

公告本

397953

申請日期	85.6.22
案 號	85107513
類 別	G06F 9/44

A4
C4

(以上各欄由本局填註)

397953

發明專利說明書

一、發明名稱	中 文	在高性能處理器中執行預測性載入指令之方法
	英 文	"METHOD FOR EXECUTING SPECULATIVE LOAD INSTRUCTIONS IN HIGH-PERFORMANCE PROCESSORS"
二、發明人	姓 名	1.亞伯特·J·洛波 2.索瑪亞·瑪利克 3.麥克·波特諾
	國 籍	1.3.美國 2.印度
	住、居所	1.美國德州希達派克市森林道1003號 2.美國德州奧斯丁市帕瑞吉班路13032號 3.美國德州奧斯丁市安瑞路8411號
三、申請人	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

(由本局填寫)

承辦人代碼：
大類：
I P C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 1996.2.6 08/597647

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

線

有關微生物已寄存於： ，寄存日期： ，寄存號碼：

五、發明說明(1)

背景

1. 技術領域

本發明通常與用來改善資料處理的一種方法有關；而且，特別是與在一處理器內，用來執行諸多推測性載入指令有關。又更特別的是，本發明與在一高性能處理器內，用來選擇性執行諸多推測性載入指令的一種方法有關；該方法使得：在一資料高速存取儲存器中之有用資料的錯誤位移能夠避免。

2. 有關技藝的描述

一種高性能處理器，能藉由提取(fetching)及分派(dispatching)諸多指令而獲得高指令輸貫量(throughput)，此乃基於假設：正確地預測諸多分支路徑，並在不需等待完成執行諸多先前指令下就容許執行諸多指令。這便是一般習稱的"推測性執行"(speculative execution)。一般說來，當處理器具有在其它情況應該是閒置(idle)的一些資源時，它就會以推測方式來執行諸多指令，使得：在最低限度或不需成本下就可執行操作。若隨後的諸多事件(events)指示：推測性指令應該未被執行；則處理器只會放棄推測性指令所產生的任何結果。

只要處理器似乎是遵循著一種簡單的循序方法，諸如：在一"純量"(scalar)處理器中的那些方法；就可用推測方式來執行大多數的操作。然而，就某些應用而言，諸多推測性操作對於處理器性能會有嚴重損害。譬如說，在執行一分支指令後之一載入指令(或習稱為"推測性載入指令"，原

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明(2)

因是：在無法確知應該採取那個分支路徑下，就用推測方式來執行載入指令)的情形中，若被預測的執行路徑不正確，則相關於在指令流中之未決的(pending)推測性載入指令，就要因從系統匯流排中請求所要的資料而付出高延遲(時間)的代價。於許多應用中，若錯誤預測(mispredicted)分支路徑的速率夠高，則以推測方式存取系統匯流排(資料)的成本就會貴得離譜。再者，因為由錯誤預測所造成之一推測性載入指令的一項錯誤執行；所以，儲存於一資料高速存取儲存器中的"必要"(essential)資料，就可能因從系統匯流排中所獲得的某些"無關"(irrelevant)資料而位移。

因此，應有必要提供：利用一種較佳的預測方案，在一高性能處理器中，用來選擇性執行推測性載入指令的一種方法。

概要

因此，本發明之一目的是：提供針對資料處理的一種改良式方法。

本發明之另一目的是：提供在一處理器內，用來執行諸多推測性指令的一種改良式方法。

本發明之又一目的是：提供在一高性能處理器內，用來執行諸多推測性載入指令的一種改良式方法；該方法使得：在一資料高速存取儲存器中之有用資料的錯誤位移能夠避免。

披露有：用來選擇性執行諸多推測性載入指令的一種方法及處理器。處理器具具有：用來儲存諸多指令的許多緩衝

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(3)

器，以及用來執行諸多指令的多種執行單元。根據本發明，當遇上針對一推測性載入指令之資料不是儲存在一資料高速存取儲存器中時，就得讀取在相關於該特定推測性載入指令之一致能推測性載入表內之一位元，以便決定位元之一狀態。若相關位元係在第一狀態中；則從一系統匯流排中起求針對推測性載入指令之資料，並隨後暫停對推測性載入指令做進一步的執行，以便等待來自一分支處理單元的諸多控制信號。若相關位元係在第二狀態中；則立即暫停對推測性載入指令的執行，以便等待來自分支處理單元的諸多控制信號。若根據來自分支處理單元的諸多控制信號而有執行推測性載入指令；則在致能推測性載入表中的相關位元，其後就會被設定到第一狀態。然而，若為回應來自分支處理單元的諸多控制信號而沒有執行推測性載入指令；則在致能推測性載入表中的相關位元，其後就會被設定到第二狀態。這樣一來，就可避免因推測性載入指令的錯誤執行而導致在資料高速存取儲存器中之有用資料的位移。

在以下的詳細書面描述中，本發明的所有目的，特點，及優點都將呈現眼前。

附圖概述

藉由參考以下針對發明的闡述，當連同諸多附圖加以閱讀時，就可更加瞭解發明本身，及使用之一說明模式，另外一些目的，以及其中的諸多優點；其中：

圖1係利用本發明方法之一處理器的一種說明實施例之方

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(4)

塊圖：

圖2A係根據本發明之一說明實施例的一種致能推測性載入表之方塊圖；

圖2B係根據本發明之一說明實施例，用來圖解說明：針對一32位元載入指令，指向一個1×256致能推測性載入表的一種索引操作方案(indexing scheme)之一方塊圖；及

圖3係根據本發明之一說明實施例，用來選擇性執行諸多推測性載入指令的一種方法之一高階邏輯流程圖。

一說明實施例的闡述

可將本發明建構於具有一種"連通管式架構"(pipelined architecture)之種種的處理器上。爲了圖解說明，將如下所述之一說明實施例建構於由美國國際商用機器公司(IBM)所製造的諸多微處理器之"威力電腦"(Power PC™)族群中之一成員上，該成員(機種)係根據諸多"精簡指令集計算"(reduced instruction set computing, RISC)技術而操作的。

現在參考諸多附圖，特別是參考圖1，圖解說明有：針對資料處理的一種處理器10之一方塊圖。在本發明之一說明實施例中，處理器10係單一積體電路超純量微處理器(superscalar microprocessor)，包括多種的：暫存器，緩衝器，執行單元，及功能單元。如圖1中所繪示的，經由在處理器10內之一匯流排界面單元(BIU)12，處理器被耦合到系統匯流排11。BIU 12控制在處理器10與被耦合到系統匯流排11的諸多其它裝置，諸如：一主記憶體(未示出)之間的資訊轉移。處理器10，系統匯流排11，以及耦合到系

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(5)

統匯流排11的諸多其它裝置一起形成一主資料處理系統。

在處理器10內，將BIU 12連接到指令高速存取儲存器14及資料高速存取儲存器16。指令高速存取儲存器14及資料高速存取儲存器16均為高速存取儲存器，它們使處理器10能獲得相對快的存取時間，而所要存取的是：先前已從主記憶體轉移到兩種高速存取儲存器14及16的諸多指令或資料之一子集合；於是，改善主資料處理系統的操作速度。將指令高速存取儲存器14進一步耦合到循序提取器(sequential fetcher) 17，在每一執行周期期間，它提取來自指令高速存取儲存器14的諸多指令。循序提取器17，雖將從指令高速存取儲存器14中所提取的諸多分支指令傳送到分支處理單元(BPU)18，以供執行之用；卻將諸多循序指令暫時儲存在指令佇列單元(instruction queue)19內，以供在處理器10內之其它執行電路的稍後執行之用。

除了BPU 18外，處理器10的執行電路包括三種執行單元，即：定點單元(fixed-point unit, FXU)22，載入／儲存單元(load/store unit, LSU)28，及浮點單元(floating-point unit, FPU)32。耐些熟習於諸多電腦技藝者皆知：三種執行單元中的每一種均能執行一個或更多種類的指令；而且，在每一處理器周期期間，所有的執行單元都會同時地操作。譬如說，FXU 22利用從諸多特定通用暫存器(GPR_S)24或GPR重新命名緩衝器(rename buffer) 25中所收到的一些原始運算元(source operands)，執行一些定點數學運算，諸如：加法，減法，及閘操作(ANDing)，或閘操作(ORing)

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(6)

，及互斥或閘操作(XORing)。繼一定點指令的執行之後，FXU 22將指令的諸多資料結果輸出到諸多GPR重新命名緩衝器25，它們是：在將一些資料結果從諸多GPR重新命名緩衝器25轉移到至少是諸多GPR₅24中的一個之前，為一些資料結果提供暫時的儲存，直到指令完成為止。相反地，FPU 32根據從諸多浮點暫存器(FPR₅)34或FPR重新命名緩衝器35中所收到的一些原始運算元，執行諸多浮點運算，諸如：浮點乘法及除法。FPU 32將由諸多浮點指令之執行所致的資料輸出到經選定的一些FPR重新命名緩衝器35，它們是：在將由諸多FPR重新命名緩衝器35所致的資料轉移到經選定的一些FPR 34之前，暫時儲存資料結果，直到諸多指令完成為止。顧名思義，LSU 28執行浮點及定點指令，它不是載入：來自記憶體(即：資料高速存取儲存器16或主記憶體)的資料到經選定的諸多GPR₅24或FPR₅34中；就是儲存：從諸多GPR₅24，GPR重新命名緩衝器25，FPR₅34，或FPR重新命名緩衝器35中經選定的一個到記憶體的資料。

處理器10對諸多指令採取"連通管"式及"不按順序進行"(out-of-order)式執行，以便進一步改善其超純量架構的性能。於是，只要能觀察資料相依性(dependencies)，就可按任何順序，由FXU 22，LSU 28，及FPU 32加以執行諸多指令。此外，按一序列的連通管階段(stages)，由FXU 22，LSU 28，及FPU 32中的每一個加以處理諸多指令。與任何高性能處理器相似，縱然是在某些設計建構例中，可

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (7)

將某些連通管階段加以減縮或合併；處理器10仍將按五個分別連通管階段：提取，解碼／分派，執行，結束(finish)及完成／返寫(completion/writeback)順序，加以處理每一指令。

在提取階段期間，循序提取器17取還(retrieve)與來自指令高速存取儲存器14的一個或更多記憶體位址相關的一個或更多指令。藉由循序提取器17，將從指令高速存取儲存器14中所提取的諸多循序指令儲存在指令佇列單元19內。在相對照下，循序提取器17將來自指令流中的諸多分支指令加以移除，並將它們加以轉遞(forwards)到BPU 18，以供執行之用。BPU 18包括一支預測機構，在一說明實施例中，它包括諸如：一"分支操作歷史表"(branch history table)之一動態預測機構，該機構使BPU 18能夠藉著預測是否將採取分支操作，而以推測方法來執行諸多未決定的(unresolved)條件分支指令。

在解碼／分派階段期間，分派單元20將來自指令佇列單元19的一個或更多指令加以解碼，並分派到三種適當執行單元：22，28，及32中的一種。並且，在解碼／分派階段期間，分派單元20為每一已分派指令的資料結果，配置(allocates)一個重新命名緩充器在諸多GPR重新命名緩衝器25或FPR重新命名緩衝器35內。

在執行階段期間，當針對諸多已指示運算的一些運算元及執行資源均可供利用時，三種執行單元：22，28，及32就會以帶投機性方式加以執行從分派單元20所收到的一

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(8)

些指令。三種執行單元：22，28，及32中的每一種，均配備有一"保留表"(reservation table)；該表儲存：一些運算元及執行資源都不能利用的諸多已分派指令。

在執行操作已經終結之後，三種執行單元：22，28，及32就會將一些資料結果儲存在：諸多GPR重新命名緩衝器25或FPR重新命名緩衝器35內，端視指令型式而定。然而，三種執行單元：22，28，及32發訊告知"完成單元"40：執行單元已經結束一指令。最後，藉著將資料結果從諸多GPR重新命名緩衝器25或FPR重新命名緩衝器35分別轉移到諸多GPR₅24或FPR₅34，而按程式順序加以完成諸多指令。

現在參考圖2A，繪示有：根據本發明之一說明實施例，用來促進(facilitating)諸多推測性載入指令之選擇性執行操作的一種"致能推測性載入"(ESL)表之一方塊圖。ESL表60係一小的記憶體位置：最好是在處理器10內，且由一載入指令位址之一較低部份加以索引。如圖2A中所示，ESL表60是：1位元寬及n位元長，其中n係用於索引操作之一載入指令的一些最低有效位元(LSB₅)數。當作是本發明之一說明實施例，用於ESL表的索引操作之一載入指令的LSB₅數n雖然是8；然而，n可以是：被認為適於一特定設計的任何整數。如圖2B中所示，根據本發明的目的，可將一32位元載入指令62的後8個位元：24到31用於索引操作。因為就8個位元而言會有256種排列(即： $2^8=256$)；所以，ESL表64的長度也是256。

五、發明說明(9)

在一說明實施例下的索引操作方案是有點率直的：一載入指令之諸LSB₅的十進制數值就會指到在一ESL表內的(相關)位元位置。譬如說：當載入指令62的8個LSB是"0000 0000"時，載入指令62就會指到ESL表64內的"第零"位元；當載入指令62的8個LSB是"0000 0001"時，載入指令62就會指到ESL表64內的"第一"位元；當載入指令62的8個LSB是"0000 0010"時，載入指令62就會指到ESL表64內的"第二"位元；...等等。

ESL表64的每一位元指示：具有相關之諸索引操作LSB之一載入指令，是否在一分支指令之後就已經先前被執行。當作是本發明之一說明實施例，縱然是一種反向指定也可接受；仍將以"0"位元表示"未被執行"，而以"1"位元表示"已被執行"。當處理器10最初被預定(initialized)時，在ESL表64中的全部位元應該都是零。如以下所述的，隨著時間的進展及採取在分支指令之後的諸多不同載入指令；於是，在ESL表64內的某些位元就會被修改。

現在參考圖3，繪示有：根據本發明之一說明實施例，用來執行諸多推測性指令的一種方法之一高階邏輯流程圖。在方塊50處開始，從分派單元20分派一載入指令。在方塊52處，計算出針對載入指令之資料的有效位址。根據計算出的有效位址，然後在方塊54處做一決定，有關：針對載入指令的資料現在是否儲存於資料高速存取儲存器16中。若資料現在是儲存於資料高速存取儲存器16中；則然後在方塊56處，從資料高速存取儲存器16中獲得資料，進而在

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明(10)

方塊99處，處理流程離開，使得：不論載入指令是否推測性，皆能執行載入指令。這是因為：即使它是一個推測性載入指令，也不會為錯誤預測的執行而付出代價。

然而，若資料現在不是儲存於資料高速存取儲存器16中，則在方塊58處必須做另一決定，有關：載入指令是否推測性。若載入指令不是推測性；則處理流程無從選擇，只得在方塊60處，從系統匯流排11中請求所要的資料，然後在方塊99處離開。在其它方面，若載入指令是推測性，則在方塊62處，處理流程會讀取ESL表。在方塊64處，隨後做一決定，有關：所要的資料是否應該從系統匯流排11中獲得。根據以上所述的索引操作方案，在相關於推測性載入指令的相關位置處，藉著讀取在ESL表內之一位元而執行這種決定，以便決定位元之一狀態。當作是在方塊64處所示之一實例，將32位元載入指令的後8個LSB(即：有效位址[24:31])用於索引操作。

若在ESL表內之相關位元在第一狀態中，該位元係由推測性載入指令加以索引的；則在方塊66處，從系統匯流排11中請求資料。當作是本發明之一說明實施例，第一狀態係一邏輯"1"。隨後，在方塊68處，暫停推測性載入指令的執行，以等待來自分支處理單元(BPU)18的某些控制信號，以便決定：推測性載入指令的執行是否應該繼續進行。在方塊70處，從BPU 18中獲得這種決定的結果。若根據來自BPU 18之諸控制信號的指示是：取消載入指令的執行(注意：這時載入指令不再是推測性)；則在方塊72處，藉

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明⁽¹¹⁾

著將相關位元從"1"反向成"0"，根據本發明之一說明實施例，將ESL表內的相關位元加以更新；進而在方塊99處，處理流程離開。在其它方面，若根據來自BPU 18之諸控制信號的指示是：清除載入指令的執行；則在方塊99處，處理流程只會離開。

若在ESL表內之相關位元係在第二狀態中，該位元係由推測性載入指令加以索引的；則在方塊74處，立即暫停推測性載入指令的執行，以等待來自BPU 18的諸多控制信號，以便決定：推測性載入指令的執行是否應該繼續進行。當作是本發明之一說明實施例，第二狀態係一邏輯"0"。在方塊76處，從BPU 18中獲得這種決定的結果。若根據來自BPU 18之諸控制信號的指示是：清除載入指令的執行(再者，這時載入指令不再是推測性)；則在方塊78處，從系統匯流排11中請求所要的資料。隨後，在方塊80處，藉著將相關位元從"0"反向成"1"，根據本發明之一說明實施例，將ESL表內的相關位元加以更新；進而在方塊99處，處理流程離開。在其它方面，若根據來自BPU 18之諸控制信號的指示是：取消載入指令的執行；則再在方塊99處，處理流程只會離開。

這時，熟習於處理器設計技藝者可能猜想到：在甚至是單一軟體程式內，也會有相當多具有相同之諸LSB的載入指令；並且，應該不知道：在ESL表內，具有諸多已匹配LSB的推測性載入指令是否正是已先前索引之相同的載入指令。換成另一方式，可能已經由具有相同之諸LSB的另

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(12)

一載入指令加以索引在ESL表中的位元。然而，這並不是一項主要的顧慮，原因是：就本發明的目的而言，將這種或許是有點罕見的錯誤發生視為可接受的。若因"錯誤索引操作"(mis-indexing)而導致推測性載入指令的錯誤執行，開始會影響到處理器的性能；則藉由增加用於索引操作的諸多LSB數，就可提高ESL表的解析度(resolution)。

如已描述的，本發明提供：在一高性能處理器內，用來選擇性執行諸多推測性載入指令的一種方法。藉由一種ESL表的利用，本發明提供一種唯一的方法來決定：當在資料高速存取儲存器中不能找到推測性載入指令時，就是否應向系統匯流排提出一項資料請求，以供一推測性載入指令的執行之用。

相對於一說明實施例，雖然已經特別加以顯示並描述本發明；但那些熟習於此技藝者應瞭解的是：在不背離本發明的精神及範圍的前提下，可以在形式及細節上做多種改變。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (12a)

圖式元件符號說明

- | | |
|----|-----------------------|
| 10 | 處 理 器 |
| 11 | 系 統 匯 流 排 |
| 12 | 匯 流 排 界 面 單 元 |
| 14 | 指 令 高 速 存 取 儲 存 器 |
| 16 | 資 料 高 速 存 取 儲 存 器 |
| 17 | 循 序 提 取 器 |
| 18 | 分 支 處 理 單 元 |
| 19 | 指 令 佇 列 單 元 |
| 20 | 分 派 單 元 |
| 22 | 定 點 單 元 |
| 24 | 通 用 暫 存 器 |
| 25 | 諸 多 GPR 重 新 命 名 緩 衝 器 |
| 28 | 載 入 / 儲 存 單 元 |
| 32 | 浮 點 單 元 |
| 34 | 浮 點 暫 存 器 |
| 35 | 諸 多 GPR 重 新 命 名 緩 衝 器 |
| 40 | 完 成 單 元 |
| 60 | ESL 表 |
| 62 | 載 入 指 令 |
| 64 | ESL 表 |

(請先閱讀背面之注意事項再填寫本頁)

訂

四、中文發明摘要(發明之名稱：

在高性能處理器中執行預測性載入指令之方法)

所揭露的是在一高性能處理器中，用來選擇性執行諸多預測性載入指令的一種方法。根據本發明，當遇上令資料不要儲存在一資料高速存取儲存器(data cache)中的一預測性載入指令時，就得讀取有關於該特定預測性載入指令之一致能預測性載入表內之一位元，以便決定該位元之一狀態。若有關位元係在第一狀態中，則從一系統匯流排中請求供預測性載入指令用之資料，並隨後暫停對預測性載入指令做進一步的執行，以便等待來自一分支處理單元的諸多控制信號。若相關位元係在第二狀態中，則立即暫停對預測性載入指令的執行，以便等待來自分支處理單元的諸多控制信號。若為回應諸多控制信號而執行預測性載入指令時，則在致能預測性載入表中的相關位元就會被設定成

英文發明摘要(發明之名稱：

"METHOD FOR EXECUTING SPECULATIVE LOAD INSTRUCTIONS IN HIGH-PERFORMANCE PROCESSORS")

A method for selectively executing speculative load instructions in a high-performance processor is disclosed. In accordance with the present disclosure, when a speculative load instruction for which the data is not stored in a data cache is encountered, a bit within an enable speculative load table which is associated with that particular speculative load instruction is read in order to determine a state of the bit. If the associated bit is in a first state, data for the speculative load instruction is requested from a system bus and further execution of the speculative load instruction is then suspended to wait for control signals from a branch processing unit. If the associated bit is in a second state, the execution of the speculative load instruction is immediately suspended to wait for control signals from the branch processing unit. If the speculative load instruction is executed in response to the control signals, then the

四、中文發明摘要 (發明之名稱：)

第一狀態。然而，若為回應諸多控制信號而沒有執行推測性載入指令，則在致能推測性載入表中的相關位元就會被設定成第二狀態。這樣一來，就可避免因推測性載入指令的錯誤執行而導致資料高速存取儲存器中之有用資料的位移。

英文發明摘要 (發明之名稱：)

associated bit in the enable speculative load table will be set to the first state. However, if the speculative load instruction is not executed in response to the control signals, then the associated bit in the enable speculative load table is set to the second state. In this manner, the displacement of useful data in the data cache due to wrongful execution of the speculative load instruction is avoided.

六、申請專利範圍

1. 一種在一處理器內用來選擇性執行諸多推測性載入指令的方法，該方法包括以下步驟：

在致能推測性載入表中讀取有關於一推測性載入指令之一位元，以便決定其狀態，以回應遇上該推測性載入指令，其中推測性載入指令的資料並非儲存於一資料高速存取儲存器中；

回應該相關位元係在第一狀態中的一項決定，利用一系統匯流排來請求該用於推測性載入指令的資料，進而暫停推測性載入指令的執行，該指令暫停來自一分支處理單元的諸多控制信號；

回應該相關位元係在第二狀態中的一項決定，暫停該推測性載入指令的執行，該指令暫停來自分支處理單元的諸多控制信號；

將在該致能推測性載入表中的該相關位元設定成第一狀態，以回應根據諸多控制信號之推測性載入指令的一項執行作業；

將在該致能推測性載入表中的相關位元設定成第二狀態，以回應根據諸多控制信號之推測性載入指令的一項非執行作業，其中避免了因推測性載入指令的錯誤執行而導致在資料高速存取儲存器中之有用資料發生位移的可能性。

2. 根據申請專利範圍第1項之在一處理器內用來選擇性執行諸多推測性載入指令的方法，其中該方法還包括：在該處理器的初始化期間，將在該致能推測性載入表中的

(請先閱讀背面之注意事項再填寫本頁)

訂

六、申請專利範圍

全部位元都設定成第二狀態的步驟。

3. 根據申請專利範圍第1項之在一處理器內用來選擇性執行諸多推測性載入指令的方法，其中回應該相關位元係在第一狀態中的一項決定，還包括：回應該相關位元係一邏輯"1"的一項決定；而且，其中回應該相關位元係在第二狀態中的一項決定，還包括：回應該相關位元係一邏輯"0"的一項決定。

4. 一種用來選擇性執行諸多推測性載入指令的處理器，該處理器包括：

用來讀取在致能推測性載入表中之相關推測性載入指令之一位元，以便決定其狀態的裝置，以回應遇上該推測性載入指令及推測性載入指令的資料並非儲存於一資料高速存取儲存器中；

利用一系統匯流排來請求用於推測性載入指令的資料的裝置，進而暫停推測性載入指令的執行，並暫停來自一分支處理單元的諸多控制信號，以回應該相關位元係在第一狀態中的一項決定；

用來暫停推測性載入指令的執行的裝置，而暫停來自分支處理單元的諸多控制信號，以回應該相關位元係在第二狀態中的一項決定；

用來將在致能推測性載入表中的相關位元設定成第一狀態，以回應根據諸多控制信號之推測性載入指令的一項執行作業的裝置；

用來將在致能推測性載入表中的相關位元設定成第二

(請先閱讀背面之注意事項再填寫本頁)

訂

六、申請專利範圍

狀態，以回應根據諸多控制信號之推測性載入指令的一項非執行作業的裝置，其中避免了因推測性載入指令的錯誤執行而導致在資料高速存取儲存器中之有用資料發生位移的可能性。

5. 根據申請專利範圍第4項之用來選擇性執行諸多推測性載入指令的處理器，其中該處理器還包括：在處理器的初始化期間，用來將在致能推測性載入表中的全部位元都設定成第二狀態的裝置。
6. 根據申請專利範圍第4項之用來選擇性執行諸多推測性載入指令的處理器，其中第一狀態是邏輯"1"，而第二狀態則是邏輯"0"。
7. 根據申請專利範圍第4項之用來選擇性執行諸多推測性載入指令的處理器，其中致能推測性載入表是1位元寬，而且係由該推測性載入指令之位址的許多最低有效位元加以索引的。
8. 一種用來選擇性執行諸多推測性載入指令的資料處理系統，該資料處理系統包括：

一記憶體；

一處理器，包括：

用來讀取在一致能推測性載入表中相關推測性載入指令之一位元，以便決定其狀態的裝置，以回應遇上該推測性載入指令及推測性載入指令的資料並非儲存於一資料高速存取儲存器中；

用來利用一系統匯流排而請求供推測性載入指令用

(請先閱讀背面之注意事項再填寫本頁)

訂

六、申請專利範圍

的資料，進而暫停推測性載入指令的執行，並暫停來自一分支處理單元的諸多控制信號再做決定的裝置，用以回應該相關位元係在第一狀態中的一項決定；

用來暫停推測性載入指令的執行，並暫停來自分支處理單元的諸多控制信號的裝置，用以回應該相關位元係在第二狀態中的一項決定；

用來將致能推測性載入表中的相關位元設定成第一狀態之裝置，用以回應根據諸多控制信號之推測性載入指令的一項執行作業；

用來將致能推測性載入表中的相關位元設定成第二狀態的裝置，用以回應：根據諸多控制信號之推測性載入指令的一項非執行作業，其中避免了因推測性載入指令的錯誤執行而導致在資料高速存取儲存器中之有用資料發生位移的可能性。

9. 根據申請專利範圍第8項之用來選擇性執行諸多推測性載入指令的資料處理系統，其中該資料處理系統還包括：在處理器的初始化期間，用來將致能推測性載入表中的全部位元都設定成第二狀態的裝置。
10. 根據申請專利範圍第8項之用來選擇性執行諸多推測性載入指令的資料處理系統，其中第一狀態是邏輯"1"，而第二狀態則是邏輯"0"。
11. 根據申請專利範圍第8項之用來選擇性執行諸多推測性載入指令的資料處理系統，其中致能推測性載入表是1位元寬，而且係由該推測性載入指令之位址的許多最低有效位元加以索引的。

85107513

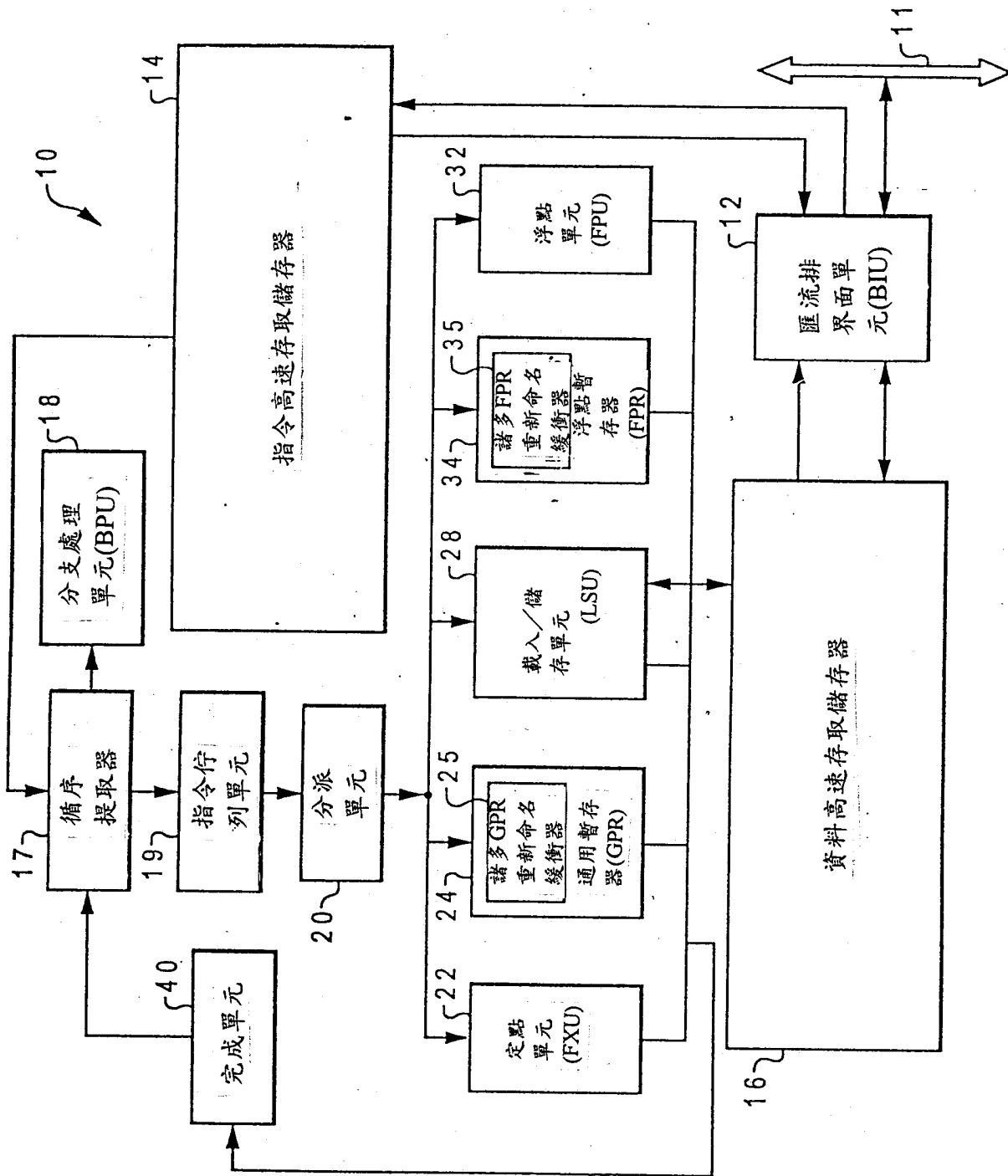


圖1

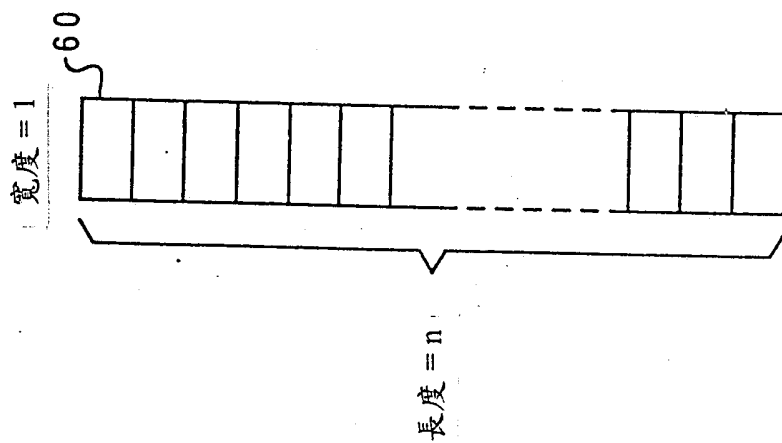


圖 2A

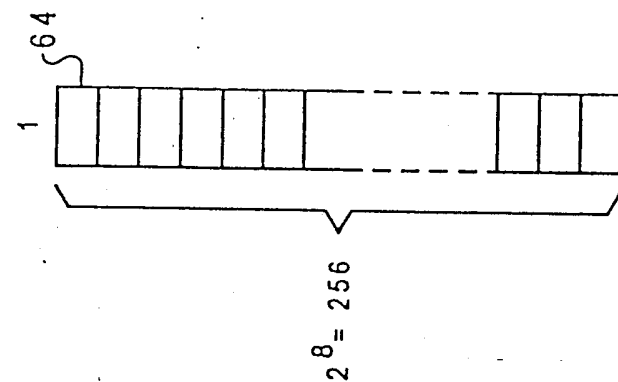
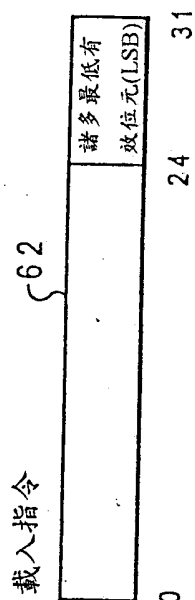
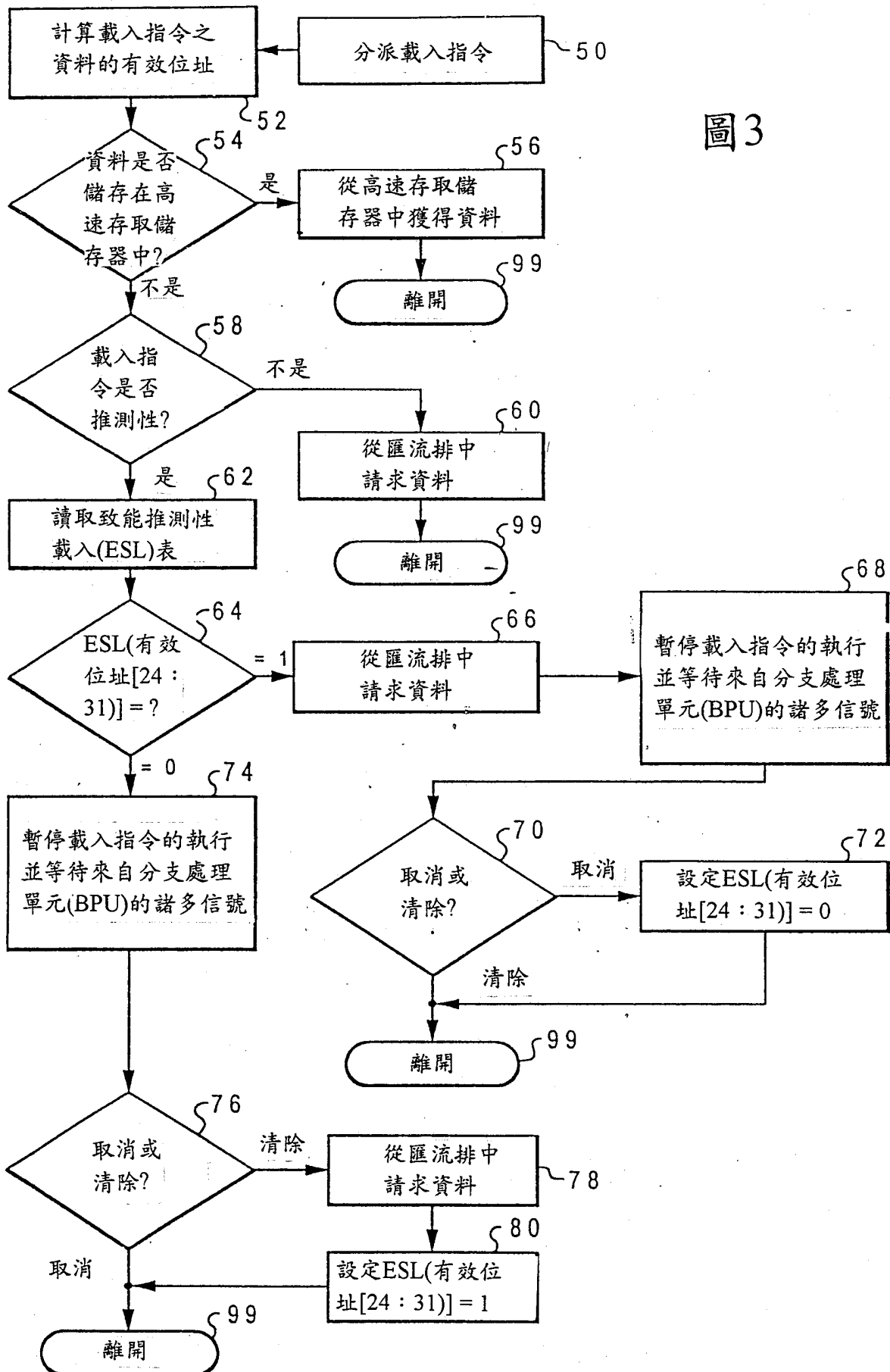


圖 2B

圖3



85107513

397953

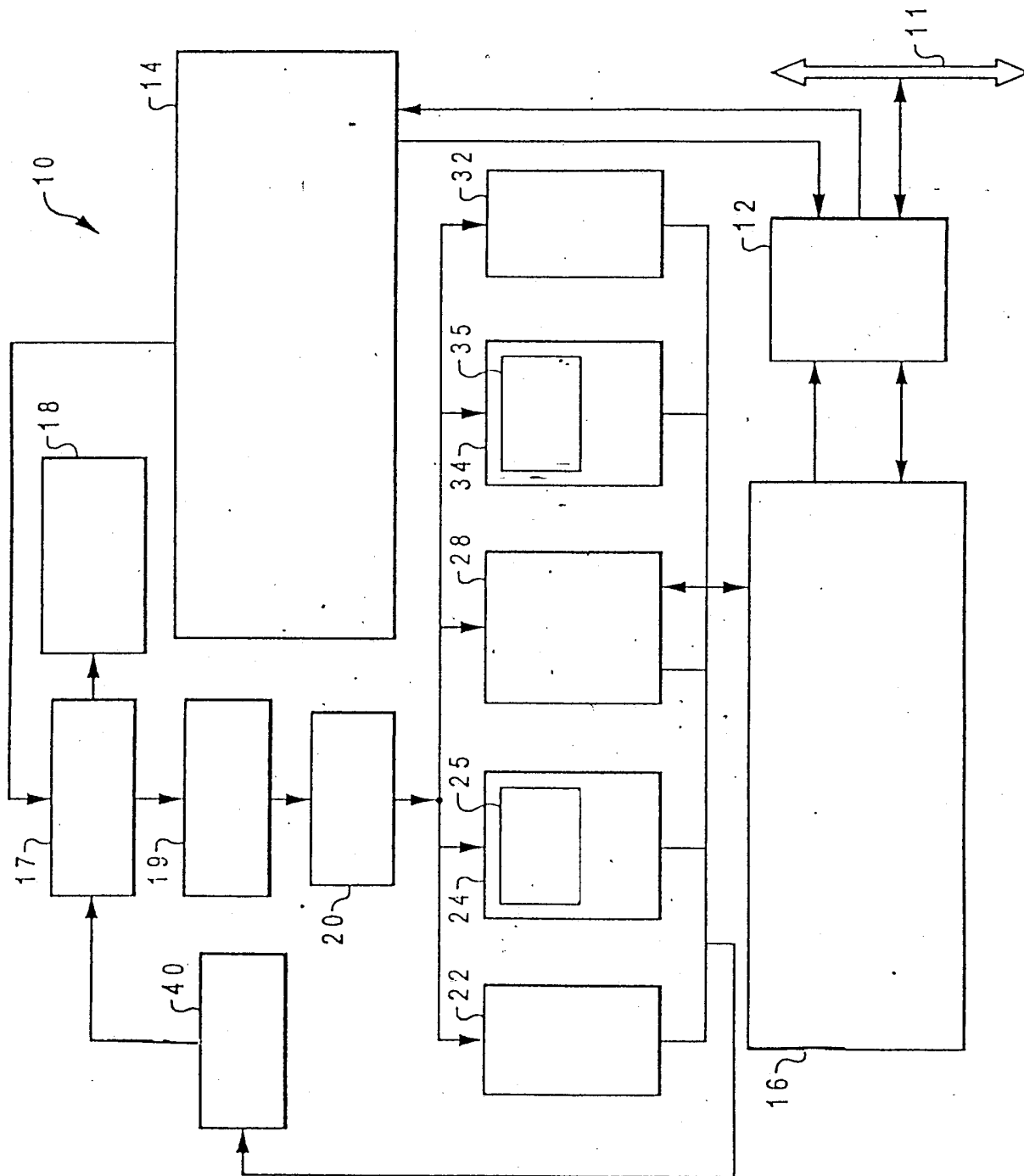


圖 1

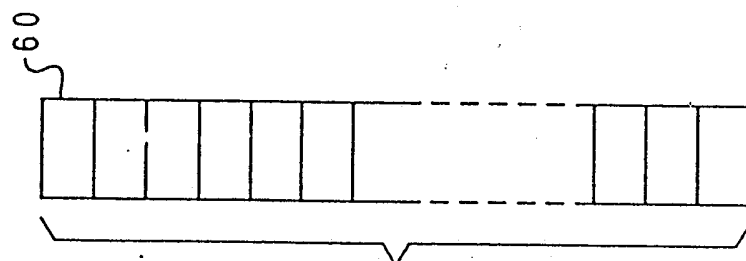


圖 2A

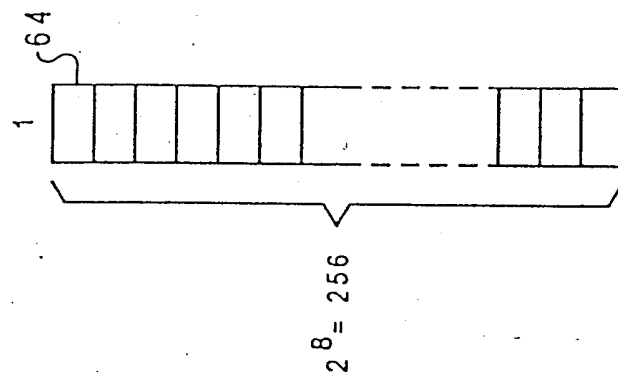
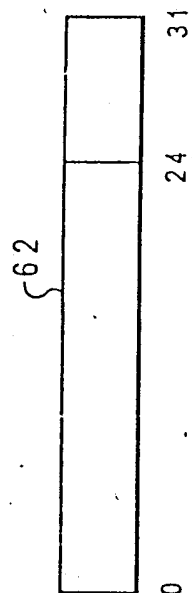


圖 2B

