

公告本

申請日期	89 年 9 月 30 日
案 號	89120405
類 別	G06F12/04, 9/315, 9/34

A4
C4

497034

(以上各欄由本局填註)

發明專利說明書

一、發明 名稱	中 文	讀入及儲存資料於電腦系統之方法
	英 文	A method for loading and storing data in a computer system
二、發明 人 創作	姓 名	(1) 大衛·薛佛爾 Shepherd, David E
	國 籍	(1) 英國
	住、居所	(1) 英國必治妥西伯利公園哈柯特路六號 6 Harcourt Road Westbury Part, Bristol, BS6 7RG, United Kingdom
三、申請人	姓 名 (名稱)	(1) 日立製作所股份有限公司 株式会社日立製作所
	國 籍	(1) 日本
	住、居所 (事務所)	(1) 日本國東京都千代田區神田駿河台四丁目六番地
	代 表 人 姓 名	(1) 庄山悅彦

裝
訂
線

經濟部智慧財產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 1999年10月1日 09/410,545 有主張優先權

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

經濟部智慧財產局員工消費合作社印製

五、發明說明(1)

發明之背景

本發明係概括關於微處理器或微控制器架構，尤指一種構造為處理未對準記憶體參考之架構。

在過去十年在電腦架構，在其中理想為在一單一操作循環進行每一指令之精簡指令集電腦（Reduced Instruction Set Computer，簡稱RISC）裝置，已變為受人歡迎。RISC架構具有優於具有標準架構及指令集之電腦之優點，因為其能遠為較高資料處理速度，由於其能在較短時間期間進行頻繁操作。RISC裝置以16-位元指令集開始，並成長至具有圖形能力之32-位元指令集架構。利用此等三十二位元指令集架構及更複雜應用，需要較大記憶體大小，例如長度二，四，或八位元組之字（亦即各為16，32，或64位元之字）。然而，某些周邊裝置及應用僅產生或接受一或二位元組之資料。此種類型資料之一項結果，為其產生一未對準字參考。其他實例包括某些壓縮資料流，其可能以需要取用未對準資料之方式合併資料。

要瞭解何為未對準字參考，需要說明對準字參考。如果一資料物件在位址A為N位元組大小，如果 $A \bmod N = 0$ ，則物件為對準。表1示資料之對準及未對準存取之實例，如果位元組偏置予以規定供低階三位元之位址（Computer Architecture A Quantitative Approach，John Hennessy and David Patterson，Morgan Kaufmann，Publishers，Inc.，Copyright 1990，96頁，在本文中稱作"Hennessy"）。

（請先閱讀背面之注意事項再填寫本頁）

訂
線

五、發明說明(2)

表 1

物件位址	藉位元組偏置	在位元組偏置
	對準	未對準
位元組(8-位元)	0, 1, 2, 3, 4, 5 , 6, 7	(決不)
字(16-位元)	0, 2, 4, 6	1, 3, 5, 7
長字(32-位元)	0, 4	1, 2, 3, 4, 5, 6 , 7
四字(64-位元)	0	1, 2, 3, 4, 5, 6 , 7

因而，供一能處理4位元組長字之機器，如果順序饋入進入資料作為資料之2位元組，後隨資料之再2位元組，便無法在一單一循環檢索或儲存資料之4位元組，因為其將會在記憶體內重疊一字邊界。因此，有些先前技藝 R I S C 裝置不是不接受在此形式之資料，在該情形必須使用特殊程序，以保證所有資料在字邊界對準，就是需要規劃程式，其使用至少二連續指令循環。保證例如所有資料在字邊界對準之一種方式，將為將額外位元加至較短長度，通常稱作位元填補之資料。不論使用位元填補或更改規劃程式，未對準參考均降低此等先前技藝 R I S C 裝置之性能。

要處理未對準字在一系統之饋入及儲存，亦即一在記憶體橫跨一字邊界之資料字(表1)，先前技藝機器也曾使用一在一字饋入或儲存位元組之對準網路，或一僅在需要

(請先閱讀背面之注意事項再填寫本頁)

訂 線

五、發明說明(3)

對準之情形使資料移位之移位器 (Hennessy , 出處同上 , 95 - 97頁) 。

圖 1 例示一先前技藝對準網路 114 。 在圖 1 中 , 記憶體 100 示八連續位元組 (亦即一位元組等於 8 位元) : Y3 , Y2 , Y1 , D4 , D3 , D2 , D1 , 及 X4 。 在記憶體 100 之每一位元組予以給定一範圍自 0 至 7 之位址 。 例如 , 在記憶體 100 , 位址 2 具有記憶體內容 Y1 。 在此及下列實例所使用之希望資料位元組 , 為在位址 3 之 D4 , 在位址 4 之 D3 , 在位址 5 之 D2 , 及在位址 6 之 D1 。 將每一此等希望之資料位元組饋入及儲存至暫存器 R 110 , 及自其饋入及儲存 。 暫存器 R 110 有 4 位元組位置 : P4 , P3 , P2 及 P1 。 記憶體 100 之記憶體切片 112 示一在位址 3 之希望資料位元組 D4 。 D4 可在位置 P4 , P3 , P2 , 或 P1 , 通過對準網路 114 自記憶體切片 112 饋入暫存器 R 115 。 在此情形 , D4 在位址 3 , 通過對準網路 114 , 自記憶體切片 112 饋入至在暫存器 R 115 之 P4 。 同樣 , 位於記憶體 100 位址 4 , 5 , 及 6 之希望資料位元組 D3 , D2 , 及 D1 , 可通過一類似對準網路 , 在暫存器 R 115 饋入至位置 P3 , P2 , 及 P1 , 以產生暫存器 R 110 。 此種類型之硬體對準網路 114 可見於 Intel 在 1970 年代後期推出之 8086 及 8088 。 Intel 8088 為字及位元組可定址 。 8088 使用一縱橫開關以交換位元組 (Structured Computer Organization , 3rd Edition , Andrew Tanenbaum , Copyright 1990 , 215 - 217 頁 , 230 - 237 頁) 。 請察知 , Intel 8088 指令集有供移位及旋轉之單獨指令 , 因為此等指令視為不同之操作 。 例如 , 左移位一

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明(4)

位元，將會捨棄最左位元，而左旋轉將會使最左位元循環至最右位元。

圖2例示一種使用移位操作，使不對準資料字不對準之先前技藝實例。一實例可見於1989年3月21日授予Hensen等人之美國專利4,814,976號，RISCComputerWithUnaligned Reference Handling And Method For TheSame（在本文稱作"Hansen"）。在位址0-3之記憶體100之內容，予以饋入暫存器A120，位置PA4至PA1。在位址4至7之記憶體100之內容，予以饋入暫存器B130，在位置PB4至PB1。暫存器A120然後予以左移位三位置，因而D4為在位置PA4。暫存器B130予以右移位一位置，因而D3為在位置PB3，D2為在PB2，及D1為在PB1。暫存器A122與暫存器B132合併144，以產生希望之資料在暫存器R110位於適當位置。合併144係以在暫存器B132之位置PB3至PB1重寫在暫存器A122之位置PA3至PA1，或以在暫存器A122之適當位置重寫在暫存器B132之適當位置。合併144之為拷貝在暫存器A132之PA4之內容至在暫存器R110之位置P4，並可拷貝暫存器B132之PB3，PB2，及PB1之內容至暫存器R110之位置P3，P2，及P1。

因此，在記憶體之未對準字在微處理器饋入及對準，並且在微處理器之對準字使用圖1之對準網路114或圖2之左移位，右移位，及合併144予以未對準及儲存在記憶體。此等技術例如使用在自一種32-位元電腦架構饋入及儲存之32-位元字。在饋入及儲存32，16，及8資料位元之64位元

(請先閱讀背面之注意事項再填寫本頁)

訂線

五、發明說明(5)

架構有新問題發生。64位元記憶體系統需要如32一位元記憶體系統兩倍多之對準路徑，供位元組及半字，以及二32一位元對準衰減器供字存取。因此，先前技藝之對準網路變成一種複雜及昂貴之解決辦法。而且，在圖2中，合併144變成更複雜，因為其必須處理多很多之移位至暫存器中之不計較116。另外，諸如Hansen等人之先前技藝，未揭示自32進至64位元字時如何完成符號擴充。圖2不是有二M一位元移位器，就是有一左移位及一右移位或一更複雜之M一位元雙向移位器。因此，當電腦架構自32至64位元，並可為128位元時，便需要有一處理包括適當符號擴充之未對準資料之較佳方法。

發明之概述

本發明揭示一種方法，供饋入儲存在若干記憶體位置之未對準資料，包括一饋入未對準資料之第一部份至一第一儲存位置，並在第一記憶體位置使第一部份自第一位置旋轉至一第二位置之步驟。其次將未對準資料之第二部份饋入一第二儲存位置，並自一位置旋轉至另一位置。然後使用一種邏輯操作進入一結果儲存位置，使第一儲存位置與第二儲存位置合併。儲存位置可例如為64一位元暫存器。邏輯操作可為一位元方式"或"操作。在未對準資料之第一部份為在第一儲存位置之第二位置時，該方法可任選包括在第一儲存位置進行掩蔽，零擴充及/或符號擴充操作。

本發明也揭示一種方法，供儲存資料至未對準之許多

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明(6)

記憶體位置，包括使在一第一儲存位置之資料之第一部份自一第一位置旋轉至一第二位置，並將位於第二位置之資料儲存至在一由第一指標所給定位址之未對準之許多記憶體位置。然後使在第二儲存位置之資料之第二部份自一第三位置旋轉至一第四位置；以及將位於在第四位置之資料儲存至在一由第二指標所給定位址之未對準之許多記憶體位置。該第一指標可例如包括一高位址及第二指標包括一低位址。

精於此項技藝者配合附圖，參閱下列詳細說明，將會明白本發明之此等及其他諸多優點及特色。

附圖之簡要說明圖1例示一先前技藝對準網路；

圖2例示一種使用移位操作，使一未對準資料字對準之先前技藝實例；

圖3例示一特定實施例，示本發明自記憶體饋入一未對準資料字至一暫存器；

圖4例示本發明之一種將一在暫存器之值儲存至一未對準記憶體位置之特定實施例；

圖5例示本發明之另一將一未對準記憶體字饋入至一暫存器，包括符號擴充之特定實施例；

圖6例示使用於本發明特定實施例之一種簡化指令格式；

圖7例示核心之頂級分區之實例；

圖8例示本發明之LSU特定實施例之簡化方塊圖；

圖9例示在本發明之特定實施例，利用快取記憶體命中

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明(7)

饋入之流水線作用；

圖 10 例示在本發明之特定實施例，利用快取記憶體命中儲存之流水線作用；

圖 11 例示對準及符號擴充區段之簡化方塊圖；

圖 12 示本發明之一種 8 位元組右旋轉器 800 之特定實施例；

圖 13 示一種 4 位元組未對準字饋入指令如何在不同末端達成對準之實例；

圖 14 示一種 4 位元組未對準儲存指令在不同末端之對準之實例；

圖 15 例示本發明之符號選擇區段 680 之簡化方塊圖；

圖 16 例示本發明之符號（零）擴充區段 690 之簡化方塊圖。

主要元件對照

100	記憶體
100'	記憶體
110	暫存器 R
112	記憶體切片
114	對準網路
115	暫存器 R
120	暫存器 A
122	暫存器 A

（請先閱讀背面之注意事項再填寫本頁）

訂
線

五、發明說明(8)

- 124 暫存器 A
- 130 暫存器 B
- 132 暫存器 B
- 134 暫存器 B
- 144 合併單元
- 150 指標
- 152 字邊界
- 200 核心單元
- 210 暫存器
- 216 指標
- 218 記憶體
- 218' 記憶體
- 218" 記憶體
- 230 記憶體
- 232 指標
- 234 高暫存器
- 23810 低暫存器
- 240 實例指令
- 242 結果暫存器
- 400 核心
- 403 匯流排介面單元
- 405 匯流排介面單元
- 410 指令流程單元
- 420 指令流程

(請先閱讀背面之注意事項再填寫本頁)

訂線

五、發明說明(9)

- 430 指令快取記憶體單元
- 440 饋入儲存單元
- 450 快取記憶體單元
- 510 管道
- 530 區段
- 554 對準器
- 6 52 解碼器
- 6 70 後相位旋轉器
- 676 選擇器
- 680 符號選擇區段
- 686 旋轉輸出
- 690 擴充區段
- 800 右旋轉器
- 801 前相位旋轉器
- 802 最高有效字
- 810 輸出
- 812 多工器
- 814 多工器
- 816 多工器
- 818 多工器
- 820 多工器
- 822 多工器
- 860 鎖存器
- 910 記憶體

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

五、發明說明(10)

- 922 暫存器
- 934 快取記憶體
- 935 低部份
- 951 暫存器
- 962 記憶體
- 1020 多工器
- 1040 控制區段
- 1042 控制區段
- 1044 控制區段
- 1046 控制區段
- 1030 多工器
- 1032 多工器
- 1034 多工器
- 1036 多工器

特定實施例之說明

本發明係概括關於在一微處理器或微控制器使資料移動至記憶體及自其移動資料，尤指饋入及儲存未對準記憶體參考。

圖3例示本發明之一種特定實施例，示自記憶體饋入一未對準資料字至一暫存器。在圖3中，記憶體100示四希望之資料位元組，D4，D3，D2，及D1，其將予以移動至暫存器R110。一指標150產生最低位元組D1之位址，亦即位址在記憶體100'（其為與圖2

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

五、發明說明(11)

中之記憶體 100 相同) 等於 6。記憶體 100' 字邊界 152 為在位址 3 與 4 之間。D4 如字之下位元組，在記憶體 100' 自位址 0 - 3，予以在位置 PA1 饋入暫存器 A124，並且然後右旋轉一位元組，以在圖 3 中產生其在暫存器 A124 之新位置 PA4。D4 右之位元組已予以零掩蔽，因而位置 PA3 直到 PA1 包含 0。同樣，在記憶體 100' 中之位元組 D3，D2 及 D1，予以在位置 PB4，PB3，及 PB2 饋入暫存器 B134，並右旋轉一位元組至在暫存器 B134 中之位置 PB3 直到 PB1。D3 左之位元組予以 0 擴充，因而位置 PB4 包含 0。暫存器 A124 利用暫存器 B134 予以位元方式 " 或 " 160，以在暫存器 R110 產生結果。圖 3 之右旋轉為較簡單，並可需要硬體少於圖 2 之雙 M - 位元移位器或 M - 位元雙向移位器。另外，圖 3 之位元方式 " 或 " 160，允許再使用已在整數多媒體單元 (Integer Multimedia Unit，簡稱 IMU) 進行之邏輯操作，而非需要一如圖 2 中之特殊合併單元 144。因此，圖 3 表示一種較之在先前技藝者簡單，自記憶體 100 饋入未對準字至暫存器 R110 之硬體實施。

圖 4 例示本發明在一暫存器儲存一值至一未對準記憶體位置之特定實施例。在圖 4 中，暫存器 210 包含 8 位元組，編號自 7 直到 0 (小末端式格式)。暫存器 210 之定址 3 直到 0 之位元組包含標記 w，x，y，及 z 之位元組 (位址 7 - 4 之內容為不計較，"*")。此等四對準

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明(12)

位元組在第一暫存器 2 1 0 將經由 2 1 4 在記憶體 2 1 8 儲存在包含 E, D, C, 及 B 之位置 4 - 1。藉指標 2 1 6 指向在記憶體 2 1 8 之第一位元組 B, 其表示一在第二暫存器之位址。在記憶體 2 1 8', 第一暫存器 2 1 0 之位元組 w 已在位置 4 重寫 E。此為一自指標 2 1 6 偏置 3 位元組之位址。在 2 2 2 例示一實例指令。在記憶體 2 1 8", 暫存器 2 1 0 之位元組 x, y, 及 z 已在位置 3 - 1 重寫 D, C, B。偏置 0 位元組之指標 2 1 6 產生供此儲存之開始點, 並且在 2 2 6 示該實例指令。因此, 在指標 2 1 6 開始自暫存器 2 1 0 至記憶體 2 1 8 之一種小末端式儲存 4 位元組, 並向左進行 4 位元組, 包括指標 2 1 6。記憶體 2 1 8" 示位於位址 4 直到 1 之位元組 w, x, y, z。

圖 5 例示本發明饋入一未對準記憶體字至一暫存器, 包括符號擴充之另一特定實施例。在圖 5 中, 記憶體 2 1 0 示在位址 8, 位址 4 及位址 0 之長字 (3 2 - 位元) 邊界。在記憶體 2 3 0 中之位元組 E, D, C 及 B 產生未對準字。一指標 2 3 2 為一置在第二暫存器之位址, 其指向將行自記憶體 2 3 0 饋入之第一位元組 B。在記憶體 2 3 0 中之位元組 E, 表示對指標 2 3 2 加位址 1 外加一偏置 3, 以在記憶體 2 3 0 產生一位址 4, 藉以所定位之最高位元組。在記憶體 2 3 0 之位元組 E 然後在位址 0 予以饋入 h i 暫存器 2 3 4, 並右旋轉至一在 h i 暫存器 2 3 4 之位址。位元組 E 右在 2 3 4 之位元組已予以 0 掩

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明 (13)

蔽，並且位元組 E 左在 h i 暫存器 2 3 4 之位元組已予以符號 - 擴充，亦即 " s s s s "。指令 2 3 6 產生饋入有偏置 3 之高未對準字至 " h i " 暫存器之實例。記憶體

2 3 0 之位元組 D，C，及 B 予以饋入暫存器 2 3 8，在位址 1 開始，並右旋轉以佔用低暫存器 2 3 8 之位址 2，1 及 0。一實例指令 2 4 0 示饋入有偏置 0 之 1 0 未對準字至一 " 1 0 " 暫存器。在 1 0 暫存器 2 3 8，D 左之位元組已予以 0 擴充。暫存器 2 4 2 示以 1 0 暫存器 2 3 8 位元方式 " 或 " h i 暫存器 2 3 4 之結果。結果為一有 E，D，C，及 B 對準在結果暫存器 2 4 2 之符號 - 擴充字。一樣本指令 2 4 4 示利用 1 0 暫存器 2 3 8 位元方式 " 或 " h i 暫存器 2 3 4，以產生結果暫存器 2 4 2。

在所例示之特定實施例，記憶體予以位元組定址。饋入及儲存指令使用小 - 末端式或大 - 末端式表示提供對資料之存取。在一種特定實施例，在電源開重設規定饋入及儲存指令之末端，並且其後不改變。

請察知，小末端式位元組次序將位址為零之位元組位址在字置在最低有效位置（小末端）。大末端式位元組次序將位址為零之位元組位址在字置在最高有效位置（大末端）。因此，在一種大末端式電腦架構，資料之位元視為自左至右排成一行，最低數及最高有效位元為在左，以及在一種小末端式架構，位元右至左予以編號，最低有效位元及最低數位在右。

在一種特定實施例，饋入及儲存指令在一暫存器與記

(請先閱讀背面之注意事項再填寫本頁)

訂

線

五、發明說明(14)

憶體之間傳送資料，其中有些饋入指令具有有符號及無符號之變型，以進行正確擴充至暫存器中。供位元組（8一位元）及字（16一位元）物件大小，有符號及無符號之饋入均可能存在。供長字（32一位元）物件大小，僅可提供有符號之饋入。此係因為不管符號，所有32一位元物件均可在暫存器保持在一種符號－擴充形式。供四字（64一位元）物件大小，在有符號與無符號之饋入間之可無差異。

在以上之特定實施例，提供二組不同之饋入及儲存指令：

第一組為：對準之饋入及儲存指令，並自然支援對準之資料。此為資料之位址為存取之寬度之確切倍數之情形。如果此等指令之一試圖未對準存取，其將會導致一種不對準之例外。

第二組為：未對準之存取支援。一實例為軟體，其可能想要存取未對準之物件（例如合併之資料結構）。指令已包括可用以構成有效率順序，供饋入未對準之物件，或有未知之未對準者。

提供指令供饋入及儲存未對準之長字及四字。供存取未對準字（16一位元）之指令可使用現有對準儲存及算術指令，例如：

/*小末端式實例*/

LD.B Rbase, #0, Rtmp0

LD.B Rbase, #1, Rtmp1

（請先閱讀背面之注意事項再填寫本頁）

訂
線

五、發明說明 (15)

SHLLI Rtmp1, #8, Rtmp1

OR Rtmp0, Rtmp1, Result

其中 LD.B 自加第一源運算元至一 10- 位元立即常數 (表 2) , 藉以所形成之有效位址饋入一位元組。位元組予以符號 - 擴充至結果暫存器 ; 以及

表 2

LD.B

100000	m	s	d	r
--------	---	---	---	---

31 26 25 20 19 10 9 4 3 0

LD.B Rm , s , Rd
基本 ← 零擴充 64 (Rm) ;
偏置 ← 符號擴充 10 (s) ;
位址 ← 零擴充 64 (基本 + 偏置) ;
結果 ← 符號擴充 (讀取記憶體 8 (位址)) ;
Rd ← 暫存器 (結果) ;

其中 SHLLI 指令使 Rm 邏輯式左移位 S<0 for 6> , 並將結果儲存在 Rd (表 3) ; 以及

(請先閱讀背面之注意事項再填寫本頁)

訂 線

經齊那智慧財產公司員工肖費合作法印製

五、發明說明(16)

表 3

SHLLI

110001	m	0001	s	d	r
--------	---	------	---	---	---

31 26 25 20 19 16 15 10 9 4 3 0

SHLLIRm, s, Rd

源 1 ← 零擴充 64 (Rm) ;
 源 2 ← 零擴充 64 (符號擴充 6 (s)) ;
 結果 ← 源 1 << 源 2 ;
 Rd ← 暫存器 (結果) ;

其中或指令進行 Rm 及 Rn 之內容之位元方式 - "或" , 並將結果儲存在暫存器 Rd (表 4) :

表 4

"OR"

000001	m	0001	s	d	r
--------	---	------	---	---	---

31 26 25 20 19 16 15 10 9 4 3 0

ORRm, Rn, Rd

源 1 ← 零擴充 64 (Rm) ;
 源 2 ← 零擴充 64 (Rn) ;
 結果 ← 源 1 或 源 2 ;
 Rd ← 暫存器 (結果) ;

圖 6 例示一使用在本發明特定實施例之簡化指令格式。在圖 6 中，位置 3 1 0 包含操作或操作碼。位置 3 1 2 包含基本暫存器 " m " ，其在加至偏置常數 " s " 時，在

(請先閱讀背面之注意事項再填寫本頁)

裝
訂
線

五、發明說明 (17)

未對準物件產生一最高或最低位元組之位址。饋入操作之結果予以儲存在一終點暫存器 " d "。供儲存操作，將行予以儲存之所獲得值為在暫存器 " y "。

在此實施例所說明之指令，可用以在 3 指令饋入一未對準長字或四字物件。儲存一未對準長字或四字花費 2 指令。

供未對準長字饋入之指令順序使一符號一擴充結果返回。如果需要零擴充，此可在軟體進行，而以一額外指令為代價。

未對準饋入順序之實例如下：

LDHI.Lptr, #off+3, Rhi

LDLO.Lptr, #off, Rlo

"OR"Rhi, Rlo, 結果

在未對準物件之最高位元組之位址，予以傳至"饋入高部份"指令 (LDHI.L)，同時在未對準物件之最低位元組之位址，予以傳至"饋入低部份"指令 (LDHI.L)。一般為，至高部份指令之立即運算元為 (n - 1) 多於至低部份指令之立即運算元，其中 "n" 為在位元組之物件大小。

一未對準儲存順序之實例如下：

STHI.L ptr, #off+3, 值

STLO.L ptr, #off, 值

就未對準饋入順序而論，傳至高部份指令 (

S T H I . L) 之位址 (例如 p t r # o f f + 3)，應該指向未對準物件之最高位元組，而傳至低部份指令 (

(請先閱讀背面之注意事項再填寫本頁)

訂 線

五、發明說明(18)

S T H O . L) 之位址，應該指向未對準物件之最低位元組（例如 p t r # o f f ）。

表 5 示一特定實施例之饋入及儲存指令實例。

（請先閱讀背面之注意事項再填寫本頁）

表

訂

線

五、發明說明(19)

表 5

未對準饋入指令	說明
LDHI.L	一未對準，有符號長字之高部份自記憶體饋入至一通用暫存器。
LDLO.L	一未對準，有符號長字之低部份自記憶體饋入至一通用暫存器。
LDHI.Q	一未對準四字之高部份自記憶體饋入至一通用暫存器（供與一暫存器相同大小之物件，符號可能不重要）。
LDLO.Q	一未對準四字之低部份自記憶體饋入至一通用暫存器（供與一暫存器相同大小之物件，符號可能不重要）。
未對準儲存指令	
STHI.L	一長字之高部份自一通用暫存器未對準儲存至記憶體。
STHLO.L	一長字之低部份自一通用暫存器未對準儲存至記憶體。
STHI.Q	一四字之高部份自一通用暫存器未對準儲存至記憶體。
STLO.Q	一四字之低部份自一通用暫存器未對準儲存至記憶體。

硬體實施之特定實施例

在一特定實施例，本發明可在一有一核心單元 200

(請先閱讀背面之注意事項再填寫本頁)

表
訂
線

五、發明說明 (20)

在零或根階層之 CPU 實施。核心 400 在階層 1 可包括六單元，加一選用之可拆開浮點單元 (Floating Point Unit，簡稱 FPU)。圖 7 例示核心 400 之頂級分區之實例。表 5 說明在核心中之每一單元之功能。

表 5

階層	單元	簡稱	說明
0	S5 核心 400	S5	頂級核心區段
1	匯流排介面單元 405	BIU	匯流排對外部模組諸如周邊模組及外部記憶體介面存取。
1	指令流程單元 410	IFU	CPU 管道之前端：提取解碼，發出及分支。也包含模式 B 仿真。
1	指令流程多媒體單元 420	IMU	處理所有整數及多媒體指令。主 CPU 資料路徑。 包含指令快取記憶體及指令
1	指令快取記憶體單元 430	ICU	翻譯後備緩衝器 (Translation 記憶體 Lookaside Buffer，簡稱 TLB)
1	饋入儲存單元 440	LSU	處理所有記憶體指令及資料快取記憶體控制。
1	資料快取 450	DCU	包含資料快取記憶體及資料記憶體單元 翻譯後備緩衝器 (TLB)
1	浮點單元 (未示)	FPU	可拆開浮點解碼器，管道控制及執行管道 (在圖 7 中未示)。

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明(21)

在一特定實施例，饋入/儲存單元 (load/store unit ，簡稱 L S U) 4 4 0 進行下列功能：

執行所有記憶體相關指令，處理來自 B I U 4 0 5 之探聽要求，處理對組態暫存器之存取，經由來自 B I U 4 0 3 之組態空間要求，處理對除錯暫存器之存取，維持資料快取記憶體，例如快取記憶體連貫性，行一填滿，及/或回寫，檢查存取許可，支援資料監視點，代表 F P U 進行饋入及儲存，及在 F P U 與核心 4 0 0 之間進行某些資料傳送。

圖 8 例示本發明之 L S U 4 4 0 之一特定實施例之簡化方塊圖。L S U 4 4 0 包括表 7 中在階層 2 所列示之 8 功能方塊。

表 7

簡稱	名稱	階層	說明
lsp 510	ls 管道	2	饋入/儲存流水線控制區段
dcc 515		2	資料快取記憶體控制器
drf 520	drefill	2	Drefill 區段
cfgrq 530		2	組態區段要求器
cfgrp 535		2	組態區段應答器
snp 540	探聽	2	探聽區段
dwp 545	資料 wp	2	資料監視點
lsdp 550	ls 資料路徑	2	LSU 資料路徑區段
lspalgn 554		3	對準,(符號/零)擴充區段

(請先閱讀背面之注意事項再填寫本頁)

訂 線

五、發明說明 (22)

l s 管道區段 l s p 5 1 0，控制饋入 / 儲存流水線。此區段協調並使所有核心記憶體存取排序。l s 管道 5 1 0 經由一路徑通過 I F U 4 1 0 接收一供目前指令之資料值，其利用在主 I M U 4 2 0 資料路徑之現有結果匯流排。l s 管道區段 5 1 0 處理 L S U 單元之 C P U 流水線方面其在流水線之 E X 1，E X 2 及 E X 3 級之開始，負責保持所需要之資料，並控制 C P U 執行之停止。在另外之跟蹤，資訊予以在 E X 4 (W B) 級傳至跟蹤通道。

d c c 區段 5 1 5 為主資料快取記憶體控制器區段。其包含 F S M，其在資料快取記憶體陣列將操作排序，以實施各種操作。其包括一自 l s 管道 / I F U 接收要求，以執行指令之 F S M。F S M 將執行指令所需要對快取記憶體及再填滿等之連串要求排序。另外，d c c 區段 5 1 5 自所有其他區段 (d r e f i l l 5 2 0，探聽 5 4 0，c f g r p 5 3 5) 接收快取記憶體要求，並在此等要求與 F S M 所產生之要求之間仲裁，以在任何既定循環確定任一資源存取快取記憶體。

代表 LSU440 之 d r e f i l l 區段 d r f 5 2 0 介接至 B I U 4 0 5，包括快取記憶體相關循環如快取記憶體行一填滿，及非可快取記憶體存取。d r f 5 2 0 可 ' 應要求 ' 存取快取記憶體資料陣列，以進行回寫及填滿活動。其包含緩衝供 2 B I U 4 0 5 存取之 4 進行中快取記憶體寫入及位址緩衝器，及 2 進行中儲存緩衝器寫入至資料陣列一這使儲存能完全流水線化。

config 要求區段 c f g r q 5 3 0，處理 L S U 與組態匯流排間

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明 (23)

之介面。組態匯流排提供機構，供 CPU 存取組態暫存器及快取記憶體 / TLB 狀態，以及允許除錯控制器經由 BIU 405 存取除錯暫存器。'要求'邊，亦即 cfgrq 區段 530，接受來自任一 ls 管道 510 或 drefill 520 區段之組態要求，並且此等位置在匯流排，並等待響應。

Config 應答器區段 cfgrp535，接收 config 要求區段 530 所作之要求，以供對資料快取記憶體組態空間之存取。cfgrp535 則 dcc 515 作相關要求，並在 config 匯流排使結果返回。

資料監視點區段 dwp545，將存取位址與二資料監視點範圍比較，並將任何命中發訊號至 ls 管道區段 510，其復依與自 IFU 410 所接收之存取關聯之控制訊號而定，確定一監視點及 / 或例外是否被觸發。位址範圍經由組態匯流排予以設定。

snp 區段 540 為一自 BfU 405 區段接收探聽要求，然後並順序通過在可能存在所供給實體位址處之快取記憶體之位置。如果其找到一命中，其將會要求 drf520 回寫任何沾污資料，並且如果有一寫入探聽要求，則其將使該線無效。

饋入 / 儲存資料路徑區段 lsdp550，包括在主執行資料路徑關於 64 位元資料之操作。資料對準器 lsalgn554，此處為連同儲存緩衝器 556 之資料元件。結果經由一與 IMU 420E3 級共用之結果匯流排予以返回至管道檔案。此路徑也用以將資料自資料路徑通過 IFU 410 傳至主 LSU 區段。lspalgn 區段 554 為位元組對準及符號 / 零擴充區段，所有記

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明(24)

憶體運算元均通過此區段。此區段也支援小末端式模式及大末端式模式。此區段554為lsdp550之一部份，並處理資料饋入及儲存操作所需要之資料對準。

在一特定實施例，LSU可包括一單一對準器，以進行饋入及儲存操作所需要之對準旋轉及符號/位元組選擇，而非在每一饋入及儲存路徑有一單獨之旋轉器。在一特定實施例，僅使用一右旋轉器，由於供64-位元架構，k位元組之左旋轉為同等於(8-k)位元組之右旋轉。在一種替代性實施例，使用一種雙向旋轉器。在該二實施例，二操作共用相同之資源，以縮減面積。此點之較少代價，為饋入及儲存之對準步驟，均必須發生在相同之流水線級，以避免資源危險。這導致一種在資料儲存至快取記憶體前，需要1額外循環之儲存操作。此額外循環被流水線化所隱藏，並在性能上少有效應，除非碼立即需要重新饋入剛才所儲存之資料。然而，編譯器應該能再使用來自原來暫存器之資料。

在LSU為主要關切之路徑，為自資料陣列所讀取之資料者，其然後可在發出至在IMU 420及管道檔案之各轉發點前，通過對準器/符號-擴充邏輯554。由於宜於使此路徑之長度最少，對準器554可置於IMU 420資料路徑內。

LSU 440可使用3流水線級，以存取快取記憶體：

- 1) E1:計算位址(IMU代表LSU所處理)。
- 2) E2:作成虛擬標籤(vtag)比較，並且供饋入操作，資料讀取。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

五、發明說明 (25)

3) E3:供饋入依據比較結果所選擇之資料，並予以對準及符號擴充。供儲存，資料予以對準，如果資料無法立即寫入，並發出至進行中儲存緩衝器。

此作用可予以完全流水線化——指令可為在每一流水線級，並且同時假設所有存取"命中"使用虛擬標籤 (vtag) 比較，在每循環一指令執行。

供在資料在 E3級之結束進入儲存緩衝器之儲存操作，資料尚未寫入至資料陣列，然而自一流水線觀點，指令為完全。

在 1s 管道區段 510 也有一 E4/WB 級，其可使用以將資料傳至跟蹤通道，並可能不有助於指令執行。

drefill 區段 520 包含一儲存緩衝器，其包含多至 2 進行中儲存。在否則不存取該循環資料陣列，以將此資料寫入至快取記憶體之情形，drefill520 及 dcc515 可使用'自由循環'。

對在此緩衝器儲存為進行中時所接收之記憶體操作檢查位址匹配——如果此隨後操作與一進行中儲存'衝突'，則其予以停止，直到儲存已寫入至快取記憶體，以保證饋入/儲存動作之正確執行。

依據位址之 64 一位元字位址 (亦即位元 [32 : 3])，完成在儲存緩衝器中之項目與一隨後記憶體存取間之相依性檢查。此可予以擴充為包括一上/下 32 一位元字接觸檢查，以有助 32 位元軟體——在此情形為一種自 64 位元字與將會允許一進行中 32 一位元儲存繼續者相

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明 (26)

反一半之饋入。

L S U 4 4 0 可處理 2 未完成 B I U 4 0 5 (亦即外部記憶體) 填滿存取 (預提取指令或快取記憶體遺漏所導致) , 或非可快取記憶體饋入。L S U 4 4 0 另外提供緩衝供回寫及儲存。

預提取指令不阻斷執行, 並且行檔案, 及如果需要, 回寫要求一旦已發出至 B I U 4 0 5 , 執行便恢復。

一饋入遺漏可使流水線停止, 直收到關鍵字。儲存可僅使流水線停止, 直到關鍵字已回寫 (假設需要回寫) 。在此點, 流水線將會在填滿之其餘部份繼續時重新開始。

在未完成要求仍然填滿一行時, 可發生另外之記憶體操作。如果此種記憶體要求導致一快取記憶體遺漏, 則可發出一第二填滿要求至 B I U 。如同第一填滿要求, 快取記憶體被自任何存取阻斷, 直到 B T U 應答第二填滿要求。C P U 管道再次停止, 直到第一關鍵字返回。可接受來自 C P U 管道之另外記憶體要求, 只要其可自快取記憶體予以服務。如果第三遺漏發生, 則此被阻斷, 直到第一填滿完成。

L S U 控制一虛擬快取記憶體。由於同義詞一虛擬位址所參考之資料, 也可能被其他同義虛擬位址所參考, 故可能有在虛擬加標籤 (v t a g) 快取記憶體所固有之問題。因此快取記憶體包含一實體標籤 (p t a g) 之單獨記錄供所有資料, 並且如果初始虛擬標籤基查找失敗, 此標籤配合 T L B 翻譯, 用以檢查同義詞命中。

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明(27)

一軟體需求為供任何二標記至相同物件之虛擬位址者(亦即實體位址)，然後虛擬索引將二者標記至相同快取記憶體組。供16kB頁及以上，可能無問題，因為無位元經歷用以索引快取記憶體之位址翻譯。然而，供4kB頁，某些索引位元經歷翻譯。因此，其軟體必要保證標記至相同實體位址之所有虛擬位址具有相同值供位元〔13:12〕。可能需要此需求，供使用虛擬索引之任何快取記憶體(虛擬或實體)。

由於快取記憶體為依據虛擬加標籤，在一(虛擬標籤)快取記憶體命中之情形，於是無需參考TLB。因此，虛擬標籤陣列也包含頁許可之拷貝，及來自適當TLB項目之其他頁基資訊，以使能與標籤比較相同時間檢查存取許可等。這保證檢測任何存取違規，而無需查詢TLB。

LSU支援控制暫存器位元所可增強之一全套LRU替代策略，其可'鎖定'若干快取記憶體方式(以一種每快取記憶體基礎)。資料可通過一特殊預提取模式饋入一種鎖定之方式。一通過儲存命中之資料更新以外之項目，無法以鎖定方式予以修改。

讀入至CPU之資料，可使其位元組重新排序，因而其予以排列如一64位元量，而例如有最高有效位元組在位元組位置7，及最低有效位元組在位元組位置0。

這保證所有64位元資料在內部呈一致之末端。對準/符號-擴充區段所進行之子字選擇將會計及系統末端，並利用適當掩模，"異或"運算用以選擇資料之位址之底

(請先閱讀背面之注意事項再填寫本頁)

訂 線

五、發明說明 (28)

3 位元，藉以選擇正確之子字。

L S U 之一特定實施例配合大小為 1 6 k B ， 3 2 k B 或 6 4 k B 之快取記憶體工作。因為此 L S U 計及所有三快取記憶體大小。例如，將會傳遞 9 位元作為一 6 4 k B 快取記憶體所需要之一組索引；一 3 2 k B 快取記憶體將會忽略最高有效位元，同時一 1 6 k B 快取記憶體將會忽略 2 最高有效位元。同樣，所傳遞供比較之標籤，可依據供一 1 6 k B 快取記憶體之需求，而在 3 2 k B 及 6 4 k B 情形有冗餘位元。組基位址抵觸檢測可進行一 7 位元比較，以處理在一 1 6 k B 快取記憶體之組抵觸，其可能在 3 2 k B 及 6 4 k B 快取記憶體報告錯誤抵觸。

圖 9 例示在本發明之一種特定實施例，饋入一快取記憶體命中之流水線作用。此為假設在前面或隨後指令無其他饋入／儲存操作。與資料抽取 6 0 6 並行完成 Vtag 比較 6 0 4，並在次一循環使用匹配訊號選擇正確資料。使資料在 3 循環返回，導致一 2 指令饋入／使用損失。此種方案示在 E 2 完成儲存緩衝器／填滿抵觸檢查 6 0 2。這保證在 E 3 級之開始可利用確定是否需要停止所需要之所有資訊。如果必要，資料在 E 3 予以對準。

在 vtag 比較有一遺漏時，發生一混淆之命中，但在隨後之 ptag 比較時（使用 DTLB 所翻譯之實體位址）使命中返回。再次假設在前面或隨後循環無饋入或儲存指令。在處理初始 vtag 遺漏時，此順序導致一 3 循環流水線停止。

供混淆命中之 vtag 行，應該予以更新為指最近之混淆

(請先閱讀背面之注意事項再填寫本頁)

訂 線

五、發明說明(29)

(如參考之局部性暗示，此為其次將用以指此資料之最可能混淆－請察知，供正確之作用，此更新為不'必要'但應該有助性能)。

在快取記憶體控制器確定有 vta 及 ptag 遺漏時，則 drefill 區段 520 要求一行再填滿。drefill 區段 520 然後控制回寫及填滿活動，同時快取記憶體控制器之流水線控制部份等待關鍵字出現。

為避免將行自快取記憶體回寫之資料，在填滿資料回寫前，被其重寫之任何危險，重要的是，在脈衝串讀取要求後 4 循環，發生供回寫之讀取－這保證無危險存在。在此特定實施例，此不是問題，因為服務任何填滿要求，將會花費約 16 循環。在一種替代性實施例，dcc 515 可警告 drefill 520 區段遺漏之或然率，以允許推測性開始第一次讀取。

在遺漏檢測之循環(在 ptag 比較後)，如果 drefill 區段指示無法接受填滿要求，dcc 狀態機將會移入至一保持循環。請察知，dcc 需要知道確定如此是否需要填滿回寫。回寫係自可能已讀取之有效位元及沾污狀態位元確定。一旦檢測到 vtag 遺漏，便使流水線停止。dcc 515 保持停止，直到其完成設定填滿。供 normal 饋入，其然後維持停止，直到收到關鍵字資料。

圖 10 示在本發明之一種特定實施例，儲存一快取記憶體命中之流水線作用。在檢測到快取記憶體命中時，則依是否需要資料陣列供一隨後饋入，或是否資料已在儲存

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明 (30)

緩衝器而定，儲存資料（連同位元組－選擇值對準至正確對準，以供儲存）不是直接寫入至資料陣列就是置入至儲存緩衝器。如果資料置於儲存緩衝器，則其將會自該處取得，並在次一可利用循環（亦即無資料陣列存取發生之次一循環）寫入至資料陣列。

供混淆之命中，如同供饋入之情形，對準資料予以發出至儲存緩衝器，代替發回至回寫。

供儲存遺漏，流水線儲存等待，直到已發生來自資料將行儲存之字之任何回寫，並且然後發出資料至儲存緩衝器。

實例位元組對準及符號－擴充區段 (lspalgn)

lspalgn 區段 5 5 4 之一特定實施例，在自快取記憶體饋入／儲存至快取記憶體之 8－位元組資料（6 4－位元）作位元組對準及符號－擴充操作，以抽取對應於一饋入－儲存指令之最後結果之有效位元組部份。因為此區段形成一關鍵定時路徑之一部份，其為一特殊資料路徑區段，並因此自 1 s 管道 5 1 0 之其餘部份分開。然而有些控制邏輯保持在 1 s 管道，並在需要其前一循環將控制訊號傳至資料路徑。

圖 11 例示對準及符號擴充區段 lsalgn554 之簡化方塊圖。在圖 1 1 中，位元組選擇解碼器 6 5 2 確定一儲存更新在終點 8 位元組字之何一位元組。例如，儲存一位元組至在該字之第三位元組，將會產生一 00000100 之位元組選擇

(請先閱讀背面之注意事項再填寫本頁)

訂 線

五、發明說明 (31)

掩模。至解碼器 6 5 2 之輸入為：s 5 _ 小末端式 6 5 4，供一小末端式系統，其為 $lsp_opc_ex2[3:0]$ ，其來自 l s 管道 5 1 0，並為在 E 2 之指令之操作碼；以及 $lsp_addr_ex2[2:0]$ 658，其為在一 8 位元組部份將行予以旋轉之諸位元組之最高 / 最低位元組之位址。輸入資料通過一自 lsp 510 選擇 6 4 一位元資料， lsp_algh 資料 6 7 2，或自 DCU 4 5 0 選擇 6 4 一位元資料，亦即 dcu_lsu_d 資料 6 7 4 之選擇器 6 7 6 到達。所選擇之資料 $lspalgn$ 資料 6 7 8，進入一後相位旋轉器 6 7 0。旋轉藉來自旋轉距離控制器 662 之位元 $rc[1:0]$ 668 予以控制。旋轉距離控制器 662 發出 lsp_dodd_ex2 664 及 lsp_dodd_ex2 666 至 DCU 450，其然後予以多工處理，以選擇 lsu_dcu_dodd 訊號。此訊號提供 $rc[3]$ 。供在 E 2 相位讀取 $vtag$ 及資料陣列之代表性情形，則其為所使用之 lsp_dodd_ex2 。然而，如果有一 $vtag$ 遺漏，及然後有一 $ptag$ 命中，則第二次讀取資料陣列（以讀取 $ptag$ 命中所選擇之資料），並且現在指令為在 E 3 級，因而將會選擇 lsp_dodd_ex3 訊號。同樣，如果有一快取記憶體遺漏，則在資料自 BIU 返回時，將會使用 lsp_dodd_ex3 訊號進行在在該路徑之主旋轉區段匹配自快取記憶體路徑在該資料之級前所需要之第一級之旋轉。

旋轉器 6 7 0 之輸出為 lsp_d 資料 - $ex3$ 686，並且供儲存操作，表示將行予以儲存之位元組，亦即 Sign Ext. 區段 6 9 0 為一穿越。供儲存無需符號或零擴充。供一饋入指令，依來自符號選擇區段 6 8 0 之 $lspalgn$ 符號位元

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

五、發明說明 (32)

6 8 4 之值而定，在 lsp_d 資料_ex3686 進行符號 (零) 擴充 6 9 0。

符號選擇區段 6 8 0 產生符號位元 6 8 4，供在符號 / 零擴充區段 6 9 0 之多工器。有一輸入 lsp_類型_ex2 682 進入區段 6 8 0，其產生該類型作為有符號 / 無符號位元組，有符號 / 無符號 (1 6 位元) 字，有符號 / 無符號 (3 2 位元) 長字或 (6 4 位元) 四字之一。符號選擇區段 6 8 0 確定結果之符號位元 (來自在資料中之位元組之符號位元及結果大小)，及符號零擴充區段 6 9 0 以一位元組基礎，選擇在該位元組位置之結果是否來自：1) 來自旋轉輸出 6 8 6 之位元組，2) 符號擴充位元組 6 8 4 (由符號位元所確定)，或 3) 0 位元組 6 8 8，亦即 8 b' 0，(供零擴充)。使用 0 位元組之一實例為供一未對準 3 2 位元饋入一長字之上部，例如位元組 H，其中下 3 位元組為 0。如果在該位址之字為 ABCDEFGH，則旋轉之輸出將為 DEFHABC (因為來自位置 0 之 H 已移動至在結果 32 位元字之正確位置)。符號選擇區段產生控制至符號 / 零擴充，因而 H 之符號位元選擇位元組 7 - 4。位元組 3 為來自旋轉器之結果 (亦即 H)，及位元組 2 - 0 為 0。亦即結果為 SSSSH000，其中 S 為 H 之符號擴充。

圖 12 示本發明之一 8 位元組右旋轉器 800 之特定實施例。輸入為在小末端式格式之 8 位元組，而有每一位元組置在一鎖存器：在 _資料 7 802，直到鎖存器在 _資料 0 804。在鎖存器在 _資料 0 804 之位元組 0，為來自位元 7 直到位元 0；在鎖

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明 (33)

存器在 _資料 7 802 之位元組 7 為來自位元 63 至位元 55 等等。
 $rc[2:0]$ 850, 852, 及 854, 為旋轉距離控制訊號。在粗線示
 供資料自在快取資料之最高有效字 802 至旋轉之資料輸出
 $o_資料$ 鎖存器 860 之路徑, 以證明在輸入中之每一位元組可
 如何切換至任何輸出位元組。例如, 在 _資料 7 802 之輸出
 810 在第一級 801 (DCU) 分開至以進行至多工器 814 及 812
 。多工器 814 之輸出然後分開成為多工器 816 及 818 (在後相
 位旋轉器 660) 。或者依選擇訊號 $rc[2]$ 850 而定, 輸出 810 可
 在 DCU 801 被多工器 812 分開至在後相位旋轉器 660 (LSU)
 之多工器 820 及 822。多工器 816 及 818 由 $rc[1]$ 852 予以控制。
 多工器 816 之輸出可進行至多工器 814 及 826, 並依選擇訊號
 $rc[0]$ 854 而定, 輸入資料位元組 802 可進行至鎖存器 $o_資料$ 7
 862 或 $o_資料$ 6 864。

在一特定實施例, 供定時原因, 8-位元組旋轉器可分
 開成為使雙倍字輸入資料在 DCU 450 旋轉之前相位旋轉器
 801, 及在 LSU 440 作位元組及字旋轉之後相位旋轉器 660。
 在一種替代性實施例, 8-位元組旋轉器不予以分開, 但包
 括在前相位旋轉器 801。在另一實施例, 8-位元組旋轉器
 不予以分開, 但包括在後相位旋轉器 670。旋轉器之輸入資
 料為 [7, 6, 5, 4, 3, 2, 1, 0], 其表示次序為位元組 7,
 位元組 6, 位元組 5...等。例如, $rc[2:0]$ 為 3'b010, 其使
 $in_資料$ [63:0] 向右旋轉 2 位元組。結果為 [1, 0, 7, 6, 5,
 4, 3, 2], 其表示 $in_資料$ 位元組 1 旋轉至 MSB, 及 $in_資料$
 位元組 2 旋轉至 LSB。表 6 中示可能之旋轉。

(請先閱讀背面之注意事項再填寫本頁)

訂
線

五、發明說明 (34)

表 6

rc[2:0]	o_資料[位元組]
3'b000	[7, 6, 5, 4, 3, 2, 1, 0]
3'b001	[0, 7, 6, 5, 4, 3, 2, 1]
3'b010	[1, 0, 7, 6, 5, 4, 3, 2]
3'b011	[2, 1, 0, 7, 6, 5, 4, 3]
3'b100	[3, 2, 1, 0, 7, 6, 5, 4]
3'b101	[4, 3, 2, 1, 0, 7, 6, 5]
3'b110	[5, 4, 3, 2, 1, 0, 7, 6]
3'b111	[6, 5, 4, 3, 2, 1, 0, 7]

圖 1 3 示一 4 - 位元組未對準字饋入指令，如何在不同末端達成對準之實例。在小末端式次序 9 0 5，在記憶體 9 1 0，在位址 1 1 0 及 0 1 1 之間有 4 資料位元組 (1, 2, 3, 4)。此資料在快取記憶體 9 1 2 (1, 2, 3, 4) 被反射。資料 9 1 4 之低部份 (4) 使 3 位元組向右 9 1 6 旋轉，及資料 9 1 8 之高部份 (1, 2, 3) 使 3 位元組向右 9 2 0 旋轉。結果為一在暫存器 9 2 2 之未對準長字。在大末端式次序 9 3 0，在此特定實施例，資料在小末端式次序儲存在快取記憶體 9 3 4。低部份 9 3 5 使 1 位元組向右 9 3 6 旋轉，及高部份 9 3 8 使 1 位元組向右 9 4 0 旋轉。表 7 及表 8 中示供 8 - 位元組及 4 - 位元組饋入不同末端之詳細旋轉距離。在 4 - 位元組情形，旋轉距離控制 rc [1 : 0] 為相似於 8 - 位元組

(請先閱讀背面之注意事項再填寫本頁)

訂 線

五、發明說明 (35)

情形，而在 L S U 有後相位旋轉器。

表 7：未對準 8 - 位元組饋入指令之右旋轉距離 ~182x30

lsp_addr_ex2[2:0] (在記憶體)	低部份 (小末端)	高部份 (小末端)	低部份 (大末端)	高部份 (大末端)
000	0	1	0	7
001	1	2	7	6
010	2	3	6	5
011	3	4	5	4
100	4	5	4	3
101	5	6	3	2
110	6	7	2	1
111	7	0	1	0

表 8:未對準 4 - 位元組饋入指令之右旋轉距離

lsp_addr_ex2[2:0] (在記憶體)	低部份 (小末端)	高部份 (小末端)	低部份 (大末端)	高部份 (大末端)
000	0	5	4	7
001	1	6	3	6
010	2	7	2	5
011	3	0	1	4
100	4	1	0	3
101	5	2	7	2
110	6	3	6	1
111	7	4	5	0

(請先閱讀背面之注意事項再填寫本頁)

表
訂
線

經濟部智慧財產局員工消費合作社印製

五、發明說明 (36)

圖 1 4 示一 4 - 位元組未對準儲存指令在不同末端對準之實例。在暫存器 9 5 1 示資料之 4 位元組 (1 , 2 , 3 , 4) 。在小末端式次序 9 5 0 , 資料 (4) 9 5 2 之低部份向右 9 5 4 旋轉 5 , 及高部份 (1 , 2 , 3) 向右旋轉 5 。然後利用導入位元組啓動訊號 lsp_db sel_ex3 , 而保持有效資料 , 亦即寫入至快取記憶體 9 6 0 , 及寫入至記憶體 9 6 2 , 在位址 1 1 0 直到 0 1 1 。在大末端式次序 9 7 0 , 低部份向右 9 7 4 旋轉 7 , 及高部份向右 9 7 8 旋轉 7 , 然後 lsp_db sel_ex3 允許儲存有效資料。表 9 及表 1 0 中示供 8 - 位元組 , 4 - 位元組儲存之不同末端之詳細旋轉距離。

表 9: 未對準 8 - 位元組儲存指令之右旋轉距離

lsp_addr_ex2[2:0] (在記憶體)	低部份 (小末端)	高部份 (小末端)	低部份 (大末端)	高部份 (大末端)
000	0	7	0	1
001	7	6	1	2
010	6	5	2	3
011	5	4	3	4
100	4	3	4	5
101	3	2	5	6
110	2	1	6	7
111	1	0	7	0

(請先閱讀背面之注意事項再填寫本頁)

訂 線

五、發明說明 (37)

表 10: 未對準 4-位元組儲存指令之右旋轉距離

lsp_addr_ex2[2:0] (在記憶體)	低部份 (小末端)	高部份 (小末端)	低部份 (大末端)	高部份 (大末端)
000	0	3	4	1
001	7	2	5	2
010	6	1	6	3
011	5	0	7	4
100	4	7	0	5
101	3	6	1	6
110	2	5	2	7
111	1	4	3	0

以下表 11, 表 12, 及表 13 中為 4-位元組, 2-位元組, 及 1-位元組儲存指令之位元組選擇。

表 11: 4-位元組儲存指令之位元組選擇

lsp_addr_ex2[2:0] (在記憶體)	低部份 (小末端)	高部份 (小末端)	低部份 (大末端)	高部份 (大末端)
000	00001111	00000001	11110000	10000000
001	00001110	00000011	01110000	11000000
010	00001100	00000111	00110000	11100000
011	00001000	00001111	00010000	11110000
100	11110000	00010000	00001111	00001000
101	11100000	00110000	00000111	00001100
110	11000000	01110000	00000011	00001110
111	10000000	11110000	00000001	00001111

(請先閱讀背面之注意事項再填寫本頁)

訂 線

五、發明說明 (38)

表 12:2 - 位元組儲存指令之位元組選擇

lsp_addr_ex2[2:0] (在記憶體)	小末端	大末端
000	00000011	11000000
010	00001100	00110000
100	00110000	00001100
110	11000000	00000011

表 13:1 - 位元組儲存指令之位元組選擇

lsp_addr_ex2[2:0] (在記憶體)	小末端	大末端
000	00000001	10000000
001	00000010	01000000
010	00000100	00100000
011	00001000	00010000
100	00010000	00001000
101	00100000	00000100
110	01000000	00000010
111	10000000	00000001

在一特定實施例，符號擴充或零擴充，為隨後位元組一旋轉操作，供饋入指令之步驟。無需作擴充供儲存指令，因為儲存寬度係依據指令替代 8 - 位元組寬度。掩模訊號

(請先閱讀背面之注意事項再填寫本頁)

訂 線

五、發明說明³⁹)

係供利用饋入指令掩蔽未對準資料之有效位元組，於是大小用以確定在一既定位元組位置之結果是否取自符號/零擴充或取自旋轉之輸出。此等訊號用以選擇產生 `lsu_結果_ex3` 之多工器之輸入。一產生低階位元組（在小末端式模式之 LDLO 及在大末端式之 LDHI）之未對準 4-位元組饋入始終零擴充。供未對準 4-位元組儲存，符號位元位於在饋入一高指令供小末端式之偏置位址，及在饋入一低指令供大末端式之偏置位址。

圖 15 例示本發明之符號選擇區段 680 之簡化方塊圖。圖 15 示在一特定實施例，如何自操作，位址及末端確定 `lspalgn_signbit 684`。至區段 1012 之輸入產生一選擇訊號，`sel[2:0]`，1016 其在來自輸入字 `lspalgn_data` 之符號位元（63，55，47，39，31，23，15，7），控制一 8-對-1 多工器 1020。請察知，如果結果類型為無符號，便使一 0 之符號位元 684 返回。

圖 16 例示本發明之符號（零）擴充區段 690 之簡化方塊圖。圖 16 示依大小，例如大小 `[2:0]1042`，掩模（供未對準饋入），例如掩模 `[1]1040`，及操作碼（訊號 `lsp_opc_ex3`，不過一至每一控制區段 1040，1042，1044，及 1046 之輸入未示）而定，如何在一多工器（1030，1032，1034，1036）自 "0" 688，旋轉輸出 686 或符號位元 684 選擇在輸出中之每一位元組。大小陣列為一種編碼：

```
'界定 LSU_SIZE_8 3'b111
```

```
'界定 LSU_SIZE_4 3'b011
```

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

線

五、發明說明⁽⁴⁰⁾

'界定 LSU_SIZE_2 3'b001

'界定 LSU_SIZE_1 3'b000

其中 LSU_SIZE_n 為大小 n 位元組之資料之編碼。掩模用以掩蔽旋轉輸出 686 (以一種每位元組基礎)，因而供未對準饋入，低於最低位元組讀取之位元組予以掩蔽至 "0"。此掩蔽在與符號 / 零擴充 690 相同之步驟予以合併。

結論

在上述詳細說明，本發明業已參照其特定例證性實施例予以說明。一般精於此項技藝者將會明白其他實施例。例如，指令可為長度 16 或 64 位元，及微處理器可配合 8，16，32，或 128 位元匯流排及字操作。因此顯然對其可作成各種修改及變化，而不偏離後附申請專利範圍及其同等者之全部範圍所闡釋之本發明之較廣義精神及範圍。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

四、中文發明摘要 (發明之名稱：饋入及儲存資料於電腦系統之方法)

本發明係概括關於微處理器或微控制器架構，尤指一種構造為處理未對準記憶體參考之架構。本案揭示一種方法，供饋入儲存在若干記憶體位置之未對準資料，包括一饋入未對準資料之第一部份至一第一儲存位置，並在第一步記憶體位置使第一部份自第一位置旋轉至一第二位置之步驟。其次將對準資料之一第二部份饋入一第二儲存位置，並自一位置旋轉至另一位置。然後使用一邏輯操作進入一結果儲存位置，使第一儲存位置與第二儲存位置合併。一儲存位置可例如為64位元暫存器。邏輯操作可為一種位元方式”或”操作。在未對準資料之第一部份為在第一儲存位置之第二位置時，該方法可任選包括在第一儲存位置進行掩蔽，零-擴充及/或符號擴充操作。

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

英文發明摘要 (發明之名稱 A METHOD FOR LOADING AND STORING DATA IN A COMPUTER SYSTEM)

The present invention relates generally to microprocessor or microcontroller architecture, and particularly to an architecture structured to handle unaligned memory references. A method is disclosed for loading unaligned data stored in several memory locations, including a step of loading a first part of the unaligned data into a first storage location and rotating the first part from a first position to a second position in the first memory location. Next a second part of the unaligned data is loaded into a second storage location and rotated from one position to another position. Then the first storage location is combined with the second storage location using a logical operation into a result storage location. The storage locations may be, for example, 64-bit registers. The logical operation may be a bit-wise OR operation. The method may optionally include, performing masking, zero-extending and/or sign extending operations on the first storage location, when the first part of the unaligned data is in the second position of the first storage location.

訂

線

六、申請專利範圍

1 . 一種方法，供饋入未對準資料儲存在許多記憶體位置，包含：

饋入該未對準資料之一第一部份至一第一儲存位置；

使在該第一儲存位置之該未對準資料之該第一部份自一第一位置旋轉至一第二位置；

饋入該未對準資料之一第二部份至一第二儲存位置；

使在該第二儲存位置之該未對準資料之該第二部份自一第三位置旋轉至一第四位置；以及

使用一邏輯操作至一結果儲存位置，

將該第一儲存位置與該第二位置合併。

2 . 如申請專利範圍第 1 項之方法，其中該第一位置為一第一暫存器，該第二儲存位置為一第二暫存器，及該結果儲存位置為一結果暫存器。

3 . 如申請專利範圍第 2 項之方法，其中該暫存器為長度 64 一位元。

4 . 如申請專利範圍第 1 項之方法，其中邏輯操作為一位元方式”或”操作。

5 . 如申請專利範圍第 1 項之方法，另包含在該未對準資料之該第一部份為在該第一儲存位置之該第二位置時，在該第一儲存位置進行一選擇自一由掩模，零一擴充，及符號擴充所組成類組之操作。

6 . 如申請專利範圍第 1 項之方法，其中該旋轉係在包含一第一相位及一第二相位之二相位進行，在第一相位進行一主要旋轉，在第二相位進行一較少旋轉。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

六、申請專利範圍

7. 一種方法，供儲存資料至未對準之許多記憶體位置，包含：使在一第一儲存位置之該資料之一第一部份自一第一位置旋轉至一第二位置，

將位於在第二位置之該資料，在一由一第一指標所給定之位址儲存在該未對準之許多記憶體位置；

使在一第二儲存位置之該資料之一第二部份自一第三位置旋轉至一第四位置；以及

將位於在第四位置之該資料，在一由一第二指標所給定之位址儲存在該未對準之許多記憶體位置。

8. 如申請專利範圍第7項之方法，其中該第一指標包含一高位址，及該第二指標包含一低位址。

9. 如申請專利範圍第8項之方法，其中該資料予以內容式在該高位址與該低位址之間儲存在該未對準之許多記憶體位置。

10. 如申請專利範圍第7項之方法，其中該旋轉係在包含一第一相位及一第二相位之二相位進行，在第一相位進行一主要旋轉，在第二相位進行一較少旋轉。

11. 如申請專利範圍第7項之方法，其中該第一儲存位置為一第一64位元暫存器，該第二儲存位置為一64位元第二暫存器，及該結果儲存位置為一64位元結果暫存器。

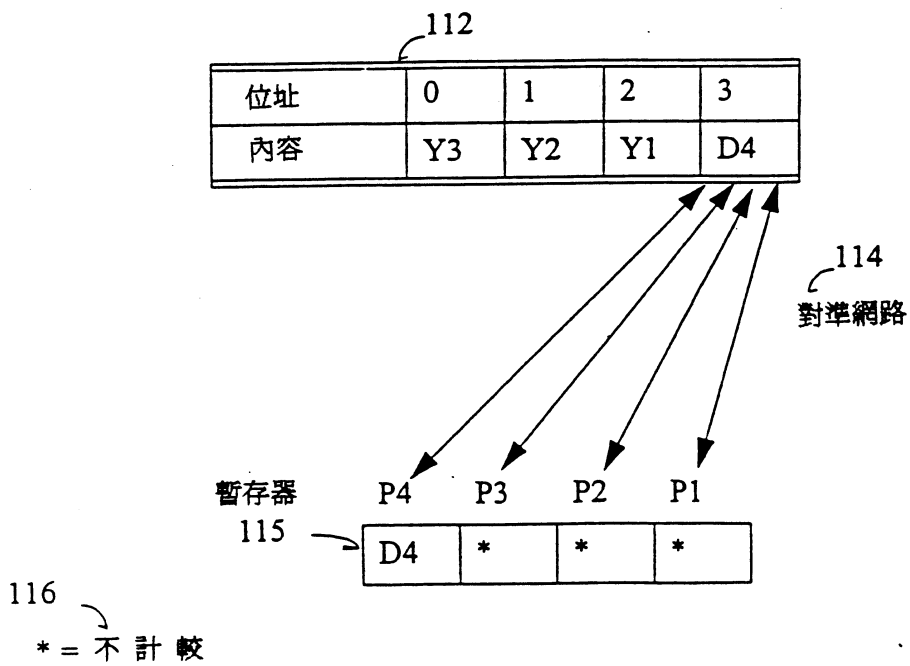
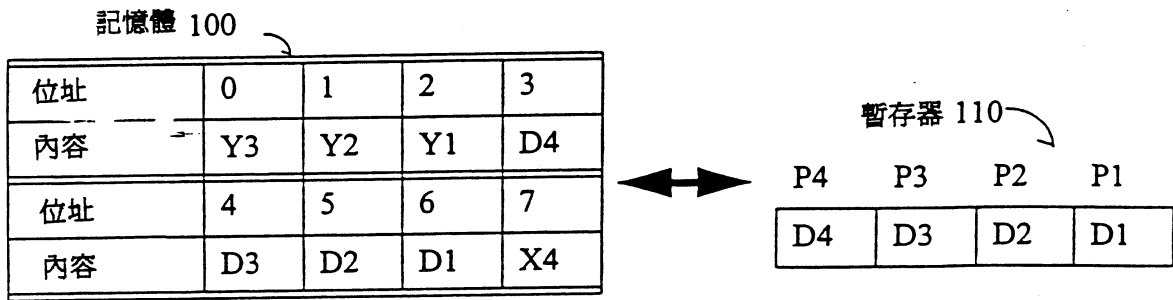
12. 如申請專利範圍第7項之方法，其中該資料係選擇自一由8，16，32，及64位元長度之資料所組成之類組。

(請先閱讀背面之注意事項再填寫本頁)

裝

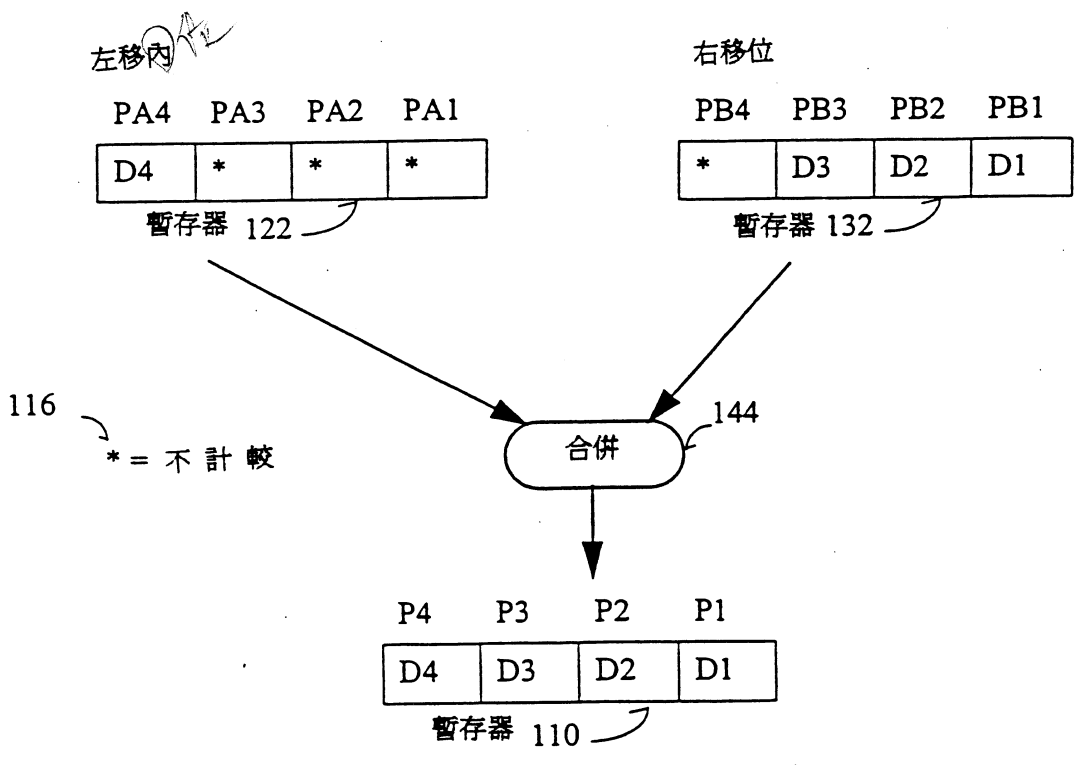
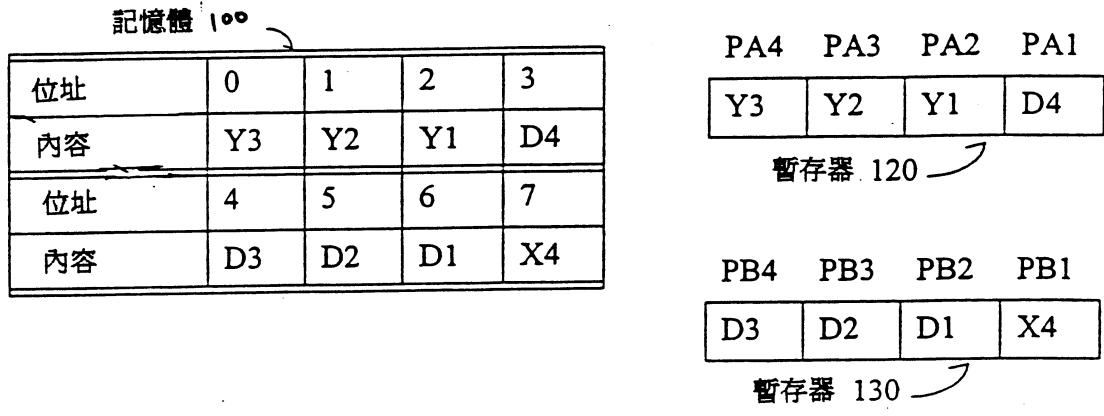
訂

線



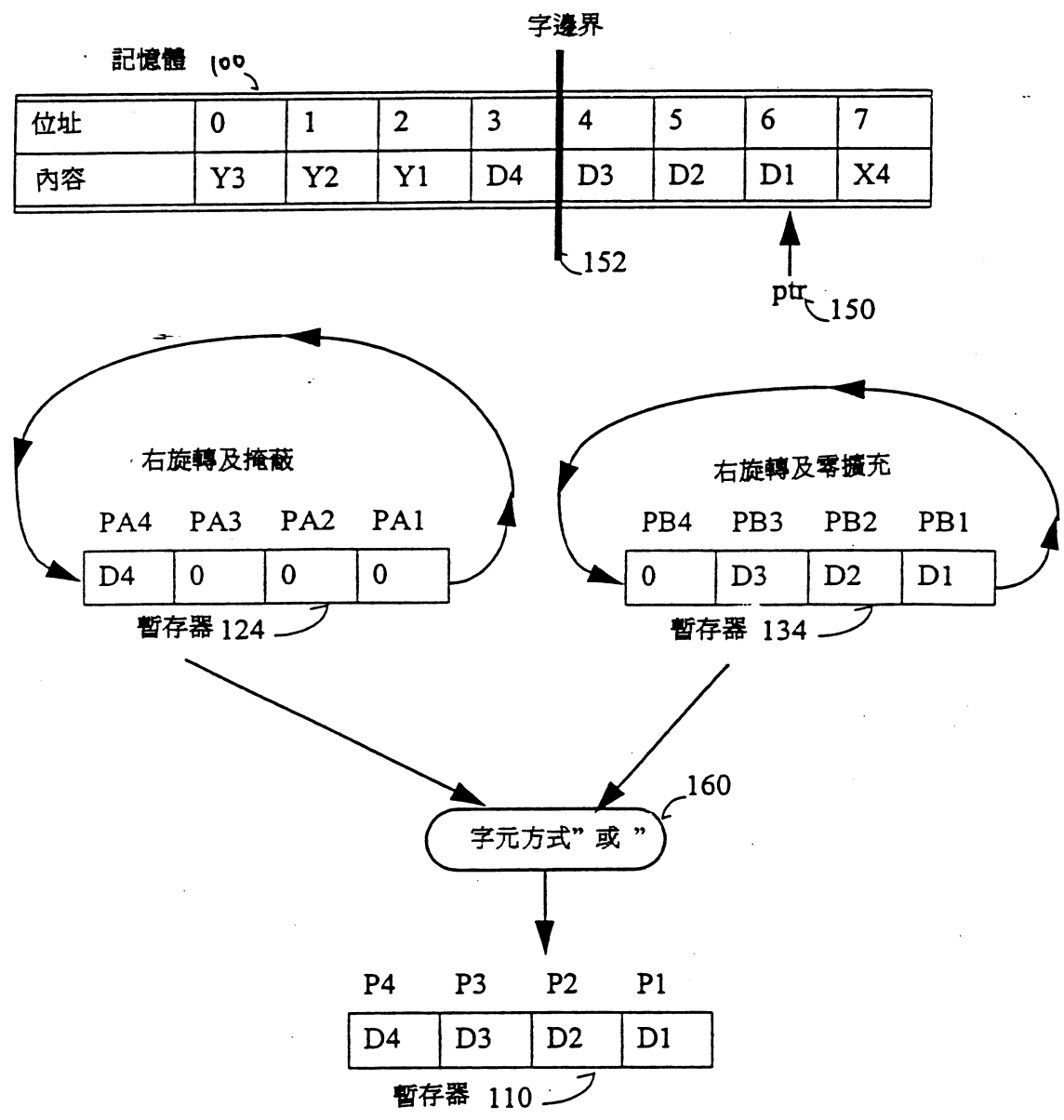
(先前技藝)

第 1 圖

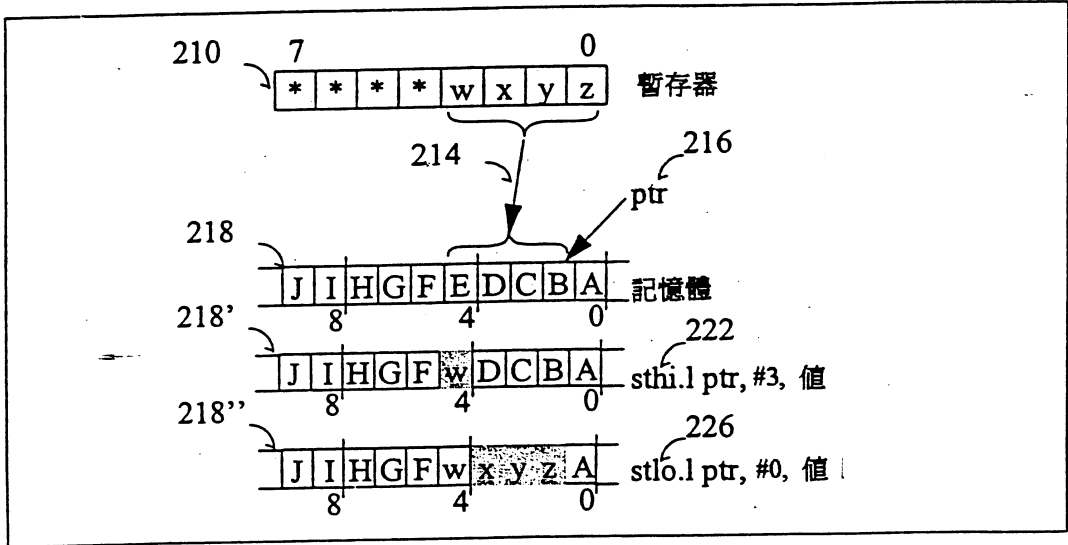


(先前技藝)

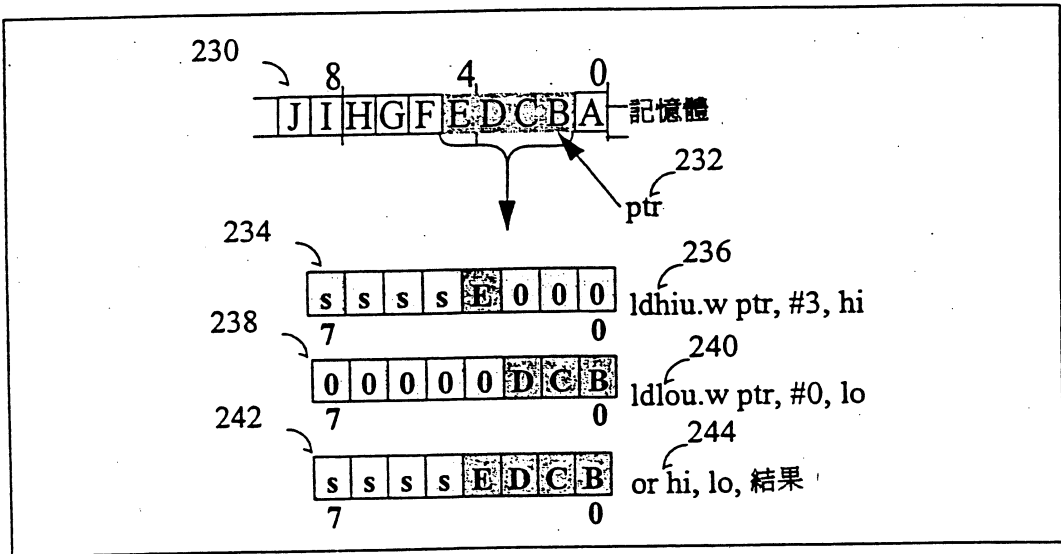
第 2 圖



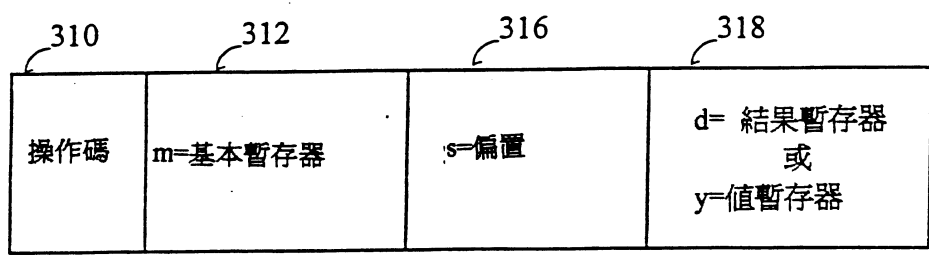
第 3 圖



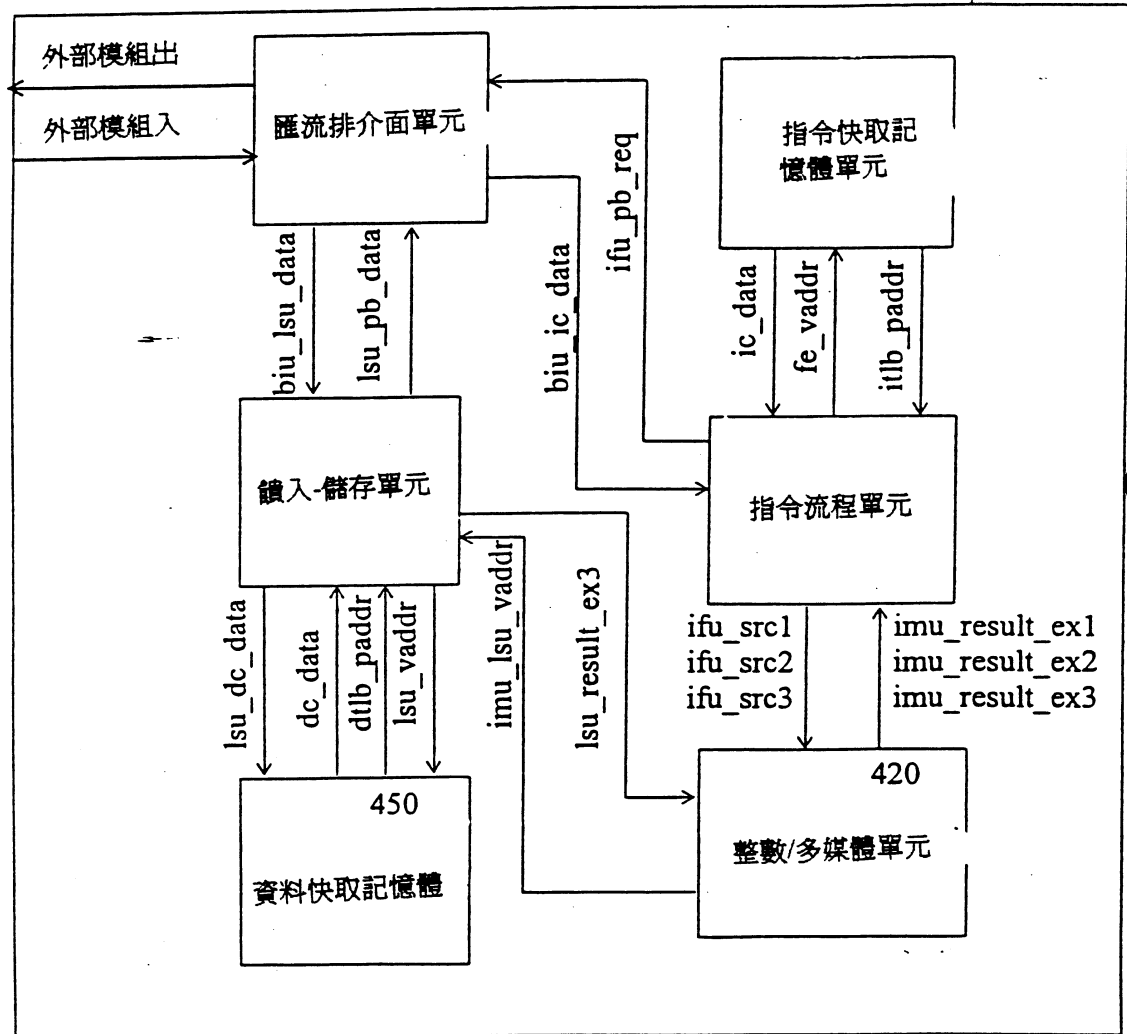
第 4 圖



第 5 圖

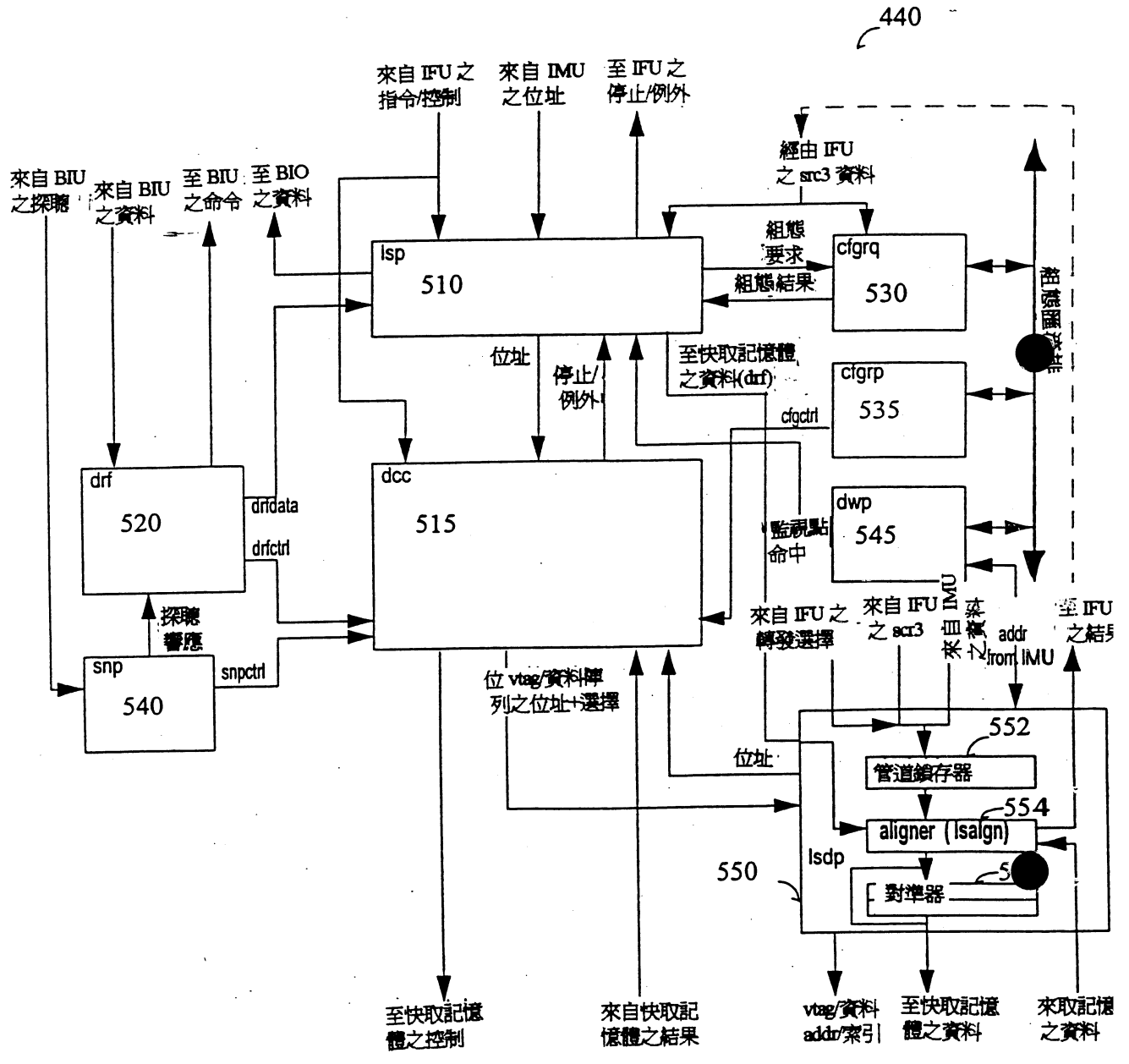


第 6 圖

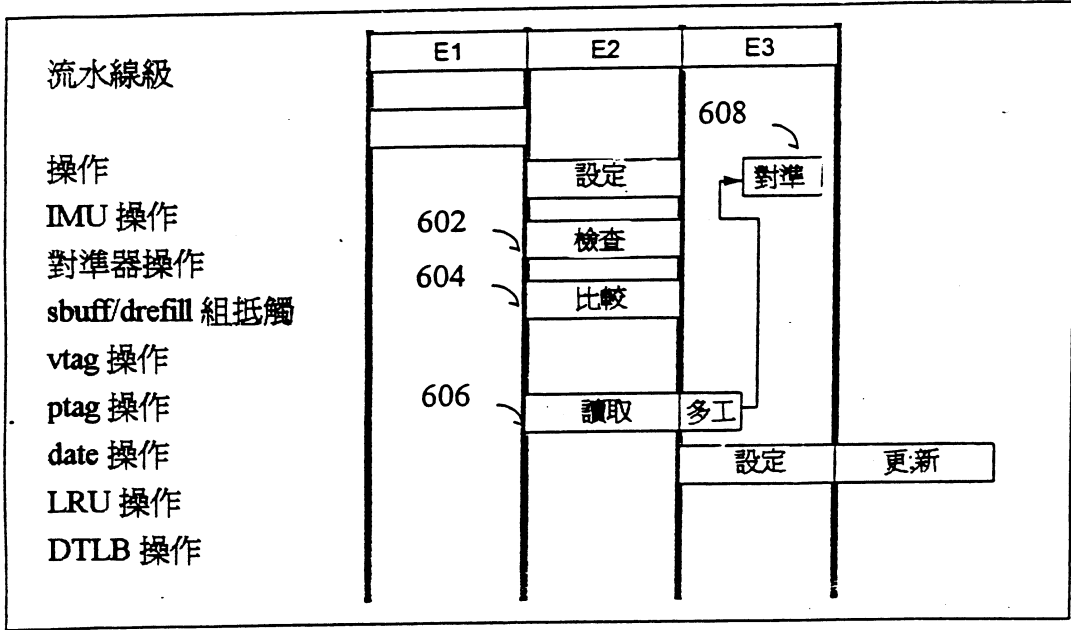


註: LSU 結果進行至利用乘法器結果予以多工處理之 IMU, 以形成在諸區段間運作之總體 imu-result_ex3. 32/64 位元匯流排

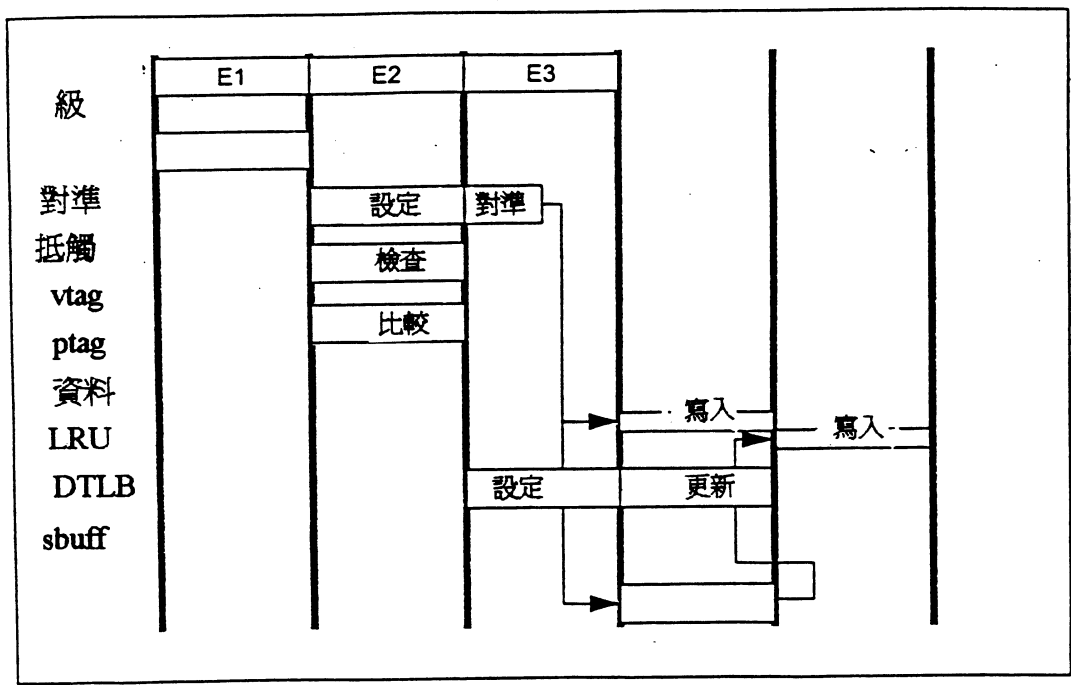
第 7 圖



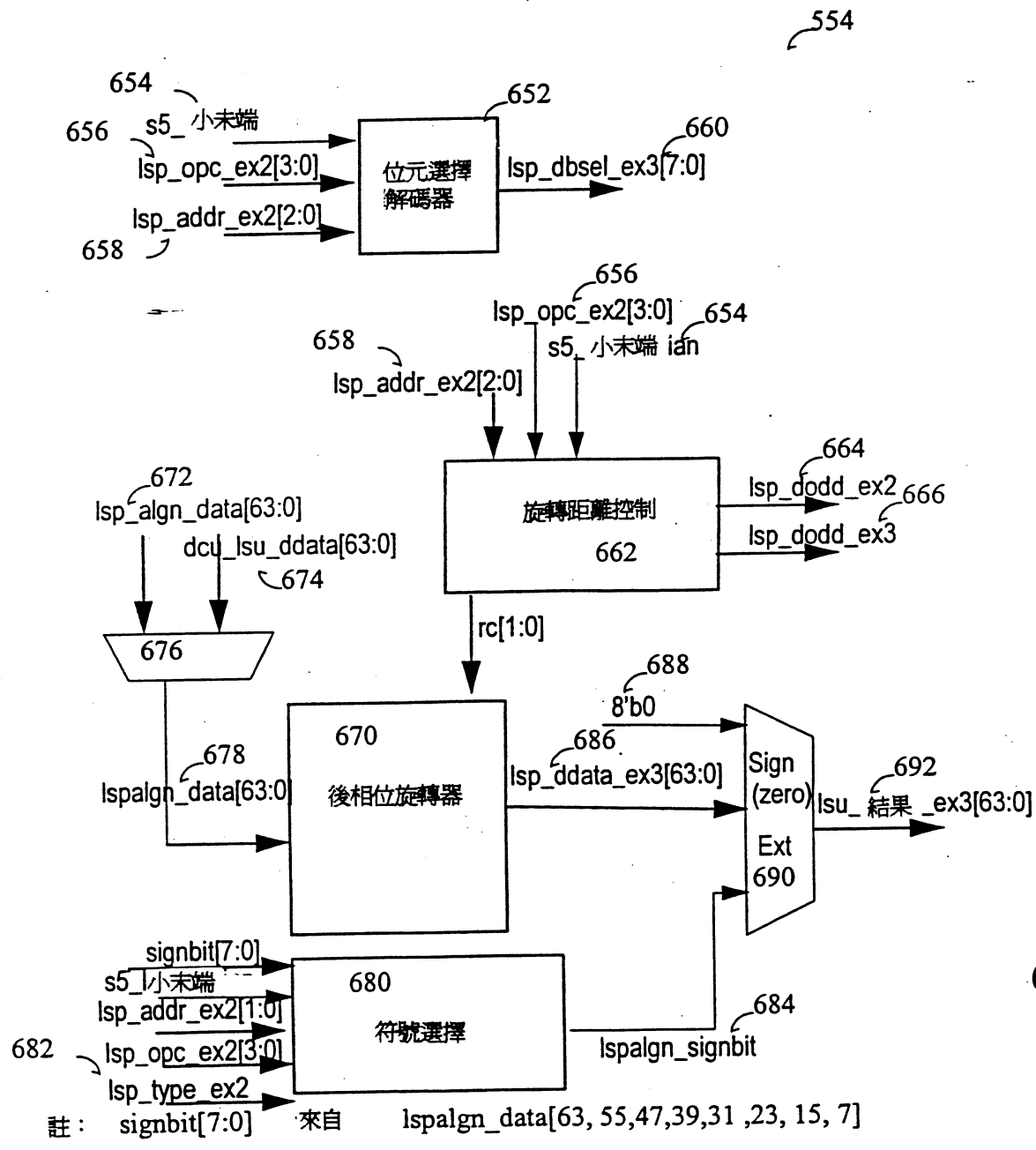
第 8 圖



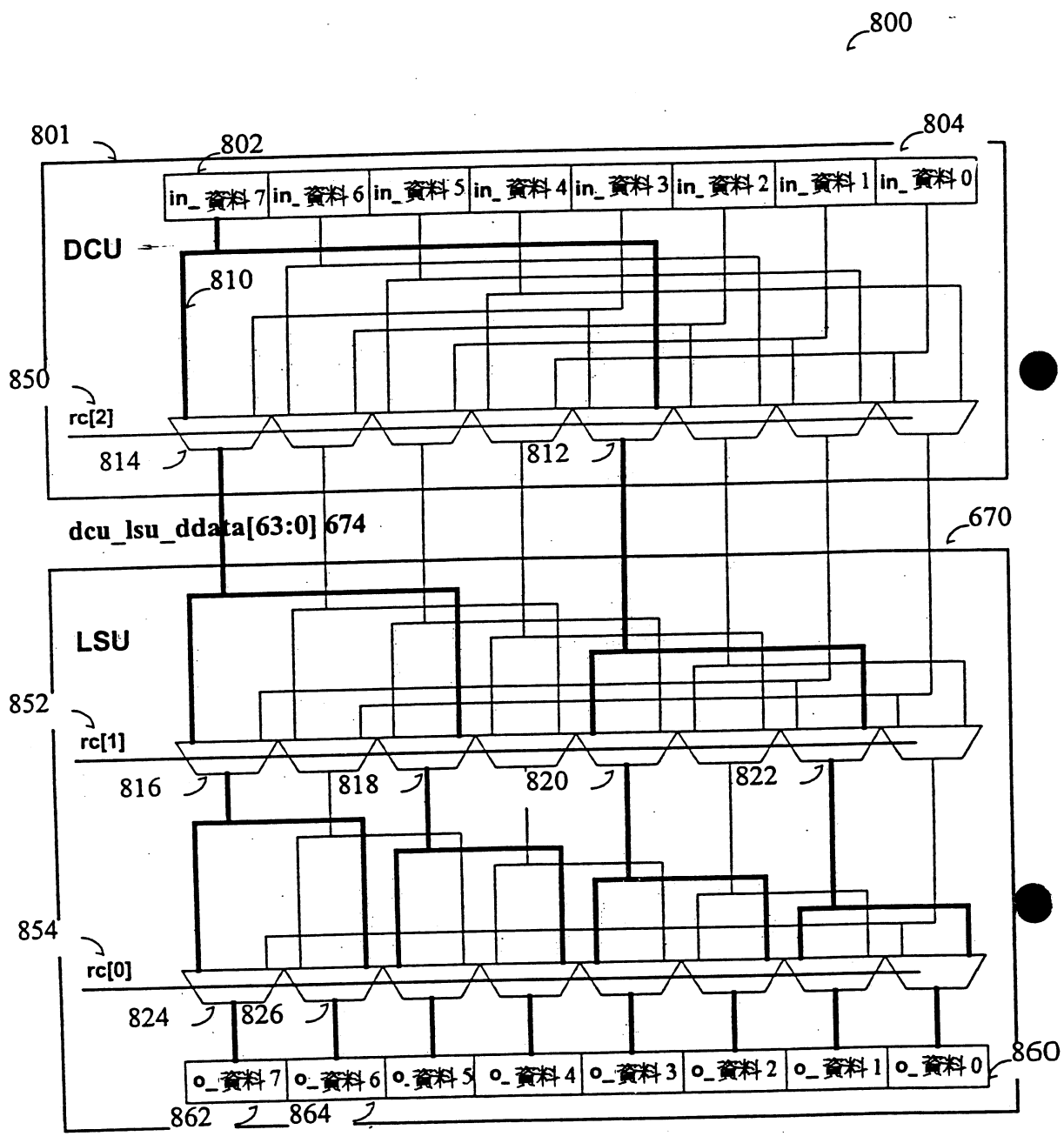
第 9 圖



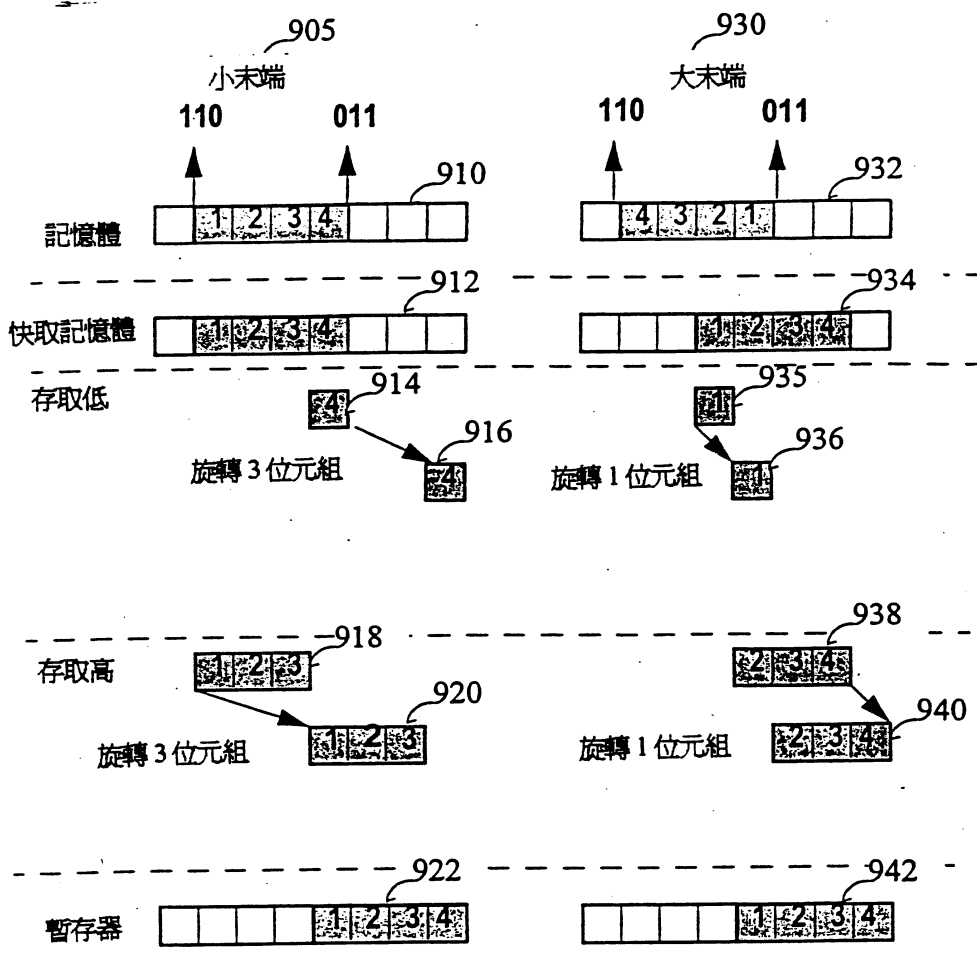
第 10 圖



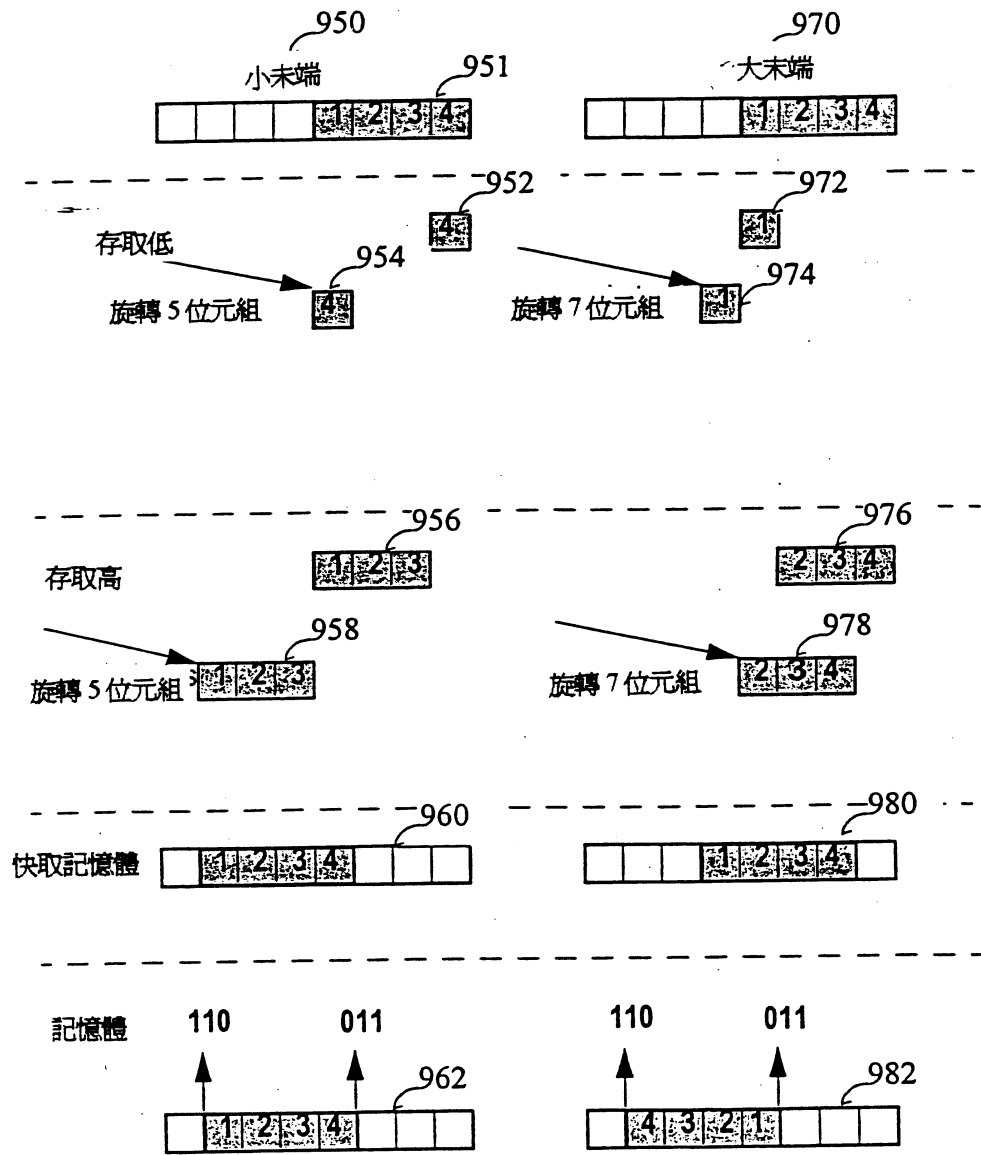
第 11 圖



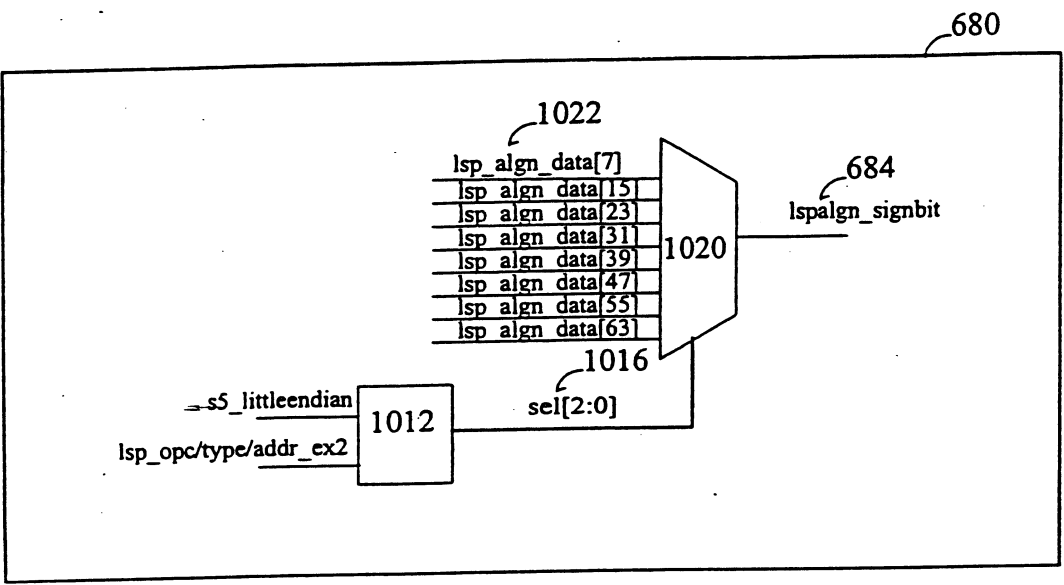
第 12 圖



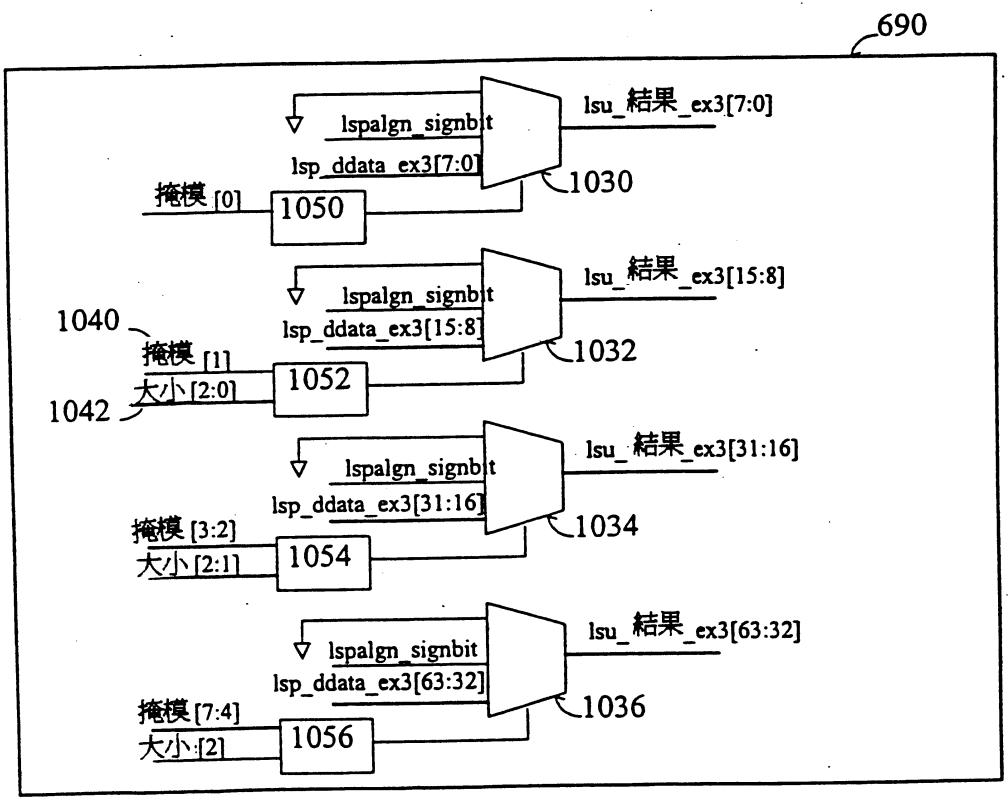
第 13 圖



第 14 圖



第 15 圖



第 16 圖