

公告本

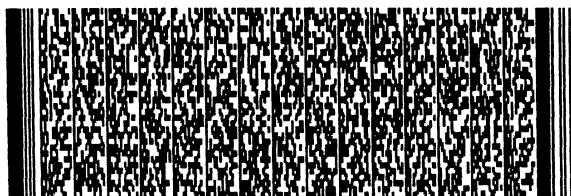
申請日期: 92.6.13	IPC分類
申請案號: 92116034	G06F 7/34

(以上各欄由本局填註)

發明專利說明書

I222017

一、 發明名稱	中 文	快取記憶體資料區段的配置與初始化機制
	英 文	CACHE DATA BLOCK ALLOCATION AND INITIALIZATION MECHANISM
二、 發明人 (共1人)	姓 名 (中文)	1. 羅德尼·胡克
	姓 名 (英文)	1. Rodney E. Hooker
	國 籍 (中英文)	1. 美國 US
	住居所 (中 文)	1. 美國德州78730奧斯汀市大景街9408號
	住居所 (英 文)	1. 9408 Big View Drive, Austin, TX 78730
三、 申請人 (共1人)	名稱或 姓 名 (中文)	1. 智慧第一公司
	名稱或 姓 名 (英文)	1. IP-FIRST, LLC
	國 籍 (中英文)	1. 美國 US
	住居所 (營業所) (中 文)	1. 美國佛蒙特市教堂巷1045號 (本地址與前向貴局申請者相同)
	住居所 (營業所) (英 文)	1. 1045 Mission Court, Fremont, CA 95054, U. S. A.
	代表人 (中文)	1. 陳鴻文
	代表人 (英文)	1. Miller Chen



一、本案已向

國家(地區)申請專利

申請日期

案號

主張專利法第二十四條第一項優先權

美國 US

2003/02/11

10/364,927

有

二、☐主張專利法第二十五條之一第一項優先權：

申請案號：

無

日期：

三、主張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

日期：

四、☐有關微生物已寄存於國外：

寄存國家：

寄存機構：

寄存日期：

寄存號碼：

無

☐有關微生物已寄存於國內(本局所指定之寄存機構)：

寄存機構：

寄存日期：

寄存號碼：

無

☐熟習該項技術者易於獲得,不須寄存。

五、發明說明 (1)

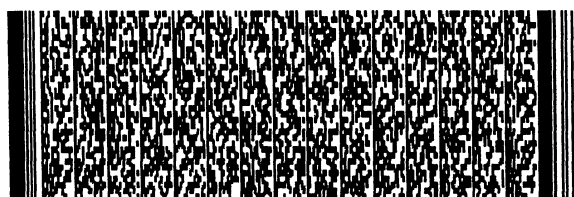
與相關申請案之對照

[0001] 本申請案與下列同在申請中之中華民國專利申請案有關，其申請日與本案相同，且具有相同的申請人與發明人。

<u>台灣申請案號</u>	<u>申請日</u>	<u>DOCKETNUMBER</u>	<u>專利名稱</u>
92116035	92/6/13	CNTR. 2157	意圖進行儲存之預取機制
92116036	92/6/13	CNTR. 2162	快取線配置與初始化之裝置及方法
92116037	92/6/13	CNTR. 2182	區段記憶體之意圖進行儲存的預取機制

【發明所屬之技術領域】

[0002] 本發明係有關微電子學的領域，尤指一種裝置



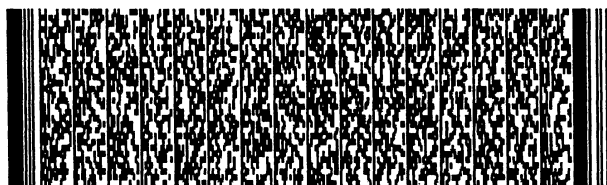
五、發明說明 (2)

及方法使程式員得以命令微處理器在其內部快取記憶體(cache)上執行配置與初始化一個區段(block)快取線(cache line)的預取運算(prefetch operation)。

【 先前技術 】

[0003] 在今日的微處理器，其內部邏輯區段(logic block)間的資料傳輸速度遠超過於其與外部記憶體的存取速度。在一個x86桌上型電腦組態中，其匯流排(bus)與系統記憶體之間的界面運作速率是以百萬赫(megahertz)的百倍計，但是其內部微處理器時脈速率卻已接近數十倍的十兆赫(gigahertz)。因此，近年來已發展出一個快取記憶體結構的層級體系，此體系使得高效能微處理器不必在每次讀取(read)或寫入(write)資料時，必須在一個緩慢的記憶體匯流排(memory bus)上執行作業(transaction)，而更能發揮其效能。

[0004] 一個機載(on-board)，或區域(local)，快取記憶體在一個管線化(pipeline)微處理器中是一個獨立的單位，在本質上，其運作方式對於在管線化微處理器中流動的指令而言是透明(transparent)的，此方式確保一個應用程式(application program)中之指令所需要的資料已經常駐在其快取記憶體內，而可以用管線速率存取，而不是以記憶體匯流排速率。不同的技術使用不同的快取記憶體架構，有些係由多層(multiple levels)快取記憶體所組成：第一層快取記憶體係非常接近處理器的執行邏輯

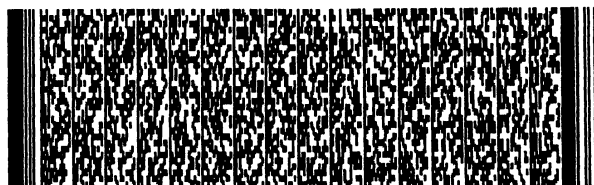


五、發明說明 (3)

(execution logic)，第二層快取記憶體可以是晶片上(on-chip)或非晶片上，係用來儲存較不常存取的資料，第三層快取記憶體則可能是在記憶卡(memory card)上，以此類推。不論應用那一種架構，熟習此領域技術者將發現使用快取記憶體的旨在於排除當匯流排作業係經由一個緩慢的記憶體匯流排所發出時的微處理器管線中指令的停滯(stalled)，此匯流排作業係為了取得一待決的(pending)讀取或寫入運算所需之資料。當此狀況發生，程式的執行將產生令人難以忍受的暫停(halt)，直到得到所需資料為止。

[0005] 今日電腦系統元件(device)間共用記憶體區域的現象使得情況更為複雜。舉例而言，主微處理器(primary microprocessor)與通訊微處理器(communications microprocessor)之間的通訊是經由在一指定記憶體區域上的讀取及寫入資料。視訊元件(video device)上顯示資料予操作員(operator)之視訊卡(video card)的微處理器與主微處理器共用一些被稱之為視訊緩衝器(video buffers)的記憶體區域，也是很常見的情形。

[0006] 在共用記憶體系統中，可能發生來自一個共用區域(region)的資料存在於兩個不同微處理器的區域(local)快取記憶體中，或是存在於連結到同一個記憶體匯流排之不同元件上。若所有元件只是單純的讀取資料，則允許它們將資料常駐於其區域快取記憶體結構，並不會

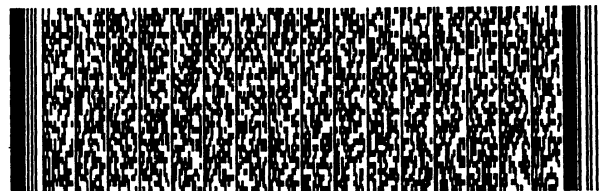
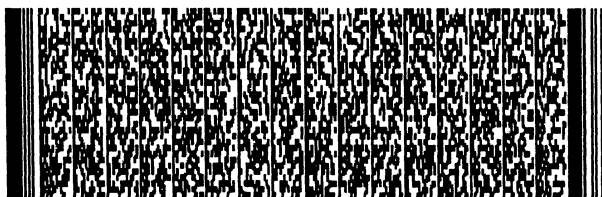


五、發明說明 (4)

造成任何傷害。但是當它們均被允許改動(modify)存在於其區域快取記憶體的资料時，即會造成不可預期的後果。

[0007] 為防止此情況發生，系統設計者開發了快取記憶體一致性協定以標示快取記憶體中資料之狀態。MESI是最普遍使用的協定。依照MESI來維護區域快取記憶體可以確保同一資料的兩個副本不會在同一時間被改動。MESI共用狀態告知區域快取記憶體一特定區段的資料是否為共用(shared)。如是共用，則區域處理器在經由一較慢的記憶體匯流排上執行作業以取得排他許可(exclusive permission)之前，不得改動資料。若欲改動資料，處理器需先取得資料的排他所有權。

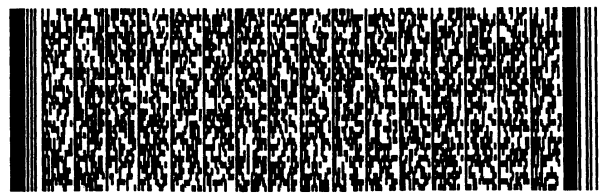
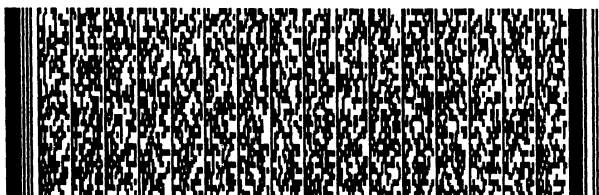
[0008] 本發明在此強調的問題是有關於欲將資料寫入記憶體時所招致的發生在程式內的延遲。熟習此領域技術者將發現快取記憶體並無合理的方法以得知何時一特定記憶體區域會最先被需要，因此當該特定記憶體區域第一次被載入到區域快取記憶體時，總是會引起記憶體匯流排延遲。認知此項事實，設計者開發了一可在微處理器上執行的預取指令。但是預取指令並不能在程式流程的運算元上運作。確切的說，預取指令命令區域快取記憶體從記憶體上載入運算元到該快取記憶體以備未來之用。而且因為快取記憶體單元與記憶體間的運作方式，對一微處理器管線內之指令流程而言，是透明的，因此審慎的方式是在需用資料之前先發出預取指令，使得快取記憶體可以在需用資料之前，從記憶體提取資料-與在主程式流程(primary



五、發明說明 (5)

program flow) 中其他指令的執行平行處理。然後當隨後的指令(subsequence instruction)出現，並且需要存取已預取之資料時，此資料已經在快取記憶體中立即可存取(readily accessible)，所以程式的執行不會因為等待從記憶體提取資料而停滯。已預取之資料在快取記憶體中立即可存取係僅指其將要被讀取而已。如果此預取之資料會被隨後的指令所改動，則程式的執行將必須延遲以等待快取記憶體單元到匯流排去請求共用資料的排他所有權，而後區域處理器始可改動此資料。

[0009] 如前所示，共用記憶體區域被廣泛的應用在今日電腦系統，以提供各元件間之資訊通訊之用。以元件間之一連接到匯流排的通訊緩衝器(communication buffer)為例，在資料區域之一些資料的存在或不存在(換言之，該指定區域被設定成同一邏輯數值，或全為一，或全為零)，對一元件而言，可表示另一元件已經準備好接收資料。相對的，在一視訊緩衝器的組態中，經由將所有視訊緩衝器項目設為同一數值，微處理器可以開始為操作員"繪製"背景顏色。為了在元件間轉移資訊，上述兩個例子皆須在資料區域內之資料實體上儲存特定資料。而這些事態皆不須在轉移資訊之前讀取該資料區域的資料。因此，在共用記憶體計算系統內之元件在轉移資料到一共用記憶體區域時，並無經由快取記憶體以前進到記憶體匯流排的有效律方法-即使經由預取運算，該共用記憶體區域早已事先載入到快取記憶體-因為該快取記憶體內之資料



五、發明說明 (6)

可能不是排他的擁有。

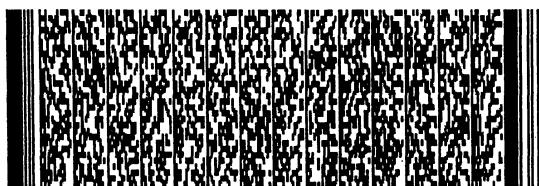
[0010] 因此，我等需要一種裝置及方法使得程式員得以命令微處理器執行預取資料，此資料係處於不求助於匯流排作業以取得資料的排他所有權，而可被隨後的運算所改動的狀態。

[0011] 除此之外，我等亦需要一種裝置及方法使得程式員得以命令微處理器將被已排他的預取之資料初始化(initialize)。

【發明內容】

[0012] 本發明如同前述其他申請案，係針對上述及其他習知技術之問題與缺點加以克服。本發明提供一種更好的技術以預取一區段的快取記憶體資料，其狀態係為使得隱含的儲存運算可立刻被公告以便該區段之資料被初始化成一指定數值。在一具體實施例中，提供一種組態為配置與初始化記憶體中一區段之資料的微處理器裝置。該裝置包括轉譯邏輯與執行邏輯。轉譯邏輯將一區段配置及初始化指令轉譯成一微指令序列，此微指令序列係用以命令微處理器預取處於排他狀態的一區段之快取線，並且將此區段之快取線初始化成一指定數值。執行邏輯係耦接至轉譯邏輯。該執行邏輯接收前述微指令序列，然後經由記憶體匯流排發出作業，要求處於排他狀態一個區段之快取線，並且將此區段之快取線初始化成一指定數值。

[0013] 本發明的一個目的係提供一種微處理器裝置以



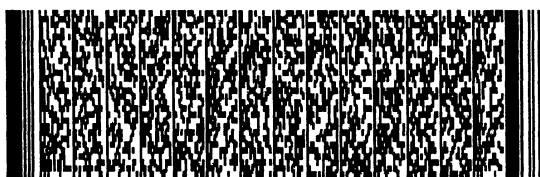
五、發明說明 (7)

執行區段配置與初始化運算。此裝置包含一區段配置與初始化指令及一轉譯器。該區段配置與初始化指令命令微處理器預取處於排他狀態之一個區段之快取線，並且將此區段之快取線初始化成一指定數值。轉譯器則是接收區段配置與初始化指令，並將此區段配置與初始化指令轉譯成相關的微指令，此相關的微指令命令在微處理器內的執行邏輯經由一記憶體匯流排發出匯流排作業，此匯流排作業係要求此區段之快取線的排他所有權及將此區段之快取線初始化成一指定數值。

[0014] 本發明的另一目的，在於提供一種配置與初始化一個區段之快取線的方法。此方法包括提取 (retrieving) 一區段配置與初始化巨集指令；轉譯該區段配置與初始化巨集指令成一微指令序列，此微指令序列係用以命令一微處理器預取處於排他狀態的一個區段之快取線，並且將此區段之快取線初始化成一指定數值；而且，回應此微指令序列，經由記憶體匯流排發出匯流排作業以讀取該處於排他狀態之快取線，並且將該區段之快取線初始化成該指定數值。

【實施方式】

[0034] 以下的說明，係在一特定實施例及其必要條件的脈絡下而提供，可使一般熟習此項技術者能夠利用本發明。然而，各種對該較佳實施例所作的修改，對熟習此項技術者而言乃係顯而易見，並且，在此所討論的一般原



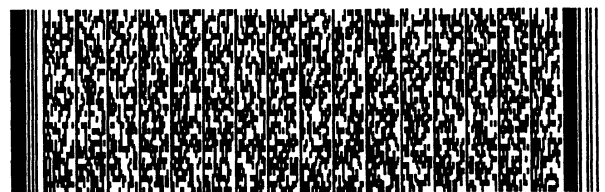
五、發明說明 (8)

理，亦可應用至其他實施例。因此，本發明並不限於此處所展出與敘述之特定實施例，而是具有與此處所揭露之原理與新穎特徵相符之最大範圍。

[0035] 前文已針對今日之管線化微處理器如何執行預取運算，作了背景的討論，有鑑於此，在圖一至三，將呈現一強調今日預取技術之限制的例子。緊接著，在圖四至十八，將呈現本發明之討論。本發明使得程式員得以命令微處理器預取處於排他MESI狀態的資料至其快取記憶體，並且將資料初始化為所規定數值，因此，得以避免因為隨後的儲存運算執行資料的初始化所造成的任何程式延遲。

[0036] 請參閱圖一，其係一方塊圖，用以說明在今日管線化微處理器100內的有效管線階段101-105。此微處理器有一個提取階段101，一個轉譯階段102，一個暫存階段103，一個定址階段104，及一個執行階段105。

[0037] 於運作時，該提取階段101從系統記憶體之一指令範圍120中提取(retrieve)巨集指令121-123以供微處理器100執行。該巨集指令121-123接著被送至轉譯階段102。該轉譯階段102將巨集指令121-123轉譯成對應的微指令(或稱原生指令)序列(未顯示)，此微指令序列係命令微處理器100執行巨集指令121-123指定之運算。非常類似於工廠產品在裝配線上流經連續的工作站之方式，該微指令亦同步於管線時脈訊號(未顯示)的流經在管線中隨後的階段103-105。根據前述方式，微指令被送至暫存階段103。如果有一特定微指令指定一運算元被儲存於暫存階



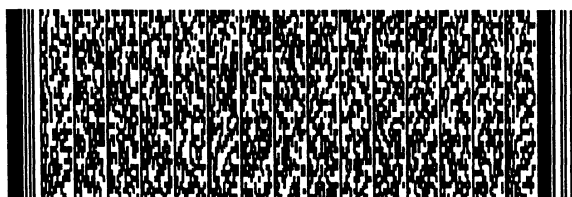
五、發明說明 (9)

段103之暫存器內，則邏輯在那一點上可存取該暫存器以提取該運算元，並且將之隨同該特定微指令一起送至定址階段104。定址階段104包含用以產生位址以存取儲存在資料記憶體108內之運算元的邏輯。相似於暫存階段103，該定址階段104亦將所產生之位址，隨著相應的微指令，傳送至執行階段105。

[0038] 執行階段105執行該微指令所指定的運算。在當今的微處理器100，運算的型式是由指令集架構(instruction set architecture)所決定，但是熟習此項技術者將發現這些運算不會超出一般的範疇，例如邏輯運算，算術運算，及記憶體存取運算(換言之，資料讀取與資料寫入運算)。由執行指定之運算所產生的結果，若不是儲存於儲存於暫存階段103之暫存器內，即是被寫入到資料記憶體108內的記憶位置。

[0039] 熟習此項技術者將發現今日的管線化微處理器100也許會有比圖一之101-105更多的階段，因為，經由分解管線中的主函數以增加管線中階段的數目是種經證實可增加管線(pipeline)中指令121-123之產出量的技術。為了簡明起見，如圖一所示之當今微處理器100的管線化階段101-105已經足以說明先前技術之缺點，而不需以不相關的細節增加讀者的負擔。

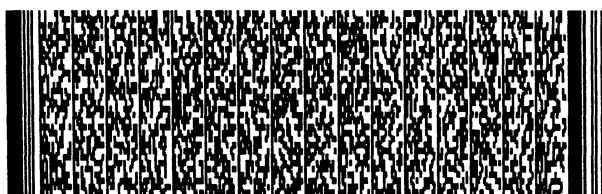
[0040] 值得注意的是在當今微處理器100中，其執行階段105除了有一資料快取記憶體107之外，還有執行邏輯106。該資料快取記憶體107的運作是與在管線化階段101-



五、發明說明 (10)

105 中指令之執行平行，此運作方式確保那些有高度可能性被一應用程式之指令121-123所存取的資料已經存在於快取記憶體107，因此當資料存取微指令(換言之，載入記憶體或儲存記憶體微指令)進行到執行階段105時，該執行邏輯106能夠在一或兩個管線時脈週期之內執行該資料的存取，而不是造成可能數以百計之時脈週期的程式延遲，只因為等待經由記憶體匯流排110到資料記憶體108以執行該資料的存取。在一有效率的快取記憶體系統組態中，資料的載入與儲存絕大部份發生在經由快取記憶體匯流排109之執行邏輯106與資料快取記憶體107之間，並且資料快取記憶體107的運作方式是相對的透明於流經管線化階段102-105的微指令流程，此運作方式確保資料實體的快取副本與系統記憶體108是同步且一致的。

[0041]MESI(修改，排他，共用，無效)(modified, exclusive, shared, invalid)是個普遍使用的協定，此協定係用以在一系統組態之記憶體108的共用區域內確保其快取記憶體項目的一致性。雖然並未在圖一中描述，但是為了使用同一資料運算之目的，在一電腦系統組態中之其他元件(未顯示)是可以共用記憶體108的一些區域。舉例而言，視訊卡可以與微處理器100共用記憶體108的一個區域，以便於存取微處理器100所產生的監視器顯示資料。另一例則是在系統匯流排110上的多重元件可以經由從資料記憶體108中之共用區域的資料讀取及資料寫入來彼此連絡。架構性研究之詳細敘述以提供使用MESI協定的



五、發明說明 (11)

動機並不在本發明的應用領域；此處只需了解MESI在確定系統記憶體108與區域快取記憶體結構107之間資料的一致性之普遍應用。

[0042]因為經由記憶體匯流排110的作業需要數以百計之時脈週期才能完成，所以資料被以包含數個位元組的區段形態在資料快取記憶體107中移進及移出。這些區段稱之為快取線。雖然快取線行寬(換言之，快取線之位元組大小)會因不同的架構而改變，在今日的系統架構非常常見的有32-位元組行寬，或64-位元組行寬，或甚至128-位元組行寬。

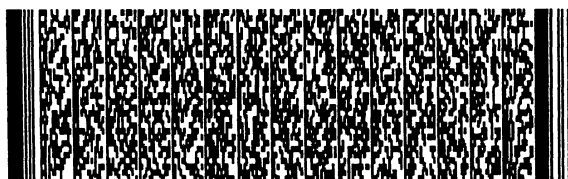
[0043]即使是最有效率的快取記憶體結構107在從記憶體108，經由記憶體匯流排110，到快取記憶體107的進行資料轉移，不可避免地會有所延遲。但是在提供一快取線予快取記憶體107之後，其隨後的對該快取線內之資料實體的存取即不會發生重大的延遲，因為快取記憶體107及其快取記憶體匯流排109的速度是與微處理器100內其他邏輯的速度(例如執行邏輯106)相近的。

[0044]依照MESI協定，在區域資料快取記憶體107內的快取線可處於下述任一四種狀態中：修改，排他，共用，及無效。一修改狀態之快取線係指在該快取線上執行一區域儲存運算後，但是尚未與主記憶體108同步化之快取線。監測來自其他元件(亦稱為匯流排元件之經由其記憶體匯流排110的記憶體作業是區域快取記憶體107的責任，因此如果匯流排元件從一修改狀態之快取線要求資



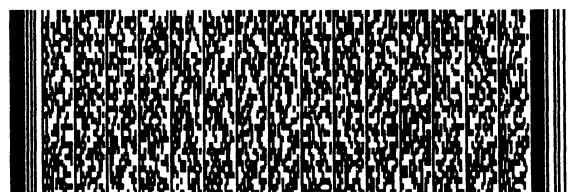
五、發明說明 (12)

料，則區域快取記憶體107會將該修改過的資料送至該要求資料之匯流排元件。此種對匯流排110的監測稱為匯流排窺視模式(bus snooping)。一排除狀態之快取線係指其區域快取記憶體107可以在該快取線上執行儲存運算之快取線。排除狀態暗示其區域快取記憶體107擁有對該快取線的排除所有權；微處理器100因此被許可修改其內容。一共用狀態之快取線係指存在於兩個或數個在匯流排110上元件的區域快取記憶體107內之快取線。因此，任一元件均可從共用的快取線讀取資料，但均不被許可去修改其內容。為能在共用的快取線上修改資料(換言之，執行儲存運算)，在修改其內容之前，元件100需先經由記憶體匯流排110與其他元件實施適當的作業以取得該快取線的排除所有權(換言之，讀取處於排除MESI狀態之快取線到其區域快取記憶體107)。一旦取得該快取線的排除所有權，即可執行儲存運算，並且將該快取線狀態變更為修改狀態。在公告(posting)儲存運算之前，先行要求快取線的排除所有權可以保證資料的一致性，因為在任一時間點，均只有一個元件100可以修改其快取線的內容。當區域快取記憶體107偵測到(經由窺視模式)經由記憶體匯流排110至其快取線的寫入作業，或是當另一元件經由記憶體匯流排110發出匯流排作業以取得該快取線的排除所有權時，該快取線狀態即變更為無效狀態。將一快取線標示為無效表示其內之資料與記憶體108不一致而無法被讀取或寫入。



五、發明說明 (13)

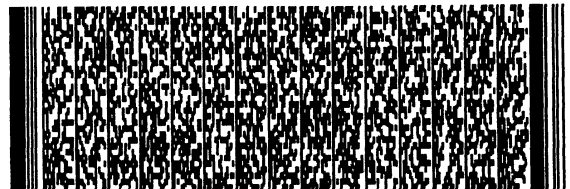
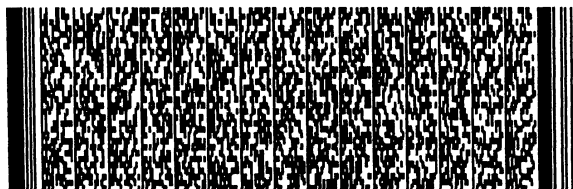
[0045] 因為一快取記憶體107的運作是與在微處理器管線中之指令流程平行，在程式流程120中，設計者需在資料的存取被要求之前，先提供預取巨集指令122以載入該資料到一快取記憶體107，因此得以克服資料必須起始的從記憶體108提取至快取記憶體107所導致之延遲的缺點。在程式流程120中通常有一預取指令122以命令其區域快取記憶體107從記憶體108載入一快取線，而此預取指令122與隨後指令的執行是平行的，因此當程式流程120的指令123要從該快取線存取資料的時候，該快取線已經存於其快取記憶體107中。在圖一的例子中，一預取指令122，`PREFETCHTO [EAX]`，命令由暫存器EAX的內容所定位之快取線被載入到快取記憶體107，使得其內容可被隨後的資料存取指令123，`MOV EBX, [EAX]`，在資料流程120中被執行時所使用，此資料存取指令123命令微處理器100從由暫存器EAX指定之位址讀取資料，並且將之移至暫存器EBX。因為x86指令已被廣泛的認知，為了簡明起見，傳統上前述資料流程120內的預取指令122及資料存取指令123之描述係根據x86的指令集架構。但是熟習此項技術者將發現，在許多其他的指令集架構亦提供預取指令122以命令微處理器100從記憶體108讀取一快取線至一區域快取記憶體107，使得隨後的指令123從該快取線執行一指定的資料讀取運算不會發生延遲。如果預取指令是很明智的置於資料流程120內，則可以有效率的克服因為在快取記憶體107的起始存取資料所造成之延遲，而因此顯著的改進程式的



五、發明說明 (14)

執行速度。當完成預取運算之一經由記憶體匯流排110的作業後，其所需求的快取線不是以排他狀態(若是該區域快取記憶體107擁有該快取線的唯一快取副本時)，即是以共存狀態(若是其他元件亦擁有該需求快取線的快取副本時)存在於快取記憶體107內。不論其在那一狀態，在該快取線內的資料實體是可被立即讀取(read)存取的。但是如前述所指明，為將資料寫入到一快取線(換言之，執行一儲存運算)，需要擁有該快取線之排他所有權。因此，如果該預取運算導致快取處於排他狀態之快取線，則一待決之儲存即可立刻對此快取線公告。但是若該來自匯流排110之快取線是於共用狀態，則一待決之儲存必須被停滯(stalled)，以等待該快取記憶體單元107經由匯流排110發出作業以取得該快取線之排他所有權。在該快取線在排他狀態下被送至快取記憶體107之後，則該待決之儲存即可被公告。

[0046] 現請參閱圖二，其係為一方塊圖200，用以描述在圖一之微處理器內執行預取運算之一界面至記憶體的快取記憶體單元。該方塊圖200顯示在微處理器201內被應用為執行預取運算的邏輯。該微處理器201內有一轉譯器210以接受一巨集指令流程202並將之轉譯成對應的微指令211。用以命令對記憶體242作資料載入及儲存運算的微指令211隨後即被送到一快取記憶體單元220。該快取記憶體單元220包括記錄邏輯221及一資料快取記憶體222。該記錄邏輯221係耦接至一匯流排單元230。該匯流排單元230

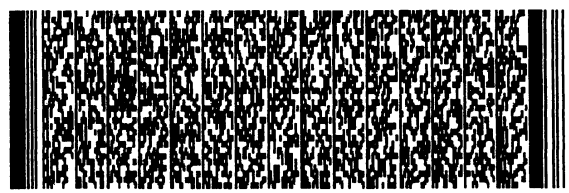
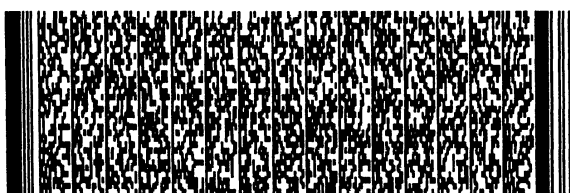


五、發明說明 (15)

係界面到一系統記憶體匯流排240，該系統記憶體匯流排並且與系統記憶體242及其他匯流排元件241耦接。

[0047] 巨集指令之示範流程202說明如何指定一預取運算及如何可在此預取之資料上執行隨後的讀取與儲存運算。一有關於此運算序列在桌上型電腦上的常見範例是記憶體內計數器的讀取及增加。一序列的預取，讀取，及儲存運算需要既可以在快取線內讀取資料，也可以在快取線內修改資料。因此，示範流程的第一個巨集指令202，`PREFETCH [EAX]`，命令微處理器201去預取一位址對應於暫存器EAX內容的快取線。第二個巨集指令202，`MOV EBX, [EAX]`，命令微處理器201去讀取一位址為暫存器EAX所指定之記憶體位置的內容，並將該內容寫入暫存器EBX。第三個巨集指令202，`INC EBX`，命令微處理器201增加暫存器EBX的內容。第四個巨集指令202，`MOV EAX, [EBX]`，命令微處理器201在對應於暫存器EAX內容的記憶體位置上儲存該暫存器EBX的內容。上述詳細的預取，讀取，及儲存運算僅僅是增加一個數值到位址由暫存器EAX所指定之記憶體。值得注意的是為了要有效率的利用該預取指令202，必須在第二個巨集指令202，`MOV EBX, [EAX]`，之前提供充分的該預取指令202，使得因為載入由EAX之內容所指定之快取線的所造成的延遲，可以被中介的巨集指令202的平行執行所吸收。但是，為了簡明起見，該中介的巨集指令202並未在方塊圖200中敘述。

[0048] 轉譯器210將該預取巨集指令202轉譯成一對應

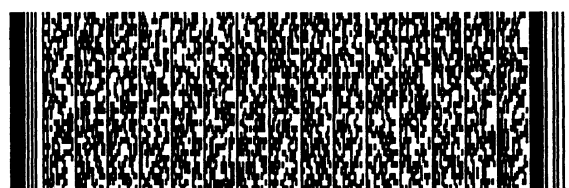


五、發明說明 (16)

的預取微指令211，`PREFETCH[EAX]`，然後送到快取記憶體單元220。記錄邏輯221詢問資料快取記憶體222以決定所要求的快取線是否已存在並且有效(換言之，即非處於無效狀態)於資料快取記憶體222內。若答案為否，則該記錄邏輯221命令其匯流排單元230，經由其系統記憶體匯流排240，發出作業以從記憶體242取得所要求的快取線。若其他的匯流排元件241均無該所要求之快取線的副本時，則該記錄邏輯221即將所要求的快取線以排他狀態送到資料快取記憶體222。若一或數個匯流排元件241擁有該要求的快取線之區域副本時，則該記錄邏輯221即將該快取線以共用狀態寫入到資料快取記憶體222。在前述任一狀態，該所要求的快取線均存在於快取記憶體222內以備隨後的存取運算使用。

[0049] 轉譯器210將第二個巨集指令202轉譯成一載入微指令211，`LD EBX, [EAX]`，命令微處理器載入位址為暫存器EAX所指定的記憶體之內容到暫存器EBX。如前圖一所討論的，微處理器內之執行邏輯(未顯示)從快取記憶體單元220要求該記憶體位址之內容。因為該快取線所有之內容，由於預取運算的結果已經存在於該資料快取記憶體222中，所以市立即可用並且載入微指令211的執行不會有任何延遲。

[0050] 第三個巨集指令202被轉譯成一對應的增加微指令211，`INC EBX`，此指令命令執行邏輯將暫存器EBX之內容增加。因為不需要新的資料，所以該載入微指令211

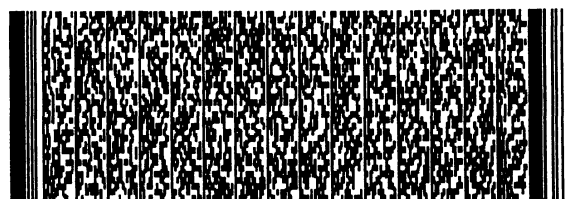
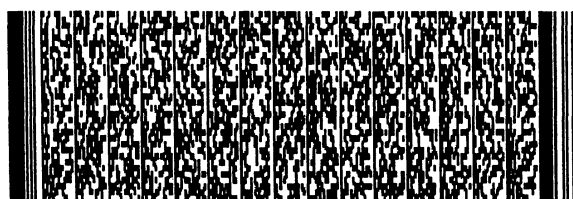


五、發明說明 (17)

不會被送到快取記憶體單元220。

[0051]最後，在流程中的第四個巨集指令202被轉譯成一儲存微指令211，ST [EAX], EBX，此指令命令執行邏輯去執行一資料儲存運算，以將暫存器EBX的內容寫入到位址由暫存器EAX內容所指定之記憶體位置。該儲存微指令211從而以待決的儲存運算型式被送到其快取記憶體單元220。於是，記錄邏輯221偵測到以待決的儲存運算為目標的快取線存在於該資料快取記憶體222。若該快取線是處於排他狀態，則此待決的儲存可立刻被公告並將該快取線之狀態變更為修改狀態。另一方面，若該快取線是處於共用狀態，則該快取記憶體單元220確定一停滯訊號223以暫停該微指令211在微處理器201的管線階段中的進行，同時該執行邏輯221則命令其匯流排單元230，經由其記憶體匯流排240，執行作業以取得該快取線的排他所有權。一旦取得排他所有權，則可許可待決的儲存對該快取線公告其資料，並且終止停滯訊號，從而繼續程式的執行。

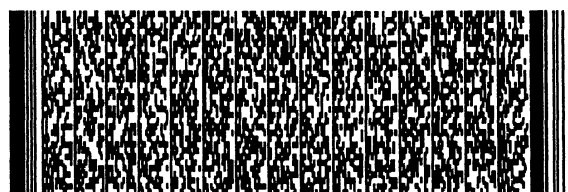
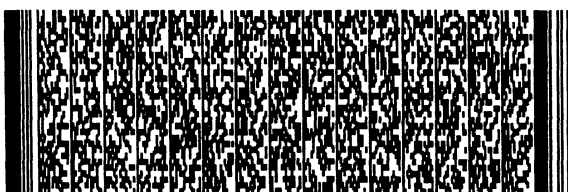
[0052]現在考慮一種運算型式只是單純的寫入資料到記憶體中而不需先讀取資料，或是一種運算型式是會先讀取資料，但是此型式確定預期有一儲存運算會隨後被公告。在這些案例中，只有對那些須先讀取資料的案例執行預先的預取指令是確定可以將程式延遲減到最小。並且在這種案例中，若預取的結果使所要求的快取線處於排他狀態，則可排除因儲存運算所導致的程式延遲。但是若預取的結果使所要求的快取線處於共用狀態，則因儲存運算所



五、發明說明 (18)

導致的程式延遲將不可避免。這是個問題，因為今日的指令集架構並未提供一方法以命令微處理器201去排他的預取一快取線到資料快取記憶體222。雖然回應一預取運算的快取線可以是排他的，但是此狀態並不能保證。這是因為預取巨集指令架構化的推定其預取的資料是會被讀取，並且其結果之經由系統匯流排的作業要求該快取線被提取，不論其是否處於共用狀態。舉例來說，在x86架構中，一x86預取指令的執行結果之經由匯流排240發出的作業作業是一資料讀取運算。該資料讀取運算要求一快取線的副本，並不管其是處於何種狀態。

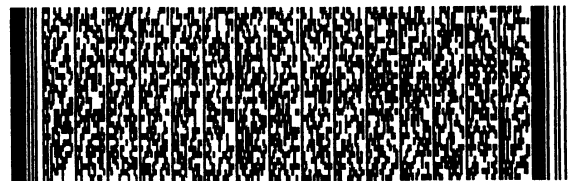
[0053] 現請參閱圖三，其顯示一時脈圖，用以描述由圖一及圖二所示之微處理器201所發出之兩個可能的匯流排作業集301，302，此匯流排作業係經由記憶體匯流排240發出以執行一預取與隨後的儲存運算。此二作業集301，302包括在微處理器201內的從匯流排單元230到記憶體匯流排240的請求作業303，與同樣在微處理器201內從記憶體匯流排240回到匯流排單元230的回應作業304。作業集301描述那些當回應一預取運算的快取線是排他狀態時所執行的作業303-304。作集302描述那些當回應一預取運算的快取線是共用狀態時所執行的作業303-304。如同圖二所述，當執行一預取指令時，記錄邏輯221命令其匯流排單元230對其記憶體匯流排240發出一資料讀取請求303，DATA READ [EAX]，要求被暫存器EAX所指定的快取線被送到其區域快取記憶體222。該資料讀取請求303在作



五、發明說明 (19)

業集301是於時間點A發出，在作業集302則是於時間點D發出。該記憶體匯流排240於是回應發出一包括該所要求的快取線的資料回應請求304回到匯流排單元230。若該快取線是在排他狀態，則在作業集301的資料回應請求304，DATA RESP [EAX].E，在時間點B被送回到匯流排單元230。若該快取線是在共用狀態，則在作業集302的資料回應請求304，DATA RESP [EAX].S，在時間點E被送回到匯流排單元230。在這時，資料可從快取記憶體222讀取而不會導致匯流排作業延遲。

[0054]當一隨後的儲存運算面對該上述作業提供之快取線時，作業集302的場景說明為了能公告該儲存運算所必然發生之作業303，304。在作業集301中，因為快取線起始即是排他狀態，欲公告該儲存運算只須在時間點C發出一資料寫入作業303，DATA WRITE [EAX]，經由匯流排240將資料寫入到記憶體242。但是如作業集302所示，在時間點H之資料寫入作業303可以被發出之前，必須先執行時間點F及G的作業303及304，以便能將快取線的所有權狀態由共用提升到排他。在時間點F，匯流排單元230發出一資料讀取與無效請求303，DATA READ/INV [EAX]，用以要求該共用狀態快取線的排他所有權。在數百時脈之後的時間點G，從匯流排240接收到一回應請求304，DATA RESP [EAX].E，以將該快取線的狀態升級到排他狀態。在時間點G接收到回應請求304之後，然後在時間點H該資料寫入作業303始可對匯流排240公告。

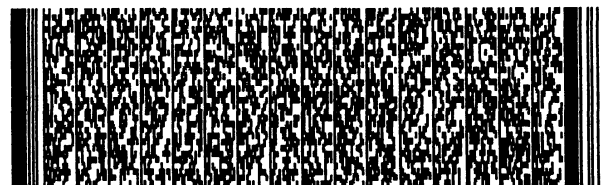
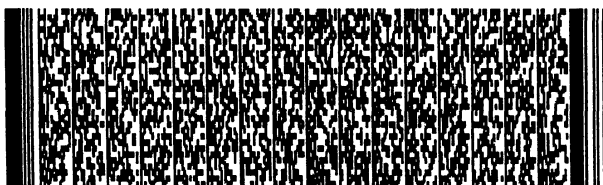


五、發明說明 (20)

[0055] 值得注意的是圖三中的作業集301，302係以一般性的方式表示，因為不同的微處理器架構應用不同的語義來說明匯流排作業303，304。除此之外，值得注意為了簡明起見，在圖三之時脈圖中已經省略了先取得對資料匯流排240存取(例如BUS REQUEST，BUS GRANT等等)之所有的作業。

[0056] 今日的發明者觀察到當今的資料預取指令受限於其並不支援確定預期的儲存運算，因此無法因為明確意圖進行對該快取線執行一儲存運算而有利的預取一快取線到快取記憶體222 - 一意圖進行儲存運算的預取 - 不論該快取線內容的讀取是否先於對該快取線公告一儲存運算。若細查作業集302中之作業303，304，很明顯的預取一處於共用狀態的快取線，只有在該快取線的讀取會早於對其公告一儲存運算的情況下才有幫助。若是一儲存運算將對該共用的快取線公告，則程式的執行必須被延遲以便將該快取線的狀態從共用提升到排他。

[0057] 程式員雖然了解今日預取指令的限制，然而仍應用它們為作意圖進行儲存情況下的預取，因為此預取指令可能(有時而非時常)於回應一資料讀取請求時，取得一快取線的排他所有權，僅只因為無其他匯流排元件擁有該要求之快取線的副本。但是，更理想的情況則是避免預取一處於共用狀態的快取線，而是命令一微處理器201去預取一處於排他狀態的快取線。本發明係指向一種裝置與方法，用以預取一處於排他MESI狀態的一單一快取線與一多



五、發明說明 (21)

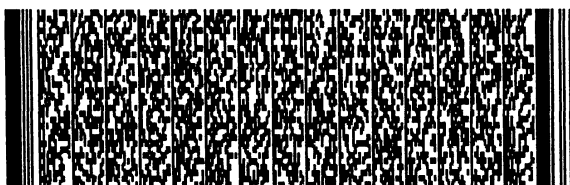
重快取線，並且隨意的初始化該預取的快取線成一指定數值。本發明現將參照圖四至圖十八進行討論。

[0058]請參閱圖四，其係為依據本發明以顯示一延伸預取指令400之方塊圖。該延伸預取指令包括一選用的多重前置碼實體401，其後為一預取運算碼402，其後則為一延伸位址指定元403。在一具體實施例中，每一個前置碼及延伸位址實體401，403都是8位元大小，而預取運算碼實體402則為一或二個位元組大小，除非本文中另行修訂，所有的實體401-403均與x86之指令集架構一致。

[0059]在運作上，該預取運算碼402係為一指定運算碼數值，用以命令一相符的微處理器執行一預取運算。在一x86的具體實施例中，其運算碼實體401的指定數值為0F18H。一個或數個選用的前置碼實體401可用來命令一相符的微處理器執行某些類型的附加運算，例如由一計數器界定次數的重複運算(例如在x86架構中的REP前置碼)，迫使執行一原子運算(例如在x86架構中的LOCK前置碼)等等。延伸位址指定元403是用以指定該特定類型之預取運算的執行。在一x86的具體實施例中，所熟知的延伸位址指定元403是ModR/M位元組403。

[0060]依據本發明，當微處理器偵測到一預取巨集指令400時，依照由延伸位址指定元403內容所指定的指示數值，該微處理器被命令去執行從記憶體預取資料到快取記憶體，其範例將在圖五中進行討論。

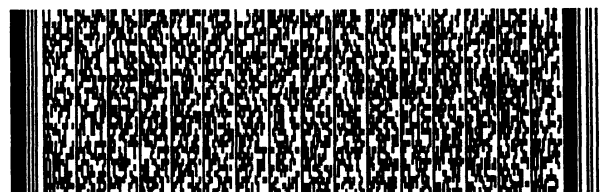
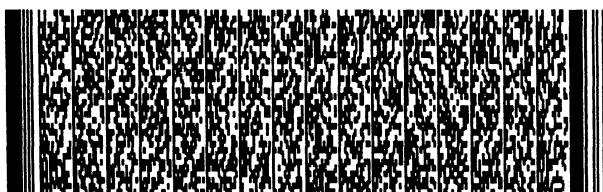
[0061]圖五係一表格500，其為圖四之延伸預取指令



五、發明說明 (22)

內的延伸位址指定元欄位403之一具體實施例，解說依據本發明該延伸位址指定元欄位403如何編碼以命令微處理器執行預取一處於排他MESI狀態的快取線。為了說明本發明起見，此處使用符合x86架構的ModR/M位元欄位，但是，可以預期的是本發明包含任何提供將一預取-排他指示編碼到指令400的工具之架構。雖然圖五的例子指向將預取-排他(或是意圖進行儲存之預取)指示編碼到一延伸位址指定元403，熟習此項技術者將發現該預取指示也可以被編碼成一在運算碼欄位401內之一指定運算碼數值。

[0062]在此編碼範例中，一x86 ModR/M位元組使用該ModR/M位元組的5:3位元編碼一由預取運算碼401所指定型式的預取運算。今日，該x86預取指令使用數值000，001，010，及011以規定意圖進行讀取運算之預取指示。所有這四個數值000-011為命令一x86微處理器，在不同程度的接近度下，預取資料到其快取記憶體。舉例來說，一個T0指示(換言之，數值001)，命令微處理器預取一快取線到快取記憶體層級體系的所有層級，而一NTA指示命令微處理器預取一快取線到一非暫時性快取記憶體結構，並且進入一接近處理器的位置，同時將快取記憶體污染減至最低程度。但是x86預取指示000-011編碼的普遍特徵是一經由匯流排發出的資料讀取請求以要求一快取線的副本，並不會在乎該快取線是處於何種MESI狀態。本發明之一具體實施例將一額外的指示編碼進延伸位址指定元中，用以命令微處理器依據本發明利用一排他(.S)指示去預取一指



五、發明說明 (23)

定的快取線。圖五顯示一用x86 ModR/M位元組的5:3位元編碼成數值100的預取-排他指示。當該prefetch.s指示依據本發明編碼到一預取指令400時，則一相符的微處理器會被偵測到經由一記憶體匯流排發出作業以預取處於排他MESI狀態的資料。在一x86的具體實施例中，如前述圖三B所示，其回應於預取指令400之prefetch.s指示而發出的特定作業是一資料讀取與無效作業。在圖三B的例子中，該資料讀取與無效作業是用以將快取線從共用狀態提升至排他狀態。

[0063] 在x86指令集架構中，數值100的5:3位元編碼在此之前是被宣告為非法的，同表格500中數值101-111的5:3位元編碼所示。一非法的ModR/M位元組編碼造成一異常。但是依據本發明，在一x86的具體實施例中，該改善一預取-排他指示的額外編碼是合法的，並且將導致前述的匯流排作業預取一處於排他狀態的快取線。

[0064] 眾人皆知因為快取記憶體結構與記憶體間的互動是不存在於微處理器管線之指令流程中，所以預取指令400只能要求的預取係依據所提供之指示來執行。若一快取線現在並不被記憶體存取所佔用時，則可執行預取運算。但若一快取線正被佔用，則預取運算須被推遲。

[0065] 現請參閱圖六，其係一方塊圖，詳述依據本發明之執行一意圖進行儲存運算之預取之微處理器600。該微處理器600有三個值得注意的階段範疇：提取，轉譯，及執行。在提取階段的提取邏輯601係用以從一指令記憶

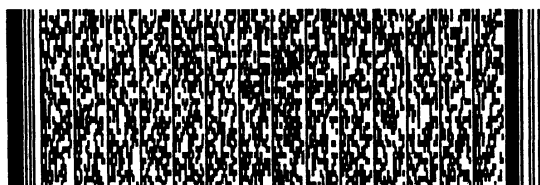


五、發明說明 (24)

體603中提取巨集指令到一指令快取記憶體602。該被提取的巨集指令然後經由一指令佇列604被送到轉譯階段。該轉譯階段的轉譯邏輯606係耦接至一微指令佇列608。該轉譯邏輯606包括延伸轉譯邏輯607。執行階段的執行邏輯609包括一延伸快取記憶體單元610。該延伸快取記憶體單元610有一資料快取記憶體611，此資料快取記憶體611係耦接至延伸記錄邏輯612。該延伸記錄邏輯612係耦接至一匯流排單元613。該匯流排單元613係耦接至一資料記憶體614。

[0066]在運作上，提取邏輯601依據本發明從指令記憶體603中提取格式化的指令到指令快取記憶體602，然後將此巨集指令依執行順序送到指令佇列604。該巨集指令被從指令佇列604提取後，被送到轉譯邏輯606。該轉譯邏輯606將每一送到此處的巨集指令轉譯成對應的微指令序列，該微指令序列係用以命令微處理器600執行由該巨集指令所指定的運算。延伸轉譯邏輯607則依據本發明去偵測延伸預取巨集指令，並且為將其轉譯成對應的延伸前置碼及位址指定元實體作準備。在一x86的具體實施例中，該延伸轉譯邏輯607被組態成偵測一x86預取指令，並且依據圖四及圖五所述之常規將該x86預取指令的ModR/M位元組轉譯成一預取微指令序列，該序列係用以命令微處理器600去排他的預取一快取線到資料快取記憶體611。

[0067]該微指令然後從微指令佇列608傳送到其執行邏輯609，在該執行邏輯中之延伸快取記憶體單元610則依

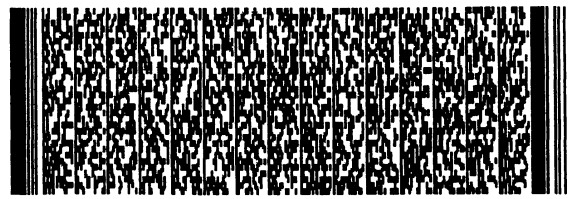


五、發明說明 (25)

據本發明被組態為執行一排除的預取運算。當該執行邏輯609執行一預取微指令序列時，其延伸記錄邏輯612命令匯流排單元613，經由記憶體匯流排605，發出作業到記憶體614要求在排除的MESI狀態下預取一指定的快取線到資料快取記憶體611內。

[0068]熟習此項技術者將發現圖六所描述的微處理器600只是一個依據本發明所簡化之後的一管線化微處理器600的代表。事實上，如前所述，今日的管線化微處理器包含許多管線階段。但是這些階段均可被概括的歸納成如圖六之方塊圖所示的三個階段群組，因此圖六之方塊圖可視為實現上述本發明之具體化所需的必要成分之說明。為了簡明起見，凡微處理器600中無關於本發明的成分均不在此描述。

[0069]請參閱圖七，其係為一方塊圖700，用以描述在圖六之微處理器內執行一預取與儲存運算之一界面至記憶體的快取記憶體單元。該方塊圖700顯示在微處理器600內被應用為執行預取運算的邏輯。微處理器701的延伸轉譯器710接收巨集指令流702，並將該巨集指令流702轉譯成對應的微指令711。微指令711在命令對記憶體742做資料載入與儲存運算之後，隨後即被送到一延伸快取記憶體單元720。該延伸快取記憶體單元720包括延伸記錄邏輯721及一資料快取記憶體722。該延伸記錄邏輯721係耦接至一匯流排單元730。該匯流排單元730係界面至一系統記憶體匯流排740，此系統記憶體匯流排740又與資料記憶體

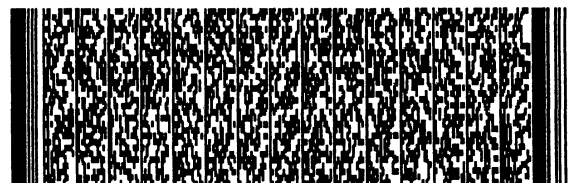
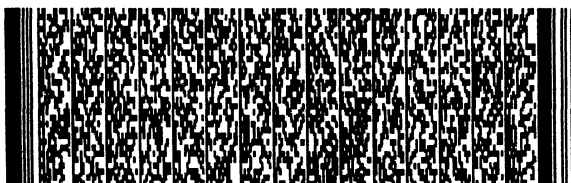


五、發明說明 (26)

742 及其他匯流排元件741 耦接。

[0070] 巨集指令之示範流程702 說明如何指定一預取指令及如何可在此預取之資料上執行隨後的讀取與儲存運算。如同圖二之敘述，一有關於此運算序列之桌上型電腦的常見範例是記憶體內計數器的讀取及增加。但是與圖一及圖二之微處理器201 不同的是，當依據本發明之一微處理器701 被命令去排他的預取一快取線時，不論此預取的資料是否會被中介的指令讀取，均可有效率的消除會關連到隨後的儲存運算的匯流排作業延遲。在圖七的巨集指令之示範流702 中預期一資料讀取運算發生早於在預取之資料上執行一儲存運算，但是熟習此項技術者將發現，沒有此一中介的資料讀取運算，資料也可以被排他的預取且隨後的被寫入。

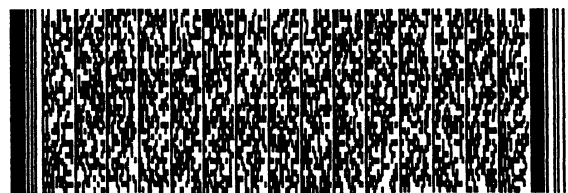
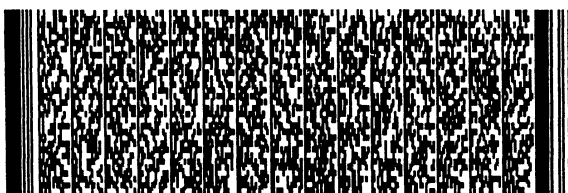
[0071] 因此，示範流程的一延伸預取指令702，
PREFETCH.S [EAX]，命令微處理器701 去排他的預取其位址對應於暫存器EAX 內容的快取線。第二個巨集指令702，
MOV EBX, [EAX]，命令微處理器701 去讀取一位址為暫存器EAX 所指定之記憶體位置的內容，並將該內容寫入暫存器EBX。第三個巨集指令，INC EBX，命令微處理器701 增加暫存器EBX 的內容。第四個巨集指令，MOV EAX, [EBX]，命令微處理器701 在對應於暫存器EAX 內容的記憶體位置上儲存暫存器EBX 的內容。值得注意的是為了要有效率的利用排他預取指令702，PREFETCH.S [EAX]，必須在第二個巨集指令702，MOV EBX, [EAX]，之前充分執行排他預取指令



五、發明說明 (27)

702，使得因為載入由EAX之內容所指定的快取線所造成的延遲，可以被中介的巨集指令702的平行執行所吸收。但是，為了簡明起見，該中介的巨集指令202的執行並未在方塊圖700中敘述。

[0072] 轉譯器710將該延伸預取巨集指令702轉譯成對應的排他預取微指令711，`PREFETCH.S [EAX]`，然將此微指令送到延伸快取記憶體單元720。該延伸記錄邏輯721詢問資料快取記憶體722以決定所要求的快取線是否已存在並且有效(換言之，即非處於無效狀態)於其資料快取記憶體722內。若答案為否，則該延伸記錄邏輯721命令匯流排單元730，經由系統記憶體匯流排740，發出作業以從記憶體742取得該所要求的快取線。若其他的匯流排元件741均無該所要求的快取線之副本，則延伸記錄邏輯721即將所要求的快取線以排他狀態送到其資料快取記憶體722。若有一匯流排元件741擁有該要求的處於排他狀態的快取線之區域副本時，則依照所應用之特定匯流排作業協定，此協定窺視在匯流排740上之作業以請求該快取線並將其區域副本變更成無效。若該區域副本已被修改，則其匯流排元件將修改的資料寫入到其匯流排740，使得微處理器701可以取得該快取線的排他所有權。若有數個匯流排共用此快取線，則這些匯流排元件均將其區域副本變更成無效，使得該快取線可以在排他狀態下被送到微處理器701。在上述任一情況，該所要求的快取線均可在在排他狀態下被送到快取記憶體722，並且可被隨後的儲存運算所使用。



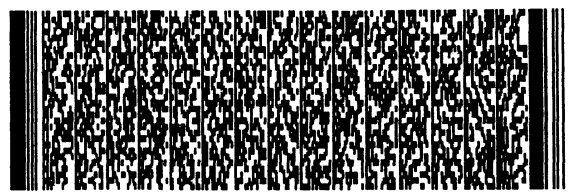
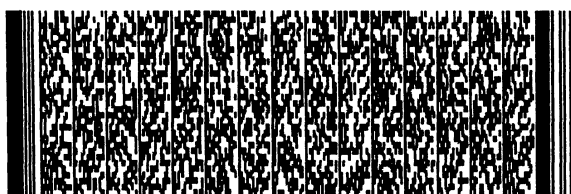
五、發明說明 (28)

[0073] 轉譯器710將第二個巨集指令702轉譯成一載入微指令711，LD EBX, [EAX]，此微指令係命令微處理器載入位址為暫存器EAX所指定的記憶體之內容到暫存器EBX。因為意圖進行儲存運算之預取的結果，該快取線所有之內容已經存在於該資料快取記憶體722中，所以該快取線立即可用，並且使得該載入微指令711的執行無任何延遲。

[0074] 第三個巨集指令702被轉譯成一對應的增加微指令271，INC EBX，此指令係命令執行邏輯將暫存器EBX之內容增加。因為不需要新的資料，所以該載入微指令711不會被送到其延伸快取記憶體單元720。

[0075] 最後，在流程中的第四個巨集指令702被轉譯成一儲存微指令711，ST [EAX], EBX，此指令命令執行邏輯去執行一資料儲存運算，以將暫存器EBX的內容寫入到位址由暫存器EAX內容所指定之記憶體位置。該儲存微指令711從而以待決的儲存運算型式被送到其快取記憶體單元720。於是，記錄邏輯721偵測到以待決的儲存運算為目標的快取線存在於其資料快取記憶體722，並且由於排他的預取之結果，該快取線是處於排他狀態。因此該儲存可不被延遲的立刻被公告。與圖二之微處理器201不同的是，依據本發明的延伸快取記憶體單元720不需確立一停滯訊號723以公告該待決的儲存，因為該目標快取線已被排他的預取。

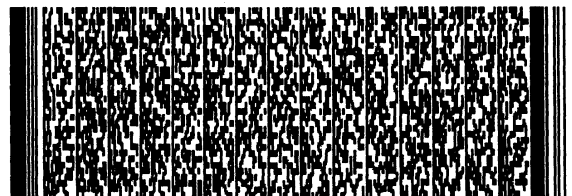
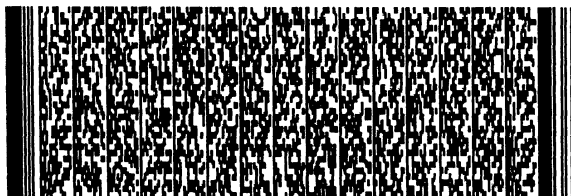
[0076] 現請參閱圖八，其顯示一時脈圖800，用以描述依據本發明由圖六及圖七所示之微處理器所發出之匯流



五、發明說明 (29)

排作業集801，802，此匯流排作業係經由記憶體匯流排740以執行意圖進行預取的儲存運算。此二作業集801，802包括在微處理器701內的從匯流排單元730到記憶體匯流排740的請求作業801，與同樣在在微處理器701內的從記憶體匯流排740回到匯流排單元730的回應作業802。時脈圖800係描述當一被要求且處於排他狀態的快取線，係為依據本發明的預取-排他巨集指令所指定的意圖進行儲存運算之預取的回應時，所執行的作業集801，802。根據上面的敘述，當執行一預取-排他指令時，延伸記錄邏輯721命令其匯流排單元730對其記憶體匯流排740發出一資料讀取與無效請求801，DATA READ/INV [EAX]，要求將被暫存器EAX所指定的快取線以排他的MESI狀態送到其區域快取記憶體722。該資料讀取與無效請求801是於時間點A發出。若所要求的快取線是處於排他的MESI狀態，則在時間點B，該記憶體匯流排740回應發出資料回應請求802，DATA RESP [EAX].E，回到匯流排單元230。在這時，一儲存運算的資料可從該快取記憶體222讀取或寫入到該快取記憶體222，而不會導致匯流排作業延遲。

[0077]同圖三所示，圖八中的作業集801，802係以一般性的方式表示，因為不同的微處理器架構應用不同的語義來說明匯流排作業801，802。圖八中描述的作業集801，802大體上依照x86的慣例，但此描述只是意圖進行說明本發明。此慣例並不會限定本發明只適用於此特定指令集架構。除此之外，值得注意的是，為了簡明起見，在



五、發明說明 (30)

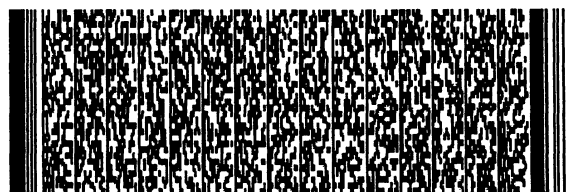
時脈圖800中已經省略了為先取得對資料匯流排740的存取(例如BUS REQUEST, BUS GRANT等等)之所有的作業。

[0078]本發明不只是考慮排他的預取一單一快取線，同時也包含需修改一個區段之資料的情況。因此，圖九至圖十一將具體指向排他的預取一個區段的資料的討論。

[0079]請參閱圖九，其為依據本發明用以顯示一延伸區段預取指令900之方塊圖。該延伸區段預取指令900包括一選用的多重前置碼實體901，其一為一重複前置碼901。該前置碼實體901之後為一預取運算碼902，其後則為一延伸位址指定元903。在一具體實施例中，每一個前置碼及延伸位址實體901，903都是8位元大小，而預取運算碼實體902則為一或二個位元組大小，除非本文中另行修訂，所有的實體901-903均與x86之指令集架構一致。在此具體實施例中，該x86的重複前置碼(REP)901被用來標示一區段預取運算。

[0080]在運作上，該預取運算碼902為一特定運算碼數值，此特定運算碼數值係用以命令一相符的微處理器執行一預取運算。在一x86的具體實施例中，運算碼實體901的特定數值為0F18H。延伸位址指定元903是用以指定執行該特定類型的預取運算的執行。在一x86的具體實施例中，該延伸位址指定元903是ModR/M位元組903。

[0081]同圖四所示，依據本發明，當微處理器偵測到一預取巨集指令900時，該微處理器被命令依據由延伸位址指定元403內容所指定的指示數值，去執行從記憶體預

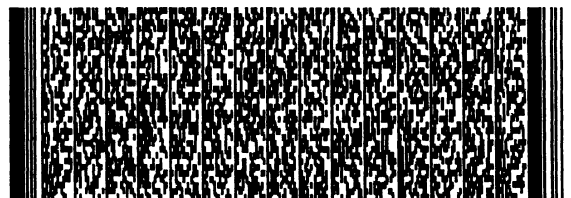
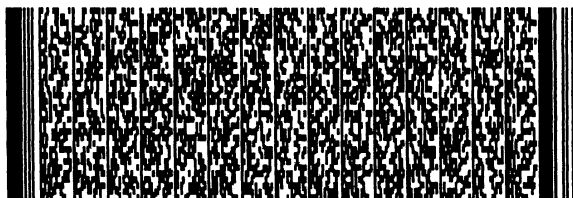


五、發明說明 (31)

取資料到快取記憶體。圖五之編碼範例也適用於描述在區段延伸位址指定元903的前置碼指示之編碼。但是若微處理器在該延伸預取指令內偵測到一重複前置碼901，則微處理器會試圖預取一指定數量在排他狀態下的快取線到其區域快取記憶體內，而此快取線的數量是由微處理器內之一架構暫存器指定。在一具體實施例中，快取線的數量是由x86相容之微處理器內之暫存器ECX所指定。

[0082]請參閱圖十，其係為一方塊圖1000，用以描述在圖六之微處理器600內執行區段預取與儲存運算之一界面至記憶體的快取記憶體單元。圖十之微處理器1001內之元件的認證與運算類似於圖七之微處理器701內之相似元件，只要將圖七的百位數圖號7用10來取代即可。為依據本發明以改進此區段預取運算，本發明使用一延伸轉譯器1010將有一重複前置碼1002的延伸預取運算轉譯成微指令序列1011，用以命令執行一排他區段預取運算。除此之外，亦使用一影子計數暫存器1013以載入自架構暫存器內之1012被預取之快取線的數量計數。並且使用延伸區段記錄邏輯(extended block fill logic)1021以命令其匯流排單元1030要求排他的預取該指定區段的快取線，並將之送到其資料快取記憶體1022。

[0083]為初始一排他的區段預取運算，第一個巨集指令1002，MOV ECX, COUNT，係用以將架構暫存器ECX內被排他的預取之區段內之快取線的數量計數初始化。延伸轉譯器1010將該第一個巨集指令轉譯成載入微指令1011，LD

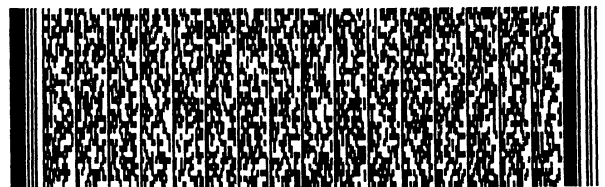


五、發明說明 (32)

ECX, COUNT, 此微指令命令微處理器將計數(count)載入 ECX。當該計數被載入到 ECX 後, 此計數亦被透明的複製到影子計數暫存器 1013, SHECX。同時, 其他的指令 1002 則可在不干擾到預取運算的計數之情況下, 修改其架構暫存器 1012。

[0084] 在計數被初始化之後, 該延伸轉譯器 1010 轉譯一延伸區段預取指令 1002, REP.PREF.S [EAX], 此指令命令微處理器 1001 預取被 ECX 所指定數量之排他狀態的快取線到其區域快取記憶體, 而該被預取的第一快取線的位址是由暫存器 EAX 所指定。以回應一命令該排他預取運算的微指令序列 1011, 該延伸區段記錄邏輯 1021 命令其匯流排單元 1030 經由其記憶體匯流排 1040 發出匯流排以要求該位址是由暫存器 EAX 所指定之處於排他狀態的快取線。該記錄邏輯 1021 在接收到這些快取線後, 即將之配置到其資料快取記憶體 1022。一但以排他狀態進入到該快取記憶體, 任一或是全部的被預取之快取線均可被修改而不會造成額外的延遲。

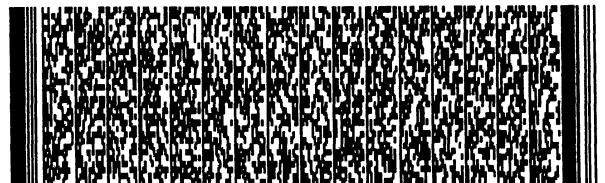
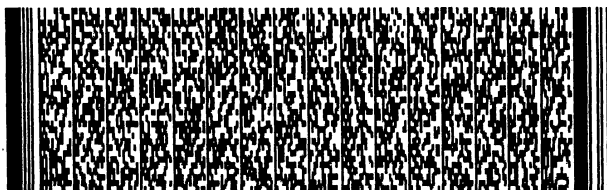
[0085] 現請參閱圖十一, 其顯示一時脈圖 1100, 用以描述依據本發明由圖六及圖十所示之微處理器 1001 所發出之匯流排作業集 1101, 1102, 此匯流排作業係經由記憶體匯流排 1040 以執行區段預取與儲存運算。為解說方便起見, 圖十一範例的系統組態使用 32 位元組快取線。但是, 熟習此領域技術者將從下述之例證中發現本發明的應用包括所有可預期之系統組態的快取線行寬。此二作業集



五、發明說明 (33)

1101，1102 包括在微處理器1001內的從匯流排單元1030到記憶體匯流排1040的請求作業1101，與同樣在在微處理器1001內的從記憶體匯流排1040回到匯流排單元1030的回應作業。時脈圖1000係描述當一被要求且處於排他狀態的一區段的快取線，係為依據本發明的包括一重複前置碼的預取-排他巨集指令所指定的意圖進行儲存運算之一區段預取的回應時，所執行的作業集1001，1002。根據上面的敘述，當執行一區段預取-排他指令時，該延伸記錄邏輯1021命令其匯流排單元1030發出一多重的資料讀取與無效請求1101，此請求並且對應於架構暫存器內所指定數量的快取線。該多重的請求包括該快取線區段內之所有快取線的位址，其位址係為架構暫存器EAX的內容所起始指定。雖然該匯流排請求1101使用遞增的位址次序，值得注意的是考慮到傳統上不同的記憶體匯流排協定，本發明亦包括遞減的排序，隨機的排序，及參差的排序。第一個資料讀取與無效請求1101是在時間點A發出，第二個請求1101是在時間點B發出，以此類推，直到最後的請求在時間點D發出。在許多種架構中，匯流排請求1102是被標記的，以致於該請求開始於時間點C早於其最後的請求的完成。在時間點C，至少有在該區段內之一快取線是可被一待決的儲存所使用。但是，為確保將延遲減到最小，最好是將該區段之快取線的儲存運算推延至時間點E，此時所有的回應1102均已到達且處於排他狀態。

[0086] 若所要求的快取線是處於排他的MESI狀態，則



五、發明說明 (34)

在時間點B，該記憶體匯流排740回應發出一資料回應請求802，DATA RESP [EAX].E，回到其匯流排單元230。在這時，一儲存運算的資料可從其快取記憶體222讀取或寫入到其快取記憶體222，而不會導致匯流排作業延遲。

[0087] 現請參閱圖十二，其為描述依據本發明執行一意圖進行儲存運算之預取方法的流程圖1200。

[0088] 流程開始於方塊1202，在此處，依據本發明，一連串的巨集指令被送到一指令佇列。流程接著進行到方塊1204。

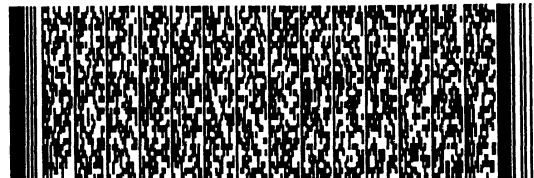
[0089] 於方塊1204中，一個隨後的巨集指令被從該指令佇列中提取，並將之送到一延伸轉譯器。流程接著進行到判斷方塊1206。

[0090] 於判斷方塊1206中，將進行一項檢查用以判斷該隨後的巨集指令是否為一延伸預取指令。若答案為是，則流程進行到方塊1208。若答案為否，則流程進行到方塊1210。

[0091] 於方塊1208中，一被偵測到的延伸預取指令被轉譯成一意圖進行儲存運算之預取的微指令序列，此微指令序列係用以命令微處理器去預取處於排他狀態之一指定的快取線。流程接著進行到方塊1212。

[0092] 於方塊1210中，該巨集指令被轉譯成一對應的微指令序列，此微指令序列係用以命令微處理器去執行一指定的運算。流程接著進行到方塊1212。

[0093] 於方塊1212中，一個隨後的微指令序列被送到



五、發明說明 (35)

微處理器內之執行邏輯。流程接著進行到判斷方塊1214。

[0094] 於判斷方塊1214中，將進行一項檢查用以判斷隨後的微指令序列是否為一意圖進行儲存運算之預取的微指令序列。若答案為是，則流程進行到方塊1216。若答案為否，則流程進行到方塊1218。

[0095] 於方塊1216中，回應意圖進行儲存運算之預取的微指令序列，匯流排作業請求被發出到一記憶體匯流排以要求一指定快取線之排他所有權。隨後該快取線以排他MESI狀態被送到微處理器，因此可為儲存運算所使用而不會發生因為提升該快取線至可被修改之狀態所導致的延遲。流程接著進行到方塊1220。

[0096] 於方塊1218中，執行該隨後的微指令序列。流程接著進行到方塊1220。

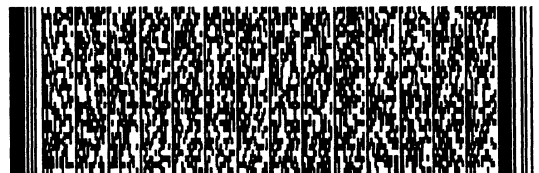
[0097] 於方塊1220中，本方法完成。

[0098] 現請參閱圖十三，其為描述依據本發明執行一意圖進行儲存運算之區段預取方法的流程圖1300。

[0099] 流程開始於方塊1302，在此處，依據本發明，一連串的巨集指令被送到一指令佇列。流程接著進行到方塊1304。

[00100] 於方塊1304中，一個隨後的巨集指令被從該指令佇列中提取，並將之送到一延伸轉譯器。流程接著進行到判斷方塊1306。

[00101] 於判斷方塊1306中，將進行一項檢查用以判斷該隨後的巨集指令是否為一延伸區段預取指令。若答案



五、發明說明 (36)

為是，則流程進行到方塊1310。若答案為否，則流程進行到方塊1308。

[00102]於方塊1310中，一被偵測到的延伸區段預取指令被轉譯成一意圖進行儲存運算之區段預取的微指令序列，此微指令序列係用以命令微處理器去預取處於排他狀態一指定數量之快取線。流程接著進行到方塊1312。

[00103]於方塊1308中，巨集指令被轉譯成一對應的微指令序列，此微指令序列係用以命令微處理器去執行一指定的運算。流程接著進行到方塊1312。

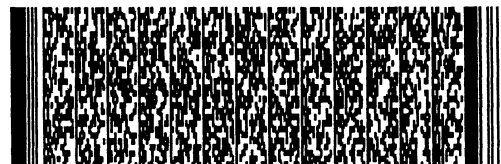
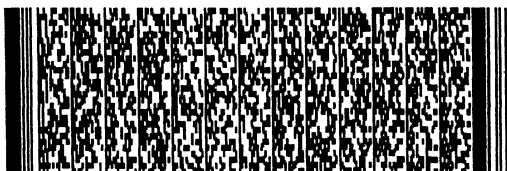
[00104]於方塊1312中，一個隨後的微指令序列被送到微處理器內之執行邏輯。流程接著進行到判斷方塊1314。

[00105]於判斷方塊1314中，將進行一項檢查用以判斷隨後的微指令序列是否為一意圖進行儲存運算之預取的微指令序列。若答案為是，則流程進行到方塊1318。若答案為否，則流程進行到方塊1316。

[00106]於方塊1316中，執行該隨後的微指令序列。流程接著進行到方塊1328。

[00107]於方塊1318中，回應意圖進行儲存運算之該區段預取的微指令序列，初始一暫時的計數器為數值0以監控將被發出之匯流排作業的數量，此匯流排作業係要求一區段的快取線之排他所有權。流程接著進行到方塊1320。

[00108]於方塊1320中，第一快取線位址被指定給第



五、發明說明 (37)

一個資料讀取與無效匯流排作業。該第一快取線位址來自於由該延伸區段預取指令所指定的位址，然後再加上一乘以方塊1318之計數的快取線行寬。由於該計數初始為零，所以該第一快取線位址等同於該延伸區段預取指令所指定的位址。流程接著進行到方塊1322。

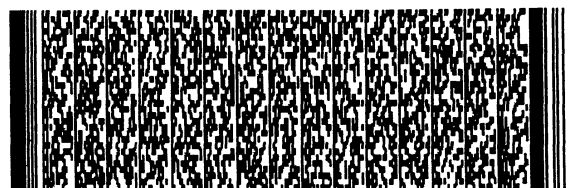
[00109]於方塊1322中，發出一經由記憶體匯流排之資料讀取與無效作業以預取該處於排他MESI狀態的該第一快取線。流程接著進行到方塊1324。

[00110]於方塊1324中，將該計數增加後，流程接著進行到方塊1326。

[00111]於判斷方塊1326中，將進行一項檢查用以判斷該增加的計數是否等同於被預取之快取線的數量，該快取線的數量係儲存於一影子暫存器(shadow register)中。若答案為否，則流程進行到方塊1320，於其中執行另一個重複(iteration)以提取一下一快取線。若計數等同於該影子暫存器的內容，則發出所有匯流排作業，同時流程進行到方塊1328。

[00112]於方塊1328中，本方法完成。

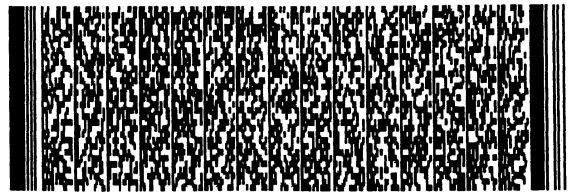
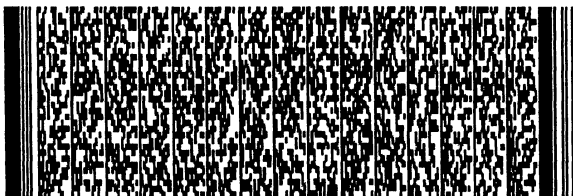
[00113]現請參閱圖十四，圖十四係為一表格500，其為解說依據本發明的另一可選用之延伸位址指定元實體的編碼方式，此編碼係用以在一區域資料快取記憶體內配置與初始化一快取線。請參閱圖五中所討論之一類似編碼方式的具體實施例，為了說明本發明起見，此處使用符合x86架構的ModR/M位元欄位，但是，可以預期的是本發明



五、發明說明 (38)

包含任何支援將一預取與初始化指示編碼到一延伸預取指令400的機制之架構。雖然圖十四的例子指向將該預取與初始化(亦稱為配置與初始化)指示編碼到一延伸位址指定元403,熟習此項技術者將發現該預取指示也可以被編碼成一在運算碼欄位401內之任何未使用指令之一特定運算碼數值。

[00114]圖十四中依據本發明的之另一可選用編碼方式的具體實施例擴展了微處理器600的能力,使其得以隱含的指定一排他預取的快取線,並且在其上執行儲存運算。在此處所討論的配置與初始化的例子對於改善應用程式的執行速度是非常有用的,因為用以初始化一快取線的儲存運算可被完全的從程式流程中排除掉。因此,圖十四顯示三種不同型式的預取與初始化指示,該三種指示使用x86的ModR/M之5:3位元的數值100-110。當任一上述三種預取與初始化指示依據本發明被編碼到一預取指令400時,一相符的微處理器將被命令發出經由一記憶體匯流排之作業,此作業係用以預取處於排他狀態之資料,並且在取得該資料後,即將之初始化成一指定數值。對數值100的編碼而言,微處理器被命令將該排他預取的資料初始化成一被架構暫存器所指定的數值,例如一x86處理器的暫存器EAX。對數值101的編碼而言,微處理器被命令去清除該預取的資料,也就是說,對該預取的資料執行儲存運算以將之設定為邏輯數值零。對數值110的編碼而言,微處理器被命令設定該預取的資料,也就是說,對該預取的資



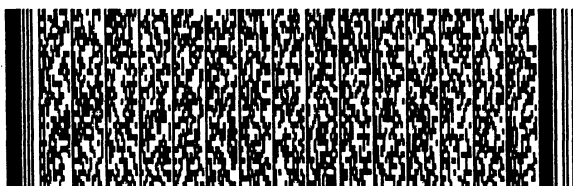
五、發明說明 (39)

料執行儲存運算以將之設定為邏輯數值一。如同圖五中討論的具體實施例所示，在一依據本發明之x86具體實施例中，這些編碼方式是合法的，並且將會導致該資料讀取與無效匯流排作業去預取一處於排他狀態之快取線。

[00115]請參閱圖十五，其係為一方塊圖1500，用以描述在圖六之微處理器內執行預取與初始化運算之一界面至記憶體之快取記憶體單元。在此一具體實施例中，一依據圖十四所編碼的預取與初始化指令的回應係為對一排他預取之快取線執行的儲存運算。方塊圖1500顯示在微處理器1500內被應用為執行預取與初始化運算的邏輯。微處理器1501的延伸轉譯器1510接收巨集指令流程1502，並將該巨集指令流程1502轉譯成對應的微指令1511。微指令1511在命令對記憶體1542做資料載入與儲存運算之後，即被送到一延伸快取記憶體單元1520。該延伸快取記憶體單元1520包括延伸記錄邏輯1521及一資料快取記憶體1522。該延伸記錄邏輯1521係耦接至一匯流排單元1530。該匯流排單元1530係界面至一系統記憶體匯流排1540，此系統記憶體匯流排1540又與資料記憶體1542及其他匯流排元件1541耦接。

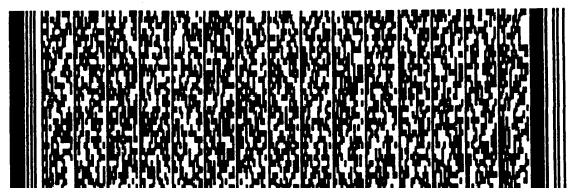
[00116]巨集指令之示範流程1502說明如何依照圖五之預取與初始化編碼100來指定一預取與初始化運算，及如何對預取的資料執行隱含的儲存運算以將之初始化。

[00117]因此，轉譯器1510將一MOV EAX, #INIT巨集指令1502轉譯成一LD EAX, , #INIT微指令，此微指令用以命



五、發明說明 (40)

令微處理器1501將數值INIT載入到架構暫存器EAX1505。該LD EAX, , #INIT被送到執行邏輯(未顯示)，此執行邏輯將INIT載入到EAX1505。緊接著，該轉譯器1510將一在此示範流程內之延伸預取與初始化指令1502，PREF.I [EBX]，轉譯成預取與初始化微指令1511，PREF.I [EBX]，此微指令係命令微處理器1501去排他的預取其位址對應於暫存器EBX之內容的快取線，並且將該快取線的數值設定成儲存在EAX1505內之數值。該預取與初始化微指令1511然後被送到其延伸快取記憶體單元1520。在此延伸快取記憶體單元1520中，延伸記錄邏輯1521詢問其資料快取記憶體1522以決定該要求的快取線是否已存在於該資料快取記憶體722內，並且處於排他狀態。若答案為是，則延伸記錄邏輯1521將該快取線初始化成EAX1505內之數值。若答案為否，則該延伸記錄邏輯1521命令匯流排單元1530發出經由系統記憶體匯流排1540之作業，以從記憶體1542取得該要求的處於MESI排他狀態的快取線。若其他的匯流排元件1541均無該要求的快取線之副本，則該延伸記錄邏輯1521即將該要求的快取線以排他狀態送到資料快取記憶體1522。若有一匯流排元件1541擁有該要求的處於排他狀態的快取線之區域副本時，則依照所應用之特定匯流排作業協定，此協定窺視在匯流排1540上之作業以請求該快取線並將其區域副本變更成無效。若該區域副本已被修改，則其匯流排元件1541將該修改的資料寫入到其匯流排1540，使得微處理器1501可以取得該快取線的排他所有



五、發明說明 (41)

權。若有數個匯流排元件1541共用該快取線，則這些匯流排元件均將其區域副本變更成無效，使得該快取線可以在排他狀態下被送到微處理器1501。在上述任一情況，該要求的快取線均可在在排他狀態下被送到快取記憶體1522，並且可立即被初始化成在EAX1505內之數值。

[00118]請參閱圖十六，其為描述依據本發明對一快取線執行配置與初始化運算方法的流程圖1600。

[00119]流程開始於方塊1602，在此處，依據本發明，一連串的巨集指令被送到一指令佇列。流程接著進行到方塊1604。

[00120]於方塊1604中，一個隨後的巨集指令被從該指令佇列中提取，並將之送到一延伸轉譯器。流程接著進行到判斷方塊1606。

[00121]於判斷方塊1606中，將進行一項檢查用以判斷該隨後的巨集指令是否為一延伸預取指令，且此延伸預取指令係編碼成對一快取線執行配置與初始化運算。若答案為是，則流程進行到方塊1608。若答案為否，則流程進行到方塊1610。

[00122]於方塊1608中，該偵測到的延伸預取指令被轉譯成一預取與初始化指令序列，此指令序列用以命令微處理器去預取一處於排他狀態的指定快取線。在接收到該指定快取線後，即將其內容修改為一指定數值。該指定數值係由該預取與初始化微指令序列內之編碼所指定。該編碼方式之一具體實施例包括清除該被配置快取線的內容



五、發明說明 (42)

(換言之，將其內容修改為邏輯數值零)。另一該編碼方式的具體實施例包括確定(asserting)該快取線的內容(換言之，將其內容修改為邏輯數值一)。又一該編碼方式的具體實施例包括將該快取線的內容設定為儲存於微處理器中一架構暫存器內的數值。流程接著進行到方塊1612。

[00123]於方塊1610中，該巨集指令被轉譯成一對應的微指令序列，此微指令序列用以命令微處理器去執行一指定的運算。流程接著進行到方塊1612。

[00124]於方塊1612中，一個隨後的微指令序列被送到微處理器內之執行邏輯。流程接著進行到判斷方塊1614。

[00125]於判斷方塊1614中，將進行一項檢查用以判斷隨後的微指令序列是否為一預取與初始化運算的序列。若答案為是，則流程進行到方塊1616。若答案為否，則流程進行到方塊1618。

[00126]於方塊1616中，回應該預取與初始化運算的序列，匯流排作業請求被發出到一記憶體匯流排以要求一指定快取線之排他所有權。隨後回應此作業，該快取線即以排他MESI狀態被送到微處理器。流程接著進行到方塊1620。

[00127]於方塊1620中，延伸記錄邏輯依據本發明將在區域快取記憶體之排他預取的快取線初始化成所指定的數值。流程接著進行到方塊1622。

[00128]於方塊1618中，執行該隨後的微指令序列。



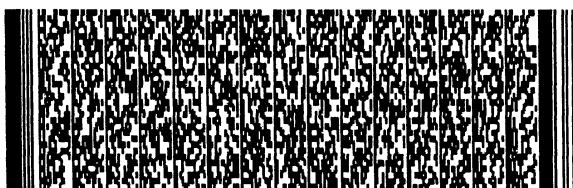
五、發明說明 (43)

流程接著進行到方塊1622。

[00129]於方塊1622中，本方法完成。

[00130]圖十七係為一方塊圖1700，用以描述在圖六之微處理器1701內執行區段預取與初始化運算之一界面至記憶體之快取記憶體單元。圖十七之微處理器1701內之元件的認證與運算類似於圖十五之微處理器1501內之相似元件，只要將圖十五的百位數圖號5用7來取代即可。依據本發明以改進此區段預取與初始化運算，本發明使用一延伸轉譯器1710將一如圖九所述之有重複前置碼1702的延伸配置與初始化指令轉譯成微指令序列1711，此微指令序列1711係用以命令執行區段預取與初始化運算。除此之外，亦使用一影子計數暫存器1713，用以載入從架構暫存器1712預取與初始化之快取線的數量計數。並且使用延伸區段記錄邏輯1721命令其匯流排單元1730要求排他的預取一指定區段的快取線，並將之送到其資料快取記憶體1722，此外並將這些快取線初始化成依照配置與初始化指令1702所指定的一指定數值。

[00131]為初始一區段配置與初始化運算，第一個巨集指令1002，MOV ECX, #COUNT，係用以將架構暫存器ECX內被排他的預取之區段內之快取線的數量計數初始化。延伸轉譯器1710將該第一個巨集指令轉譯成載入微指令1711，LD ECX, #COUNT，此微指令命令微處理器將計數載入ECX 1712。當該計數被載入到ECX 1712後，此計數亦被透明的複製到影子計數暫存器1713，SHECX。同時，其他



五、發明說明 (44)

的指令1702則可在不干擾到預取與初始化運算的計數之情況下，修改其架構暫存器1712的內容。

[00132]在計數被初始化之後，第二個巨集指令1702，MOV EAX, #INIT被送到轉譯器1710，此巨集指令1702係用以命令微處理器1701將一指定數值，INIT，載入到架構暫存器EAX1705。依據本發明之一具體實施例，其包含在EAX1705內容的指定數值將會使得在被預取之快取線區段中的所有快取線均被初始化。此一配置與初始化指令的特定編碼方式之具體實施例已於圖十四中討論過。其他的具體實施例有清除或設定該區段的快取線，回應這些具體實施例的配置與初始化指令之編碼的命令，其延伸快取記憶體單元1720內之延伸記錄邏輯1721則據以清除或設定該快取線區段。轉譯器1710將第二個巨集指令1702轉譯成一載入微指令，LD EAX, , #INIT微指令，此微指令用以命令微處理器1701將數值INIT載入到EAX1705。

[00133]在將數值INIT載入到EAX1705之後，緊接著，該延伸轉譯器1710轉譯一區段配置與初始化指令1702，REP.PREF.S [EBX]，此指令命令微處理器1701預取被ECX所指定數量之處於排他狀態的快取線到區域快取記憶體，而該第一被預取的快取線其位址是由暫存器EBX(未顯示)所指定，並且將該快取線的內容初始化成儲存在EAX1705的數值。以回應命令該排他配置與初始化運算的微指令序列1711，該延伸區段記錄邏輯1721命令其匯流排單元1730經由其記憶體匯流排1740要求排他的預取一指定區段的快



五、發明說明 (45)

取線，此區段第一快取線位址係由暫存器EBX所指定。記錄邏輯1721在接收到這些快取線後，即將之配置到資料快取記憶體1722，並將之初始化成由暫存器EAX 1705的內容所指定的數值。

[00134] 現請參閱圖十八，其為描述依據本發明執行一快取記憶體資料區段配置與初始化運算方法的流程圖1800。

[00135] 流程開始於方塊1802，在此處，依據本發明，一連串的巨集指令被送到一指令佇列。流程接著進行到方塊1804。

[00136] 於方塊1804中，一個隨後的巨集指令被從該指令佇列中提取，並將之送到一延伸轉譯器。流程接著進行到判斷方塊1806。

[00137] 於判斷方塊1806中，將進行一項檢查用以判斷該隨後的巨集指令是否為一區段配置與初始化指令。若答案為是，則流程進行到方塊1810。若答案為否，則流程進行到方塊1808。

[00138] 於方塊1810中，一被偵測到的區段配置與初始化指令被轉譯成一區段配置與初始化微指令序列，此微指令序列係用以命令微處理器去預取一指定數量之處於排他狀態的快取線，並且將該被預取快取線的內容修改成一指定數值。不同的區段配置與初始化運算的具體實施例可類同於如圖十六所述的單一快取線的配置與初始化具體實施例，只須稍作如圖九所述的重複前置碼的修改。流程接



五、發明說明 (46)

著進行到方塊1812。

[00139]於方塊1808中，該巨集指令被轉譯成一對應的微指令序列，此微指令序列用以命令微處理器去執行一指定的運算。流程接著進行到方塊1812。

[00140]於方塊1812中，一個隨後的微指令序列被送到微處理器內之執行邏輯。流程接著進行到判斷方塊1814。

[00141]於判斷方塊1814中，將進行一項檢查用以判斷該隨後的微指令序列是否為一意圖進行區段配置與初始化微指令序列。若答案為是，則流程進行到方塊1818。若答案為否，則流程進行到方塊1816。

[00142]於方塊1816中，執行該隨後的微指令序列。流程接著進行到方塊1830。

[00143]於方塊1818中，為回應該意圖進行儲存運算之區段預取的微指令序列，將一暫時的計數器初始化成數值零以監測所發出的匯流排作業請求的數量，該匯流排作業請求係要求一區段之快取線的排他所有權並將其初始化。流程接著進行到方塊1820。

[00144]於方塊1820中，第一快取線位址被指定給第一個資料讀取與無效匯流排作業。該第一快取線位址來自於由該區段配置與初始化指令所指定的位址，然後再加上一乘以方塊1818之計數的快取線行寬。由於該計數初始為零，所以該第一快取線位址等同於該區段配置與初始化指令所指定的位址。流程接著進行到方塊1822。



五、發明說明 (47)

[00145] 於方塊1822中，經由記憶體匯流排發出一資料讀取與無效作業，用以預取該之處於排他的MESI狀態的第一快取線。流程接著進行到方塊1824。

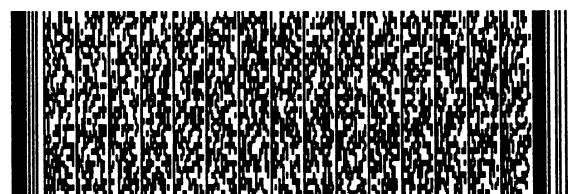
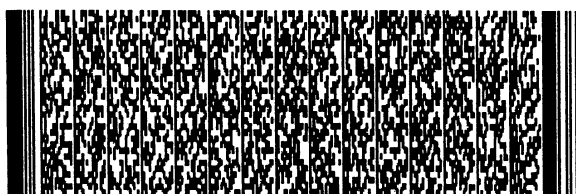
[00146] 於方塊1824中，將在區域快取記憶體中被預取之快取線初始化成該指定數值。流程接著進行到方塊1826。

[00147] 於方塊1826中，將該計數增加後，流程接著進行到方塊1828。

[00148] 於判斷方塊1828中，將進行一項檢查用以判斷該增加後的計數是否等同於被預取與初始化的快取線的數量，該快取線線的數量係儲存於影子暫存器中。若答案為否，則流程進行到方塊1820，於此處再進行另一個重複(iteration)以提取並初始化一下一快取線。若計數等同於該影子暫存器的內容，則發出所有匯流排作業，同時流程進行到方塊1830。

[00149] 於方塊1830中，本方法完成。

[00150] 雖然本發明及其目的、特徵與優點已詳細敘述，其他具體實施例仍涵蓋在本發明之範圍內。例如，在此描述了依照MESI協定時，本發明對於待決的儲存運算或消除的(eliminating)儲存運算所具有的優點。選擇MESI作為本發明的例子是因為其在本技術領域中的廣泛應用。但是，值得注意的是本發明提供預取的資料到一區域快取記憶體，其型式或狀態使得該資料可立即被修改，而不需發出作業到記憶體匯流排。該型式或狀態是否依照MESI



五、發明說明 (48)

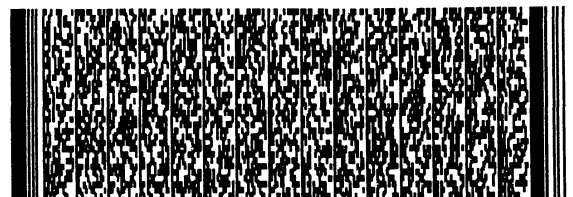
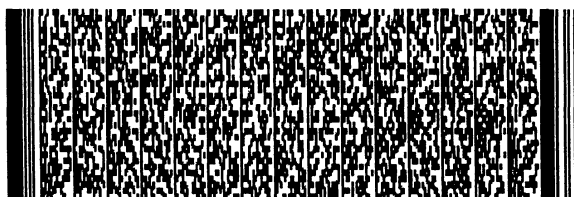
並不重要。

[00151] 如前所述，不同的架構使用不同的快取線行寬。在桌上型電腦系統中，今日普遍使用32-位元組的快取線。但是在本發明中使用的敘述並不限制本發明不能應用於32-，64-，128-，或甚至於256-位元組的快取線。相反的，本發明預期可應用在任何限制其區域快取線之修改且不提供直接預取這些快取線的系統架構，以至於該快取線可立即被修改而不必需依靠匯流排作業以取得修改許可。

[00152] 此外，本發明使用相符於x86架構的具體實施例作為例證。無疑的，x86-相容的微處理器可以從應用本發明得到好處，但值得注意的是本發明的應用範疇並不只限於x86-相容的環境，因為仍有許多不同的架構所應用的預取指令亦不能保證其結果為排他的資料。

[00153] 最後，值得注意的是，雖然在此處應用一位址指定元以指定被預取之快取線的位址，此種指定方式並不需是顯示的(explicit)。一個依據本發明之預取指令的具體實施例可以隱含的(implicitly)指定一包含該位址的架構暫存器，此處係由一先前執行的指令將該位址載入到該架構暫存器。

[00154] 總之，以上所述者，僅為本發明之較佳實施例而已，當不能以之限定本發明所實施之範圍。大凡依本發明申請專利範圍所作之均等變化與修飾，皆應仍屬於本發明專利涵蓋之範圍內，謹請 貴審查委員明鑑，並祈惠



五、發明說明 (49)

准，是所至禱。



圖式簡單說明

【發明圖示說明】

[0015] 本發明之前述與其他目的、特徵及優點，在配合下列說明及所附圖示後，將可獲得更好的理解：

[0016] 圖一係為一解說當今微處理器的有效管線化階段的方塊圖；

[0017] 圖二係為一方塊圖，其描述在如圖一所述之微處理器內執行一預取運算之一界面至記憶體的快取記憶體單元；

[0018] 圖三係為一時脈圖，其說明由如圖一及圖二所述之微處理器所發出，經由記憶體匯流排以執行之預取運算的兩種可能作業集；

[0019] 圖四係為本發明之延伸預取指令的方塊圖；

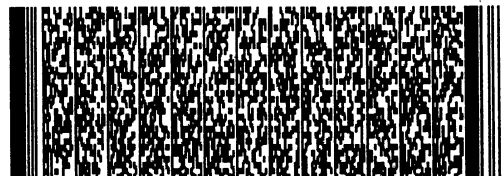
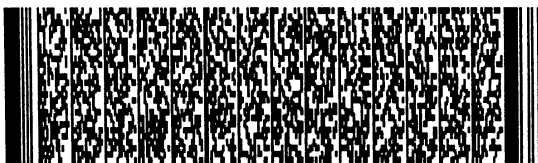
[0020] 圖五係為一表格，其說明如何將如圖四所示之延伸預取指令的延伸位址定位元欄位編碼，用以命令微處理器執行預取一處於排他MESI狀態的快取線；

[0021] 圖六係為一方塊圖，其詳述本發明執行意圖進行儲存運算之預取的一種微處理器；

[0022] 圖七係為一方塊圖，其描述在如圖六所述之微處理器內執行一意圖進行儲存運算之預取之一界面至記憶體的快取記憶體單元；

[0023] 圖八係為一時脈圖，為說明本發明由如圖六及圖七所述之微處理器所發出，經由記憶體匯流排以執行意圖進行儲存運算之預取的匯流排作業；

[0024] 圖九係為本發明之延伸區段預取指令的方塊



圖式簡單說明

圖；

[0025] 圖十係為一方塊圖，其描述在如圖六所述之微處理器內執行一區段預取及儲存運算之一界面至記憶體的快取記憶體單元；

[0026] 圖十一係為一時脈圖，為說明本發明由如圖六及圖十所述之微處理器所發出，經由記憶體匯流排以執行一區段預取及儲存運算之匯流排作業；

[0027] 圖十二係為解說本發明用以執行意圖進行儲存運算之預取方法的流程圖；

[0028] 圖十三係為解說本發明用以執行意圖進行儲存運算之區段預取方法的流程圖；

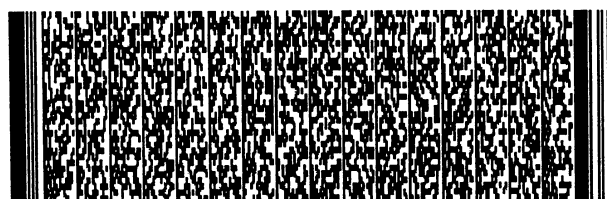
[0029] 圖十四係為一表格，其說明如何將如圖四所示之延伸預取指令的延伸位址定位元欄位編碼，用以命令微處理器執行一快取線的預取及初始化；

[0030] 圖十五係為一方塊圖，其描述在如圖六所述之微處理器內配置及初始化一快取線之一界面至記憶體的快取記憶體單元；

[0031] 圖十六係為解說本發明用配置及初始化一快取線之方法的流程圖；

[0032] 圖十七係為一方塊圖，其描述在如圖六所述之微處理器內配置及初始化一個區段的快取線之一界面至記憶體的快取記憶體單元；

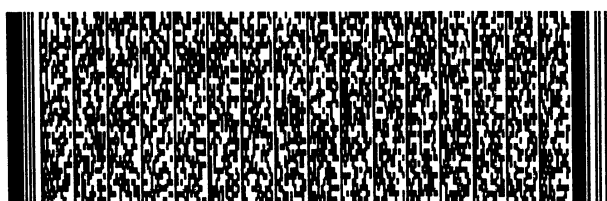
[0033] 圖十八係為解說本發明配置及初始化一快取記憶體資料區段之方法的流程圖。



圖式簡單說明

圖號說明：

100	管線化微處理器	101	提取程序
102	轉譯階段	103	暫存階段
104	定址階段	105	執行階段
106	執行邏輯	107	資料快取記憶體
108	記憶體	109	快取記憶體匯流排
110	記憶體匯流排	120	程式流程
120~123	巨集指令		
200	快取記憶體單元界面	201	微處理器
202	巨集指令	210	轉譯器
211	微指令	220	快取記憶體單元
221	記錄邏輯	222	資料快取記憶體
223	停滯訊號	230	匯流排單元
240	系統記憶體匯流排		
241	匯流排元件匯流排元件		
242	資料記憶體		
301~302	作業指令集		
303~304	匯流排作業		
400	延伸預取指令	401	前置碼
402	預取運算碼	403	延伸位址指定元
600	微處理器	601	提取邏輯
602	指令快取記憶體	603	指令記憶體
604	指令佇列	606	轉譯邏輯



圖式簡單說明

607	延伸轉譯邏輯	608	微指令佇列
609	執行邏輯	610	延伸快取記憶體單元
611	資料快取記憶體	612	延伸記錄邏輯
613	匯流排單元	614	資料記憶體
615	記憶體匯流排		
700	方塊圖	701	微處理器
702	巨集指令	710	延伸轉譯器
711	微指令	720	延伸快取記憶體單元
721	延伸記錄邏輯	722	資料快取記憶體
723	停滯訊號	730	匯流排單元
740	匯流排		
741	匯流排元件匯流排元件		
743	資料記憶體		
800	時脈圖	801~802	匯流排作業
900	延伸區段預取指令	901	前置碼
902	重複前置碼	903	預取運算碼
904	延伸位址指定元		
1000	方塊圖	1001	微處理器
1002	巨集指令	1010	延伸轉譯器
1011	微指令序列	1012	架構暫存器
1013	影子計數暫存器	1020	延伸快取記憶體單元
1021	延伸區段記錄邏輯	1022	資料快取記憶體
1023	停滯訊號	1030	匯流排單元
1040	記憶體匯流排		



圖式簡單說明

- 1041 匯流排元件匯流排元件
- 1042 資料記憶體
- 1100 時脈圖
- 1101~1102 匯流排作業
- 1200~1220 意圖進行儲存運算之預取的方法之流程
- 1300~1328 意圖進行儲存運算之區段預取的方法之流程
- 1500 方塊圖
- 1501 微處理器
- 1502 巨集指令
- 1510 延伸轉譯器
- 1505 架構暫存器
- 1511 微指令
- 1520 延伸快取記憶體單元
- 1521 延伸記錄邏輯
- 1522 資料快取記憶體
- 1523 停滯訊號
- 1530 匯流排單元
- 1540 系統記憶體匯流排
- 1541 匯流排元件匯流排元件
- 1542 資料記憶體
- 1600~1622 外取線配置與初始化的方法之流程
- 1700 方塊圖
- 1701 微處理器
- 1702 巨集指令
- 1710 延伸轉譯器
- 1705 架構暫存器
- 1711 微指令
- 1712 架構暫存器
- 1713 影子計數暫存器
- 1720 延伸快取記憶體單元
- 1721 延伸區段記錄邏輯
- 1722 資料快取記憶體
- 1723 停滯訊號
- 1730 匯流排單元
- 1740 記憶體匯流排
- 1741 匯流排元件匯流排元件
- 1742 資料記憶體
- 1800~1830 快取記憶體資料區段配置與初始化的方法之
流程



四、中文發明摘要 (發明名稱：快取記憶體資料區段的配置與初始化機制)

本發明提供一種對記憶體內一區段之快取線執行排他的預取與初始化的微處理器裝置。該裝置包含轉譯邏輯與執行邏輯。該轉譯邏輯將區段配置與初始化指令轉譯成一微指令序列，用以命令微處理器預取處於排他狀態的一區段之快取線，並且將該區段之該快取線初始化成一指定數值。該執行邏輯係耦接至該轉譯邏輯。該執行邏輯接收該微指令序列發出作業至要求處於該排他狀態區段之快取線的一記憶體匯流排上，並且將該區段之該快取線初始化成該指定數值。

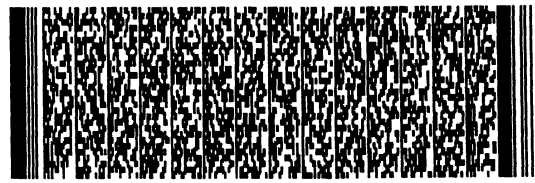
五、(一)、本案代表圖為：第 圖七 圖

(二)、本案代表圖之元件代表符號簡單說明：

700	方塊圖	701	微處理器
702	巨集指令	710	延伸轉譯器
711	微指令	720	延伸快取記憶體單元

六、英文發明摘要 (發明名稱：CACHE DATA BLOCK ALLOCATION AND INITIALIZATION MECHANISM)

A microprocessor apparatus is provided that enables exclusive prefetch and initialization of a block of cache lines from memory. The apparatus includes translation logic and execution logic. The translation logic translates a block allocate and initialize instruction into a micro instruction sequence that directs a microprocessor to prefetch a block of cache lines in an exclusive



四、中文發明摘要 (發明名稱：快取記憶體資料區段的配置與初始化機制)

721	延伸記錄邏輯	722	資料快取記憶體
723	停滯訊號	730	匯流排單元
740	匯流排	741	匯流排元件
742	資料記憶體		

六、英文發明摘要 (發明名稱：CACHE DATA BLOCK ALLOCATION AND INITIALIZATION MECHANISM)

state and to initialize the block of cache lines to a specified value. The execution logic is coupled to the translation logic. The execution logic receives the micro instruction sequence, and issues transactions over a memory bus that requests the block of cache lines in the exclusive state. Upon receipt, the execution logic initializes the block of cache lines to the specified value.



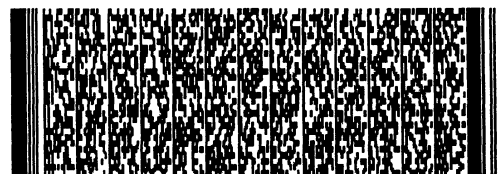
六、申請專利範圍

1. 一種微處理器裝置，用以執行來自記憶體之一區段資料的配置與初始化，該裝置包含：
一轉譯邏輯，組態成將一區段配置及初始化指令轉譯成一微指令序列，用以命令一微處理器預取處於排他狀態之一指定數量之快取線，並且將該指定數量之該快取線初始化成一指定值；以及
一執行邏輯，係耦接至該轉譯邏輯，組態成接收該微指令序列，發出作業至要求處於該排他狀態之該指定數量快取線之一記憶體匯流排上，並將該指定數量快取線初始化成該指定值。
2. 如申請專利範圍第1項所述之微處理器裝置，其中該指定數量之快取線的配置及初始化與在一應用程式內一程式流程之其他指令的執行平行。
3. 如申請專利範圍第2項所述之微處理器裝置，其中該指定數量之快取線的配置及初始化消除對應於執行隨後的儲存運算之程式延遲，要不然即會要求在該程式流程中將該指定數量之快取線初始化成該指定值。
4. 如申請專利範圍第1項所述之微處理器裝置，其中該區段配置與初始化指令包含對在一現存指令集內之一現存預取指令的修改，並且其中該現存預取指令原本不提供該指定數量之快取線之配置與初始化。
5. 如申請專利範圍第4項所述之微處理器裝置，其中該現存指令集包含x86指令集，並且其中該現存預取指令包含此x86的預取指令。



六、申請專利範圍

6. 如申請專利範圍第4項所述之微處理器裝置，其中該區段配置與初始化指令包含在一延伸位址指定元實體之一重複前置碼與一預取運算碼欄位，其中該預取運算碼欄位之一指定數值命令該微處理器將一第一快取線配置與初始化成該指定數值，並且其中該預取運算碼欄位之其他數值命令該微處理器依照該現存指令集執行其他類型之預取運算。
7. 如申請專利範圍第6項所述之微處理器裝置，其中該預取運算碼欄位包含在一x86預取指令之ModR/M位元組內的5：3位元。
8. 如申請專利範圍第7項所述之微處理器裝置，其中該重複前置碼欄位命令該微處理器配置與初始化該指定數量之快取線，並且其中該指定數量之快取線包含該第一快取線。
9. 如申請專利範圍第8項所述之微處理器裝置，其中該指定數量係由該微處理器內之一架構暫存器的內容所指定。
10. 如申請專利範圍第1項所述之微處理器裝置，其中，回應於該微指令序列，該執行邏輯命令一匯流排單元經由該記憶體匯流排發出該些作業。
11. 如申請專利範圍第10項所述之微處理器裝置，其中該些作業包含一多重的資料讀取與無效作業，此些作業係要求該指定數量之快取線之排他所有權。
12. 如申請專利範圍第11項所述之微處理器裝置，其中該



六、申請專利範圍

第一架構暫存器的內容包含該指定數量，該指定數量係透明的複製到一影子暫存器，並且此處該執行邏輯應用該影子暫存器以計數該多重的資料讀取與無效作業。

13. 如申請專利範圍第10項所述之微處理器裝置，其中該記憶體匯流排與x86的架構相符。
14. 如申請專利範圍第1項所述之微處理器裝置，其中該指定數值係隱含的由該配置及初始化指令之編碼方式所指定。
15. 如申請專利範圍第14項所述之微處理器裝置，其中該指定數值包含邏輯數值零。
16. 如申請專利範圍第14項所述之微處理器裝置，其中該指定數值包含邏輯數值一。
17. 如申請專利範圍第1項所述之微處理器裝置，其中該指定數值係由該微處理器內之一第二架構暫存器的內容所提供。
18. 如申請專利範圍第1項所述之微處理器裝置，其中，緊接在接收到一處於該排他狀態的一個第一快取線之後，該執行邏輯命令延伸記錄邏輯將該第一快取線初始化成該指定數值。
19. 一種微處理器裝置，係用以執行一區段配置與初始化運算，裝置包含：

一區段配置與初始化指令，係組態成命令微處理器預取處於排他狀態之一指定數量之快取線，並且將



六、申請專利範圍

該指定數量之快取線初始化成一指定數值；以及

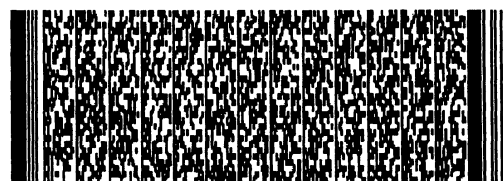
一轉譯器，係組態成接收該區段配置與初始化指令，並將該區段配置與初始化指令轉譯成相關的微指令，在其中該相關的微指令命令在微處理器內的執行邏輯經由一記憶體匯流排發出匯流排作業，用以要求該指定數量之快取線的排他所有權，並且將該指定數量之快取線初始化成一指定值。

20. 如申請專利範圍第19項所述之裝置，其中該指定數量之快取線的配置及初始化係與在一應用程式內程式流程之其他指令的執行平行。
21. 如申請專利範圍第19項所述之裝置，其中該區段配置與初始化指令包含對在一現存指令集內之一現存預取指令的修改，並且其中該現存預取指令原本不提供該指定數量之快取線之配置與初始化。
22. 如申請專利範圍第21項所述之裝置，其中該現存指令集包含x86指令集，並且其中該現存預取指令包含x86之含有一重複前置碼的預取指令。
23. 如申請專利範圍第21項所述之裝置，其中該區段配置與初始化指令包含在一延伸位址指定元實體之一重複欄位與一預取運算碼欄位，並且其中該預取運算碼欄位之一指定數值命令微處理器預取一處於該排他狀態的第一快取線，並將該第一快取線初始化成該指定數值，並且其中該預取運算碼欄位之其他數值命令微處理器依照該現存指令集執行其他類型之預取運算。



六、申請專利範圍

24. 如申請專利範圍第23項所述之微處理器機制，其中該重複前置碼欄位命令該微處理器預取該指定數量之快取線，並且其中該指定數量之快取線包含該第一快取線。
25. 如申請專利範圍第24項所述之微處理器機制，其中該指定數量係由微處理器內之一第一架構暫存器的內容所指定。
26. 如申請專利範圍第23項所述之裝置，其中該預取運算碼欄位包含在一x86預取指令之ModR/M位元組內的5：3位元。
27. 如申請專利範圍第19項所述之裝置，其中，回應於該相關的微指令，該執行邏輯命令一匯流排單元經由該記憶體匯流排發出該些匯流排作業。
28. 如申請專利範圍第27項所述之裝置，其中該些匯流排作業包含一多重的資料讀取與無效作業，此些作業係要求該指定數量之快取線之排他所有權。
29. 如申請專利範圍第28項所述之微處理器裝置，其中該架構暫存器的內容包含該指定數量，該指定數量係透明的複製到一影子暫存器，並且此處該執行邏輯應用該影子暫存器以計數該多重的資料讀取與無效作業。
30. 如申請專利範圍第19項所述之裝置，其中該記憶體匯流排與x86的架構相符。
31. 如申請專利範圍第19項所述之裝置，其中該指定數值係隱含的由該區段配置及初始化指令之編碼方式所指



六、申請專利範圍

定。

32. 如申請專利範圍第31項所述之裝置，其中該指定數值包含邏輯數值零。

33. 如申請專利範圍第31項所述之裝置，其中該指定數值包含邏輯數值一。

34. 如申請專利範圍第19項所述之裝置，其中該指定數值係由該微處理器內之一第二架構暫存器的內容所提供。

35. 一種配置與初始化一區段之快取線的方法，包含：

提取一區段配置與初始化之巨集指令；

將此區段配置與初始化巨集指令轉譯成微指令序列，其中該微指令序列係用以命令一微處理器預取處於排他狀態的該區段之快取線，並且將該區段之快取線初始化成一指定數值；以及

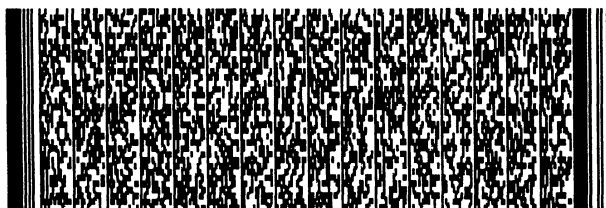
回應此微指令序列，經由記憶體匯流排發出匯流排作業以讀取該處於排他狀態之快取線，並且將該區段之快取線初始化成該指定數值。

36. 如申請專利範圍第35項所述之方法，其中該發出動作包含：

許可該微處理器平行的執行該發出與隨後的指令。

37. 如申請專利範圍第35項所述之方法，其中該提取包含：

提供該區段配置與初始化指令作為對在一現存指



六、申請專利範圍

令集內之一現存預取指令的修改，其中該現存預取指令原本不提供該區段快取線之配置與初始化。

38. 如申請專利範圍第37項所述之方法，其中該提供動作包含：

修改一x86預取指令以致能一配置與初始化運算的規格。

39. 如申請專利範圍第38項所述之方法，其中該延伸預取指令包含在一延伸位址指定元實體內之一重複前置碼與一預取運算碼欄位，其中該預取運算碼欄位之一指定數值命令該微處理器預取處於該排他狀態的該區段之快取線，並將該區段之快取線初始化成該指定數值，並且其中該預取運算碼欄位之其他數值命令該微處理器依照現存指令集執行其他類型之預取運算。
40. 如申請專利範圍第38項所述之方法，其中該預取運算碼欄位包含在一x86預取指令之ModR/M位元組內的5：3位元。
41. 如申請專利範圍第39項所述之方法，其中該重複前置碼命令該微處理器預取一指定數量之快取線，並且其中該指定數量之快取線等同於該區段之快取線內的快取線數目，並且其中該第一快取線係為該區段之快取線內的該指定數量快取線之一。
42. 如申請專利範圍第35項所述之方法，更包含：
- 依照一第一暫存器的內容來設定在該區段之快取線內的快取線數量。



六、申請專利範圍

43. 如申請專利範圍第42項所述之方法，更包含：

透明的複製該第一暫存器的內容到一影子暫存器。

44. 如申請專利範圍第35項所述之方法，其中該發出動作包含：

提供經由記憶體匯流排之一多重的資料讀取與無效作業，此作業係要求該區段之快取線的排他所有權。

45. 如申請專利範圍第35項所述之方法，其中該提取動作包含：

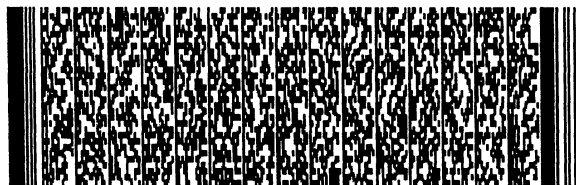
在該區段配置與初始化指令內隱含的指定一指定數值，該指定數值係用以初始化該區段之快取線。

46. 如申請專利範圍第45項所述之方法，其中該指定數值包含邏輯數值零。

47. 如申請專利範圍第45項所述之方法，其中該指定數值包含邏輯數值一。

48. 如申請專利範圍第35項所述之方法，其中該提取動作包含：

提供該微處理器之一第二暫存器內之該指定數值。



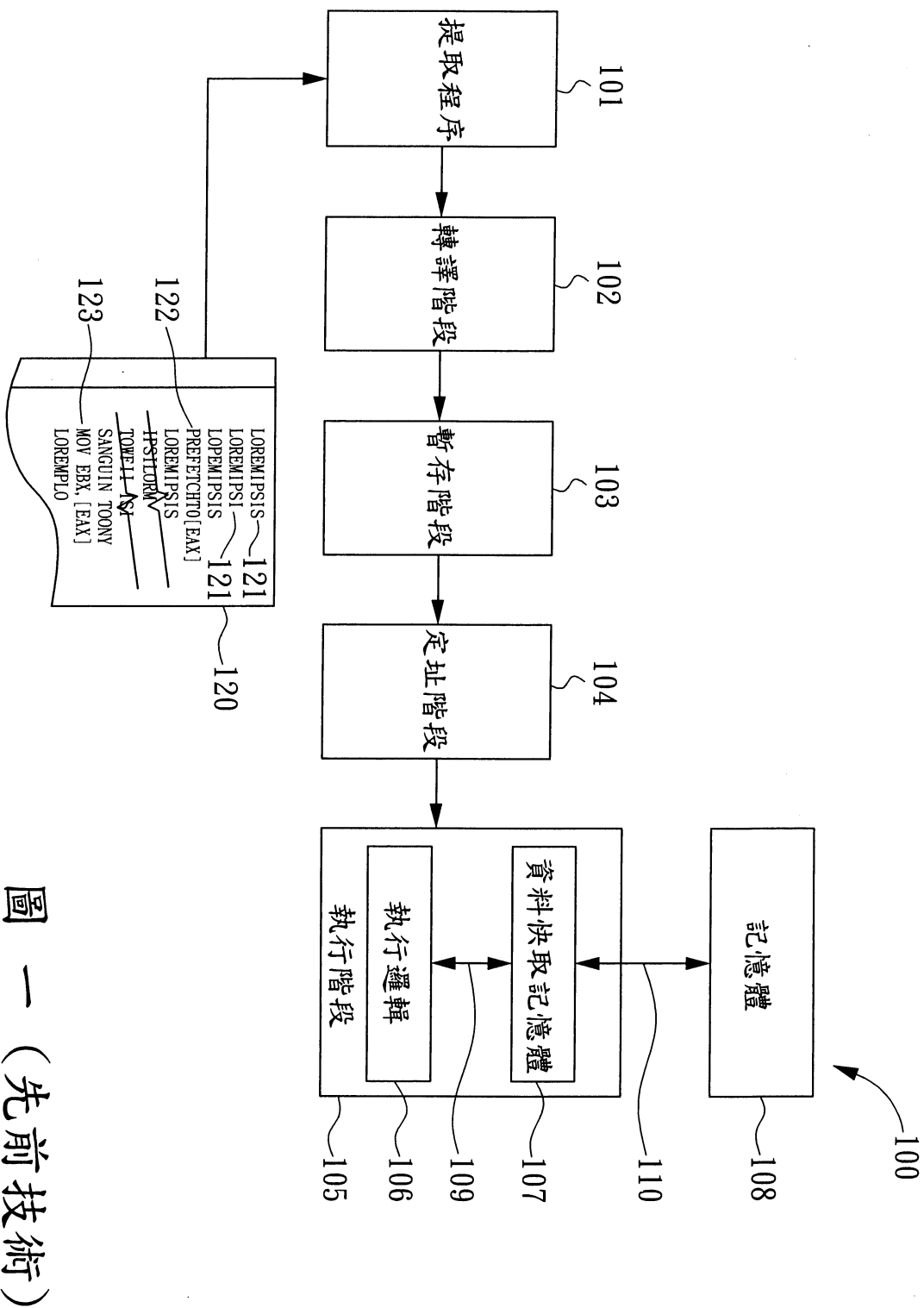


圖 一 (先前技術)

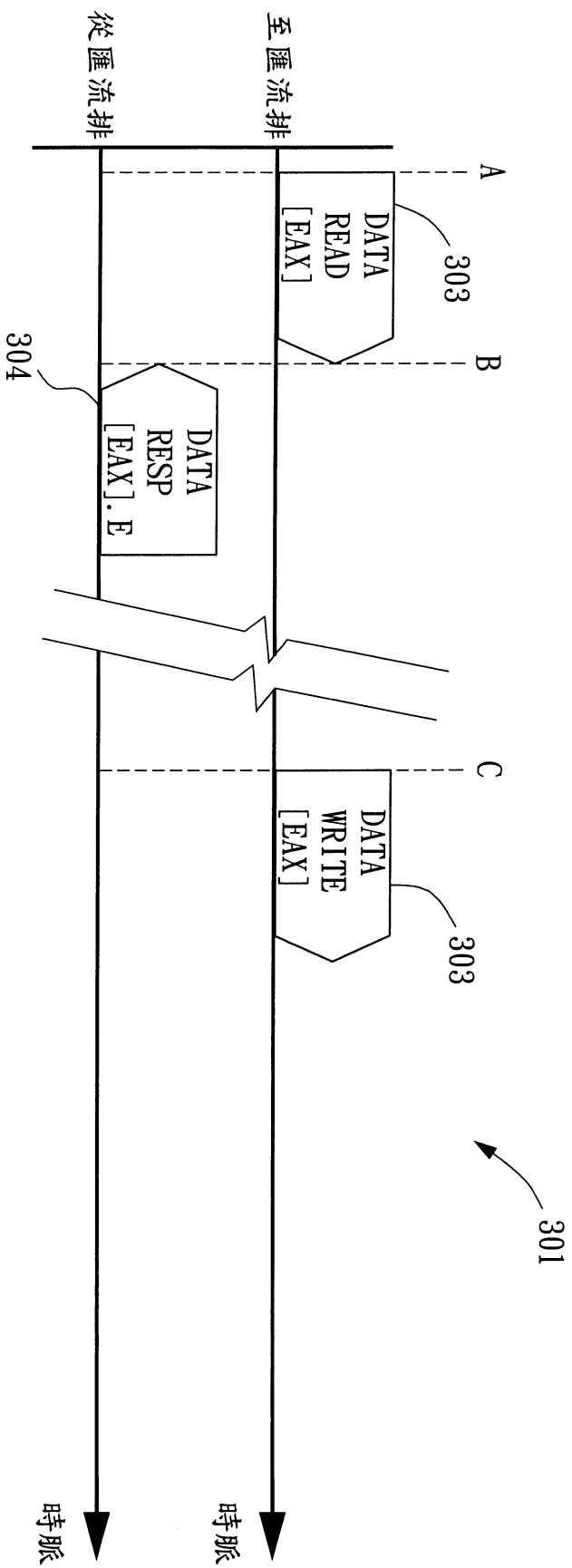


圖 三 A
(先前技術)

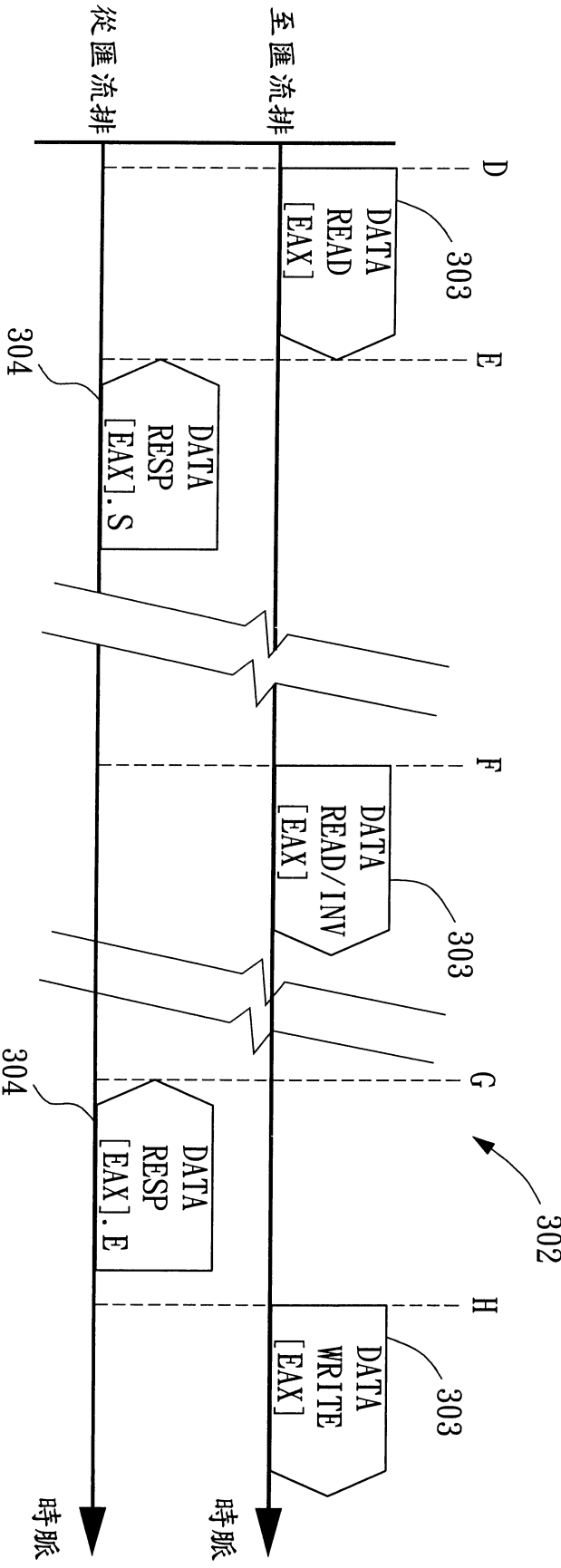


圖 三 B
(先前技術)

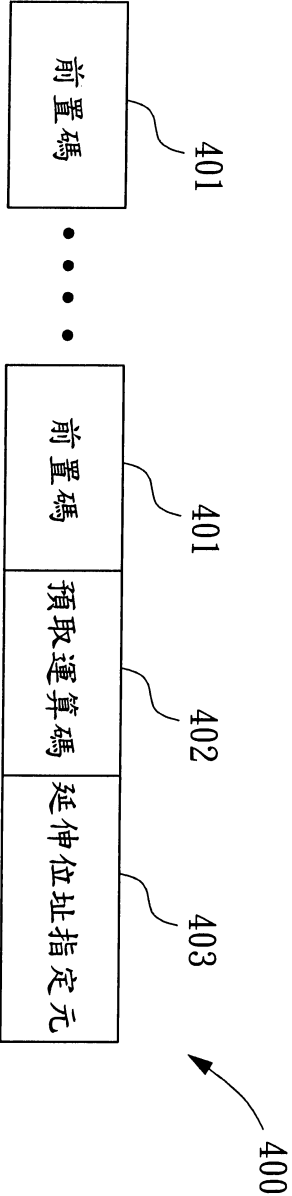


圖 四

延伸位址指定元之編碼

5:3 位元	運算	敘述
000	PREFETCHNTA	使用NTA指示預取特定資料
001	PREFETCHT0	使用T0指示預取特定資料
010	PREFETCHT1	使用T1指示預取特定資料
011	PREFETCHT2	使用T2指示預取特定資料
100	PREFETCH.S	使用.S指示預取特定資料
101	ILLEGAL	不合法運算碼
110	ILLEGAL	不合法運算碼
111	ILLEGAL	不合法運算碼

圖 五

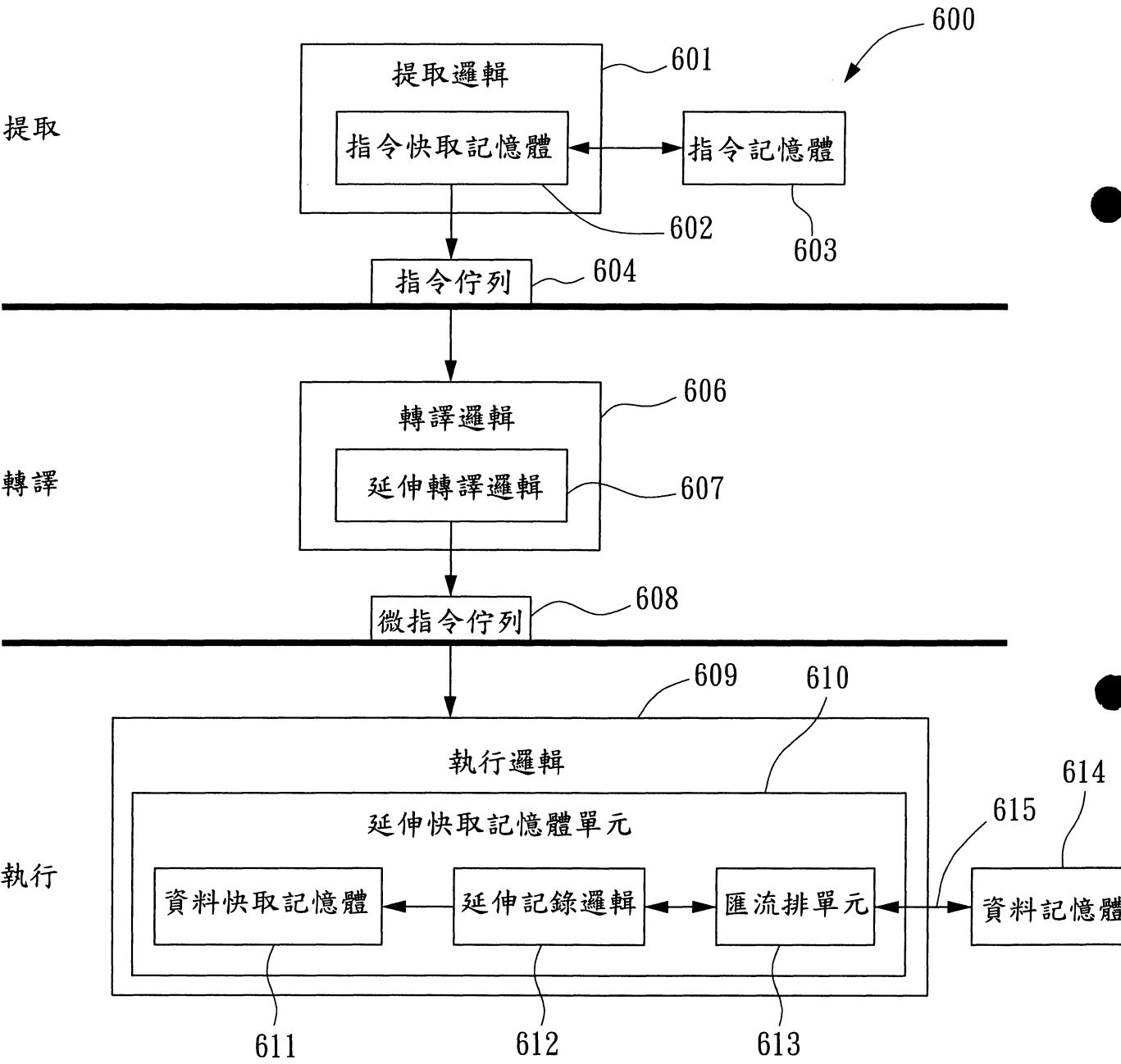


圖 六

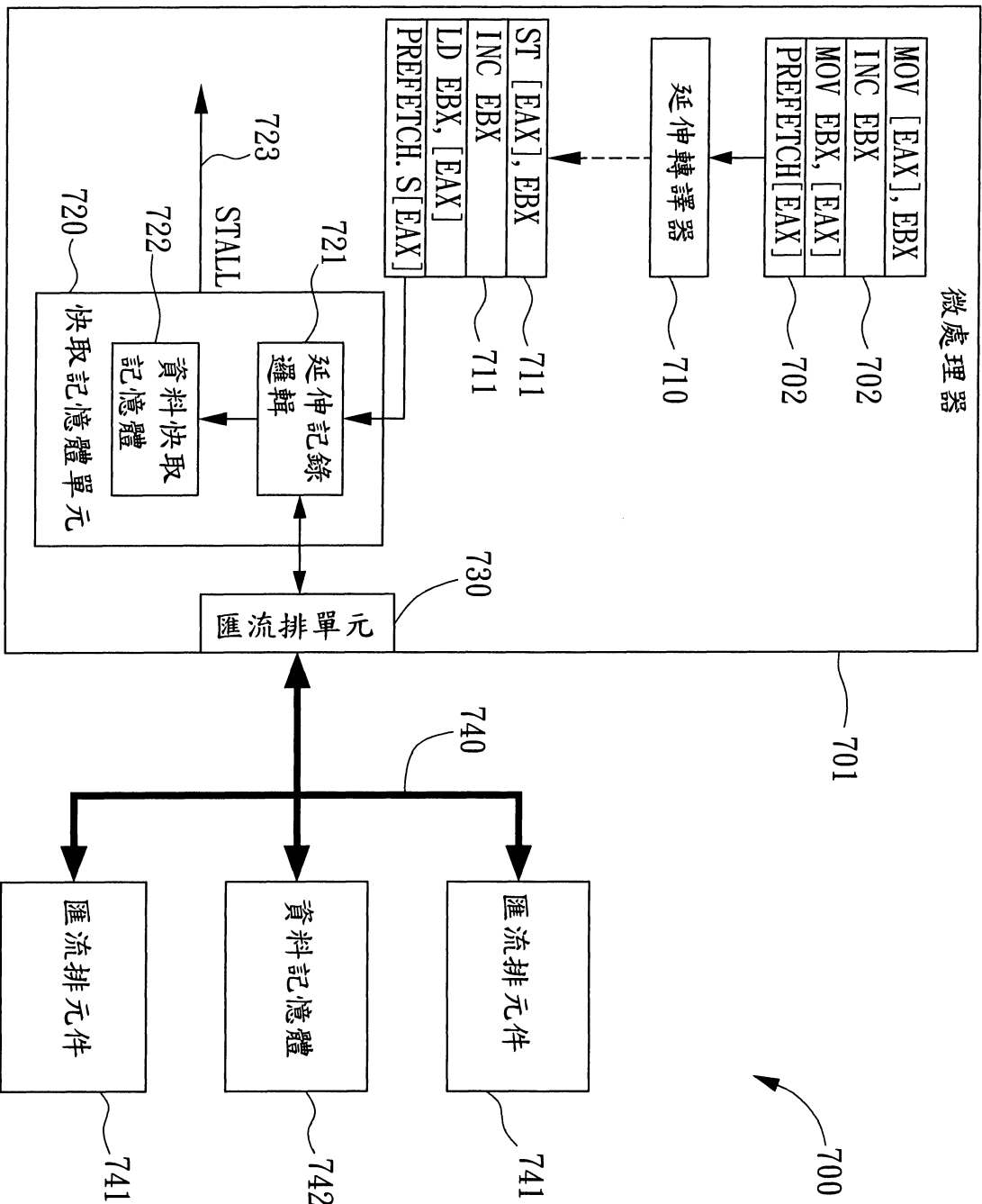


圖 七

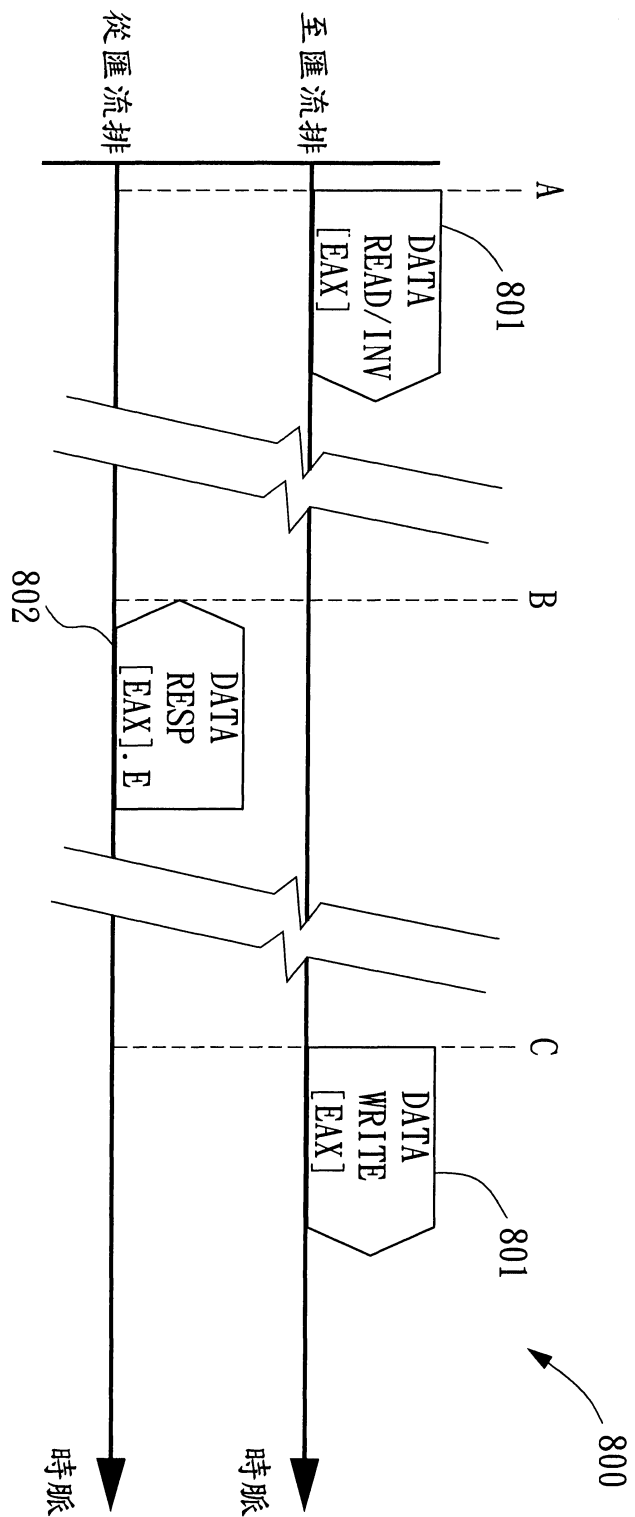


圖 八

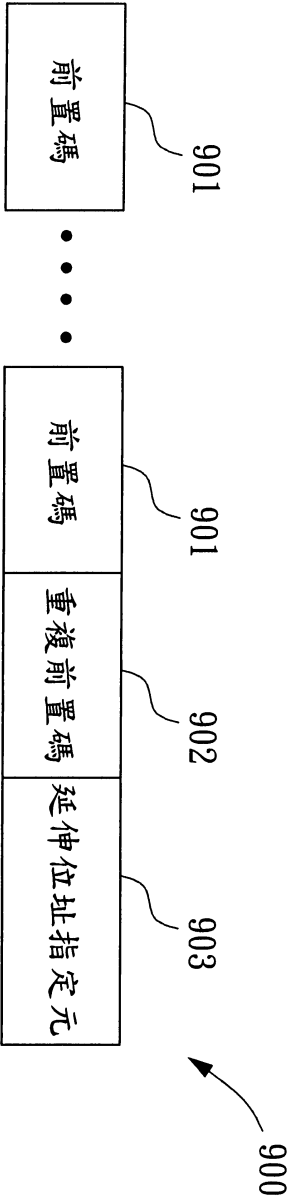


圖 九

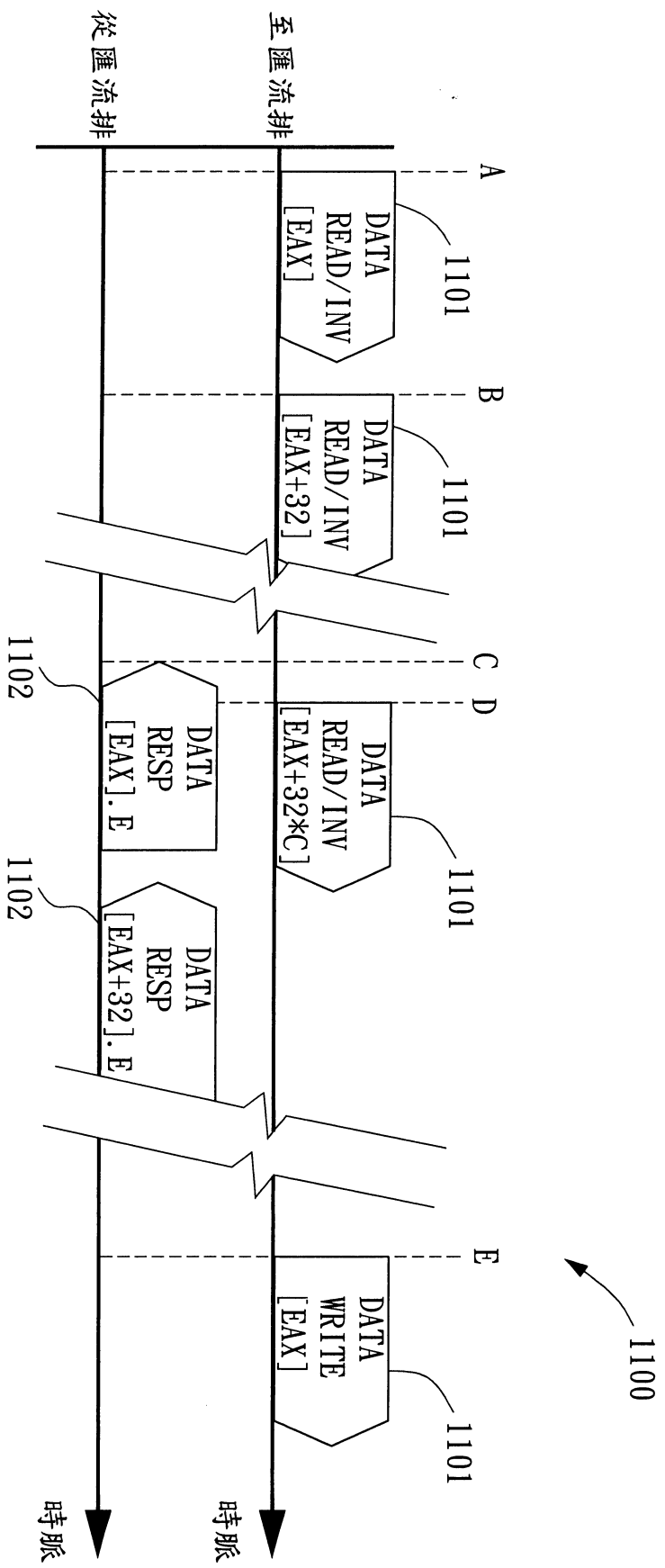


圖 十一

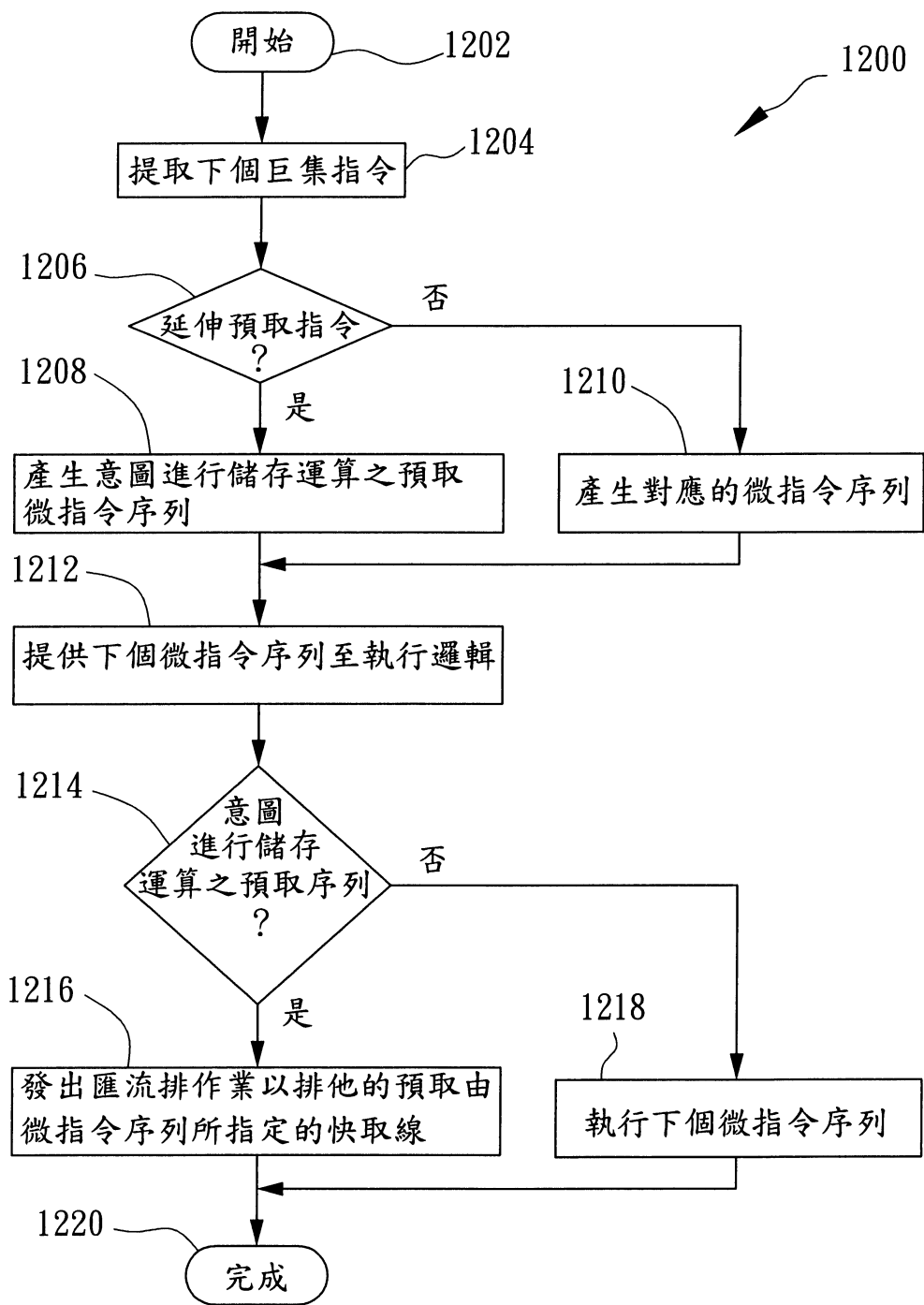


圖 十二

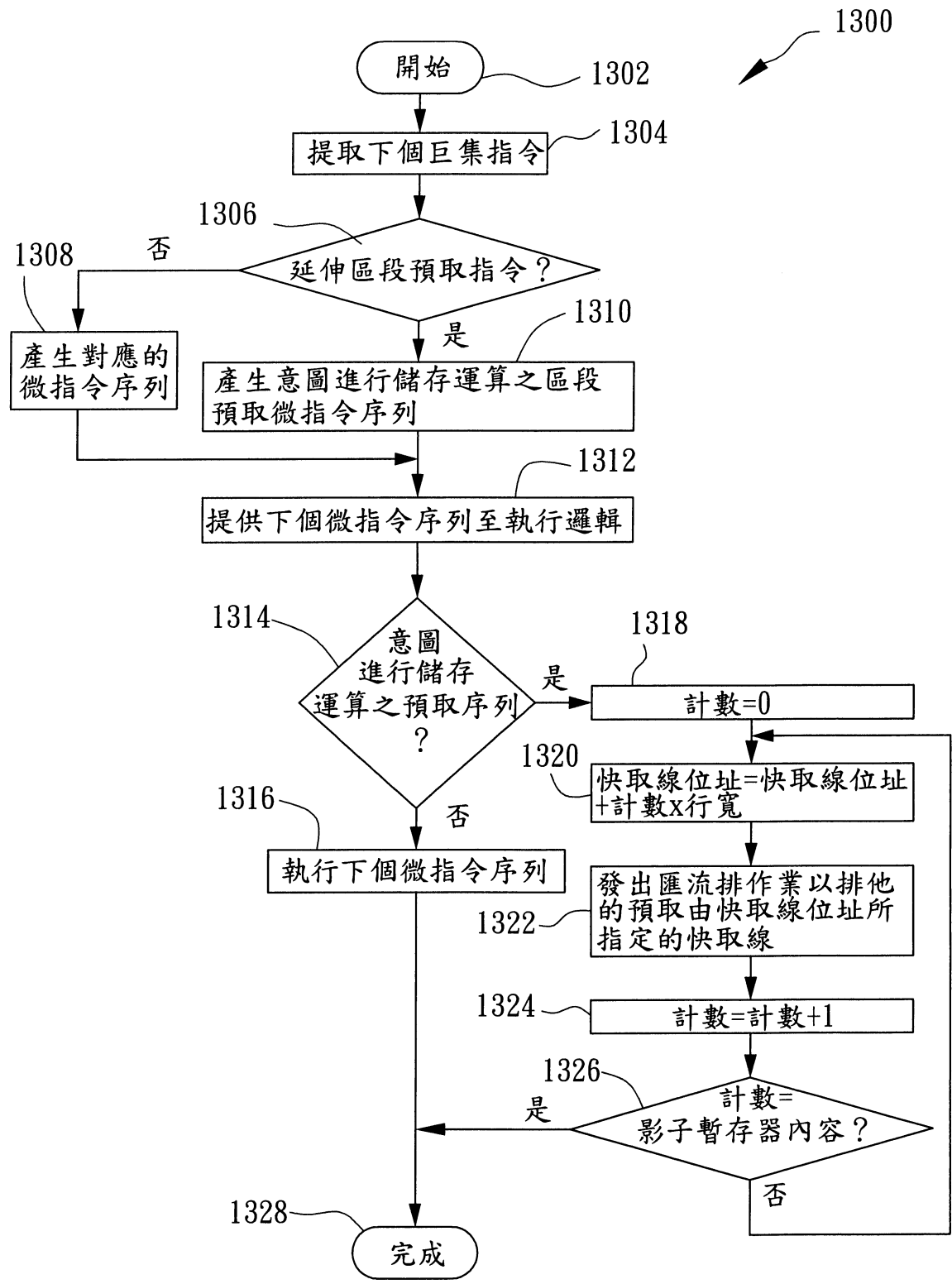


圖 十三

5:3 位元	運算	敘述
000	PREFETCHNTA	使用NTA指示預取特定資料
001	PREFETCHT0	使用T0指示預取特定資料
010	PREFETCHT1	使用T1指示預取特定資料
011	PREFETCHT2	使用T2指示預取特定資料
100	PREFETCH.I	使用.I指示預取與初始特定資料
101	ILLEGAL	使用.I指示預取與清除特定資料
110	ILLEGAL	使用.I指示預取與設定特定資料
111	ILLEGAL	不合法運算碼

圖 十四

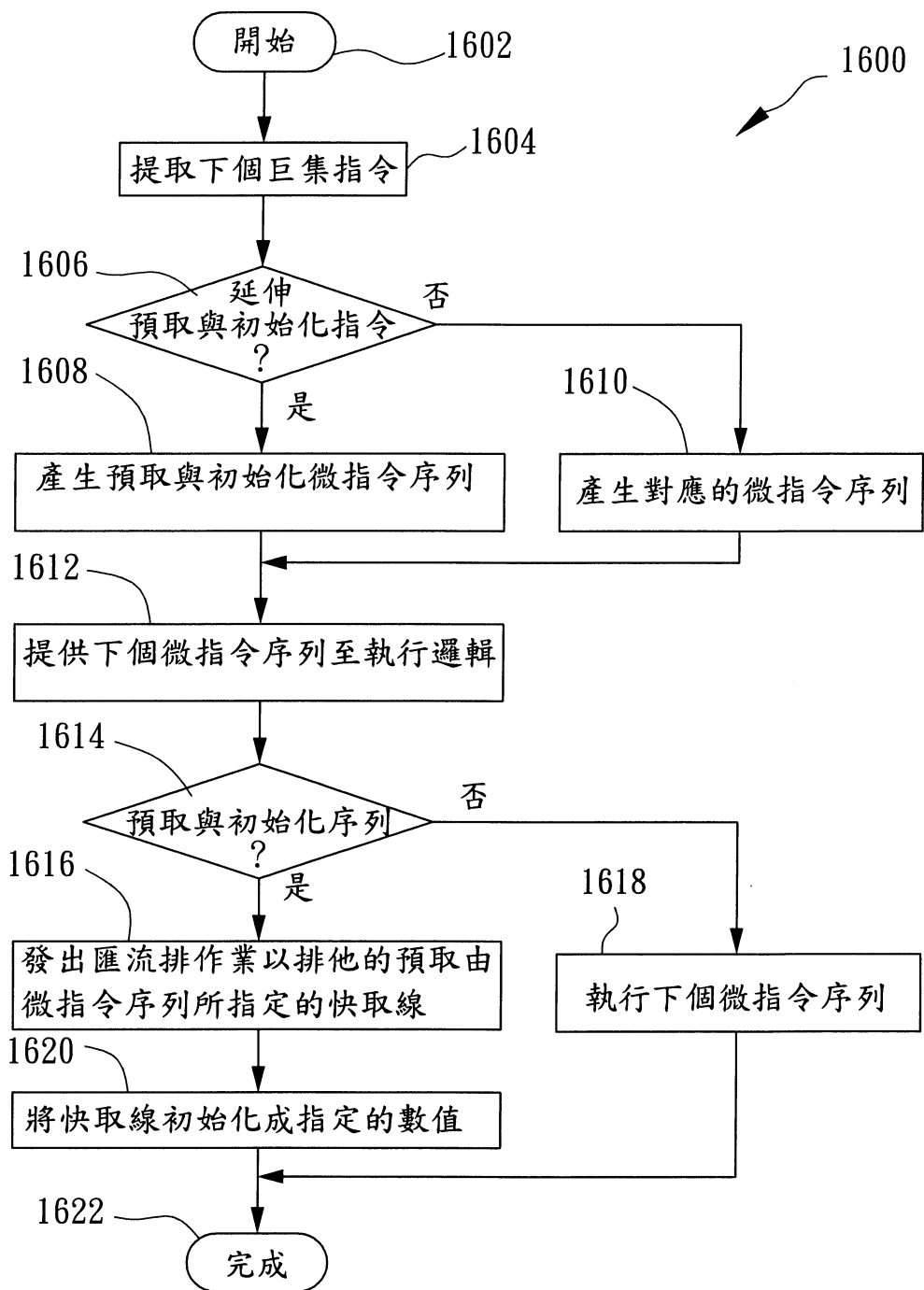
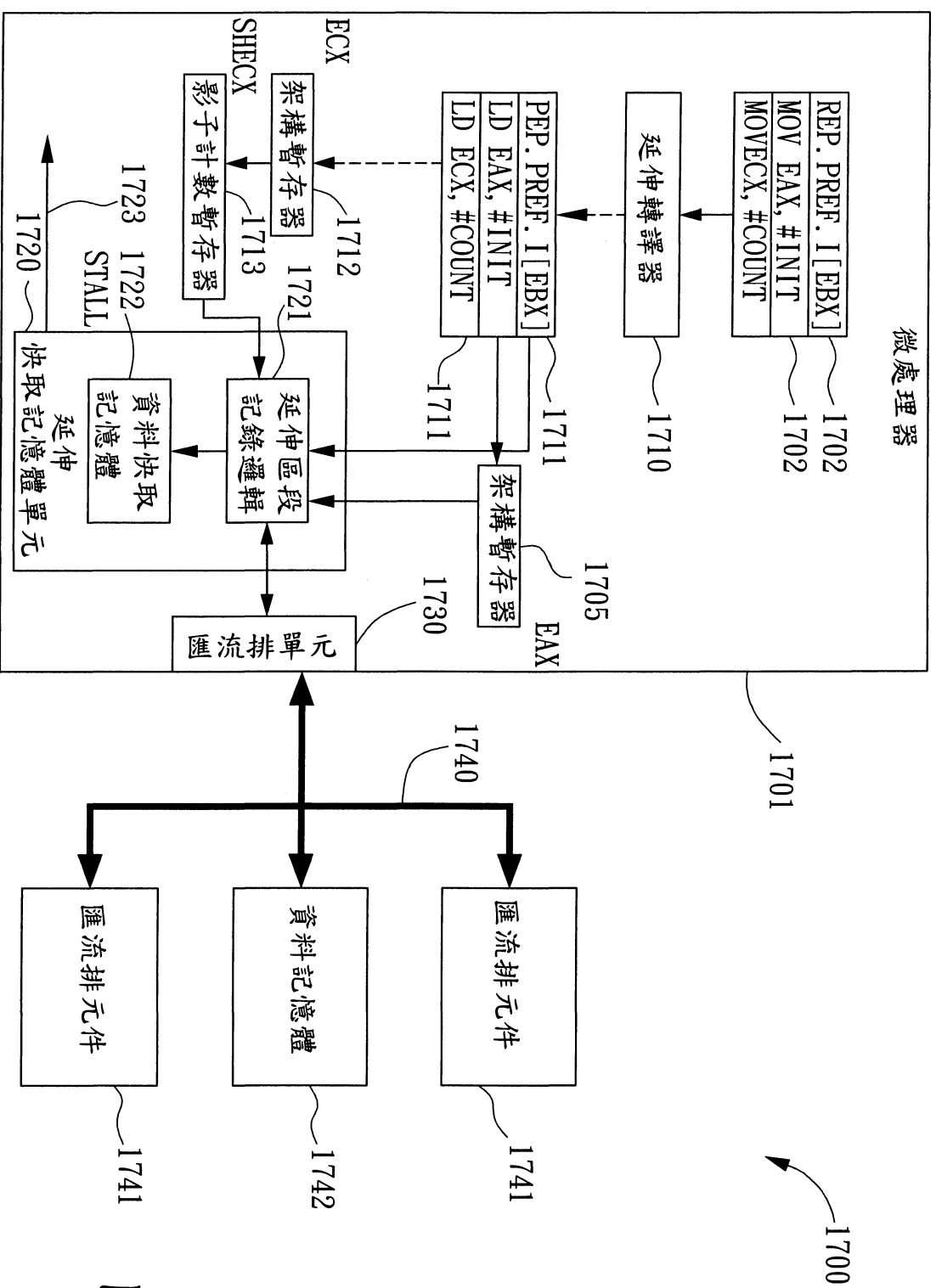


圖 十六



四十七

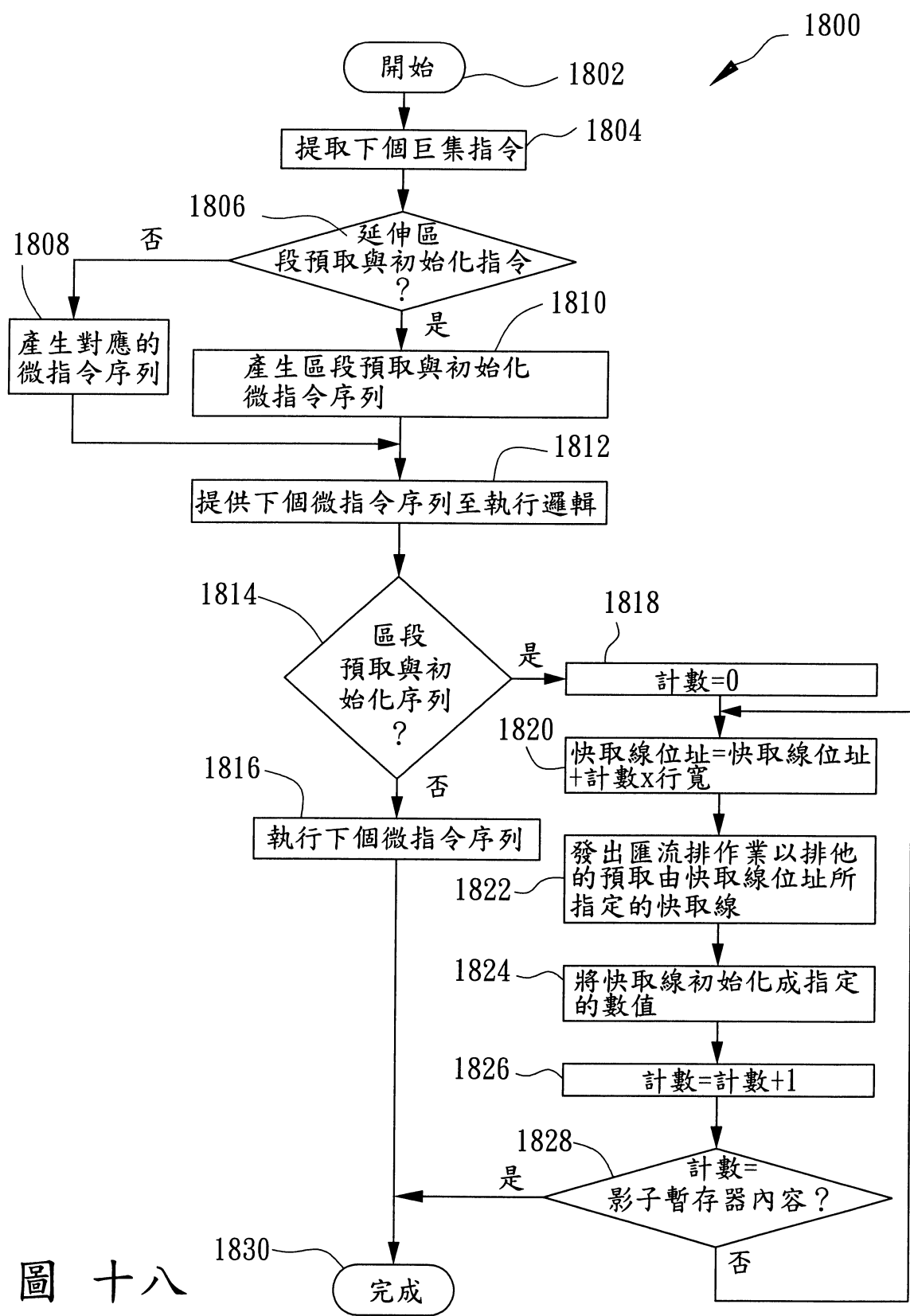


圖 十八