，以保存線的載入指令；

儲存緩衝器，以保存線的儲存指令；和

比較電路，以比較要被執行的載入指令中之一的一位址，與儲存緩衝器中至少一儲存指令的位址；

資料路徑控制邏輯，以決定載入指令是否由記憶體或由儲存緩衝器中的一個讀取資料。



六、申請專利範圍

15. 一種處理器包含：

一記憶體排序緩衝器包括：

載入緩衝器以保存線的載入指令；

儲存緩衝器以保存線的儲存指令；和

比較電路，以比較要被執行儲存指令中之一的一位址與狀態位元，與載入指令中至少一者之一位址與狀態位元；和

偵測邏輯，以根據比較決定是否要重演一或多個載入指令。



圖式

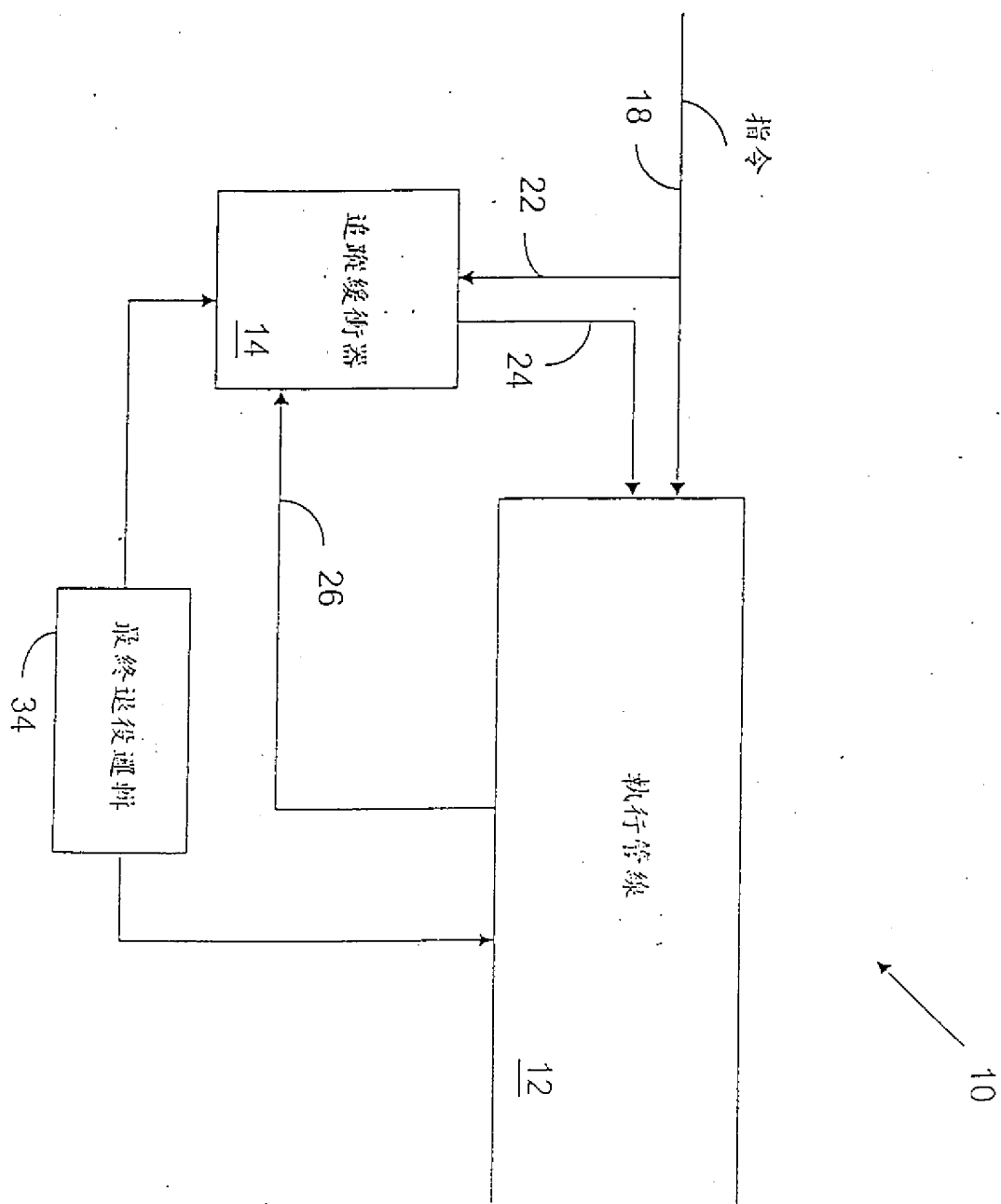


圖 1

圖式

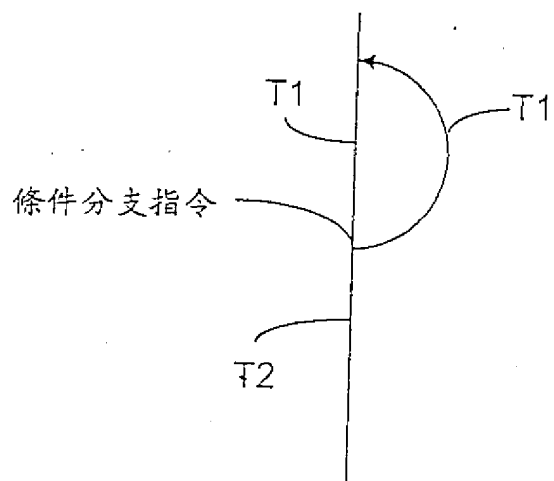


圖 4

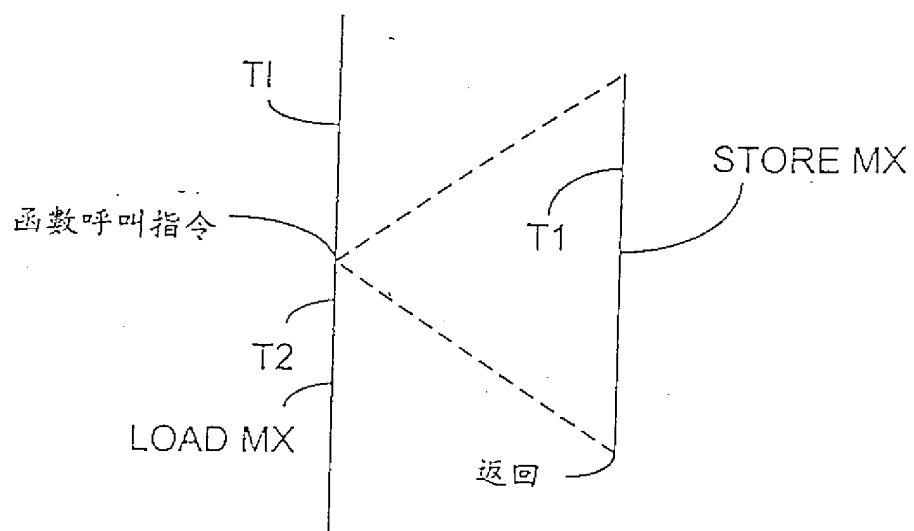


圖 5

圖式

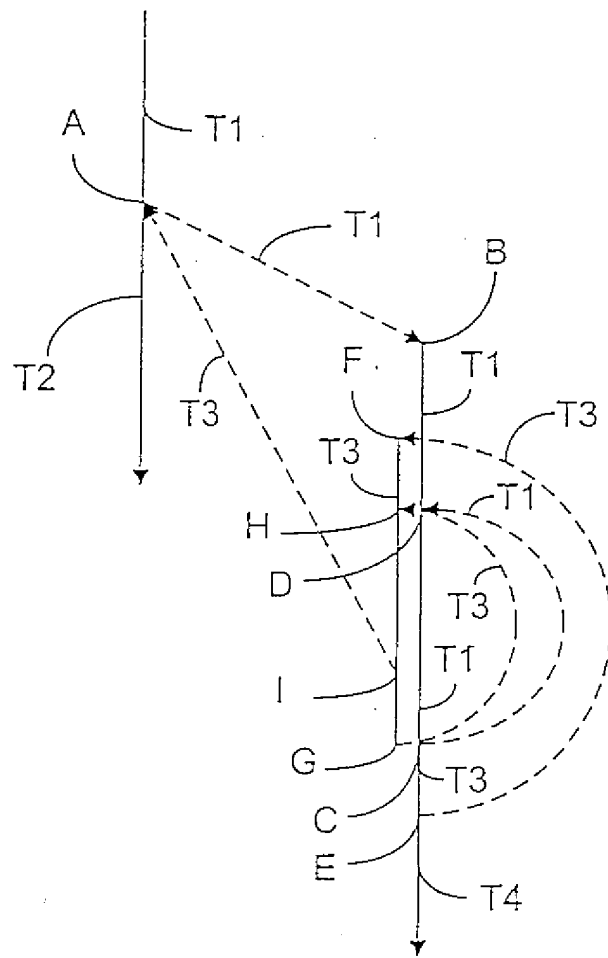


圖 6

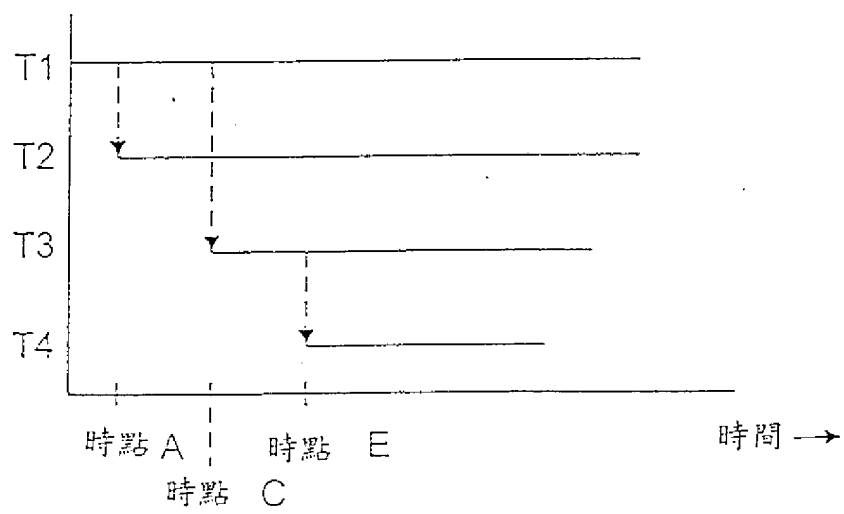


圖 7

圖式

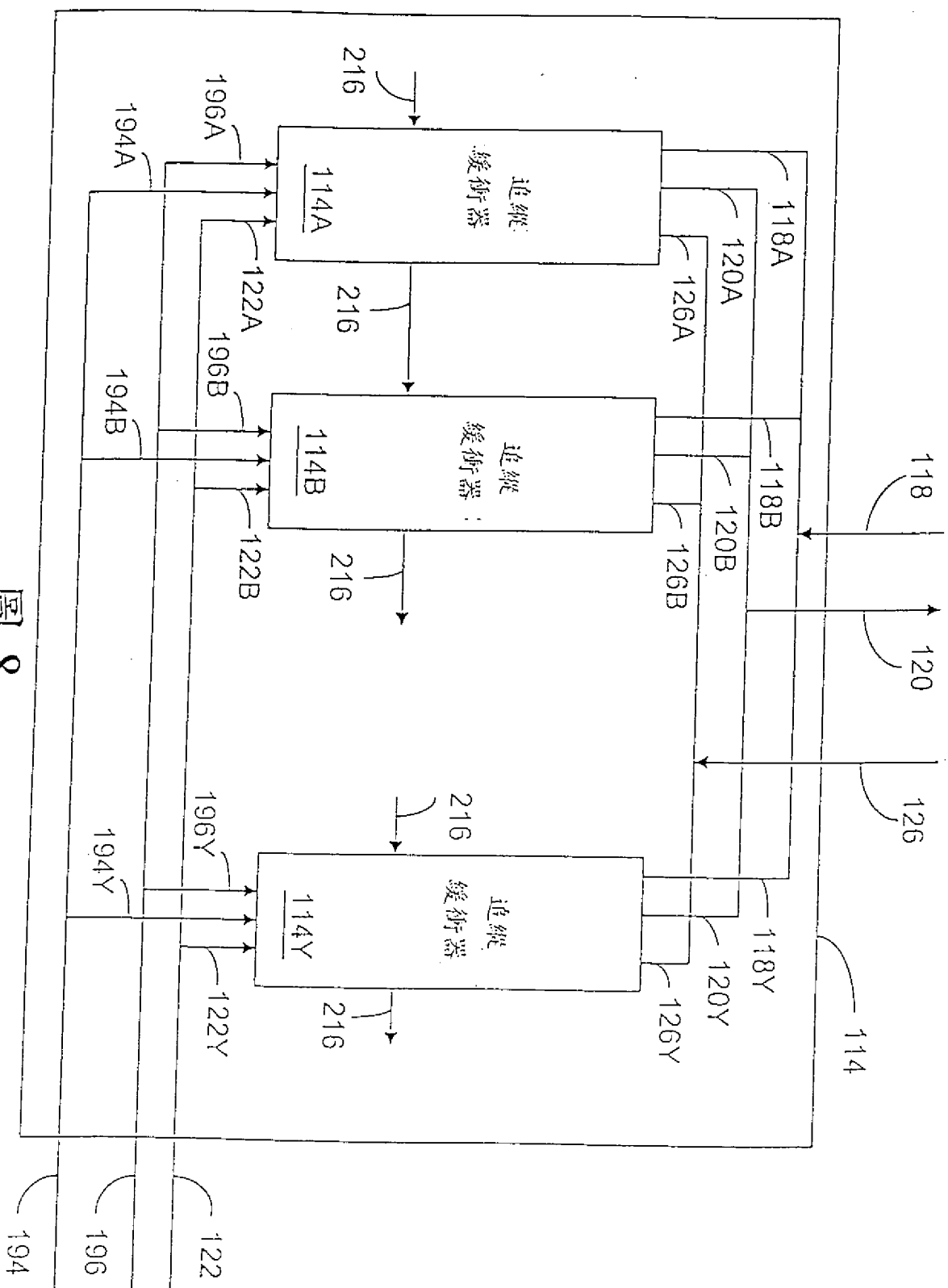


圖 8

圖式

程式和
退役順序

	線 ID 號碼	
	線 ID	號碼
1	線 ID 1	(T1)
2	線 ID 3	(T3)
3	線 ID 2	(T2)
4	線 ID 4	(T4)

在時點 T1

程式和
退役順序

	線 ID 號碼	
	線 ID	號碼
1	線 ID 3	(T3)
2	線 ID 2	(T2)
3	線 ID 4	(T5)
4	線 ID 1	(T6)

在時點 T2

圖 9

圖式

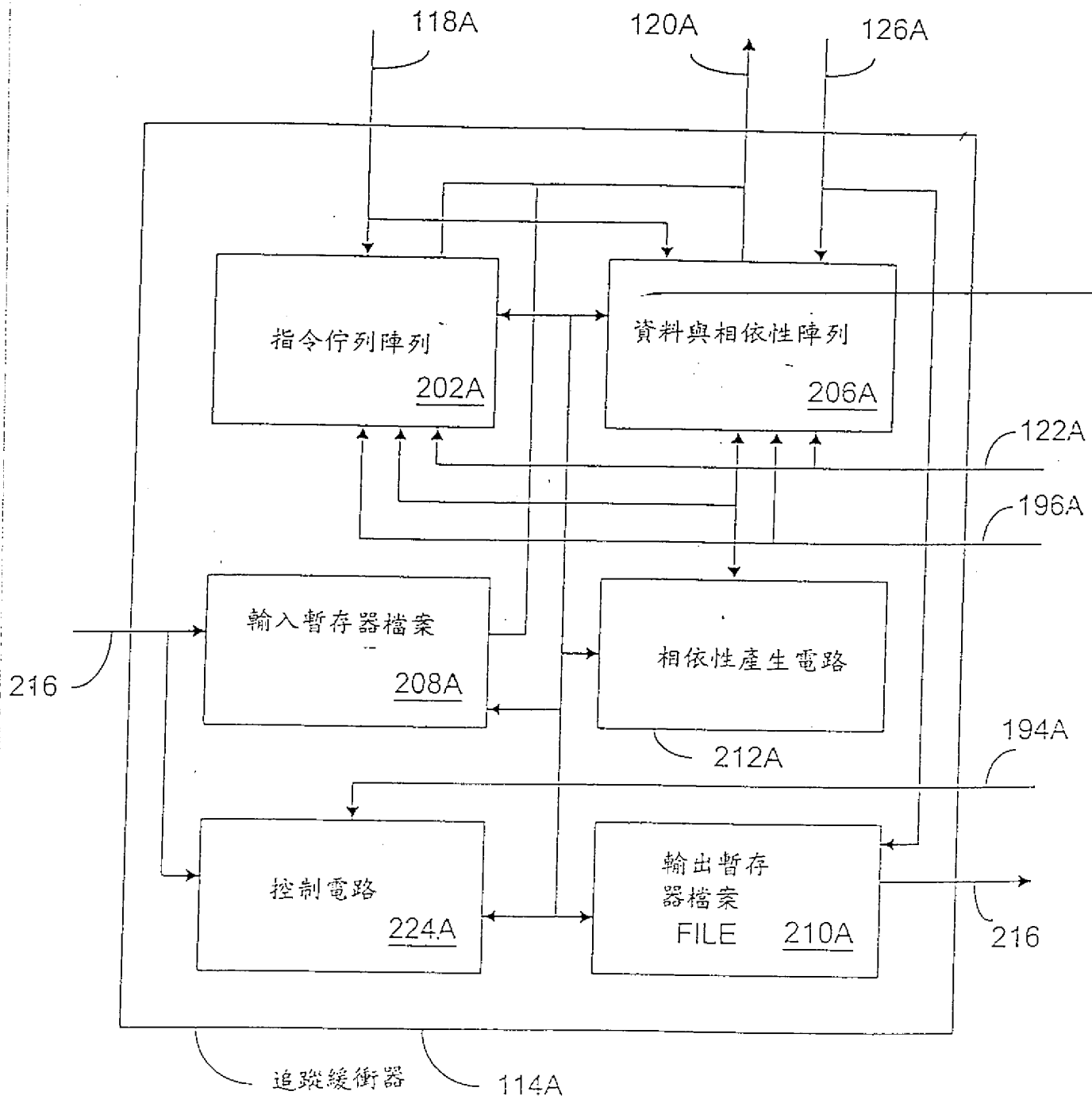


圖 10

圖式

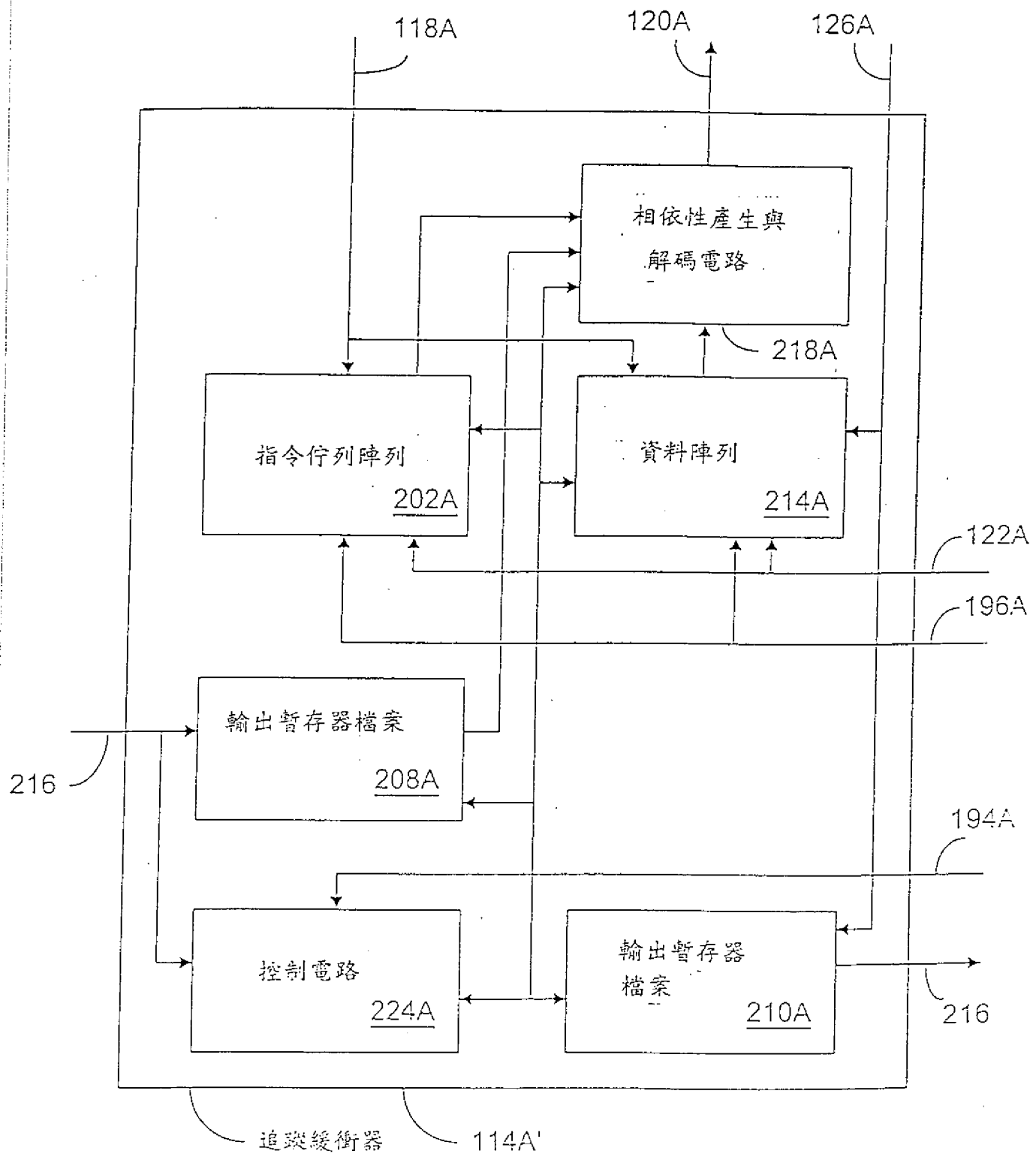


圖 11

圖式

INSTR ID	操作碼	目標	來源 1	來源 2	來源 1 的索引	有效 1	來源 2 的索引	有效 2	SBID	LBID
0	MULT	R1	R1	R2	X	0	X	0		
1	MULT	R2	R3	R4	X	0	X	0		
2	ADD	R1	R1	R2	0	1	1	1		
3	ADD	R4	X	R1	X	0	2	1		
4	STORE	MX	X	R2	X	0	1	1		
5	STORE	MY	X	R1	X	0	2	1		

指令序列陣列

圖 12

202A

圖式

INSTR ID	值或 PRID	狀態 (2 個位元)	重演 計數	相依性欄位			
				R4	R3	R2	R1
0				0	0	1	1
1				1	1	0	0
2				1	1	1	1
3				1	1	1	1
4				1	1	0	0
5				1	1	1	1

資料與
相依性障列

206A

圖 13

圖式

修正過的暫存器					修正元				
	R4	R3	R2	R1	240	R4	R3	R2	R1
	0	0	0	0		X	X	X	X
						236	234	232	230
I0	0	0	0	1		X	X	X	0
I1	0	0	1	1		X	X	1	0
I2	0	0	1	1		X	X	1	2
I3	1	0	1	1		3	X	1	2
I4	1	0	1	1		3	X	1	2
I5	1	0	1	1		3	X	1	2

圖 14

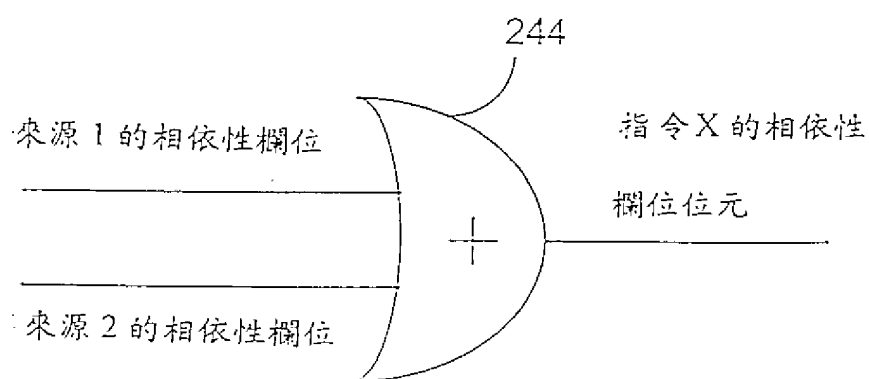


圖 15

圖式

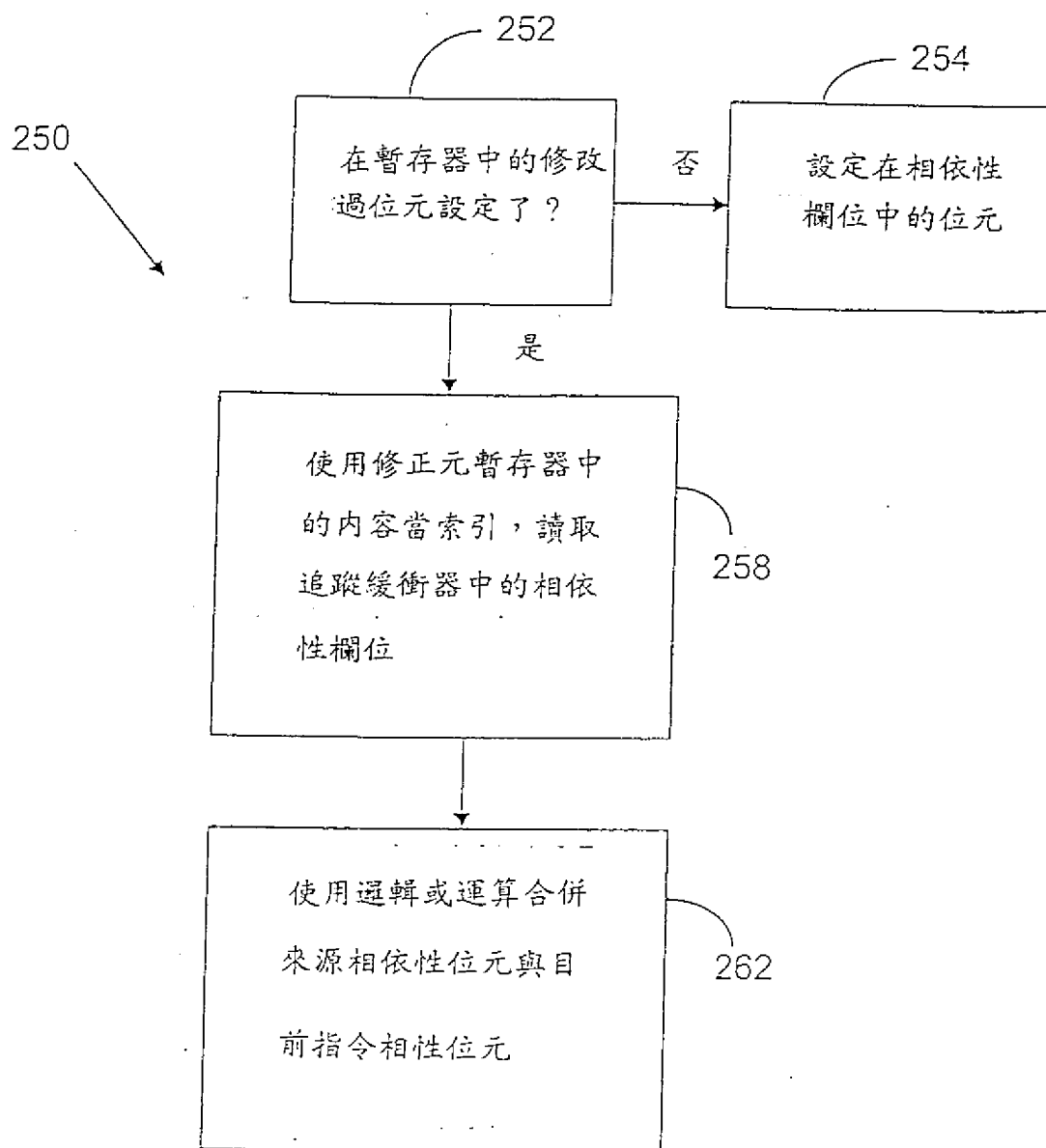


圖 16

圖式

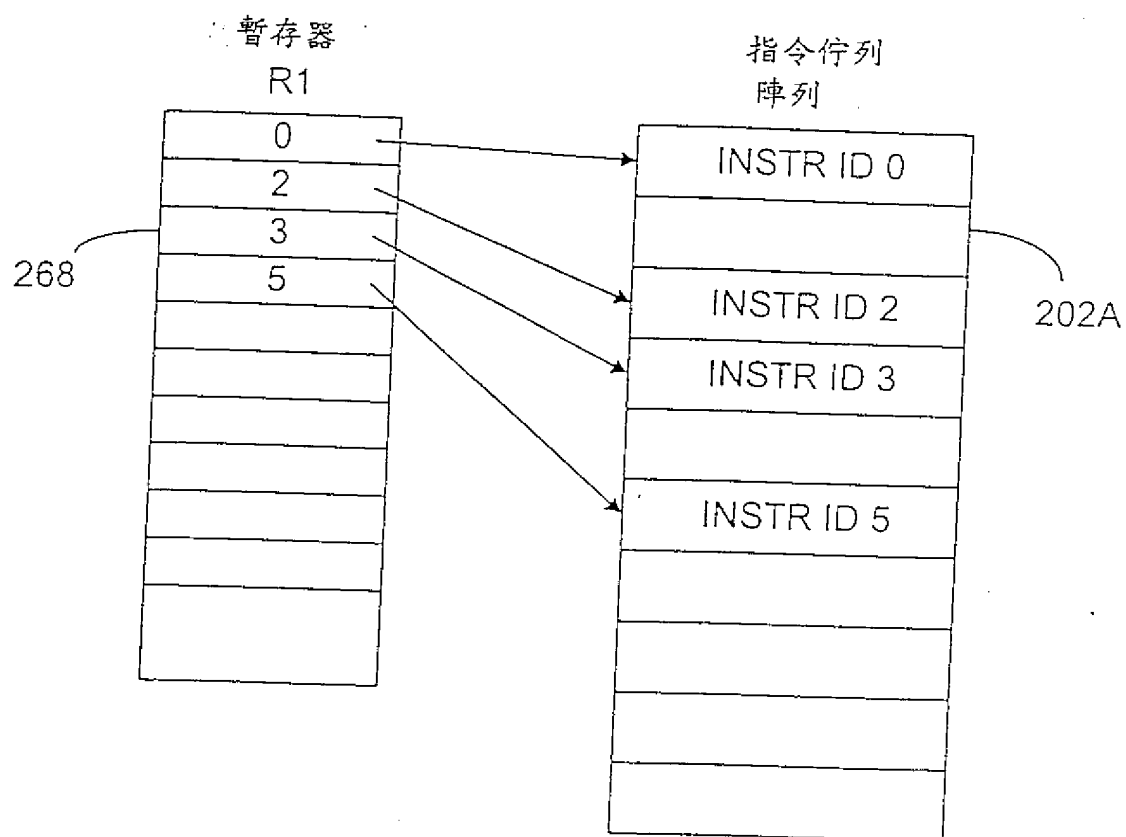


圖 17

圖式

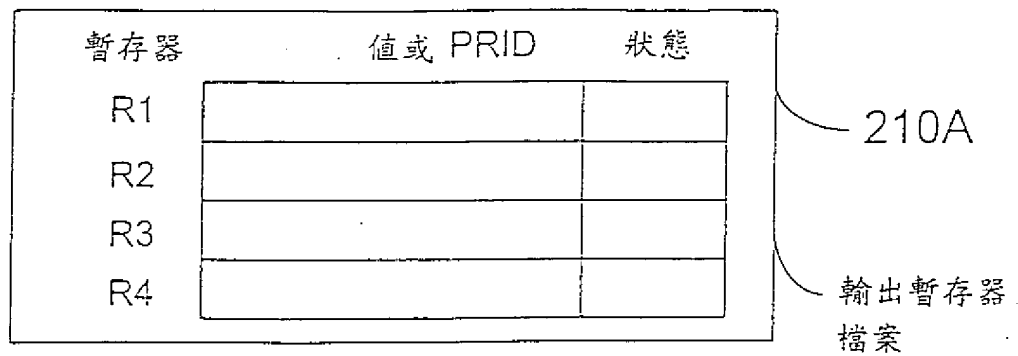


圖 18

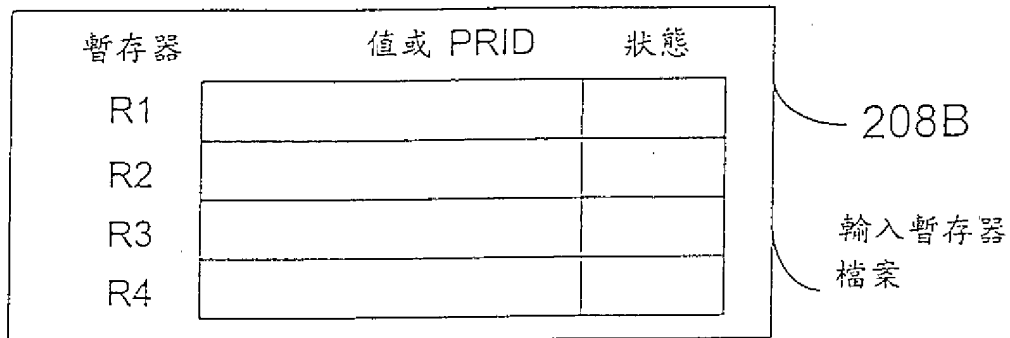


圖 19

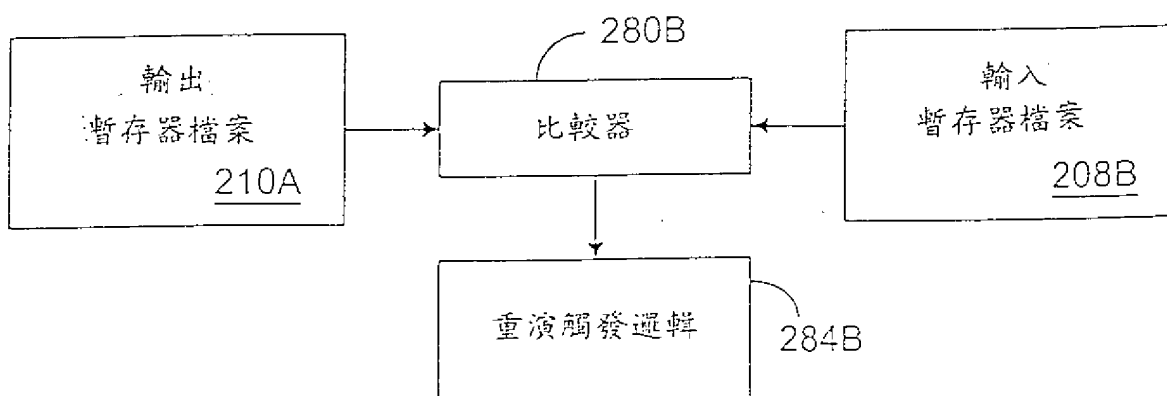


圖 20

圖式

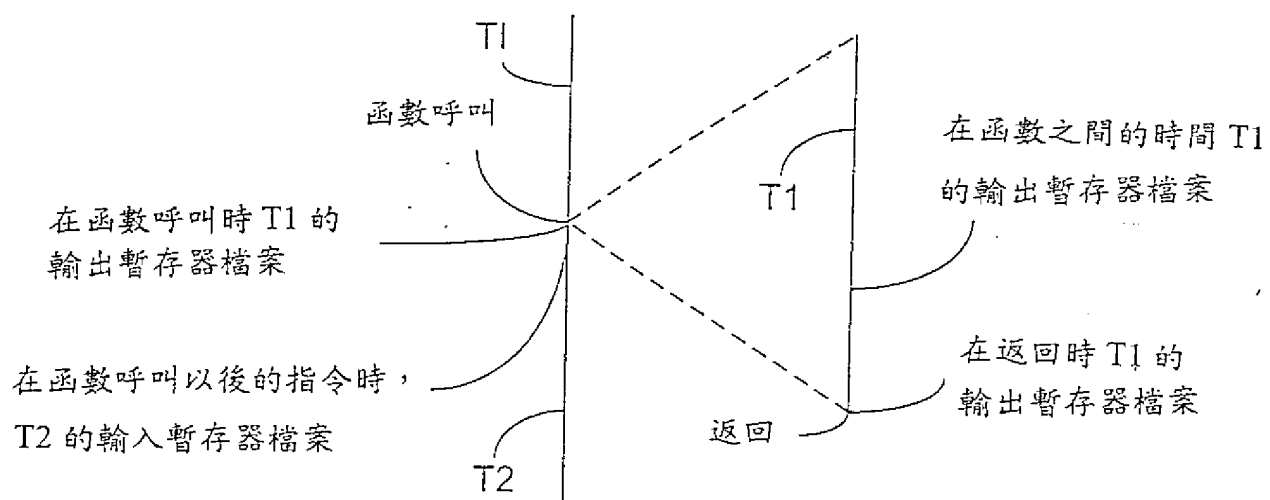


圖 21

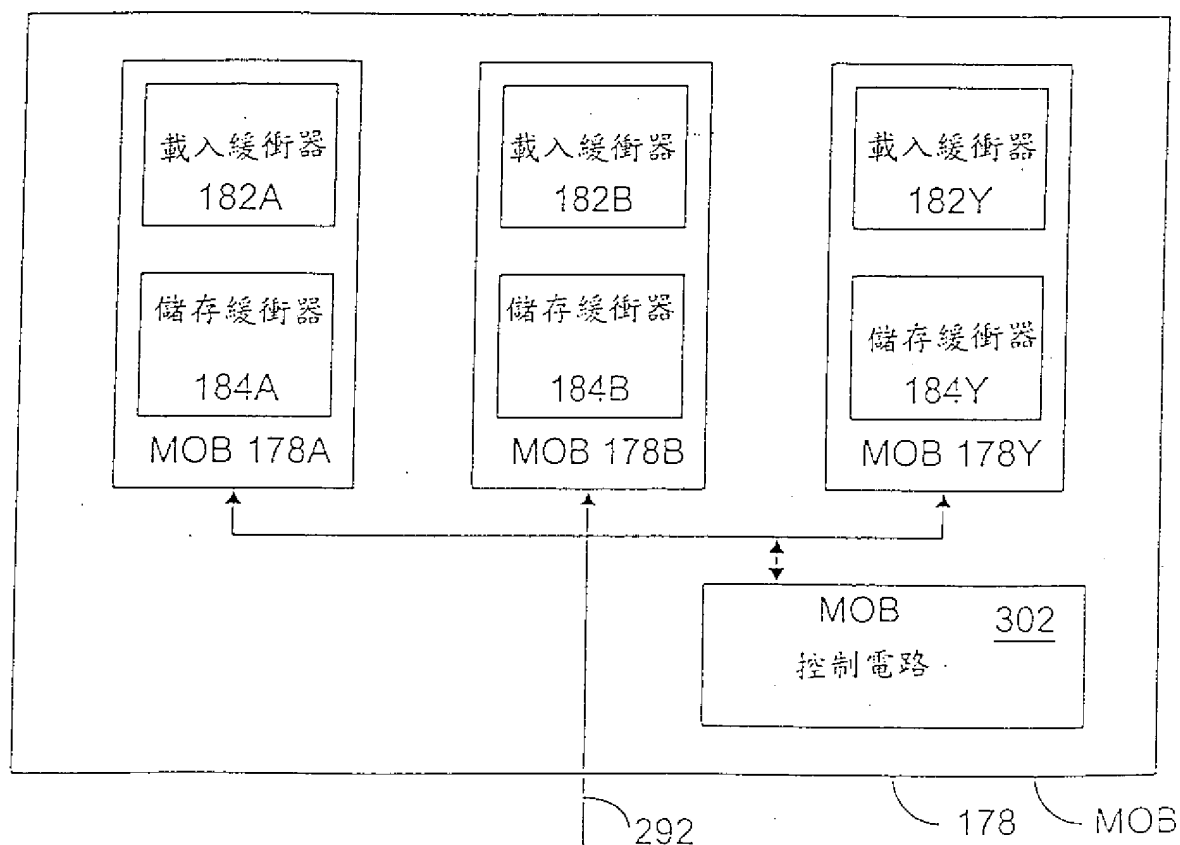


圖 22

圖式

SBID	LBID	INSTR ID	指令碼	儲存位址	SB位址有效	資料	SB資料有效	退役	重演計數
0	0		STORE 0						
1	0		STORE 1						
2	1		STORE 2						
3	2		STORE 3						
4	2		STORE 4						

184A

儲存緩衝器

圖 23

LBID	SBID	INSTR ID	指令碼	載入位址	項目有效	PRID	SB HIT	SBID	狀態線 ID	重演計數
0	1		LOAD 0							
1	2		LOAD 1							
2	4		LOAD 2							
3										

182A

載入緩衝器

圖 24

圖式

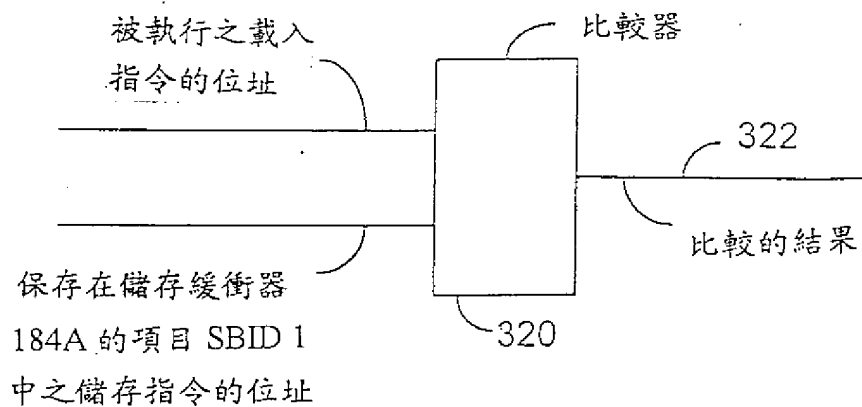


圖 25

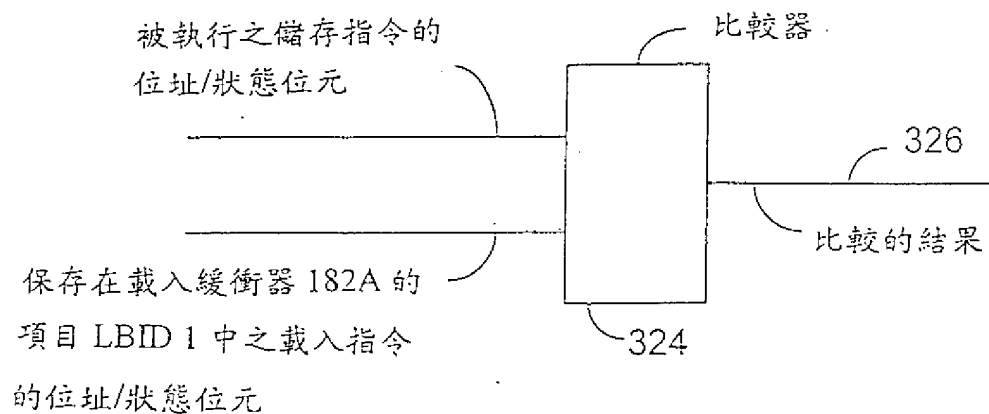


圖 26

圖式

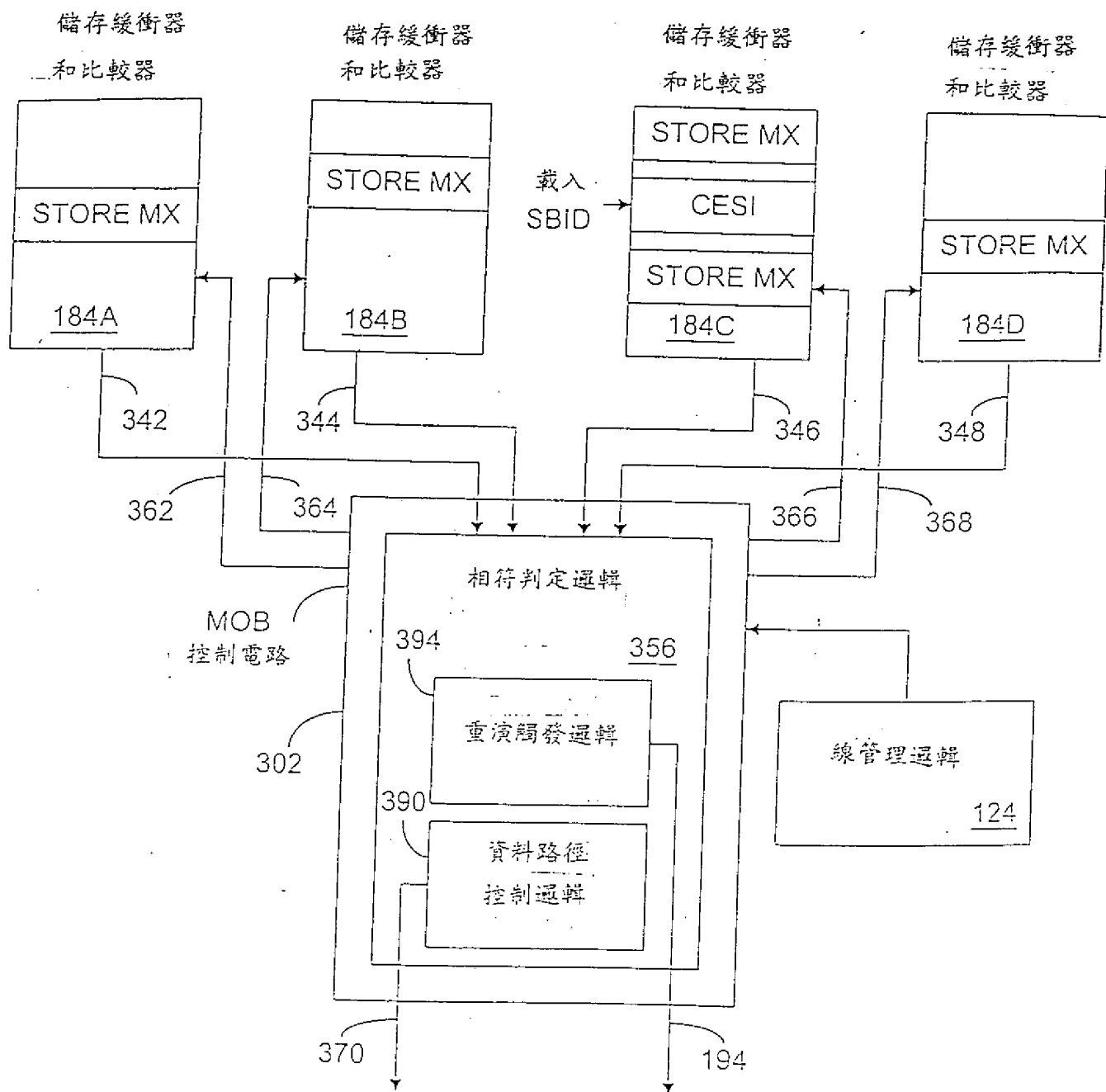


圖 27

圖式

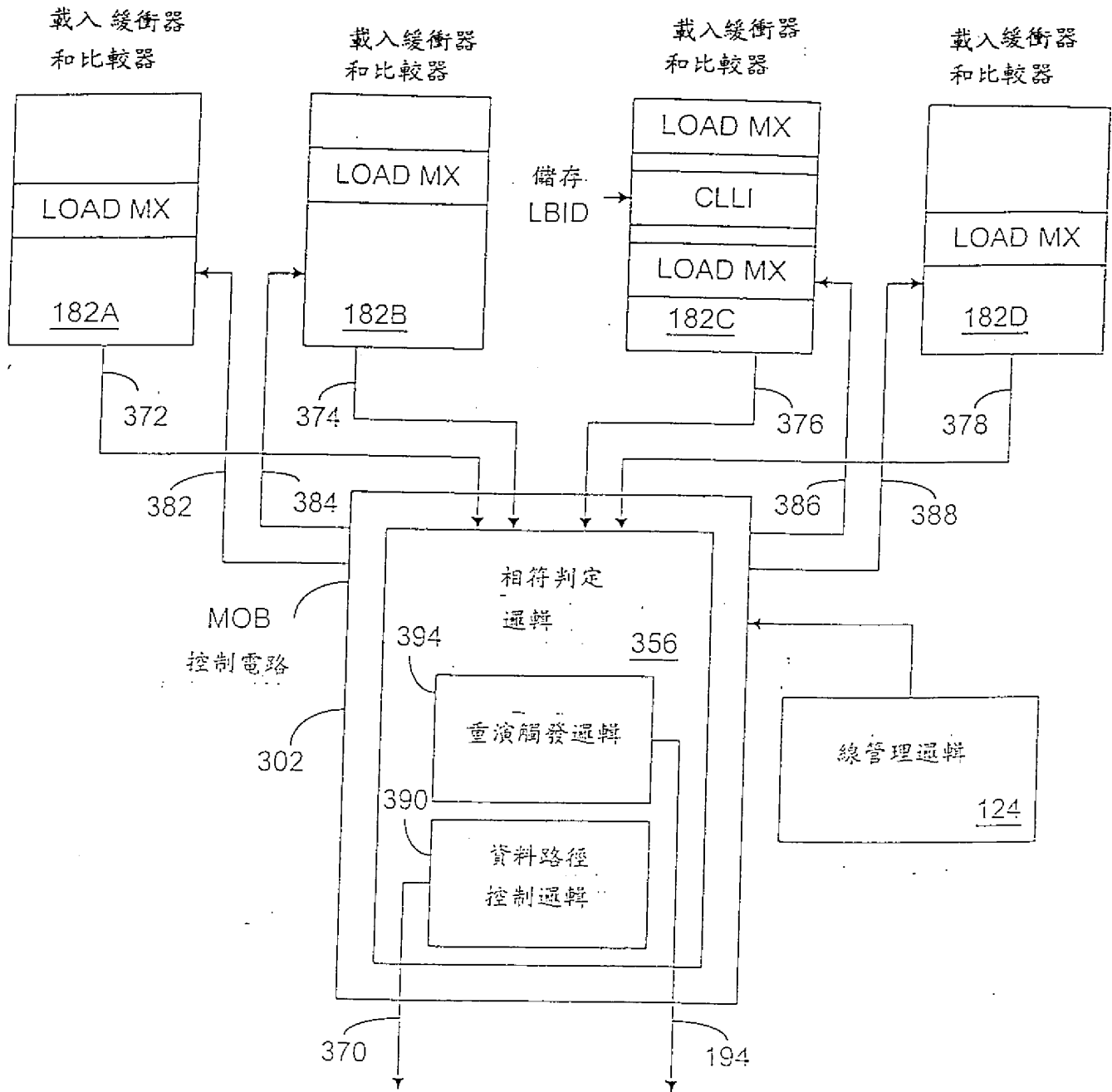


圖 28

圖式

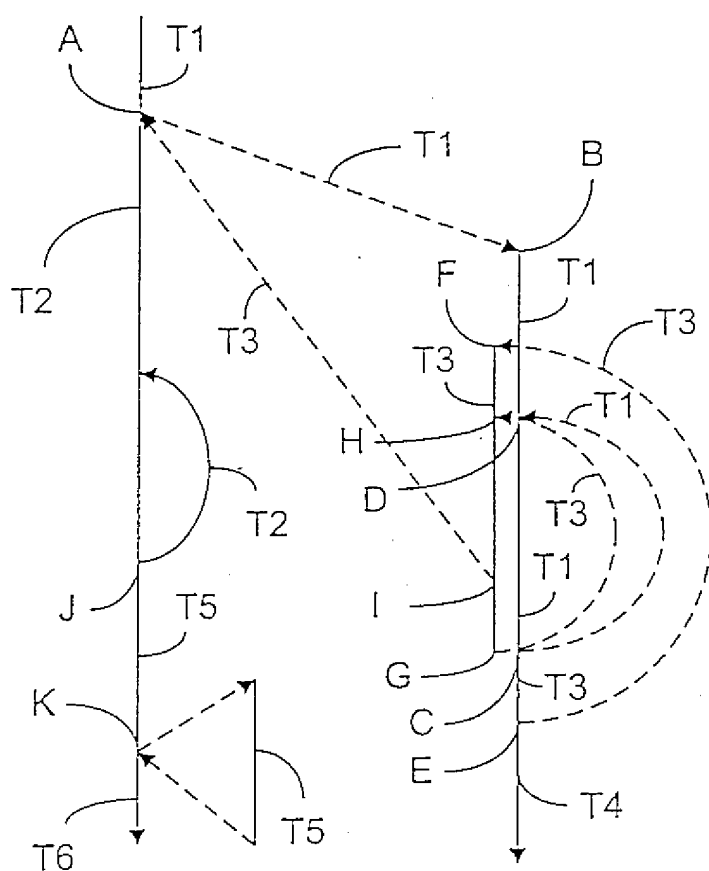
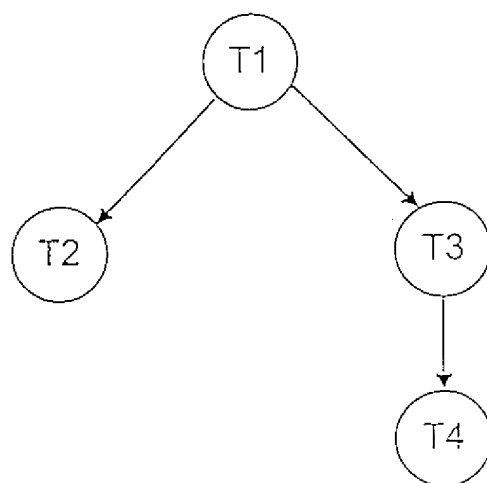


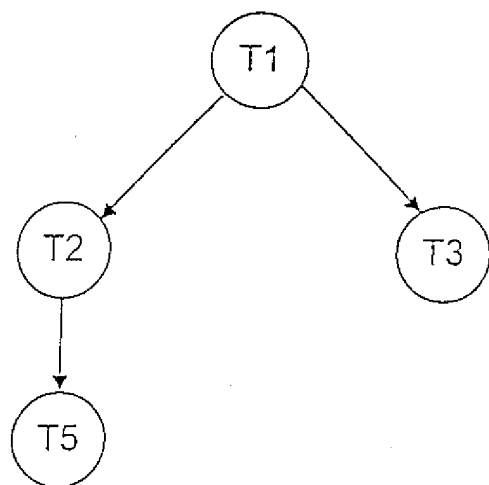
圖 29



在時點 t1 的樹

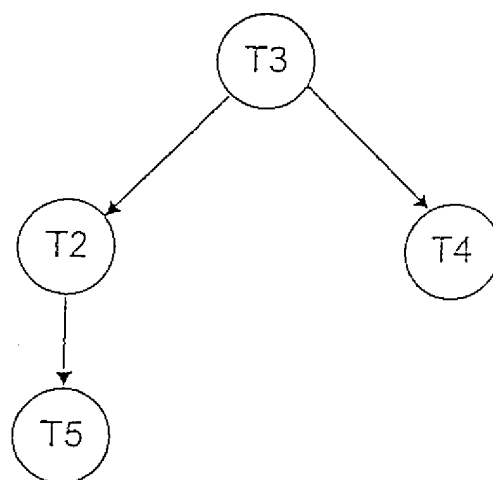
圖 30

圖式



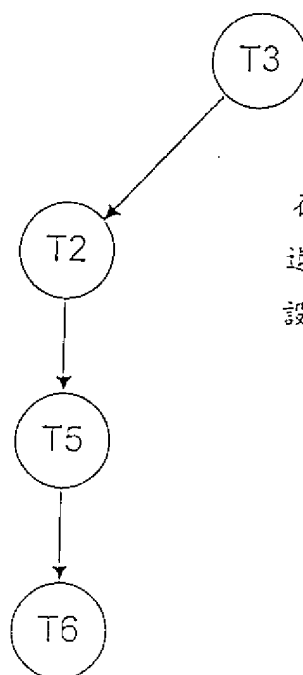
在時點 t_2 假設線
T4 在線 T1 退役
前被重設的樹

圖 31



在時點 t_2 假設線
T1 在線 T4 被重
設之前退役的樹

圖 32



在時點 t_3 線 T1
退役且線 T4 被重
設之後的樹

圖 33

圖式

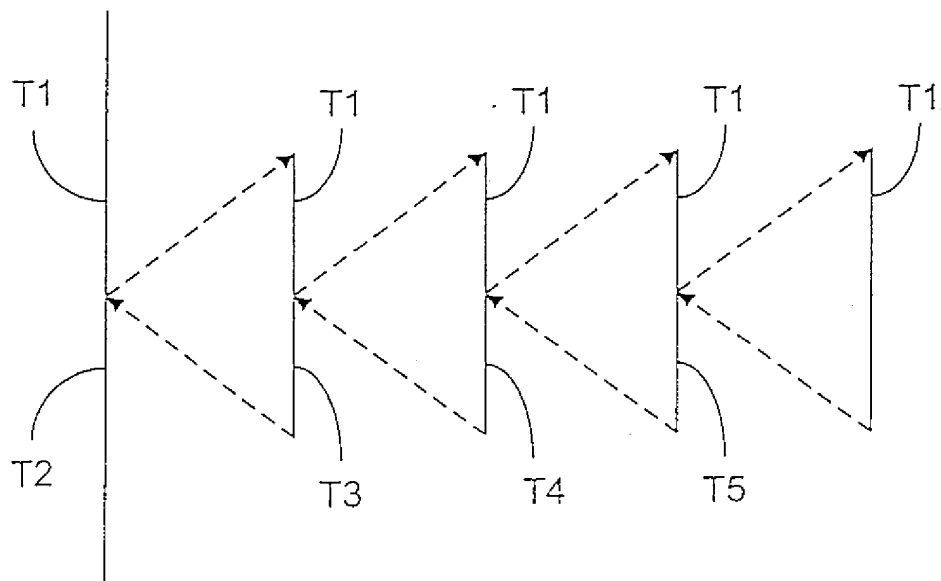
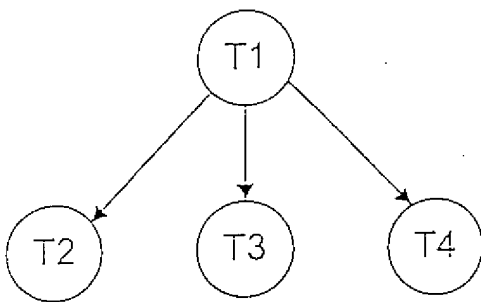
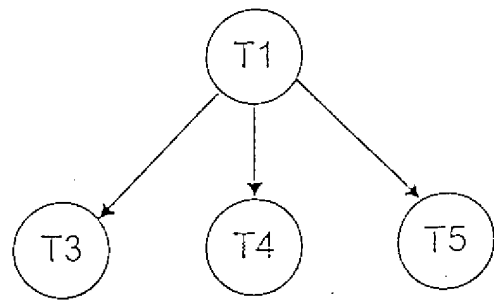


圖 34



在時點 t1 的樹

圖 35



在時點 t2 的樹

圖 36

圖式

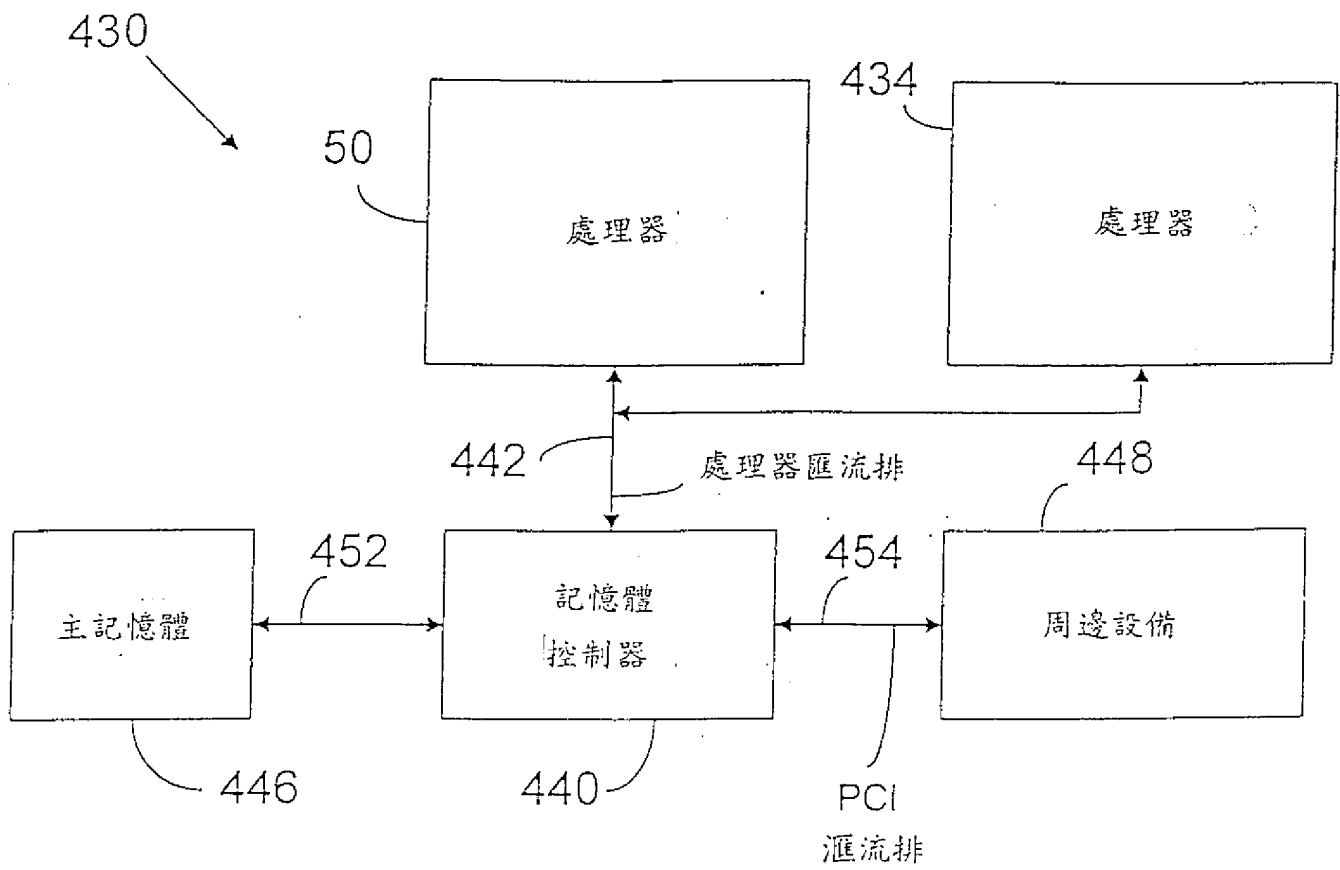


圖 38

五、發明說明 (14)

追蹤緩衝器114A中的線T6。線T6與線ID 1有關。在時點t2，程式與退役順序是線T3，T2，T5和T6。(如果線T1在線T4被重新設定之前退役，則線T5與T6將會有不同的線ID，但程式與退役順序將不變)。依所使用演算法而定，可能一開始在陣列198中線T2在線T3之前，但程式與退役順序會被更正，如在時點t1的陣列198。

如所述及，線的程式順序是在一循序處理器上線將會被執行的順序。指令的程式順序是在一循序處理器上指令將會被執行的順序。線管理邏輯124不一定要在一開始決定線的真正程式順序。然而，線管理邏輯124最後還是決定真正的程式順序。

參照圖8，追蹤緩衝器114A，114B,..., 114Y透過連接到導線118之導線118A, 118B,...118Y接收指令。在導線118A, 118B,..., 118Y與導線118之間可能有解多工電路。或者，致能訊號可控制哪個追蹤緩衝器被啟動。又或者，可能有足夠平行導線以處理平行的交易。追蹤緩衝器114A, 114B,..., 114Y經由連接到導線120的導線120A, 120B...120Y，提供重演的指令與相關資訊給管線108。要注意的是來自追蹤緩衝器114的多個指令可能同時地通過導線120與MUX 110以再執行。同時，來自解碼器106的多個指令也可能第一次通過MUX 110。線ID和指令ID(instr ID)伴隨著每個指令通過管線。重演計數也可能伴隨著該指令。在載入與儲存指令的情況中，一載入緩衝器ID(LBID)與一儲存緩衝器ID(SBID)也可能伴隨著該指



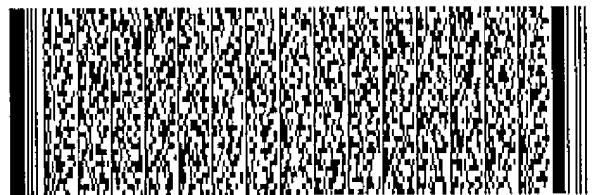
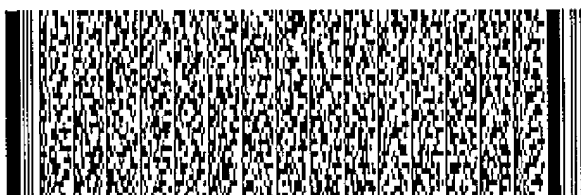
五、發明說明 (36)

相反的，如果依照時間順序store MX在load MX之後執行，那麼load MX的猜測性執行不會依照store MX相關之正確順序，而且不確定load MX將會載入正確的值。正確的值出現在記憶體位置 MX中(或儲存緩衝器項目的資料欄位保存store MX直到store MX最終退役)只會是巧合。為了要確保執行的終極正確性，MOB 178包括各種機制以確使記憶體資料在線之間的一致性。

a. 載入指令的執行

在載入指令執行之前，它的位址與儲存指令的位址一起比較以決定哪一，如果有的話，儲存指令是最接近先前相符儲存指令(CEMSI)。“相符”指的是與載入指令有相同的位址。“先前”指的是CEMSI在程式順序中在載入指令之前。“最接近”指的是在CEMSI和將被執行的載入指令之間沒有其他相符的儲存指令。如果只有一個先前相符儲存指令，它就是CEMSI。

如果有CEMSI，載入指令由CEMSI的資料欄位讀取它的資料。如果沒有CEMSI，載入指令由記憶體取它的資料，例如資料快取176，L2快取，或主記憶體。來自儲存緩衝器184或記憶體的資料經由MUX 192傳送，並寫到追蹤緩衝器114中由線ID和instr ID所指定的項目中。資料也可能被寫到暫存器檔案152中PRID所指定的暫存器中。資料也可能依快取規則(舉例來說，寫回，寫過等)被儲存在資料快取176中。因為能略過記憶體，例如資料快取176，L2快取，或主記憶體，所以MUX 192是一旁路。

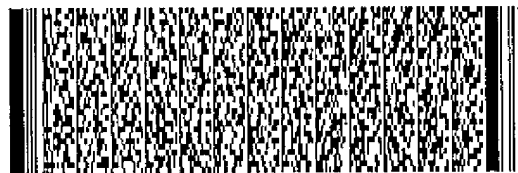
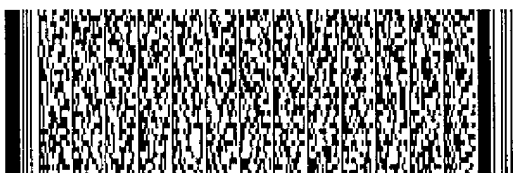


五、發明說明 (51)

任何修正。

元件符號說明

10	處理器	12	執行管線
14	追蹤緩衝器	34	最終退役邏輯
50	處理器	100	處理器
104	指令快取	106	解碼器
108	管線	110	MUX
114	追蹤緩衝器	112A-112X	程式計數器
124	線管理邏輯	134	最終退役邏輯
150	重新配置／指派單元		
152	暫存器檔案	156	分配排程／發送單元
158	執行單元	162	寫回單元
164	重排序緩衝器	172	包含位址產生器
176	資料快取	178	記憶體排序緩衝器
182	載入緩衝器	184	儲存緩衝器
188	DEMUX	192	MUX
202A	指令佇列陣列	206A	資料與附屬性陣列
208A	輸入暫存器檔案	208B	輸出暫存器檔案
210A	輸出暫存器檔案	212A	附屬性產生電路
214A	資料陣列	218A	附屬性產生與解碼電路
224A	控制電路	280B	比較器
284B	重演觸發邏輯	300	追蹤緩衝器
302	MOB控制電路	310	MOB
312	載入緩衝器	314	儲存緩衝器



五、發明說明 (52)

320	比較器	324	比較器
356	相符判定邏輯	390	資料路徑控制邏輯
394	重演觸發邏輯	400	多管線單元
402	多管線單元	430	系統
434	處理器	440	記憶體控制器
446	主記憶體	448	週邊設備

