

公告本

408269

申請日期	87. 7. 21
案 號	87111879
新 Int. Cl.	G06F 13/16

A4
C4

(以上各欄由本局填註)

發 明 專 利 說 明 書

一、發明 名稱	中 文	在一處理系統中處理多重儲存指令完成之方法及系統
	英 文	A METHOD AND SYSTEM FOR HANDLING MULTIPLE STORE INSTRUCTION COMPLETIONS IN A PROCESSING SYSTEM
二、發明 人	姓 名	1. 百瑞 D. 威廉森 2. 吉姆 E. 菲立普 3. 蒂魁 努彥
	國 籍	均美國
三、申請人	住、居所	1. 美國德州圓岩市水草路807號 2. 美國德州圓岩市品恩歐克巷1602號 3. 美國德州奧斯丁市威登巷14308號
	姓 名 (名稱)	美商萬國商業機器公司
	國 籍	美國
	住、居所 (事務所)	美國紐約州阿蒙市
	代 表 人 姓 名	費羅普

408269

裝

訂

線

經濟部中央標準局員工消費合作社印製

(由本局填寫)

承辦人代碼：
大類：
IPC分類：

A6
B6

本案已向：

國(地區) 申請專利, 申請日期: 案號: , 有 無主張優先權

美國 1998年1月30日 09/016,654 有 無主張優先權

有關微生物已寄存於: , 寄存日期: , 寄存號碼:

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

線

經濟部中央標準局員工消費合作社印製

五、發明說明(4)

根據本發明的儲存指令，其中單一 LSU 存取 GPR 與 FPR。

圖 5 的流程說明根據本發明而完成儲存指令的資料流動。

圖 6 的時序圖在說明圖 4 的資料流動操作。

圖 7 的圖形在說明第二資料流動實施例，用以完成許多根據本發明的儲存指令，其中單一 LSU 存取 GPR 與 FPR。

圖 8 的時序圖在說明圖 7 的資料流動操作。

圖 9 的方塊圖在說明根據本發明的處理器，其可存取 GPR，FPR 與更名暫存器。

圖 10A-10F 說明用於圖 9 處理器所示實施例使用的各行列的邏輯內容。

圖 11A-11B 的流程圖在說明根據本發明實施例的處理器之操作。

圖 12 的時序圖在說明根據本發明實施例的處理器操作。

元件符號說明

10	處理器	28	載入/儲存單元
11	系統匯流排	30	浮點單元
12	匯流排介面單元	32	一般目的設計暫存器
14	指令快取	34	更名緩衝器
16	資料快取	36	浮點設計暫存器
18	循序器單元	38	更名緩衝器
20	分支單元	39	記憶體
22	固定點單元	40	特殊目的暫存器
24	固定點單元	42	進位位元暫存器
26	複固定點單元	46	發送單元

五、發明說明(1)

發明範圍

本發明大致有關於處理系統，尤其有關於在處理系統中提供多重儲存指令完成。

發明背景

在繼續開發更快與功能更強的電腦系統下，已利用一顯著的微處理器稱為減縮指令組電腦(RISC)處理器，隨著RISC處理器領域的進步已導引開發出超純量處理器。超純量處理器依名稱所述，執行習知純量微處理器中不常見的功能。在這些功能中包括的是能依程式順序而隨意執行指令，執行結果顯示出是依程式順序發生的，以便維持正確的資料一致性。

超純量處理器性能的共同瓶頸的在一已知時間處理器中的指令數目會很大，通常指令有移包括一佇列，其顯示該等指令的數目，若到達極大值則佇列通常會中止任何未來指令的發送。

一種可以減慢完成的指令是儲存指令，一儲存指令可以減慢完成的理由有多種，例如儲存指令因極大的儲存數目而減慢完成，該極大的儲存數可以在一周期中完成，以及因為可以在各周期中更新快取記憶體儲存數目。習知的超純量處理器通常僅於每一周期中完成一儲存指令，這時常會導致發送中止，因此需要一種系統其能迅速且有效的克服這些問題，並減少因缺少儲存指令完成所導致的發送單元中止的數目，以增強總處理器性能。

發明之概述

(請先閱讀背面之注意事項再填寫本頁)

不

訂

89.6.12

408269

五、發明說明 (4a)

47	指令緩衝器	114	移動單元
48	完成單元	114'	移動單元
100	減少發送瓶頸的電路	116	固定單元
102	指令單元	116'	固定單元
102'	指令單元	118	浮點單元
103	發送單元	118'	浮點單元
103'	發送單元	120	完成佇列
104	指令快取	122	資料快取
104'	指令快取	122'	資料快取
105	完成單元	130	儲存佇列
105'	完成單元	306	載入/儲存單元
106	載入/儲存單元	308	指令佇列
108	指令佇列	310	EA 單元
110	有效位址單元	312	儲存佇列
112	結束佇列		

詳細說明

本發明關於增加一超純量處理器中可完成的儲存指令數以減少發送停止。以下的說明可使一熟於此技術者製造並使用本發明，而且以符合申請專利規定的方式提出。熟於此技術者可以對本文揭示的一般原理與特徵及較佳實施例作各種修正。

圖 1 是根據本發明而處理資訊的處理器系統 10 的方塊圖，在較佳實施例中，處理器 10 是單一積體電路超純量微處理器，如 Texas, Austin 的 IBM 公司生產的 PowerPC 處理器。因此詳如以下所述，處理器 10 包括各種單元，暫存器，

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(2)

因此本發明意欲減少一周期中可完成的儲存指令數，並揭示一種在停止情況後處理一處理系統中多重儲存指令之方法及系統。處理系統包括：一指令單元，該指令單元包括一發送單元與完成單元，一移動單元與至少一執行單元。一載入儲存單元包含一指令佇列用以從發送單元接收許多指令；至少一有效位址(EA)單元用以從指令佇列接收許多指令，與一儲存佇列。儲存佇列連接移動單元，至少一完成單元與至少一EA單元。儲存佇列於完成許多指令中的每一者之前，從至少一完成單元接收與許多指令中的每一者相關的資料及實際位址資訊。

依此作法，可減少習知系統的瓶頸(即指令單元可以發送的極大數目指令)。

根據本發明的實施例而提供一種在超純量處理器處理器中使用的電路，其允許快速完成在完成表中等待的儲存指令。根據本發明的此實施例，提供一種資料佇列其儲存該儲存指令所需之資料，儲存指令執行後以及可使用其有效位址時，該電路即從GPR或更名暫存器中決定儲存指令所需的資料位置。若指令，其開始時產生該儲存所需的資料，已順利完成，則依指令是固定或浮點運算而決定在一般目的暫存器(GPR)或浮點暫存器(FPR)中設計資料。爲了目前的討論目的而假設是浮點指令。可了解的是熟於此技術者可以容易的將本發明應用於浮點指令。

若指令，其產生該儲存指令所需的資料，尚未完成，但是已由相關執行單元處理到最後結果，則資料會有效的儲

(請先閱讀背面之注意事項再填寫本頁)

不

訂

89.6.12

五、發明說明 (14)

有效信號，在周期 5 提供指令 2 的資料有效信號，在周期 6 提供指令 3 的資料有效信號，在周期 7 提供指令 4 的資料有效信號。在周期 3 的後半周期提供 RA1，在周期 4 的後半周期提供 RA2，在周期 5 的後半周期提供 RA3，而在周期 6 的後半周期提供 RA4。

在此實施例中，於周期 4 至 7，儲存佇列 312 分別載有指令 1 至 4 的實際位址值，如 514 所示。各指令 1-4 的儲存佇列結束發生於儲存的實際位址載入儲存佇列中的同一周期，如 518 所示，而且如 510 所示提供 FXU 116' 資料有效信號。因此若符合以下 2 種情況則可發生儲存佇列結束，即 1) RA 與資料載入指令 1-4 的儲存佇列，或者 2) 載入 RA 而資料轉移到指令 1-4 的儲存佇列。接著指令 1-4 的資料分別載入儲存佇列 312(若有)，如 516 所示。

停止發生後，如 521 所示，可同時完成 4 個指令如 522 所示。這與圖 3 的系統成對比，其中必須循序完成 4 個指令。完成單元 105' 接著在周期 $n+1$ 期間以便指令 1-4 的完成信號，而且因為已將所有的指令標記為完成，因此發送單元 103' 可以在同一時間(周期 $n+2$)發送指令 5-8，如 524 所示。

若處理器已到達每周期 $n+2$ 的未完成指令的極大數目的極限，則此實施例允許每一周期有 4 個指令在周期 $n+2$ 中發送，然而習用實施例僅允許發送一指令。

圖 7 顯示本發明較佳實施例的儲存指令的資料流動，此資料流動支援每一周期有 2 個指令。該等元件與圖 5 的類

(請先閱讀背面之注意事項再填寫本頁)

張

訂

五、發明說明(3)

存在更名暫存器中。若指令尚未產生該儲存指令所需的資料，則更名暫存器會接收資料以由處理器標記為有效。

一旦儲存指令所需的資料可以使用，不論是在 GPR 或是在有效的更名暫存器中，都將資料送入資料佇列的資料中。更名暫存器上設有讀取埠以便於此運算，因此在本發明的本版本中，必須在程式順序中發生完成，將資料送入資料佇列的步驟會在完成佇列指標指向儲存指令的很久之前即發生，以完成次一指令。若於儲存指令發出之前一指令已停止完成，則這會特別重要。當儲存要完成時，該儲存所需的資料即已儲存在資料佇列並送入快取記憶體，習知處理器不要求存取 GPR。這允許在單一時脈周期中完成多重儲存指令，因為在 GPR 埠中沒有瓶頸，因為儲存指令在完成窗中等待處理時資料已預載入了。當然在習知處理器中這是不可能的，該處理器在 GPR 上只有單一讀取埠。可以增加更多的埠，但是這需要額外設計，晶片實際位置及複雜性。熟於此技術者可以由以下本發明的詳細實施例揭示中更明了本發明的其它目的與優點。

配合以下詳細說明與附圖，即可更明了本發明特徵之這些與其他優點。

附圖之簡單說明

圖 1 是根據本發明的電腦系統的方塊圖。

圖 2 是處理系統中完成儲存指令的習知系統。

圖 3 的時序圖在說明圖 2 的資料流動操作。

圖 4 的圖形在說明第一資料流動實施例，用以完成許多

(請先閱讀背面之注意事項再填寫本頁)

不

訂

五、發明說明(4)

根據本發明的儲存指令，其中單一 LSU 存取 GPR 與 FPR。

圖 5 的流程說明根據本發明而完成儲存指令的資料流動。

圖 6 的時序圖在說明圖 4 的資料流動操作。

圖 7 的圖形在說明第二資料流動實施例，用以完成許多根據本發明的儲存指令，其中單一 LSU 存取 GPR 與 FPR。

圖 8 的時序圖在說明圖 7 的資料流動操作。

圖 9 的方塊圖在說明根據本發明的處理器，其可存取 GPR，FPR 與更名暫存器。

圖 10A-10F 說明用於圖 9 處理器所示實施例使用的各行列的邏輯內容。

圖 11A-11B 的流程圖在說明根據本發明實施例的處理器之操作。

圖 12 的時序圖在說明根據本發明實施例的處理器操作。

元件符號說明

10	處理器	28	載入/儲存單元
11	系統匯流排	30	浮點單元
12	匯流排介面單元	32	一般目的設計暫存器
14	指令快取	34	更名緩衝器
16	資料快取	36	浮點設計暫存器
18	循序器單元	38	更名緩衝器
20	分支單元	39	記憶體
22	固定點單元	40	特殊目的暫存器
24	固定點單元	42	進位位元暫存器
26	複固定點單元	46	發送單元

五、發明說明 (4a)

47	指令緩衝器	114	移動單元
48	完成單元	114'	移動單元
100	減少發送瓶頸的電路	116	固定單元
102	指令單元	116'	固定單元
102'	指令單元	118	浮點單元
103	發送單元	118'	浮點單元
103'	發送單元	120	完成佇列
104	指令快取	122	資料快取
104'	指令快取	122'	資料快取
105	完成單元	130	儲存佇列
105'	完成單元	306	載入/儲存單元
106	載入/儲存單元	308	指令佇列
108	指令佇列	310	EA 單元
110	有效位址單元	312	儲存佇列
112	結束佇列		

詳細說明

本發明關於增加一超純量處理器中可完成的儲存指令數以減少發送停止。以下的說明可使一熟於此技術者製造並使用本發明，而且以符合申請專利規定的方式提出。熟於此技術者可以對本文揭示的一般原理與特徵及較佳實施例作各種修正。

圖 1 是根據本發明而處理資訊的處理器系統 10 的方塊圖，在較佳實施例中，處理器 10 是單一積體電路超純量微處理器，如 Texas, Austin 的 IBM 公司生產的 PowerPC 處理器。因此詳如以下所述，處理器 10 包括各種單元，暫存器，

五、發明說明(5)

緩衝器，記憶體與其它部分，其都由積體電路形成。此外在較佳實施例中，處理器 10 根據減縮指令組計算(RISC)方法來操作。在圖 1 中，系統匯流排 11 接到處理器 10 的匯流排介面單元(BIU) 12。BIU 12 控制處理器 10 與系統匯流排 11 之間的資訊傳輸。

BIU 12 接到一指令快取 14 與處理器 10 的快取 16。指令快取 14 輸出指令到循序器單元 18，為了回應用以指令快取 14 的這些指令，循序器單元 18 選擇性的輸出指令到處理器 10 的其它執行電路。

除了循序器單元 18，其包含發送單元 46 與完成單元 48 的執行單元，在較佳實施例中處理器 10 的執行電路包括多重執行單元，即分支單元 20，固定點單元 A(FXUA) 22，固定點單元 B(FXUB) 24，複固定單元(CFXU) 26，載入檔儲存單元(LSU) 28 與浮點單元(FPU) 30。FXUA 22，FXUB 24，CFXU 26 與 LSU 28 從一般目的設計暫存器(GPR) 32 與固定點更名緩衝器 34 輸入其來源運算元資訊。此外 FXUA 22 與 FXUB 24 從進位位元(CA)暫存器 42 輸入一進位位元。FXUA 22，FXUB 24，CFXU 26 與 LSU 28 輸出其運算結果(目的運算元資訊)以儲存在固定點更名緩衝器 34 中的選定位置。此外 CFXU 26 與特殊目的暫存器(SPR) 40 互傳來源運算元資訊與目的運算元資訊。

FPU 30 從浮點設計暫存器(FPR) 36 與浮點更名緩衝器 38 輸入其來源運算元資訊，FPU 30 輸出其運算結果(目的運算元資訊)以儲存在浮點更名緩衝器 38 的選定位置中。

五、發明說明(6)

循序器單元 18 與 GPR 32 及 FPR 36 互傳資訊，分支單元 20 從循序器單元 18 輸入指令與信號以顯示處理器 10 的目前狀態。爲了回應這些指令與信號，分支單元 20 向(循序器單元 18)輸出信號以顯示適合的記憶體位址以儲存一序列指令供處理器 10 執行。爲了回應來自分支單元 20 的這些信號，循序器單元 18 從指令快取 14 輸入該顯示的指令序列。若一或多個指令序列未儲存在指令快取 14 中，指令快取 14 接著從接到系統匯流排 11 的系統記憶體 39 中輸入(透過 BIU 12 與系統匯流排 11)這些指令。

爲了回應從指令快取 14 輸入的指令，循序器單元 18 選擇性的透過發送單元 46 而發送指令至執行單元 20，22，24，26，28，30 中的選定者。各執行單元執行一特殊指令類別的一或多個指令。例如 FXUA 22 與 FXUB 24 在來源運算元上執行固定點數學運算的第一類如加，減，AND，OR 與 XOR。CFXU 26 在來源運算元上執行固定點數學運算的第二類如固定乘法與除法。FPU 30 在來源運算元上執行浮點運算元，如浮點乘法與除法。

處理器 10 在每一執行單元 20，22，24，26，28，30 上同時處理多重指令以達成高性能，因此將各指令當成一序列級來處理，各與其它指令級平行的執行。這種方法稱爲管線，在較佳實施例的顯接著特徵中，指令通常是在 6 個級中處理，即擷取，解碼，發送，執行，完成與寫回。

在較佳實施例中，各指令需要一機器周期以完成指令處

(請先閱讀背面之注意事項再填寫本頁)

表

訂

五、發明說明(7)

理的各級，然而一些指令(如 CFXU 26 執行的複固定點指令)會需要一個以上的周期。因此在一特殊指令的執行與完成級之間會發生變動延遲，以回應完成先前指令所需的時間變化。

為了回應載入指令，LSU 28 從資料快取 16 輸入資訊並將該資訊複製到更名緩衝器 34，38 中的選定者。若該資訊未儲存在資料快取 16 中，則資料快取 16(透過 BIU 12 與系統匯流排 11)而從接到系統匯流排 11 的系統記憶體 39 中輸入該資訊。此外資料快取 16 能(透過 BIU 12 與系統匯流排 11)而從接到系統匯流排 11 的資料快取 16 中輸出資訊。為了回應儲存指令，LSU 28 從 GPR 32，FPR 36，更名緩衝器 34，38 的選擇者輸入資訊並複製該資訊到資料快取 16 或記憶體中。

在執行單元如 FXUA 22，FXUB 24，更名緩衝器 34，發送單元 46 之中的互動例子中，從發送單元 46 發送指令"相加 a,b,c"到 FXUA 22。發送單元 46 提供具有運算元 a 與 b 的標籤的 FXUA 22，以告知 FXUA 22 從那裡擷取運算元的資料，這是熟於此技術者能了解的。例如在具有 6 個更名緩衝器的系統中，發送單元 46 會適當的將運算元 a 標籤為位於具 6 位元標籤 100000 的更名緩衝器 1 之中。標籤 010000 接著適當的使用以顯示運算元 b 在更名緩衝器 12 中。因為 FXUA 22 不寫入 GPR 32 之中，發送單元 46 必須序列一更名緩衝器標籤作為運算目標如 001000，並將相加指令的結果置於更名緩衝器 3。

五、發明說明(8)

如上所述，習用系統的一個問題是在一周期中僅能完成一儲存指令。爲了說明習用系統的此問題，現在要參考圖2並配合說明書來敘述。

圖2是一習知系統用以完成處理系統中的儲存指令。在此實施例中，指令單元102包括發送單元103與完成單元105，指令單元102從發送單元103中的指令快取104接收指令，發送單元103接著將接收自指令快取的儲存指令提供給載入檔儲存(LDST)單元106。發送單元103也傳送儲存指令到完成單元105，若LDST 106中有先前指令，則提供儲存指令給指令佇列108，若LDST 106中沒有先前指令，則旁通指令佇列108並可以直接提供儲存指令給有效位址(EA)單元110。在任一情況下，EA單元110接著接收一特殊指令的位址資訊。

提供有效位址的控制指令給儲存佇列130的結束佇列(FQ)112，而位址資訊則提供給移動單元114。移動單元114將EA資訊轉移到實際位址(RA)並提供RA給FQ 112。FQ 112接著提供結束有效信號給完成單元105，完成單元105接著提供完成有效信號給FQ 112，FQ 112接著將位址送入固定單元(FXU)116與浮點單元(FPU)118。FXU與FPU接著提供這些位址的相關資料到完成佇列(CQ)120，FQ 112也提供RAA資訊給CQ 120，CQ 120接著能提供儲存指令給資料快取122。

現在參考圖3，其中顯示固定單元(FXD)單元116的許多儲存指令1-4的資料流動的時序圖，在此實施例中，如數

(請先閱讀背面之注意事項再填寫本頁)

不

訂

五、發明說明(9)

字 202 所示，可看到指令 1 在周期 1 中發送，指令 2 在周期 2 中發送，指令 3 在周期 3 中發送，指令 4 在周期 4 中發送。然後，如數字 204 所示，LDST 單元 106 在周期 2 的前半周期中執行指令 1 的有效位址(EA1)，在周期 3 的前半周期中執行指令 2 的有效位址(EA2)，在周期 4 的前半周期中執行指令 3 的有效位址(EA3)，在周期 5 的前半周期中執行指令 4 的有效位址(EA4)。指令 1-4 的門位址的存取如數字 206 所示，指令 1 的位址存取是在周期 2 的後半周期與周期 3 的前半周期之間，而指令 2 的位址存取是在周期 3 的後半周期與周期 4 的前半周期之間，而指令 3 的位址存取是在周期 4 的後半周期與周期 5 的前半周期之間，而指令 4 的位址存取是在周期 5 的後半周期與周期 6 的前半周期之間。

接著從門單元提供實際位址，如數字 208 所示。在周期 3 的後半周期提供指令 1 的實際位址(RAA1)，在周期 4 的後半周期提供指令 2 的實際位址(RAA2)，在周期 5 的後半周期提供指令 3 的實際位址(RAA3)，在周期 6 的後半周期提供指令 4 的實際位址(RAA4)。

在此實施例中，如上所述結束佇列載有實際位址值以分別供周期 4-7 中的指令 1-4 使用，如 210 所示。指令 1-4 可以在周期 4-7 中完成，如 212 所示。即使可完成指令 1-4，在指令之間有一種結構的相依性，這是因為設計暫存器檔上的單一讀取埠之故，其限制處理器的性能，如 213 所示。亦即指令 1 在周期 5 中完成，若完成指令 1 則指令

(請先閱讀背面之注意事項再填寫本頁)

表

訂

五、發明說明 (10)

2 可以在周期 6 中完成，若完成指令 1 與 2 則指令 2 可以在周期 7 中完成，最後指令 1-3 都在周期 B 中完成時，指令 4 才可以完成。

一停止情況如 214 所示之後，若指令完成單元已滿，在周期 $n+1$ 開始，接著指令 1 至 4 如 216 所示，可以一次完成一個，這是因為上述的結構相依性。這是透過完成單元 105 而完成，其啓始—完成信號分別供周期 $n+1$ 至 $n+4$ 期間的完成 1 至 4 使用。因此可存取固定單元一般目的暫存器 (GPR) 以便分別從周期 $n+1$ 至周期 $n+4$ 得到指令 1 至 4 的資料如 218 所示。也分別從周期 $n+1$ 至周期 $n+4$ 提供各指令的資料有效信號，如 220 所示。

然後，將指令 1 至 4 載入各完成的儲存佇列，指令 1 在周期 $n+2$ 於完成的儲存佇列中載入，指令 2 在周期 $n+3$ 於完成的儲存佇列中載入，指令 3 在周期 $n+4$ 於完成的儲存佇列中載入，指令 4 在周期 $n+5$ 於完成的儲存佇列中載入。接著將指令 1 至 4 的資料分別於指令單元中的周期 $n+2$ 至 $n+5$ 期間，每次一指令的傳送到資料快取如 224 所示。

若處理器已達到其未完成指令的極大數目的極限，則在周期 $n+2$ 中僅可以發送一指令。這也適用於周期 $n+3$ 至 $n+6$ ，因此指令 5-8 如 226 所示，必須循序發送因為完成單元已滿，因此發送時間中的此障礙會影響處理器的總性能。

此資料流動的問題是因為如上所述的結束佇列/完成佇列的互動，若發送停止發生於完成單元 105 已滿時，則指令

(請先閱讀背面之注意事項再填寫本頁)

表

訂

五、發明說明 (11)

發送有明顯的延遲。一種克服此問題的習知方法是在完成單元上增加額外的埠，這種方法會增加系統的額外消耗與複雜性，因此需要一種方法與系統來改善處理器的總性能，方法是在發送停止已發生後才完成多重儲存指令，這比目前方法都簡單。

根據本發明所述的系統與方法藉由發生停止情況以及完成緩衝器已滿之後才完成多重儲存指令以實質改善處理器的性能，此外完成緩衝器的填充速度也比習知處理器慢，在本發明中，儲存佇列是單一佇列而非結束佇列與等待之合併，此外，與完成有關的資料是在完成指令之前而不是在完成之後存取。當完成符合以下 2 種情況時即可接著完成儲存指令，而讀取位址(RA)與資料則載入儲存佇列中，或者 RA 載入儲存佇列而資料則傳送到儲存佇列。藉由較早存取資料，即可實質上同時完成多重儲存指令。因此當完成緩衝器滿了即可於發送停止之後發送多重指令。依此作而提供一種系統其能使許多儲存指令同時完成，藉由釋出完成緩衝器中的資料以實質減少指令發送延遲。

以下的說明將進一步敘述本發明，圖 4 是儲存指令資料流動的第一實施例，其根據本發明而於每周期中提供單一儲存指令執行。有數個類似於圖 2 所示元件的元件，這些元件的圖號與圖 2 類似，惟，仍有一些重要差異。第一是載入/儲存單元 306 的儲存佇列 312 是單一佇列而不是結束佇列與完成佇列之合併。在此配置下，完成單元 105 完成儲存指令，而儲存佇列 312 於完成時在適當資料上作標記

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(12)

。在較佳實施例中，儲存佇列 312 是 16 個資料深，亦即可以在儲存佇列中儲存 16 個儲存指令。而且在較佳實施例中，各資料中的位元是根據指令單元的完成信號而動作。而且此資料流動的操作與圖 2 所述的儲存指令資料流動的不同，如上所述資料是在完成儲存指令之前存取，當符合以下 2 種情況之一時即完成儲存指令：1) 實際位址與資料載入儲存佇列，或者 2) 實際位址載入儲存佇列而資料傳送到儲存佇列。

爲了進一步說明這些差異，現在參考圖 5 其係一流程圖以大致說明圖 4 的資料流動操作。

首先，經由步驟 402 而從發送單元 103 提供許多儲存指令給載入/儲存單元 306 中的有效位址 EA 單元 310，EA 單元 310 經由步驟 404 提供控制資訊給儲存佇列 312。從移動單元 114 讀取與許多儲存指令有關的實際位址(RA)，並經由步驟 408 而從 FXU 單元 116 讀取與許多儲存指令有關的資料，經由步驟 409 而發生發送停止後，完成單元即滿了。

經由步驟 410，完成單元 105' 根據接收自儲存佇列 312 的結束有效信號而在其完成窗中完成許多儲存指令。儲存佇列經由步驟 412 而將其資料標記爲完成。然後，經由步驟 416 而從發送單元 105 可以發送相同數目的新指令，並經由步驟 418 而提供給完成單元 105'，依此作，本發明可大幅減少發送停止。

爲了進一步說明本發明的操作以及與圖 2 的資料流動的

(請先閱讀背面之注意事項再填寫本頁)

表

訂

五、發明說明 (13)

時序圖對照，現在配合圖 3 來說明圖 6。圖 6 是圖 4 的資料流動時序圖，在圖 6 的實施例中，如 502 所示，於周期 1 發送指令 1，於周期 2 發送指令 2，於周期 3 發送指令 3，於周期 4 發送指令 4，這方式與圖 3 的 202 類似。然後，類似於圖 3 的 204，載入儲存單元在周期 2 的前半周期執行指令 1(EA1)的有效位址，在周期 3 的前半周期執行指令 2(EA2)的有效位址，在周期 4 的前半周期執行指令 3(EA1)的有效位址，在周期 5 的前半周期執行指令 4(EA1)的有效位址，大致如圖 6 的 504 所示。

然後，發生指令 1-4 移動的快取移動存取，其方式與圖 3 的 206 類似。在周期 2 的後半周期與周期 3 的前半周期之間發生指令 1 的移動位址的快取存取，在周期 3 的後半周期與周期 4 的前半周期之間發生指令 2 的移動位址的快取存取，在周期 4 的後半周期與周期 5 的前半周期之間發生指令 3 的移動位址的快取存取，而且在周期 5 的後半周期與周期 6 的前半周期之間發生指令 4 的移動位址的快取存取，如圖 6 的 506 所示。

惟與圖 3 相比，若可使用資料則接著存取設計的暫存器檔(GPR)，如 508 所示。在周期 3 存取指令資料 1，接著在周期 4 存取指令資料 2，在周期 5 存取指令資料 3，在周期 6 存取指令資料 4。

接著從移動單元提供實際位址(RA1-RA4)的資料，而從 FXU 116' 提供資料有效信號給儲存佇列 312，分別如 510 與 512 所示。在此實施例中，在周期 4 提供指令 1 的資料

(請先閱讀背面之注意事項再填寫本頁)

家

訂

89.6.12

五、發明說明 (14)

有效信號，在周期 5 提供指令 2 的資料有效信號，在周期 6 提供指令 3 的資料有效信號，在周期 7 提供指令 4 的資料有效信號。在周期 3 的後半周期提供 RA1，在周期 4 的後半周期提供 RA2，在周期 5 的後半周期提供 RA3，而在周期 6 的後半周期提供 RA4。

在此實施例中，於周期 4 至 7，儲存佇列 312 分別載有指令 1 至 4 的實際位址值，如 514 所示。各指令 1-4 的儲存佇列結束發生於儲存的實際位址載入儲存佇列中的同一周期，如 518 所示，而且如 510 所示提供 FXU 116' 資料有效信號。因此若符合以下 2 種情況則可發生儲存佇列結束，即 1) RA 與資料載入指令 1-4 的儲存佇列，或者 2) 載入 RA 而資料轉移到指令 1-4 的儲存佇列。接著指令 1-4 的資料分別載入儲存佇列 312(若有)，如 516 所示。

停止發生後，如 521 所示，可同時完成一個地址的 522

(請先閱讀背面之注意事項再填寫本頁)

張

訂

五、發明說明 (15)

似，而共同元件已用類似數字表示，但是有一些不同，惟如圖所示，有 2 個 EA 單元 3310a 與 310b，也有 2 個 FXU 116' 的埠，因為 FXU 116" 上有二個埠，接著移動單元 114' 提供各埠的移動信號，而儲存佇列提供各埠的結束有效信號。

現在參考圖 8，在此實施例中，因為每一周期可處理 2 個指令，指令 1，2 在周期 1 發送而指令 3，4 在周期 2 發送如 602 所示。接著 EA1，EA2 的載入/儲存執行發生於周期 3 的前半周期如 604 所示。接著在周期 2 的後半周期與周期 3 的前半周期之間存取指令 1，2 的移動位址的快取，周期 3 的後半周期與周期 4 的前半周期之間存取指令 3，4 的移動位址的快取，如 606 所示。

接著存取設計的暫存器檔(GPR)如 608 所示。在周期 3 存取指令 1，2 的資料，而在周期 4 存取指令 3，4 的資料。

接著從移動單元提供實際位址(RA1-RA4)的資料，並分別提供指令 1-4 的資料有效信號給儲存佇列 312，如 610 與 612 所示。在此實施例中，在周期 4 提供指令 1，2 的資料有效信號，在周期 5 提供指令 3，4 的資料有效信號。在周期 3 的後半周期提供 RA1，RA2，而在周期 4 的後半周期提供 RA3，RA4。

在此實施例中，於周期 4 至 5 儲存佇列 312' 分別載有指令 1 至 4 的實際位址值，如 614 所示。各指令 1-4 的儲存佇列結束發生於載入儲存佇列的實際位址的同一周期，如

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (16)

618 所示。接著分別在周期 5, 6, 指令 1-4 的資料載入儲存佇列 312' (若有), 如 616 所示。

停止發生後, 如 621 所示, 可同時完成 4 個指令如 622 所示。指令單元接著在周期 $n+1$ 中啓始指令 1-4 的完成信號, 而且因為已將所有的指令標記為完成, 因此發送單元可以在同一時間 (周期 $n+2$) 發送指令 5-8, 如 624 所示。

若處理器已到達每周期 $n+2$ 的未完成指令的極大數目的極限, 則可發送高達 4 個指令, 然而習用實施例僅允許發送一指令。當發送指令 8 時這可以省下 3 個指令。

在根據本發明的系統與方法中, 因為多重指令是一起處理, 因此可以一起完成指令, 藉以實質減少完成儲存指令的周期數。指令單元可發送的指令的極大數目瓶頸即可以減輕, 方法是不用習知系統所需的完成佇列資料, 藉以使指令單元發送新的指令。

現在參考圖 9, 其中顯示根據本發明的另一實施例而用以減少發送瓶頸的電路 100, 在此實施例中, 除了 GPR 114 以外, 更名暫存器 116 也設有讀取埠, 其允許它經由多工器而將資料送入資料佇列 122。電路 100 也包括標籤佇列 104 以儲存電路 100 使用的指令標籤, 用以在各佇列中將一特殊儲存指令所屬的位址與資料對齊。標籤佇列 104 中的各欄最好是 n 位元寬, 其中選定的 n 必須使足夠數目的標籤能一對一的識別出處理器中所有未定的儲存指令。TAGQ 之中的各欄也與一有效位元結合以指示標籤欄中包含的資料是否為一有效的未定指令。LSU 執行指令時即開

(請先閱讀背面之注意事項再填寫本頁)

衣

訂

五、發明說明(17)

始將有效位元加入，而標籤則送入 TAGQ 之中的位置。對應儲存指令的資料從資料佇列送入快取後即刪除有效位元，詳如以下所述。當刪除有效位元時，後續儲存指令即可再使用該位置。圖 10a 說明 TAGQ 之 104 中的個別資料。

電路 100 也都得之指標佇列(PTRQ) 106，其包含用以選擇一特殊 GPR 或更名暫存器檔位置的資訊，其包含一特殊儲存指令所需的資料。圖 10b 顯示典型指標佇列位置中包含的欄位。特別的，儲存指標佇列是一種機制以確保資料載入儲存資料佇列 122 的正確位置。因為資料可以從 FXU 或 FPU 執行單元中隨意返回，因此需要儲存指標佇列。在一實施例中，儲存指標佇列一向下 n 層深的佇列。由 LSU 邏輯 102 檢查最底部的 2 個位置以便從能 FXU 或 FPU 執行單元要求資料。用設計的暫存器檔指標來定址 FXU (GPR) 或 FPU (FPR) 暫存器檔，而用更名暫存器檔指標來定址 FXU(更名)或 FPU(更名)暫存器檔。設計的與更名暫存器檔都是平行存取，由相關 PTRQ 位置中的 R/A 位元決定是否將資料送入 DATAQ 122，若設定 R/A 位元，則使用設計的複製，然而若沒設定 R/A 位元，則由多工器 118 選定更名資料。在發送時間從更名單元指派 R/A 位元的初始值，並且與指令一起送入載入/儲存執行單元。這是動態的位元其必須由 LSU 邏輯維持，因為儲存指令的來源資料位置可以在儲存指標能存取資料之前，從更名暫存器檔移到設計的暫存器檔。換言之開始時判定儲存是在更名(R/A=0)中，但是在儲存指標能從更名存取資料之前，指令單元已

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(18)

完成指令，其修正 GPR 而成爲儲存指令的來源。這使得更名資料載入設計的暫存器檔，LSU 邏輯又將相關 PTRQ 位置中的 R/A 位元從 0 修正爲 1。

當資料從更名或設計的暫存器檔中存取資料時，用儲存指標標籤欄將儲存資料佇列中的資料與儲存標籤佇列中的對應位置對齊，v 位元僅顯示儲存指標佇列位置是否有效。

在每一周期，掃描指標佇列中的最底的 2 個位置以及包含在 GPR 的位置中的資料，而更名檔即送入多工器 118。控制邏輯檢查 GPR 與更名檔位置的有效位元。若位置中的資料不是有效，則不將它送入資料佇列。大致上將資料丟棄並且在次一周期再掃描指標佇列位置。最後，儲存指令所需的資料會寫入更名暫存器，並且會加入更名有效位元，其對應該特殊的更名暫存器。當資料送入資料佇列時，控制邏輯即在指標佇列位置中將該有效位元刪除。這允許後續儲存指令再使用指標佇列位置，因爲指標佇列(與結束佇列)與 TAGQ 之沒有對齊。

電路 100 也包含儲存結束佇列(FINQ) 110，儲存結束佇列 110 的功能爲完成儲存指令，這也是向下佇列。結束佇列 110 保有位址產生周期期間產生的唯一儲存標籤，但是唯有標記爲每 IID 的最後要求，其中 IID 是指指令識別碼其於發送時指派給各指令，而且用以追蹤處理器中指令。尤其是，在本發明的一特定實施例中，任何不對齊指令(即儲存指令要求的資料在快取邊界是不對齊的)在每一 IID 中具

(請先閱讀背面之注意事項再填寫本頁)

張

訂

五、發明說明 (19)

有 2 個快取存取要求，而且僅有第二要求標籤置於結束佇列。在對齊指令中，各要求標記為最後而且將其標籤置於結束佇列。當然，不對齊指令的處理是極度依賴本發明使用的特殊結構。

在本發明的此版本中，LSU 邏輯掃瞄到的最舊 2 個位置要察看其是否可以結束。依時序的考量，在各周期中本發明易於改為掃瞄更多或更少的位置，要了解的是儲存指令可以隨意完成，即若最舊的位置因某種原因而不能結束時，但是第二舊的可以。儲存結束佇列 110 中的 2 個最舊指令的標籤與儲存 TAGQ 之 104 中的標籤比較，以判定應該檢查儲存 EA/控制佇列 108 中的那一位置(詳如以下所述)。一旦其實際位址與資料已分別載入實際位址佇列 112 與儲存資料佇列 122，即可結束儲存指令，或者位址載入實際位址佇列而資料載入資料佇列中。一旦結束佇列可以結束一指令，接著其 IID，其從儲存 EA/控制佇列 108 中取得，並傳送結束信號到指令完成單元(未示)。

再度，V 位元僅顯示儲存結束佇列位置是否有效，當執行的指令於開始時即載入結束佇列，則加入有效位元。當資料在實際位址佇列中而對應儲存指令的資料佇列是有效的，則傳送結束報告給完成單元並丟棄結束佇列有效位元。換言之操作結束佇列以確保實際位址與儲存指令所需的資料在可以完成儲存指令之前都可以使用。結束報告包括得自對應指令的 EA/CNTLQ 位置的 IID，以及任何例外資訊。

五、發明說明(20)

電路 100 又包含有效位址控制佇列 EA/CNTLQ 108，圖 10 D 顯示一典型的 EA/CNTLQ 位置。儲存 EA/CNTLQ 108 是 n 層深的向下佇列，它保有有效的位址，其將要與實際位址一起傳送到資料快取，以及將儲存資料傳送到更新記憶體。此佇列也包含 IID，其將由儲存佇列使用以完成指令完成單元的儲存指令。e 位元是例外位元，其防止若執行期間發生例外時將此資料傳送到資料快取。處理器上的各種來源可以處理該 e 位元，這是一種設計選擇。例如在本發明的一實施例中，由某一事件如儲存中斷上的 DCMMU 的位址移動邏輯設定該 e 位元。

c 位元是完成位元其表示儲存指令已由指令完成單元成功的完成，而記憶體的儲存應該發生。僅將完成的儲存傳送到資料快取，僅當 TAGQ 中的對應位置，即資料佇列與實際位址佇列在儲存指令中有效，才將 c 位元加入一特殊 EA/CNTLQ 位置中，而完成單元顯示已完成此指令。資料已從資料佇列傳送到快取後，才丟棄對應 EA/CNTLQ 位置中的有效位元，同時 TAGQ 之，資料佇列與實際位址佇列中的有效位元也在對各自位置中丟棄。

在此，熟於此技術者會更了解本發明的一重要優點，因為一儲存指令所需的位址與資料已在位址與資料佇列之中，要完成指令時，即指令的 IID 成爲完成表中次一要完成的資料時，即可不必從結構或更名暫存器檔執行讀取即可完成指令。反之，資料僅從資料佇列送入快取，在本發明的此版本中，會假設資料佇列僅設置有單一讀取埠，接著

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(21)

僅有一儲存指令在每一周期能將其資料移到快取。惟，因為不必讀取暫存器檔，本發明允許在單一周期將數個 IID 從完成單元送入 EA/CNTLQ，設定各相關位置的 c 位元。一旦已傳送 IID，完成表中的資料即可由後續指令再使用，這可以在單一周期於完成表中釋出多重資料，並防止發送停止的發生(若完成表已滿)。

依本發明結構上的實例可以為各種目的而設置其它各種控制位元，例如儲存格式化器以便將資料格式化成為與記憶體對齊。v 位元顯示儲存 EA/CNTLQ 佇列位置是有效的，而且當載入一資料時即加入，以及如上所述的丟棄。

電路 100 使用的另一佇列是實際位址佇列(RAQ) 112，此佇列儲存實際位址供儲存指令的快取存取，其在完成緩衝器(未示)中未定。此佇列中的位址與對應的儲存資料一起送入快取，以實現實際寫入記憶體。

最後電路 100 具有一資料佇列 122，其儲存各儲存指令所需的資料，其在完成緩衝器(未示)中未定，資料佇列 122 從 GPR 或更名暫存器檔接收資料，這是依相關儲存指令執行時資料的位置而定。

本發明的一版本的操作如下，發送時，將一儲存指令從發送單元(未示)送入載入/儲存單元邏輯 102。發送單元指派一識別標籤(IID)給指令，其用以追蹤整個處理器中的指令。在以下的時脈周期中，LSU 102 計算指令的有效位址，以及也指派唯一的標籤給指令用以識別。注意，IID 與 LSU 產生的標籤是有區別而且儲存於 TAGQ 104 中。在此

(請先閱讀背面之注意事項再填寫本頁)

本

訂

五、發明說明(22)

實施例中的標籤識別碼僅於 LSU 內部使用，這簡化邏輯設計因為傳送原始儲存至資料快取之前，可以再發出由發送單元指派給儲存指令的 IID。雖然電路一定允許使用 IID 以對齊 LSU 佇列中的資料，此實施例使用的另一標籤則是設計選擇的事。LSU 邏輯 102 接著將標籤送入標籤佇列 (TAGQ) 104，其可用以維持電路 100 中的各種佇列之間的對齊，其方式詳如以下所述。

在相同周期中，LSU 邏輯也將指令與其標籤送入指標佇列 (PTRQ) 106 儲存 6，此實施例的 PTRQ 106 是簡單的向下佇列，PTRQ 中的各位置包含：標籤欄 (T)，GPR 指標，更名指標，更名/設計位元 (R/A)，與有效位元 (V) 如圖 10B 所示。T 欄位包含指令的識別標籤其在先前周期中由 LSU 邏輯 102 指派。GPR 欄位包含發送時指令相關的 GPR 指標，類似的，更名指標包含發送時指令相關的更名指標，雖然此資訊來自更名控制邏輯，因為它不是指令解碼的一部分。R/A 位元識別出位置其包含指令相關的資料，即資料是否在更名暫存器之中，或者它是否設計在 GPR 之中。在一實施例中，若設定 R/A 位元，接著指令的資料即儲存在 GPR 位置中。若儲存指令所需的資料在 GPR 中，則不必更新 R/A 位元因為可以確定設計的暫存器不會改變直到儲存指令要求完成該資料之後。

惟若沒有設定 R/A 位元，則可以在更名暫存器位置中找出資料，或者將可以找出。在此情況下必須在 PTRQ 106 中更新 R/A 位元，如以下所述。LSU 邏輯 102 包含來自更

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (23)

名暫存器控制邏輯(未示)的 R/A 位元，並將它同時與其它資訊一起送入 PTRQ 106。若儲存指令的資料要從更名暫存器中得到，則更名位置的有效位元顯示該更名資料是否有效，即產生該資料的指令是否已將它寫更名暫存器，或者產生指令是否仍未決定。R/A 位元與更名位置的有效位元一起使用以允許在每一周期將儲存指令(若有)送入 DATAQ 122。

尤其是，LSU 邏輯 102 在每一周期掃瞄 PTRQ 106，該邏輯 102 是因此目的而設置。掃瞄 PTRQ 位置時，PTRQ 位置中的更名指標即與完成單元的寫回向量(也是更名指標)比較，以判定是否應該設定對應的 R/A 位元。尤其是，完成單元中的各資料包含一寫回向量欄位，其對應更名暫存器，其中可找出一完成指令的資料。處理器上的邏輯用寫回向量將更名暫存器中的資料寫入設計的暫存器，其由產生資料的指令設定目標。在指標佇列位置中的更名指標與完成表中的寫回向量之間比對相符時，這表示產生儲存指令所需資料的指令已完成而且現在設計在 GPR 位置中。因此該特殊儲存指令所需的資料不會在更名暫存器中找到，但是會設計在 GPR 中。因此 LSU 邏輯在指標佇列位置中將 R/A 位元切換。

也掃瞄 PTRQ 106 以判定是否有任何位置具有準備送入 DATAQ 122 的資料，為了時序理由，可期望僅掃瞄部分的 PTRQ 106，如最後的 2 個位置，雖然這是設計的選擇自由。掃瞄 PTRQ 106 允許指令將其資料隨意寫入 DATAQ 122

(請先閱讀背面之注意事項再填寫本頁)

衣

訂

五、發明說明(24)

。例如若掃瞄 PTRQ 106 中的最後(即最舊)的位置，並發現其資料儲存在更名暫存器中，但是更名有效位元顯示資料尚未從先前資料產生指令收到資料，則 PTRQ 106 中的最舊指令的資料不能送入 DATAQ 122。邏輯接著掃瞄第二舊的位置，若設定此位置的 R/A 位元，則此位置的資料包含在 GPR 中。在此情況下 LSU 邏輯用 GPR 指標一起 R/A 位元來選擇適當的 GPR 位置，並儲存資料在資料佇列中。該特殊 DATAQ，其接收 GPR 的資料，如上所述是對應 TAGQ 中相同位置者。換言之處理器比指標佇列中的 T 欄位與相關儲存指令的 TAGQ 中的 T 欄位。若相關儲存指令的標籤儲存在 DATAQ 中的第四位置，則從 GPR 收到的資料會寫入 DATAQ 中的第四位置。依此標籤在 DATAQ，RAQ，EA/CNTLQ 中維持一致。至於各特殊的儲存指令，TAGQ，EA/CNTLQ，RAQ，DATAQ 中的位置會具有與該指令相關的資訊。

選定位置的資料則通過多工器(MUX) 118 而到達門 120，要注意的是用 MUX 118 來選擇一特別者是否從 GPR 或更名暫存器位置送入門 120，尤其是，提供 PTRQ 106 位置的 R/A 位元以輸入到 MUX 118 選擇線，若設定位元，則選擇 GPR，若重設位元，則選擇更名暫存器資料。

在以下的時脈周期中，資料從門 120 送入 DATAQ 122，為了維持正常的資料一致，用標籤來選擇 DATAQ 122 中的適當的位置以接收資料。DATAQ 122 中的此位置會進一步對應儲存在 RAQ 112 中的適當實際位址，由以下例子來詳

(請先閱讀背面之注意事項再填寫本頁)

張

訂

五、發明說明(25)

細說明它，假設 LSU 102 指派一指令給標籤識別碼 06，又假設當標籤從 LSU 102 通過時，第四位置是 TAGQ 中的次一可用位置。因此，LSU 102 會將標籤 06 寫入 TAGQ 104 的第四位置中。位址移動後，DCMMU(未示)將儲存指令設定目標的快取位置的實際位址以及標籤一起送入標籤 112。LSU 控制邏輯比較標籤俾將實際位址送回儲存在 TAGQ 104 的標籤中。此例的比較顯示如上所述標籤 06 存在 TAGQ 104 的第四位置中。因此 LSU 控制邏輯會將儲存資料指令指派標籤 06 的實際位址送回以寫入 RAQ 112 的第四位置。類似的，當具有標籤 06 的儲存指令的資料寫入 DATAQ 122 時，LSU 控制邏輯首先掃瞄 TAGQ 104 以判定標籤 06 是否儲存在標籤的第四位置，並接著將門 120 的資料寫入 DATAQ 122 的第四位置中。當然，熟於此技術者將可了解的是維持資料一致的其它方法也可容易的使用，而這是設計選擇的自由。

在此，會了解的是上述此實施例中的電路是管線的以允許多重儲存指令將實際位址與資料分別排入 RAQ 112 與 DATAQ 122 的位置中。接著在完成時，儲存指令不必存取 GPR，反之，在完成時，將適當的儲存佇列位置標記為完成，而且可以循序傳送到快取。所需的資料從適當的 DATAQ 122 位置，與儲存在 RAQ 112 中的對應快取位址，一起傳送到快取。

圖 11A, 11B 顯示的流程圖在說明本發明的實施例之操作，在步驟 1102，發送儲存指令給 LSU 邏輯，解碼的指令包

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (26)

含設計的暫存器，其具有儲存指令所需的資料。若對應的 GPR 位置尚未更新此資料，則如上所述從更名邏輯得到對應的更名暫存器指標。流程接著跳到步驟 1104 以執行指令，並且從解碼的指令資料產生有效位址。在此同時 LSU 控制邏輯也產生標籤。在步驟 1106，將有效位址與標籤傳送到快取記憶體管理單元 DCMMU 供 DCMMU 計算出實際位址。DCMMU 已計算出實際位址後，即與標籤在步驟 1108 一起返回到 LSU 控制邏輯，其在步驟 1110 將 DCMMU 的標籤與 TAGQ 比較並載入對應的實際位址佇列位置，其具有返回自 DCMMU 的實際位址。換言之，若該特殊儲存指令的 TAGQ 儲存在第四 TAGQ 位置中，則從 DCMMU 返回的實際位址，在步驟 1110 會儲存在實際位址佇列的第四位置中。

而且在完成步驟 1104 之後，處理器跳到步驟 1112 其中 TAGQ 與指標佇列載有步驟 1104 中產生的標籤與設計/更名檔指標。通常 EA/CNTLQ 與結束佇列是分別在步驟 1116，1118 中載入。

在步驟 1114，LSU 控制邏輯檢查相關儲存指令的儲存指標佇列位置中的 R/A 位元以判定指令所需的資料是否儲存在 GPR 位置中。若不是，則系統跳到步驟 1120 以檢查以了解對應所需資料的更名位置是否有效。若儲存指令所需的資料不是在 GPR 中而且無效，則在步驟 1122，處理器返回步驟 1120。若在步驟 1120，更名位置中的資料是有效，則流程跳到步驟 1124，而且用指標佇列的更名指標來

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (27)

讀取更名暫存器檔。最後在步驟 1114，若儲存指令所需的資料在 GPR 中，則流程跳到步驟 1126，而且用指標佇列的 GPR 指標來讀取對應的 GPR 檔。

讀取更名或 GPR 暫存器檔之後，流程跳到步驟 1128 並且根據 PTRQ 的 R/A 位元而選擇對應的更名/GPR 檔供對應的儲存指令用。PTRQ 中的標籤與 TAGQ 比較以判定對應的 DATAQ 位置是否要從多工器接收資料。再度，若相關儲存指令的標籤在 TAGQ 的第四位置中，則讀取自更名或 GPR 的資料會寫入 DATAQ 的第四位置中，同時指標佇列資料會從指標佇列中移走。

在步驟 1130，資料寫入 DATAQ 位置後，處理器接著跳到步驟 1132，其中判定儲存指令是否準備完成。至於此操作時，LSU 邏輯掃瞄結束佇列標籤並比較該等標籤與 TAGQ 中的標籤。在本發明的一版本中，於一周期僅掃瞄底部的 2 個 FINQ 位置，惟這是設計的選擇自由，而且在單一周期可掃瞄更多或較少的 FINQ 位置。一特殊結束佇列的標籤與 TAGQ 標籤比較以判定標籤的對應 TAGQ 位置是否擷取自結束佇列。再度，若結束佇列的特殊標籤可以在第四佇列位置中找到，則處理器檢查 EA/CNTLQ，RAQ，DATAQ 位置中對應的第四位置。至於這些對應位置中的每一者，處理器都判定有效位元是否在步驟 1134 中設定，若沒設定，則處理器返回步驟 1132 並重覆步驟 1132，1134，直到設定有效位元，供上述 3 個佇列中的每一者的對應位置用。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(28)

然後流程跳到步驟 1136，其中藉由比較 EA/CNTLQ 的 IID 與完成單元佇列中的對應 IID 而完成儲存指令。同時移走對應的結束佇列位置，在步驟 1138，處理器等待完成單元的完成，完成時，對應位置具有某一控制位元設定以顯示處理器準備將資料從 DATAQ 傳輸到快取記憶體。要注意的是，在本發明的此實施例中，從 DATAQ 至快取的實際資料傳輸是在對應的儲存指令標記為完成之後的數個周期，這是因為有多個會發生在處理器上的因素。在步驟 1140，設定其 c 位元也可將適當的 EA/CNTLQ 位置標記為完成。

在步驟 1142，處理器接著傳送實際位址，有效位址，指令 ID，以及從 DATAQ 傳輸資料到快取所需的其它必備控制信號，至 DCMMU。最後在步驟 1144，對應完成儲存指令的 TAGQ，EA/CNTLQ，RAQ，DATAQ 位置都從其各自的佇列中移走以釋出空間供後續發送的儲存指令用。

圖 12 的時序圖在說明上述實施例的電路操作，也會參考圖 1，2 的處理器的方塊圖以說明圖 12 的時序圖。在周期 1 中，從指令快取 14 發送儲存指令，並且由發送單元 46 解碼。接著將解碼的指令送入載入/儲存單元(LSU) 28，在周期 2 的前半周期中，LSU 28 計算指令的有效位址，這有時在此稱為位址產生或 "agen"，或執行。熟於此技術者會熟悉記憶體定址的不同模式以及有效與實際位址之間的差異，將可了解的是本發明可適用於許多記憶體定址設計，而使用其中的任一種則是設計的選擇自由。以下藉由參

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(29)

考 IBM 公司生產的 "PowerPC" 族處理器所使用的典型記憶體定址設計來說明本發明，這種記憶體定址模式的詳細內容可參考市面上的公開資料如 IBM 的出版品編號 MPRPPCFPE-01 中所述的 "PowerPC 微處理器族：程式環境"，其完全在此併供參考。

而且也在周期 2 中，由 LSU 28 產生指令的標籤，將有效位址與指令標籤送入記憶體管理單元以便於周期結束時作實際位址計算。

在周期 3，LSU 28 更新指標佇列 106 與 TAGQ 之 104，送入這些佇列的資訊則如上所述，而且實際位址從資料快取 16 的記憶體管理單元中返回。而且也在周期 3，存取 GPR(假設儲存指令的資料是在 GPR)，為了時序理由，可期望於將資料送入 DATAQ 之前門鎖該資料一個周期。因而在此實施例中，LSU 控制邏輯在周期 4 產生 FXU 儲存有效管線門信號以便將資料門鎖在門 120 中。

在周期 150，產生結束指令信號，如上所述用該結束佇列來結束儲存指令。它包含唯一的儲存標籤，掃描結束佇列中的各位置，比結束佇列位置的標籤以便於 TAGQ 中找出匹配。若對應的 TAGQ 位置具有 DATAQ 與 RAQ 有效位元組，則完成 EA/CNTLQ 的對應 IID。在一實施例中，將結束佇列與指標佇列作類似的處理，並掃描底部的 2 個位置。惟，如上所述，儲存能隨意結束。將結束信號傳送到完成單元以指示當它是完成佇列中的完成指令的次一者時，儲存指令即可完成。

五、發明說明 (30)

仍在周期 5，將儲存指令資料從門 120 送入 DATAQ 122，一種將資料與 DATAQ 122 中的適當位置對齊的適當方法已於以上說明。在此，儲存指令準備在周期 5 中完成，因為資料是於周期 4 從多工器中載入，即其中完成指令的周期。

完成單元能在一信號周期中完成多重指令，如上所述完成僅在儲存佇列完成中標記 4 個位置，其表示這些可以送入 DATAQ，各位置的完成位元維持在 EA/CNTLQ 中，當可以使用快取時，LSU 將最舊的完成的儲存佇列位置傳送到快取，(即若較高優先的要求不忙碌時即送入快取)。

因此熟於此技術者將可了解的是藉由存取更名暫存器檔，緊密連接迴路中的資料儲存指令可以更迅速的完成，藉以減少因完成緩衝器滿了而導致的發送停止數目，雖然已根據一些實施例來說明本發明，一熟於此技術者將可了解的是實施例是可以有變化，而這些變化是在本發明的精神與範圍之中。因此雖然已用一時完成 4 個儲存指令來說明本發明，一熟於此技術者將可了解的是可以完成任何數目的儲存指令，而且其也在本發明的精神與範圍之中。因此在不違反本發明的精神與範圍下，一熟於此技術者可以作許多修正，而該範圍是由以下申請專利範圍所定義。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

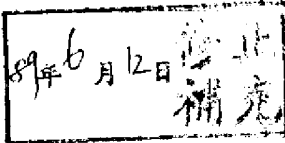
四、中文發明摘要(發明之名稱：在一處理系統中處理多重儲存指令完成之方法及系統)

本發明之一特徵係關於一種用以在一超純量處理器上處理一儲存指令之裝置，該處理器使用依序完成指令之方式，該處理器具有：一指令發送單元，一設計暫存器檔，一更名暫存器檔，一載入儲存單元，一完成單元與快取記憶體。在本發明一實施例中，該裝置包括：一指標佇列，其具有對應儲存指令之位置，該位置包含一指標以指向該設計與更名暫存器檔中之位置，以包含儲存指令所需之資料；及一多工器，連接至該設計與更名暫存器檔上之讀取埠，俾資料能從一暫存器檔送入一資料佇列中之位置，該資料佇列連接至快取記憶體。

英文發明摘要(發明之名稱：A METHOD AND SYSTEM FOR HANDLING MULTIPLE STORE INSTRUCTION COMPLETIONS IN A PROCESSING SYSTEM)

One aspect of the invention relates to an apparatus for processing a store instruction on a superscalar processor that employs in-order completion of instructions, the processor having an instruction dispatch unit, an architected register file, a rename register file, a load store unit, a completion unit and cache memory. In one embodiment of the invention, the apparatus includes a pointer queue having an entry corresponding to the store instruction, the entry containing a pointer to the entries in the architected and rename register files that contain data required by the store instruction; and a multiplexer coupled to read ports on the architected and rename register files so that data can be passed from one of the register files into an entry in a data queue, the data queue being coupled to the cache memory.

六、申請專利範圍



1. 一種在超純量處理器中用以處理儲存指令之方法，該方法包含：

從一發送單元發送儲存指令；

執行儲存指令以判定一實際位址記憶體位置，以儲存被儲存指令定為目標之資料，及包含資料之暫存器檔中之位置；

在儲存指令完成之前將暫存器檔中位置之資料送入一資料佇列；以及

在完成儲存指令時，將資料佇列中之資料送入快取記憶體。

2. 如申請專利範圍第1項之方法，其中處理器在一時脈周期期間，將資料佇列中之多重位置中之許多儲存資料送入快取記憶體中之對應位置。

3. 如申請專利範圍第1項之方法，其中從暫存器檔傳送之資料包含將設計之暫存器檔與更名檔，其對應儲存指令定為目標之資料，多工在資料佇列中。

4. 一種用以在一超純量處理器上處理一儲存指令之裝置，該處理器使用指令之順序完成，該處理器具有：一指令發送單元，一設計暫存器檔，一更名暫存器檔，一載入儲存單元，一完成單元與快取記憶體，該裝置包括：

一指標佇列，其具有對應儲存指令之位置，該位置包含一指標以指向該設計與更名暫存器檔中之位置，以包含儲存指令所需之資料；以及

一多工器，連接至該設計與更名暫存器檔上之讀取

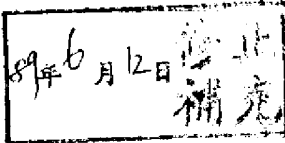
(請先閱讀背面之注意事項再填寫本頁)

裝

訂

修正本有無效之內容，應予修正。
89年6月12日修正之。

六、申請專利範圍



1. 一種在超純量處理器中用以處理儲存指令之方法，該方法包含：

從一發送單元發送儲存指令；

執行儲存指令以判定一實際位址記憶體位置，以儲存被儲存指令定為目標之資料，及包含資料之暫存器檔中之位置；

在儲存指令完成之前將暫存器檔中位置之資料送入一資料佇列；以及

在完成儲存指令時，將資料佇列中之資料送入快取記憶體。

2. 如申請專利範圍第1項之方法，其中處理器在一時脈周期期間，將資料佇列中之多重位置中之許多儲存資料送入快取記憶體中之對應位置。

3. 如申請專利範圍第1項之方法，其中從暫存器檔傳送之資料包含將設計之暫存器檔與更名檔，其對應儲存指令定為目標之資料，多工在資料佇列中。

4. 一種用以在一超純量處理器上處理一儲存指令之裝置，該處理器使用指令之順序完成，該處理器具有：一指令發送單元，一設計暫存器檔，一更名暫存器檔，一載入儲存單元，一完成單元與快取記憶體，該裝置包括：

一指標佇列，其具有對應儲存指令之位置，該位置包含一指標以指向該設計與更名暫存器檔中之位置，以包含儲存指令所需之資料；以及

一多工器，連接至該設計與更名暫存器檔上之讀取

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

修正本有無效之內容，應予修正。
89年6月12日
修正

六、申請專利範圍

- 埠，俾資料能從一暫存器檔送入一資料佇列中之位置，該資料佇列連接至快取記憶體。
5. 如申請專利範圍第4項之裝置，其中指標佇列中之位置更包含一位元，其表示儲存指令所需之資料是否儲存在設計或更名檔之位置中。
 6. 如申請專利範圍第5項之裝置，其中多工器之選擇線回應指標佇列位置中之位元。
 7. 如申請專利範圍第4項之裝置，其中當結束指令，惟在完成指令之前，儲存指令即從多工器送入資料佇列中之位置。
 8. 如申請專利範圍第7項之裝置，其中完成儲存指令時，指令所需之資料從資料佇列中之位置送入快取記憶體。
 9. 如申請專利範圍第8項之裝置，其中在單一時脈周期期間，將多重儲存指令中之資料送入資料佇列。
 10. 如申請專利範圍第4項之裝置，更包含一標籤佇列，其包含一標籤可獨特地識別出儲存指令。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

- 埠，俾資料能從一暫存器檔送入一資料佇列中之位置，該資料佇列連接至快取記憶體。
5. 如申請專利範圍第4項之裝置，其中指標佇列中之位置更包含一位元，其表示儲存指令所需之資料是否儲存在設計或更名檔之位置中。
 6. 如申請專利範圍第5項之裝置，其中多工器之選擇線回應指標佇列位置中之位元。
 7. 如申請專利範圍第4項之裝置，其中當結束指令，惟在完成指令之前，儲存指令即從多工器送入資料佇列中之位置。
 8. 如申請專利範圍第7項之裝置，其中完成儲存指令時，指令所需之資料從資料佇列中之位置送入快取記憶體。
 9. 如申請專利範圍第8項之裝置，其中在單一時脈周期期間，將多重儲存指令中之資料送入資料佇列。
 10. 如申請專利範圍第4項之裝置，更包含一標籤佇列，其包含一標籤可獨特地識別出儲存指令。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

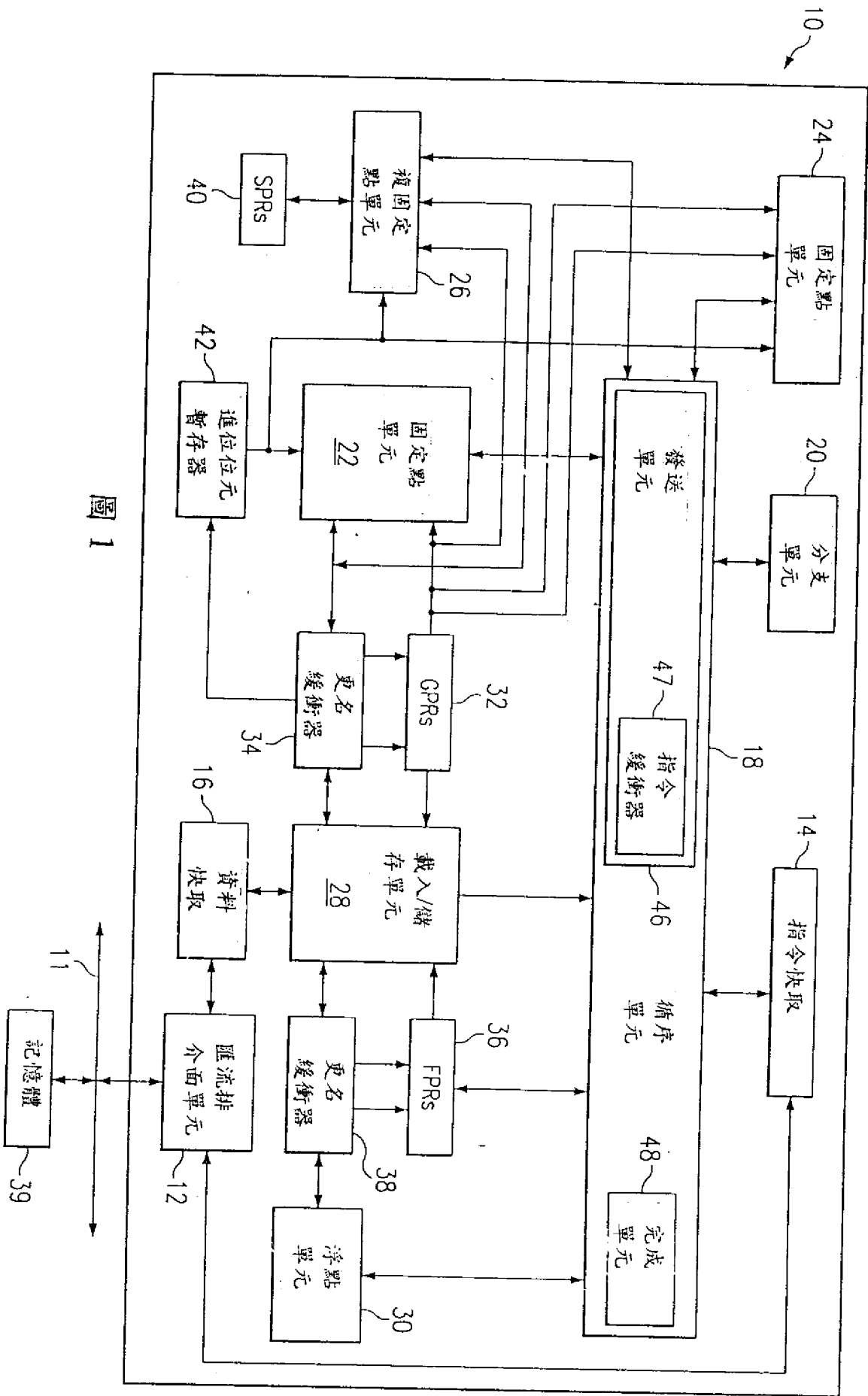


圖 1

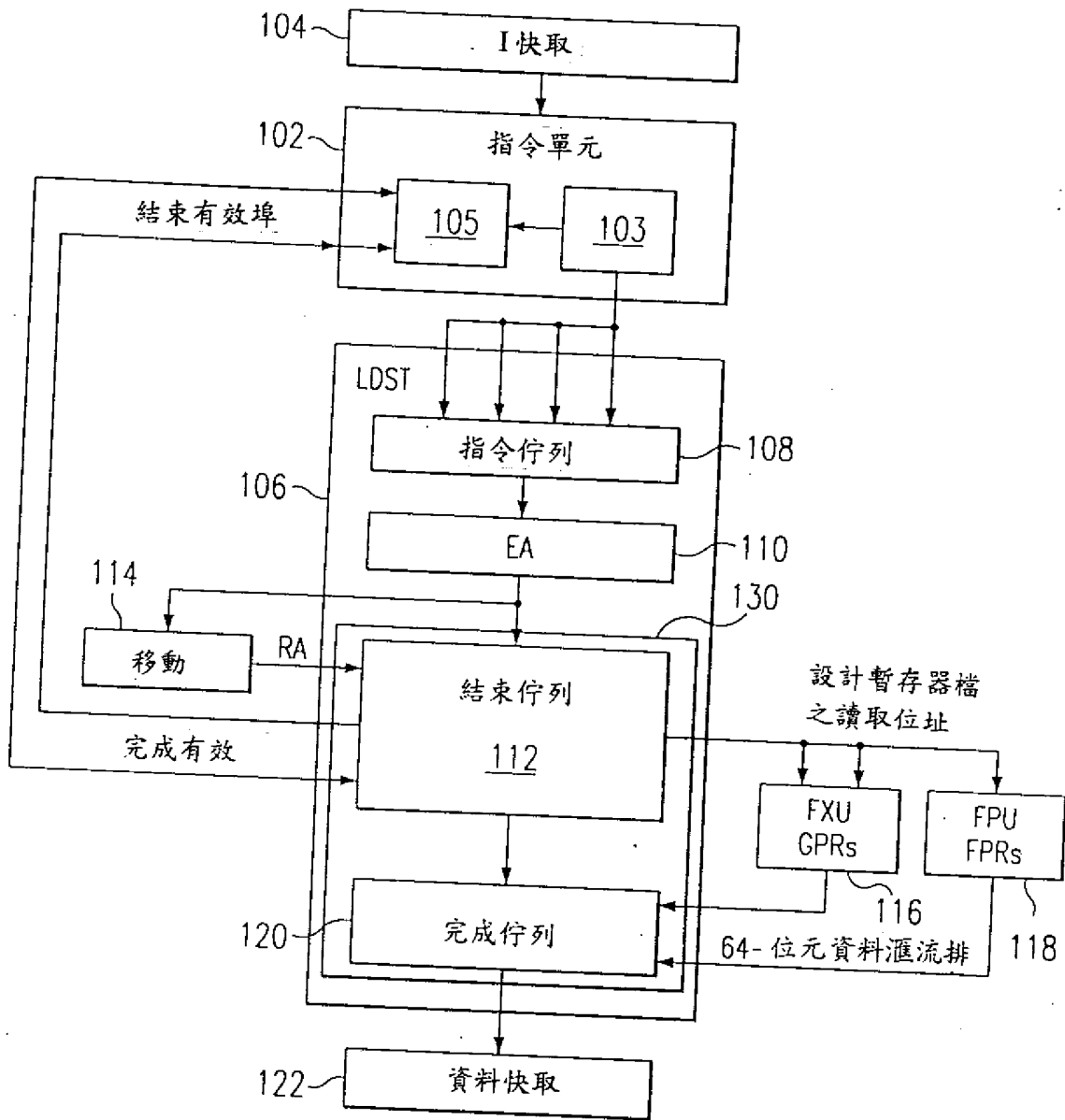


圖 2

儲存時序圖

214

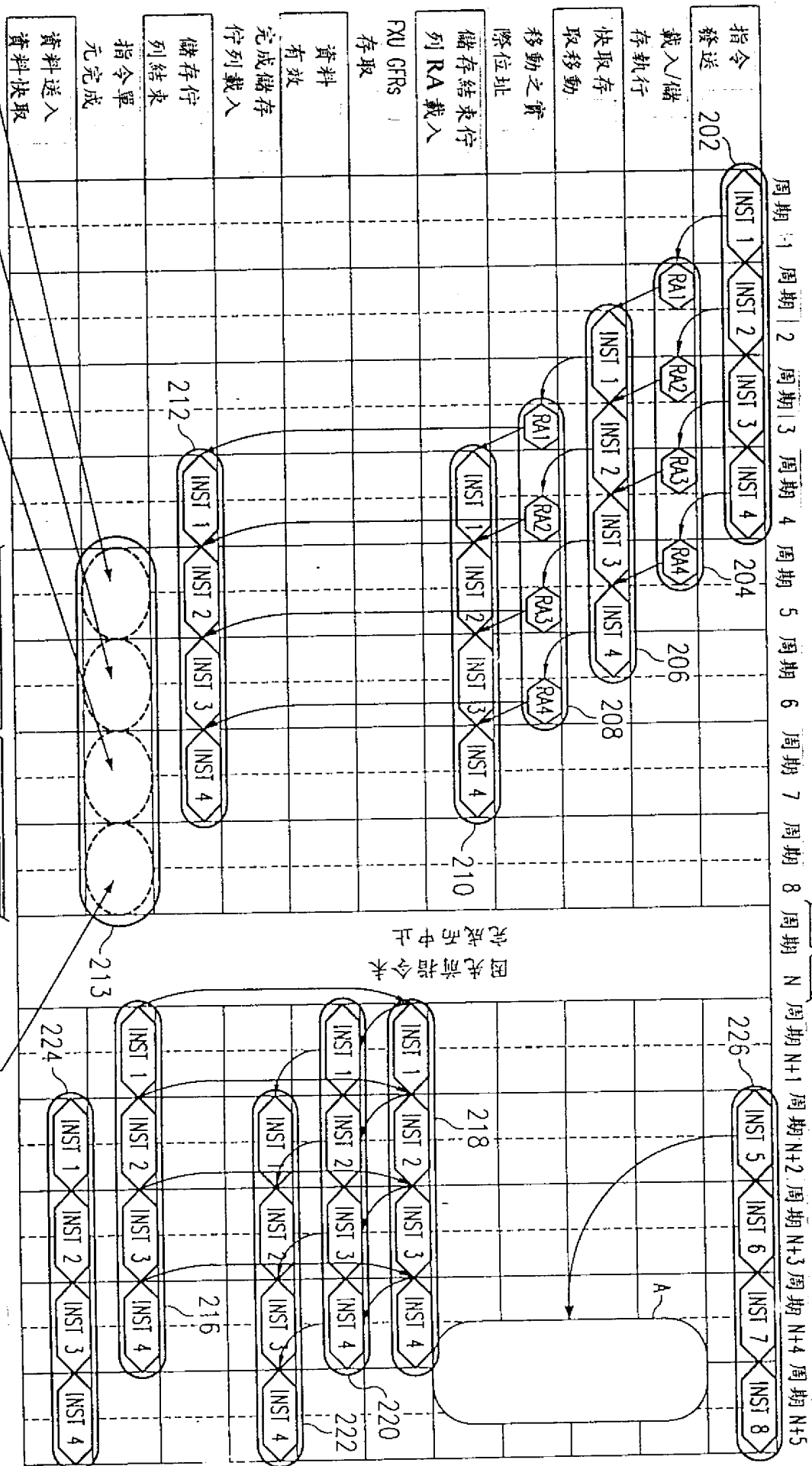


圖 3

(先前技術)

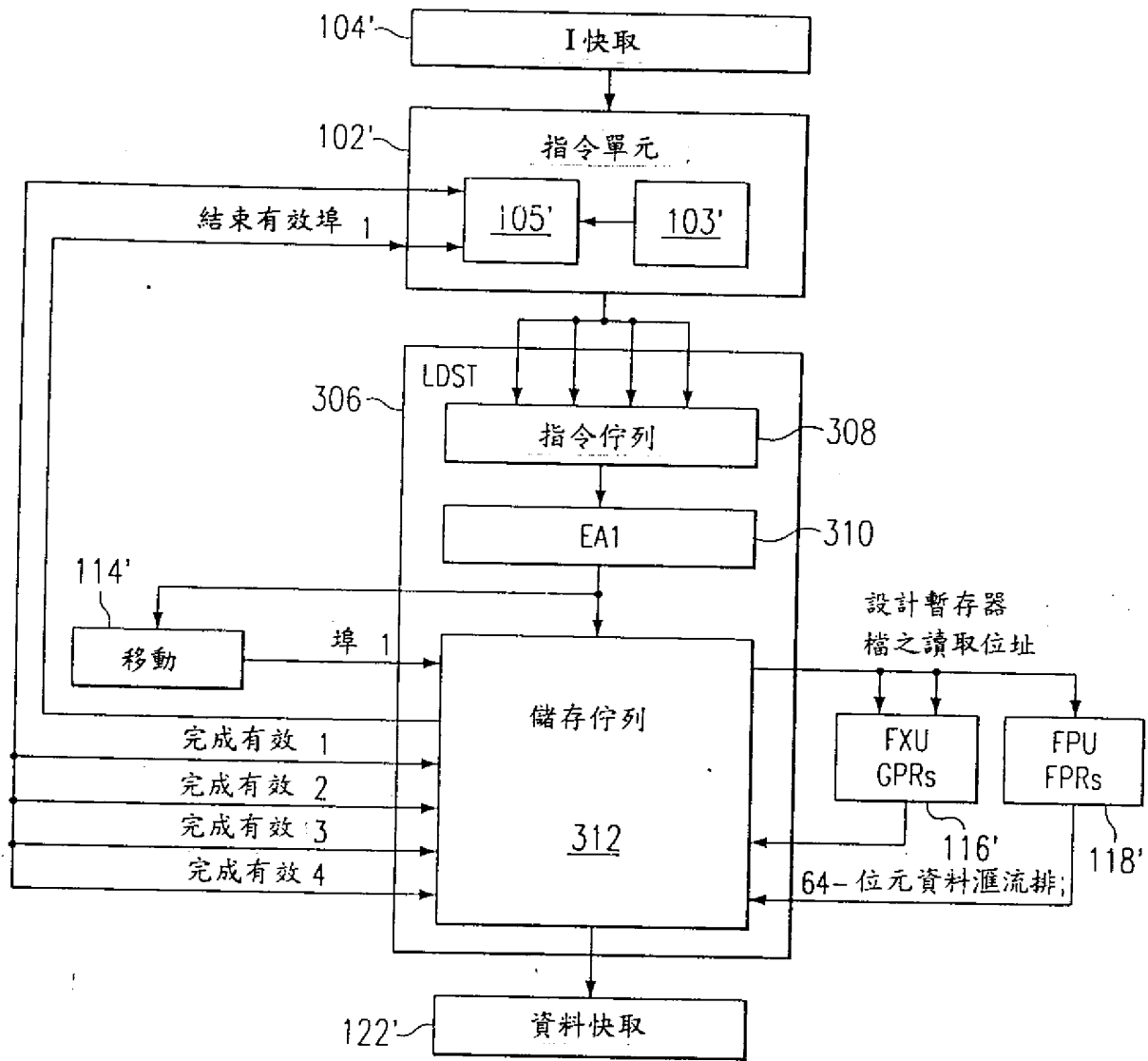


圖 4

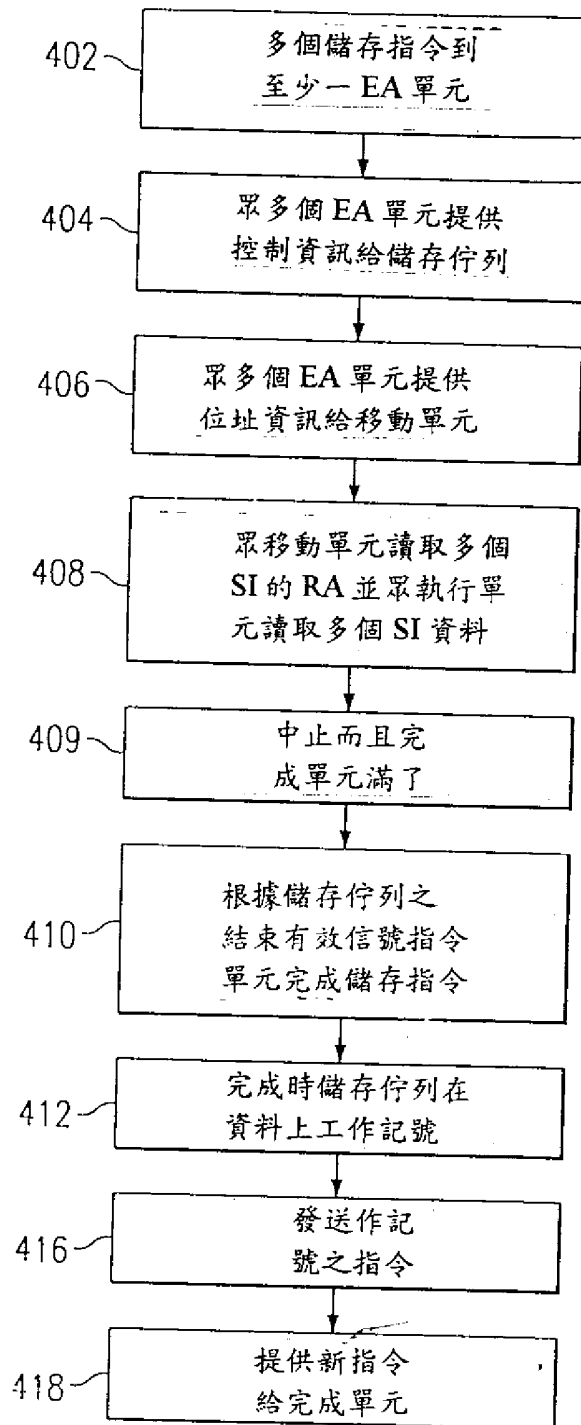


圖 5

儲存時序圖(每一周期有單一儲存執行)

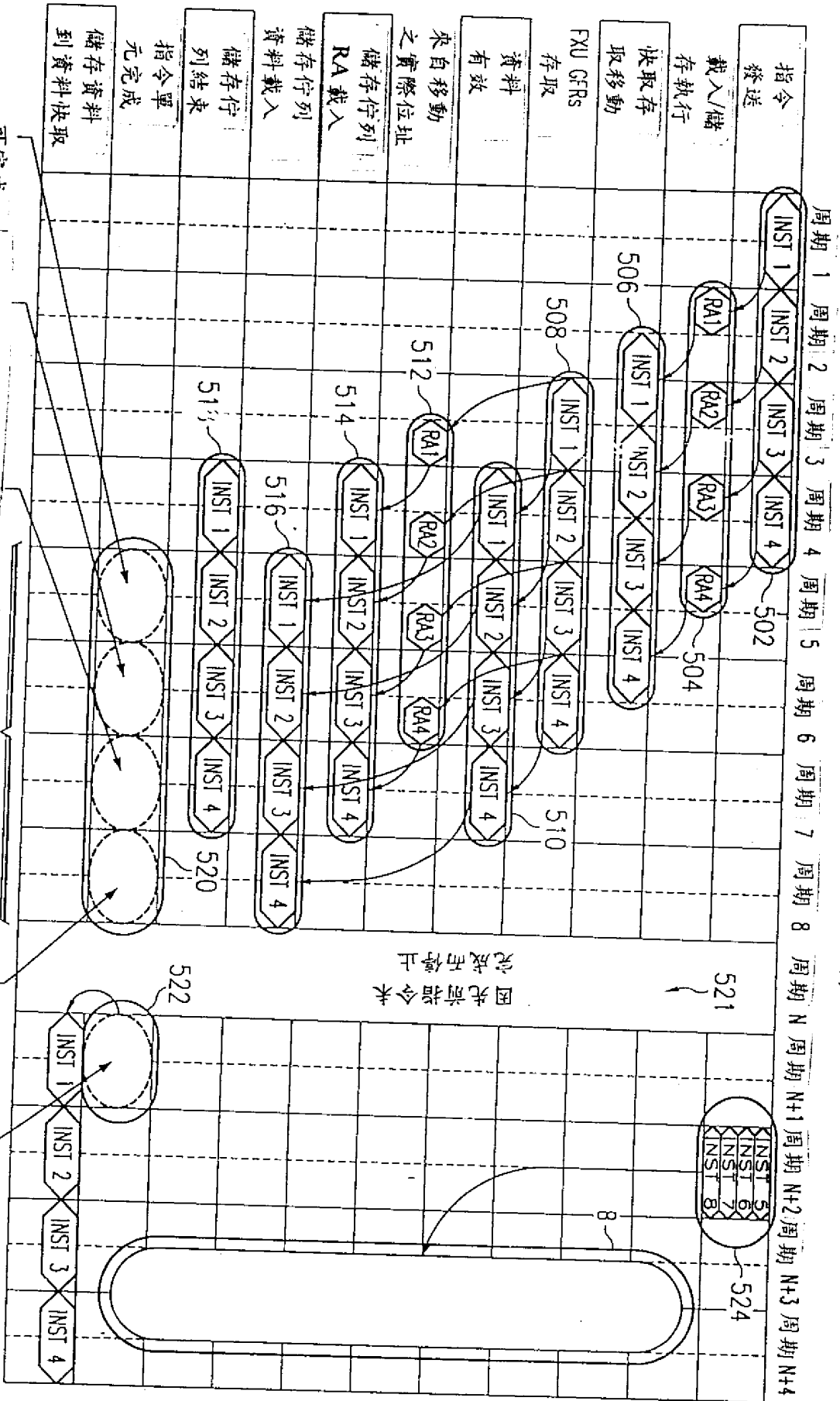


圖 6

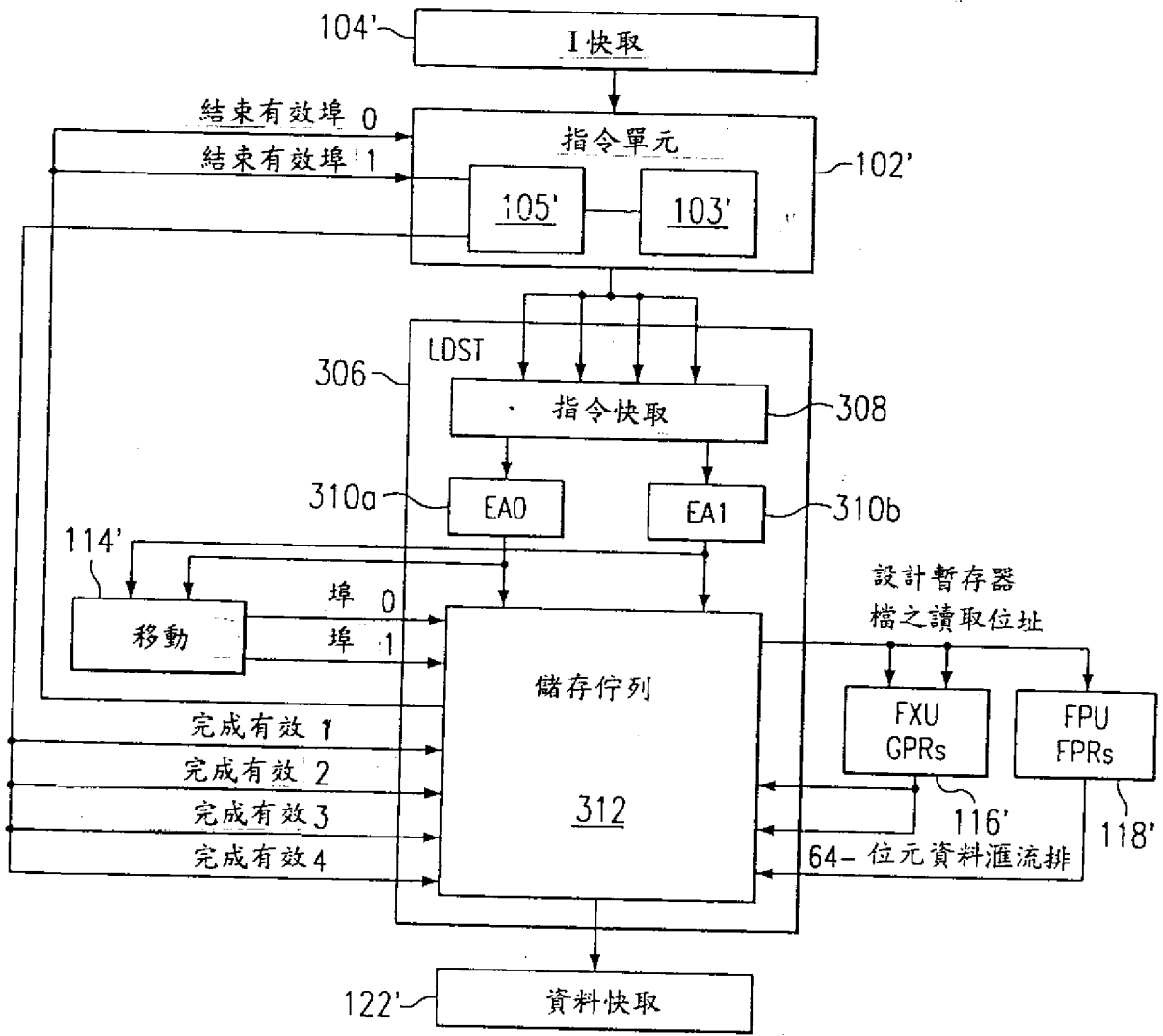


圖 7

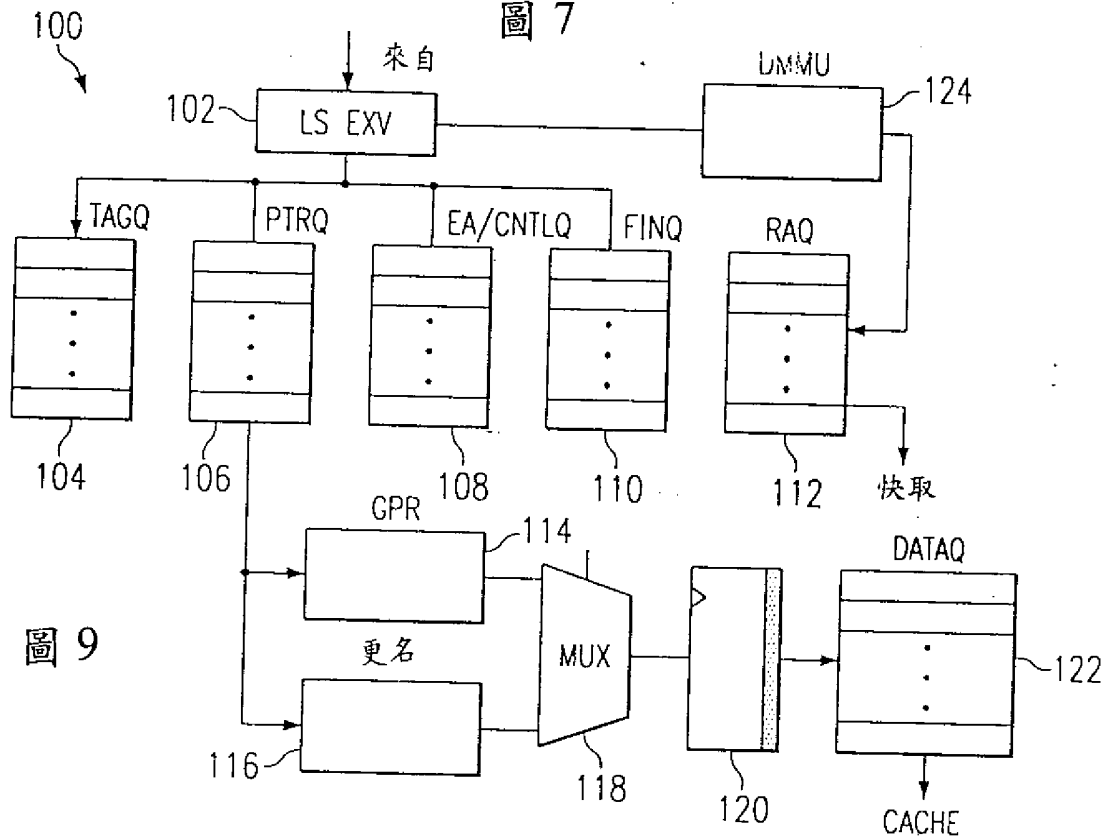


圖 9

儲存時序圖(每一周期支援 2 個儲存執行)

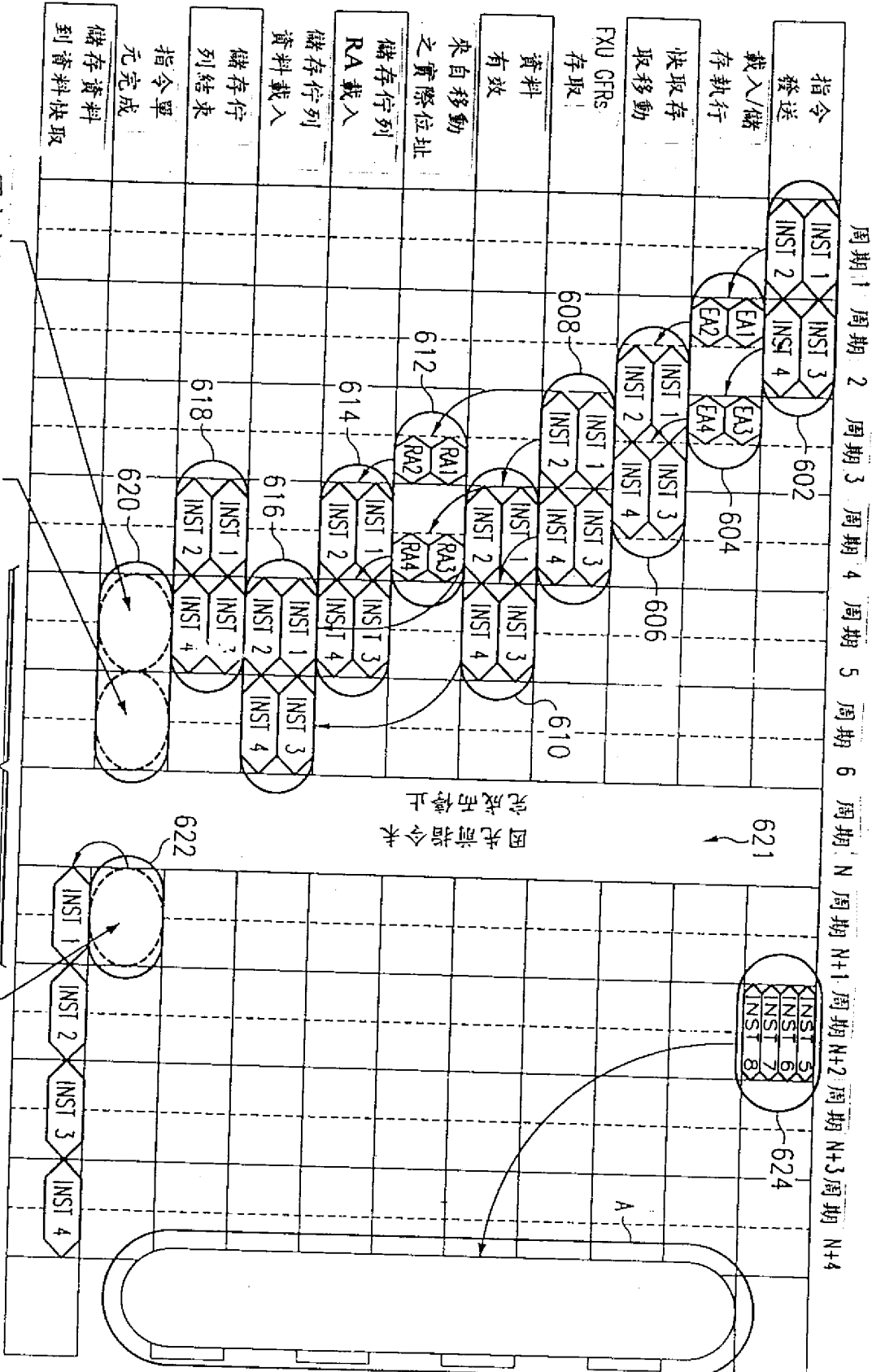


圖 8

資料佇列位置



圖 10F

標籤佇列位置

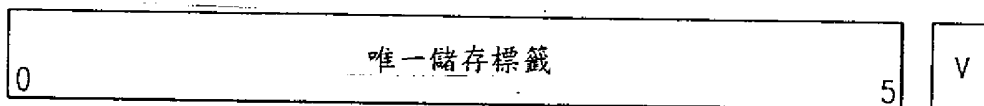


圖 10A

儲存指標佇列位置

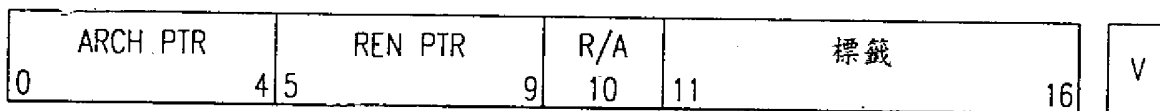


圖 10B

結束佇列位置

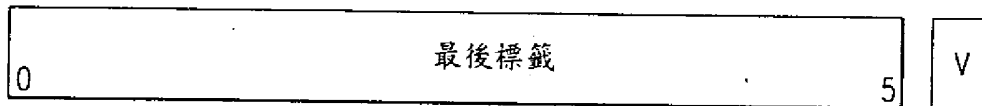


圖 10C

位址控制位置

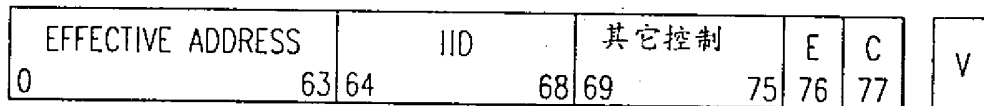


圖 10D

RA 佇列位置

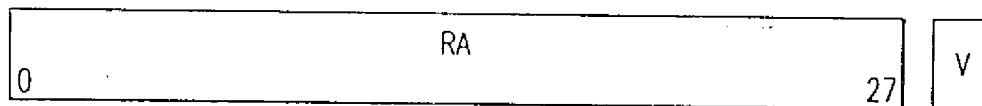


圖 10E

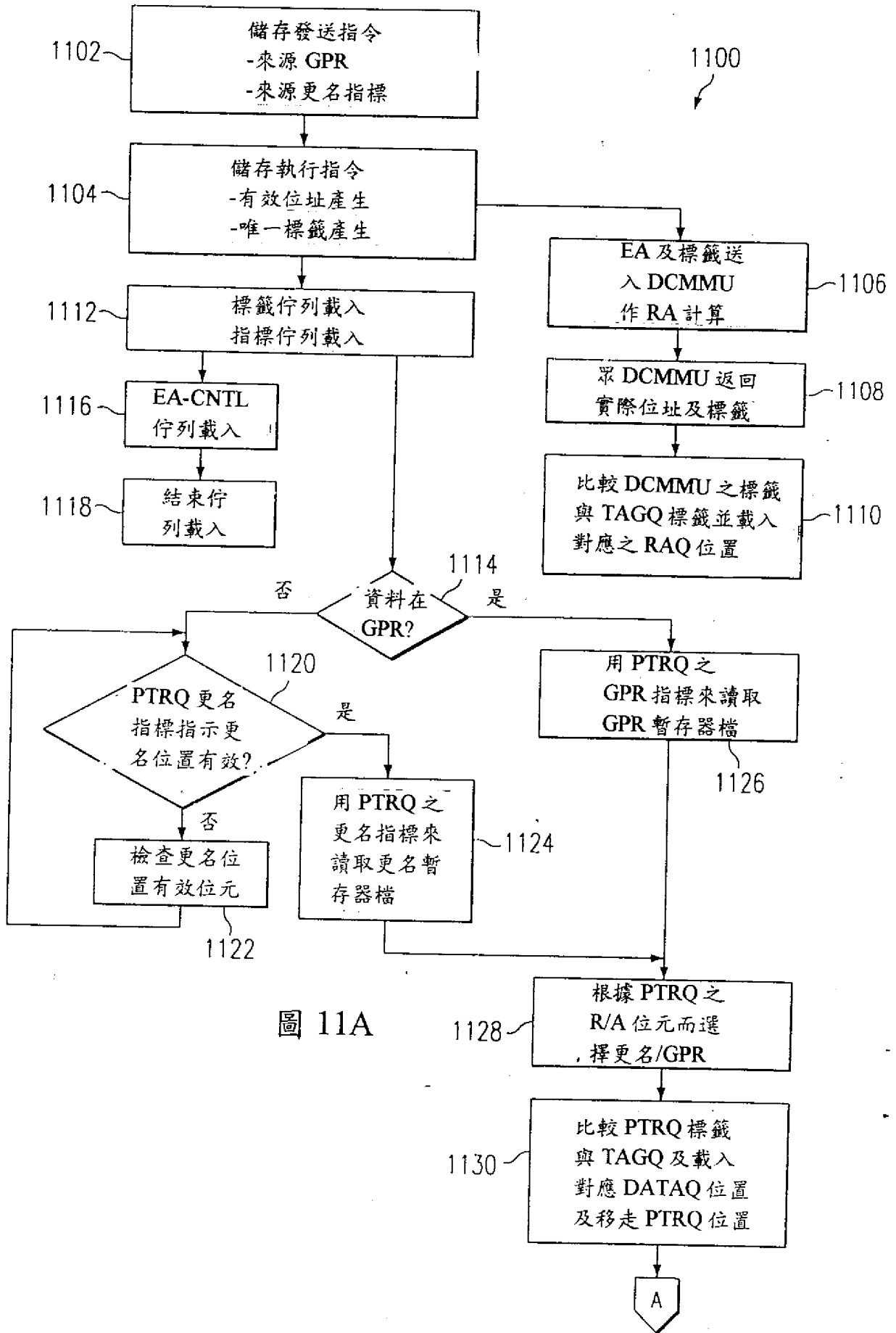


圖 11A

至圖 11B

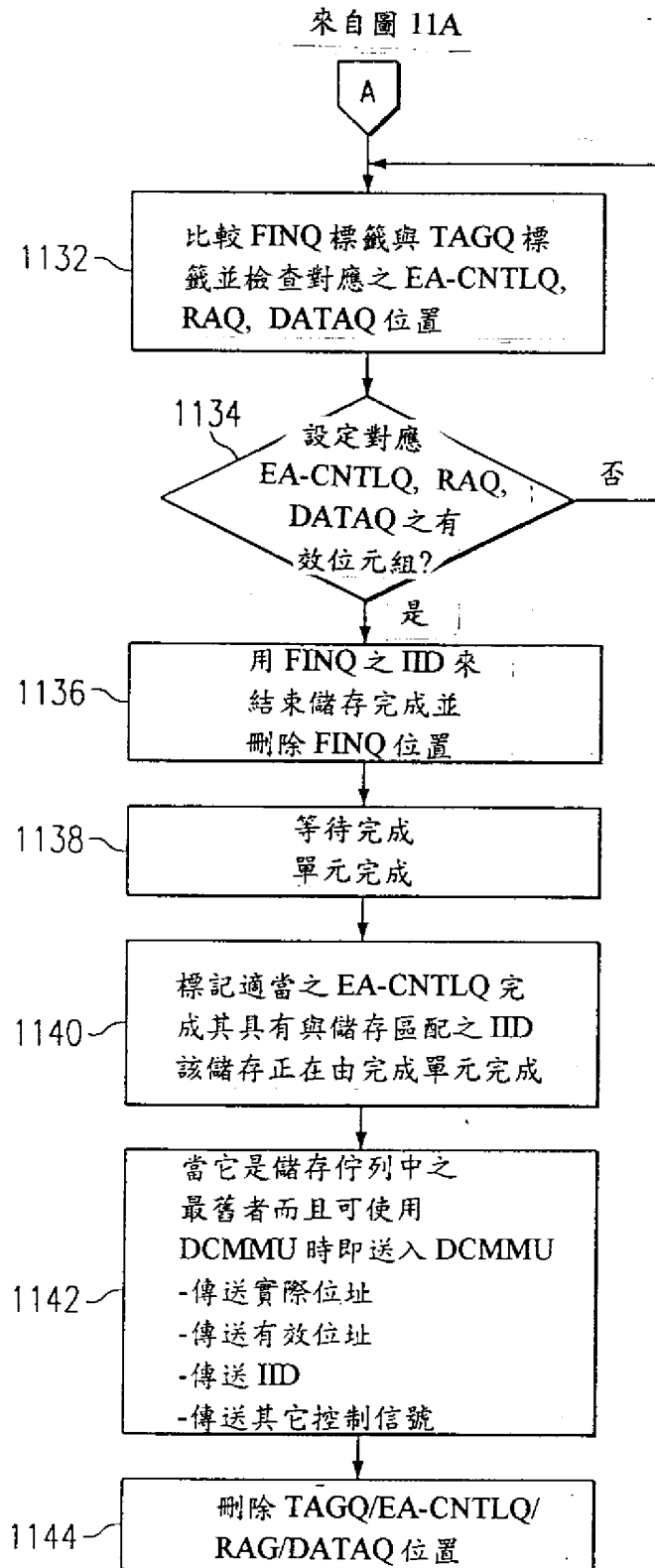


圖 11B

儲存時序圖

周期1 周期2 周期3 周期4 周期5 周期6 周期7 周期8

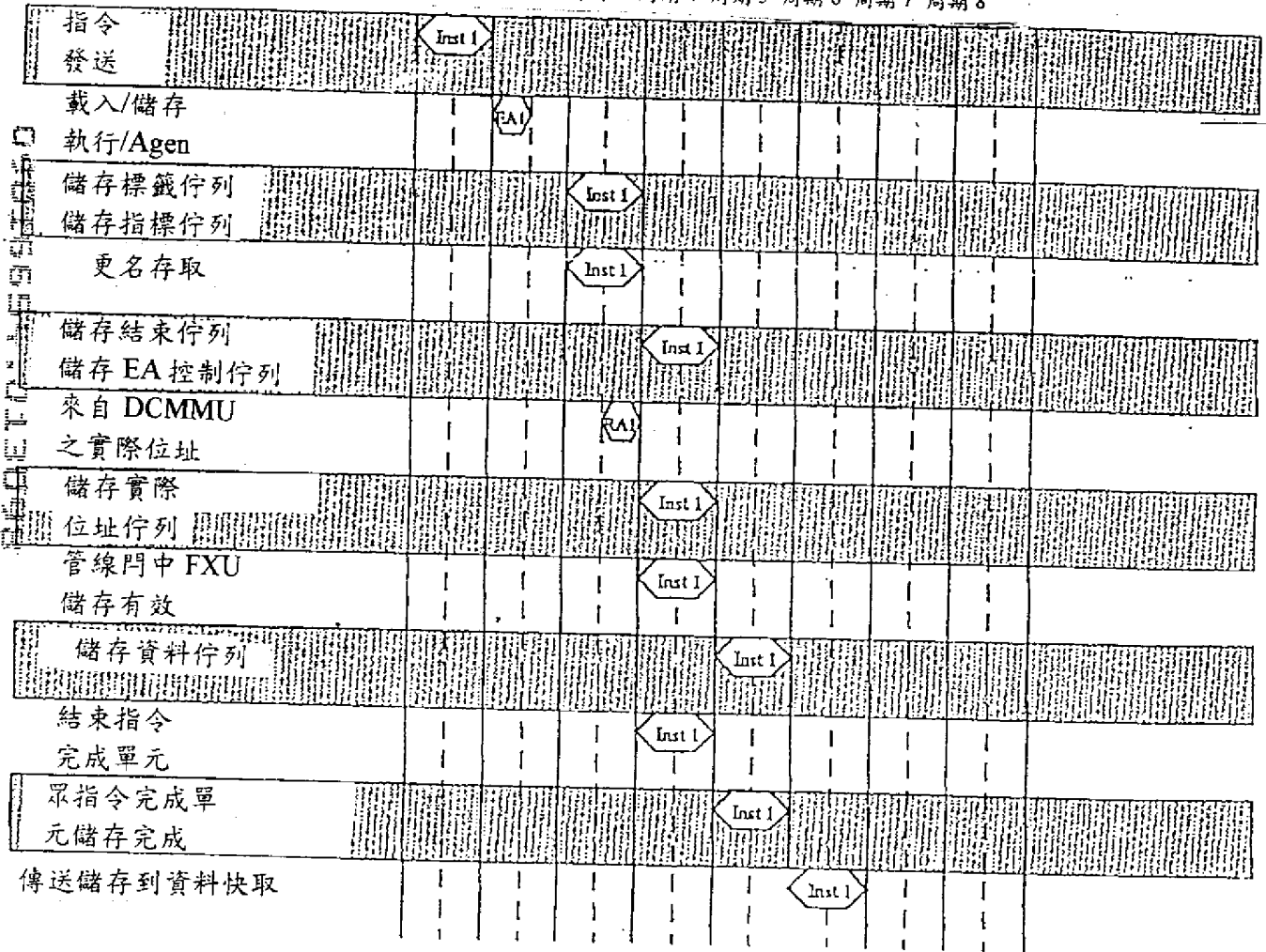


圖 12