

發明專利說明書 200404205

(本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※ 申請案號：92113724

※ 申請日期：92.5.21

※ Ip C 分類：G06F 12/00

壹、發明名稱：(中文/英文)

處理器之位置產生單元

AN ADDRESS GENERATION UNIT FOR A PROCESSOR

貳、申請人：(共1人)

姓名或名稱：(中文/英文)

荷蘭商皇家飛利浦電子股份有限公司

KONINKLIJKE PHILIPS ELECTRONICS N.V.

代表人：(中文/英文)

J.L. 凡德渥

J.L. VAN DER VEER

住居所或營業所地址：(中文/英文)

荷蘭愛因和文市格羅尼渥街1號

GROENEWOUDSEWEG 1 5621 BA EINDHOVEN

NETHERLANDS

國籍：(中文/英文)

荷蘭 THE NETHERLANDS

參、發明人：(共 2 人)

姓 名：(中文/英文)

1.柯尼利斯 赫曼諾斯 凡 伯克

CORNELIS HERMANUS VAN BERKEL

2.派崔克 彼德 伊利莎白 謬偉森

PATRICK PETER ELIZABETH MEUWISSEN

住居所地址：(中文/英文)

1.2. 均荷蘭愛因和文市普羅何斯蘭路 6 號

PROF. HOLSTLAAN 6, 5656 AA EINDHOVEN, THE
NETHERLANDS

國 籍：(中文/英文)

1.2.均荷蘭 THE NETHERLANDS

肆、聲明事項：

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，其日期為： 年 月 日。

本案申請前已向下列國家（地區）申請專利：

1. 歐洲專利機構 2002年05月24日 02077034.3
2. 歐洲專利機構 2002年09月04日 02078619.0
- 3.
- 4.
- 5.

主張國際優先權(專利法第二十四條)：

【格式請依：受理國家（地區）；申請日；申請案號數 順序註記】

1. 歐洲專利機構 2002年05月24日 02077034.3
2. 歐洲專利機構 2002年09月04日 02078619.0
- 3.
- 4.
- 5.

主張國內優先權(專利法第二十五條之一)：

【格式請依：申請日；申請案號數 順序註記】

- 1.
- 2.

主張專利法第二十六條微生物：

國內微生物 【格式請依：寄存機構；日期；號碼 順序註記】

國外微生物 【格式請依：寄存國名；機構；日期；號碼 順序註記】

熟習該項技術者易於獲得，不須寄存。

玖、發明說明：

【發明所屬之技術領域】

本發明係關於一用於一處理器的位址產生單元。

【先前技術】

在信號處理中，有相當高比率的演算法使用迴圈，通常具有相當多的重複計數並包括相當少的指令和/或在相當少的資料上運算，例如一直列像素區塊。為改良處理速度，數位信號處理(DSP)配置有所謂的位址產生單元(AGU)。這些單元從一目前位址產生該下一位址。某些單元支援數個定址模式。該單元視該定址模式而決定使用多少暫存器來計算這些位址。已知位址產生單元可用於產生資料位址(有時也被稱為位址計算單元(ACU))及產生指令位址(有時也被稱為迴圈控制單元)。

WO 01/04765敘述一用於信號處理之VLIW處理器。該處理器包括四個相同的處理元素，每一個皆配置有許多功能單元。該處理元素中的每一個元素包括一當做功能單元的位址產生單元(AGU)。該AGU支援七個定址模式，即，直接定址；基底加補償定址；間接/索引定址；基底加索引定址；循環索引定址；和處理元素相對定址。對每一個定址模式而言，暫存器是用於計算該位址。對於該定址模式，計算下一個位址的該演算法，該演算法所使用的暫存器和範例硬體實現等的細節，讀者可參考WO 01/04765。該VLIW指令包括用於每一個處理元素的一指令槽(slot)。該定址模式被標示為該指令的一部份。該暫存器為該處理器環境的

一部份，並可被載入、儲存、和復原就如同其它也為該環境一部份的暫存器。用該VLIW處理器結合一寬記憶體來儲存該等VLIW。每一記憶體直列儲存一VLIW指令。每一個被直接提取和饋入該解碼邏輯的指令存取該記憶體，以平行控制多重執行單元的執行。

該週知VLIW處理器的處理元素為單一指令多重資料流(SIMD)處理器，也稱為向量處理器。通常一VLIW向量處理器對信號處理的效能相當高。需要此一效能的信號處理任務，像是用於第三代移動式通信標準的一軟體數據機，通常包括許多可向量化的子任務。該向量處理的確使此子任務能快速完成。在此環境中完成的也包括已處理一資料方塊以及稍後(通常在固定週期內)再繼續處理一新的資料方塊的情況。結果，也經常發生子任務間的交換。一環境交換需要儲存一或多個ACU的目前暫存器，該ACU係用於目前保留的任務；並且將該所儲存的暫存器載入該有關ACU的相關暫存器內，該ACU係用於最近激發或重新激發的任務。每一ACU，例如，可能包括四個暫存器。對一ACU而言，該環境交換因而可包括存檔/恢復總數為8個的暫存器。

【發明內容】

本發明目標之一為提供一更適於高效能任務的處理器架構，特別是用於移動式通信系統的信號處理。進一步目標為將此架構用於向量處理器。

為達此目標，該處理器包括：一記憶體埠，在一位址的控制下存取一實體記憶體；至少一處理單元，用於執行儲

存在該記憶體內的指令和/或用於在儲存在該記憶體內的資料上運作；一位址產生單元(此後稱"AGU")，用於產生一控制該記憶體存取的位址；與一複數個N暫存器有關的該AGU，該暫存器在位址產生機制的控制下致能該AGU產生該位址；和一記憶體單元，被一運算激發，運轉以儲存/載入該N暫存器中的k個暫存器，其中 $2 \leq k \leq N$ ，其中該記憶體單元包括一連接器，將該k暫存器連接至一記憶體字組，該字組經由該記憶體埠寫入該記憶體；和一分離一字組的分離器，該字組經由該記憶體埠從該記憶體讀取至該k暫存器內。

本發明人已了解到：隨著該等處理器的效能增加，花在該AGU暫存器架構上的時間逐漸成為瓶頸。在傳統處理器架構中，一記憶體週期只能儲存或復原一暫存器，導致浪費許多時間等待正確的資料啟動(建構)該AGU。為克服此缺點，一運算要能儲存激發至少二暫存器，例如一環境交換或明確的指令，以儲存或復原某些或所有的AGU暫存器。為達此目的，該處理器的記憶體單元包括一連接器和分離器，將一複數個該AGU暫存器映成至一記憶體字組。本質上，如本發明之該AGU和記憶體單元可用於任何處理器，例如一數位信號處理(DSP)。然偏好將該AGU和記憶體單元用於一向量處理器。截至目前為止，向量處理器尚未廣泛用於信號處理。部份原因係由於所謂的"Amdahl法則"，使得該傳統向量處理器架構應用於非100%向量化方面效率很低。此法則說明一配置有P個處理元素的向量處理器經

向量化後所得到的整體加速，為一可被向量化的碼(f)的分數函數，等於 $(1-f + f/P)^{-1}$ 。此意味著：當可向量化50%的碼時，整體加速將小於2(而非理論上的最大加速32)。這是因為無法向量化所剩餘的50%的碼，並因而對此部份的碼而言並未達到任何加速。即使可向量化90%的碼，該加速仍小於8的因數。對消費性電子應用方面，特別是可移動式通信，若可達到一顯著的加速，才有理由接受一向量處理器的額外成本。如本發明之該AGU和記憶體單元，透過對該資料處理和/或指令迴圈提供最佳支援並有效地處理跳躍和環境交換，來打破Amdahl法則。

如在該附屬申請專利範圍第2項中所述，該記憶體單元運轉以於該實體記憶體的一讀取/寫入週期內執行儲存/載入。以此法，可快速產生該AGU架構。偏好如在該附屬申請專利範圍第3項中所述，所有的AGU暫存器皆在一運算中被儲存/載入。

如在該附屬申請專利範圍第4項中所述，該處理器具有數組暫存器，其中每一組致能一AGU，以產生一位址。隨著效能增加，該處理器可平行執行更多任務，並因而可從使用一組以上的暫存器中獲益。為有效處理資料或指令迴圈，可使用每個迴圈一組暫存器的結構。為加速建構該複數組，透過將數組暫存器連接至一記憶體字組，可在一運算中儲存一組以上的暫存器。如在該附屬申請專利範圍第5項中所述，該處理器可配置有數個AGU，每一個皆具有自己的一組暫存器。該不同的AGU在功能上可相同(並因而具

有相同數目的暫存器)。若偏好如此，不同的AGU可被指派至不同的位址計算計劃並因而具有不同數目的暫存器。偏好使用一記憶體字組使至少二AGU的所有暫存器能在一運算中被儲存/載入。或者，如在該附屬申請專利範圍第6項中所述，一AGU也可與一組以上的暫存器相關，其中每一組致能一位址的產生。在此實例中，可選擇性地將該AGU連接至一組，以執行一新的位址計算。偏好可將至少二組暫存器的所有暫存器儲存在一記憶體字組內。

如在該附屬申請專利範圍第8項中所述，偏好可選擇需被建構的該組暫存器。以此法，可以最佳方式將AGU和/或數組暫存器配置給任務，其中對牽涉到該任務的一環境切換起反應，可輕易選擇所牽涉到的AGU和/或數組暫存器，並達成再架構。為簡化該選擇，AGU和/或該數組暫存器可分成單獨可選擇的群組。然後可依任務自由指派該群組。再架構發生的是一次至少一群組。

如該附屬申請專利範圍第9項中所述，該記憶體字組的寬度為該處理器可在其上運作的該最小字組的倍數。該暫存器被儲存在處理器字組界限上的該記憶體內。以此法可輕易改變該暫存器的值而不需額外的指令來設定該AGU暫存器。

如在該附屬申請專利範圍第8項中所述，偏好該處理器可在一複數個M資料元素上平行運作(例如，該處理器為一單一指令複數資料(SIMD)/向量處理器)，並且該記憶體為寬的，以便能在一記憶體字組內儲存所有M資料元素。以此

法可在一運算內儲存/載入AGU的許多暫存器。

本發明的這些和其它觀點從後述之具體實施例中將很明顯，並將參考該具體實施例來加以說明。

【實施方式】

該位址產生單元(AGU)和記憶體單元最好用於一處理器內，該處理器已最佳化以用於信號處理。此一處理器可為一數位信號處理(DSP)或任何其它適合的處理器/微控制器。以下敘述描述在一功能十分強大的純量/向量處理器內使用該單元。可獨立或結合另一處理器來使用此一處理器。圖1圖示一可使用該純量/向量處理器之較佳架構。此架構係透過一匯流排110連接三主要元件。連接這三個元件之該匯流排110可為任何適合的匯流排，例如一AMBA高速匯流排(AHB)。該主要元件有：

該可程式化純量/向量處理器120，包括數個功能單元和一區域資料記憶體(在圖1中被稱為向量記憶體)；

一微控制器或數位信號處理(DSP)子系統130，包括有限的建於晶片上的(on-chip)程式和資料記憶體；

一介面方塊140。

該純量/向量處理器120主要係用於一般"負載量大"的處理，特別是內部迴圈的處理。該純量/向量處理器包括向量處理功能。當如此，它對要執行的碼的可向量化部份提供大規模平行。大部份的所有信號處理將由該純量/向量處理器的向量區間執行。該處理器透過採用一例如32個相同處理元素的陣列執行相同指令來提供大量平行。結合一32字

組寬的記憶體介面，使得該處理器能以低成本和適度的耗電量達到所未有的可程式化效能水準。然而，由於許多演算法無法以正確形式呈現足夠的資料平行，因而要窮究此平行是不可行的。根據Amdahl法則，將該碼直接可向量化的部份向量化後，大部份的時間都是花在該剩餘的碼上面。該剩餘的碼可分成四種：

與位址相關的指令(例如使用餘數運算定址，將一指示器遞增至一循環緩衝器內)

規則的純量運算(亦即，對應該向量處理器主迴圈的純量運算)

迴圈

不規則的純量運算。

對這些種類中的每一個種類而言，碼的該部份十分依賴所執行的演算法。例如，該Golay相關器(用於P-SCH搜尋)需要許多與位址相關的指令；但對例如Rake等其它演算法而言則可能不一樣。透過使用如本發明之該AGU/記憶體單元，可最佳化與位址相關的指令和迴圈的效能。透過嚴密整合一處理器內的純量和向量處理，可最佳化該規則的純量運算掃描。本發明人對所有與第三代數據機相關的演算法的研究揭露出不規則純量運算的部分十分有限。此特質可分離該純量/向量處理器120和該微控制器或數位信號處理(DSP)130之間的任務，其中該分離的微控制器或數位信號處理(DSP)130執行該不規則的任務，並且，最好也控制該純量/向量處理器。在此較佳架構中，該純量/向量處理器

120充當一可程式化的輔助處理器(在後面也稱為CVP,輔助向量處理器)。該純量/向量處理器120和該微控制器130之間的介面處理通信(例如經由共享的記憶體)和同步化(例如經由共享的記憶體和狀態信號)。該介面最好為記憶體映成式的介面。

該介面方塊140可讓該處理器與剩餘的系統互動。在該較佳具體實施例中,使用該純量/向量處理器做為第二代/第三代移動式網路的軟體數據機(transceiver)。對此一軟體數據機功能而言,該介面方塊140可包括專用硬體,做為一前端,其主任務之一為在該微控制器130的控制下將控制和資料字組傳送至該向量記憶體,例如DMA。接著由該純量/向量處理器處理在向量記憶體內的資料。

該純量/向量處理器120可為該匯流排110的從屬處理器,而該微控制器130和該介面方塊140(可包括一DMA單元)則充當一主處理器。所有與該CVP的通信,不管是程式、資料或控制,最好皆為記憶體映成式的。該記憶體可為一非建於晶片上的(off-chip)動態隨機存取記憶體(DRAM),並且此DRAM也可被該純量/向量處理器用來做為(非)交錯式記憶體。

在敘述中主要使用"位址計算單元"或ACU。為了該敘述的目的,這也被視為與"位址產生單元"或AGU相等。該敘述主要專注於使用此單元來計算資料位址。熟諳先前技藝者也能使用相同功能來計算指令位址("迴圈控制")。

圖2圖示如本發明該處理器之主要結構。該處理器包括一

管道化式向量處理區間210。為支援該向量區間的運算，該純量/向量處理器包括一純量處理區間220，被裝置成能平行於該向量區間運算。最好該純量處理區間也為管道化式。為支援該向量區間的運算，該向量區間的至少一功能單元也提供該純量區間對應部份的功能。例如，一位移功能單元的向量區間在功能上也能位移一向量，其中一純量元件係由該位移功能單元的純量區間供應(或被遞送至該區間)。如此，該位移功能單元涵蓋該向量和該純量區間二者。因而，至少某些功能單元不僅具有一向量區間也具有一純量區間，其中該向量區間和純量區間可透過交換純量資料而合作。一功能單元的向量區間提供該原始處理能量，其中該對應的純量區間(亦即該相同功能單元的純量區間)透過供應和/或消耗純量資料而支援該向量區間的運算。用於該等向量區間的向量資料係由一向量管道化供應。

在圖2的較佳具體實施例中，該純量/向量處理器包括以下七個特殊功能單元。

指令分散單元(IDU 250)。該IDU包括該程式記憶體252，讀取連續的VLIW指令並將每一指令的7個區段分配給該7個功能單元。最好，它包括一迴圈單元，該迴圈單元能支援多達三個巢狀階層的零負荷迴圈。在該較佳具體實施例中，該單元並不支援分支、跳躍或中斷。該起始程式計數器係從該限制描述子載入，詳述如下。

向量記憶體單元(VMU 260)。該VMU包括該向量記憶體(未顯示於圖2)。在每一指令期間，該單元可從該向量記憶

體傳送一列或一向量，或接收一列至該向量記憶體。該相同指令可額外指定一純量傳送運算和/或一接收運算。該VMU為唯一連接至外界的功能單元，亦即至該外部匯流排110。

該碼產生單元(CGU 262)。該CGU擅長有限欄位計算/數學。例如，可用該CGU產生CDMA碼小段向量以及相關功能，例如頻道編碼和CRC。

算術邏輯-乘及加單元(ALU-MAC)單元(AMU 264)。該AMU擅長規則的整數和定點計算/數學。它支援向量內的運算，其中計算/數學係在多向量上以順著元素的方向執行。在一較佳具體實例中，該AMU也提供某些向量間運算，其中計算/數學係在一單一向量內的數個元素上執行。

ShuFfle單元(SFU 266)。該SFU可根據一特定的弄亂樣式重新安排一向量的元素。

向左位移Shift-Left單元(SLU 268)。該SLU可將該向量的數個元素移位一單位，例如向左移一個字組，二個字組或四個字組。所產生的純量被提供給其純量區間。取決於所發布的SLU向量運算的類型，該所消耗的純量可以是0，或者取自其純量區間。

向右位移Shift-Right單元(SRU 270)。該SRU類似SLU，但向右位移。此外，它具有合併連續結果的能力，該等結果得自該AMU上的向量間運算。

下表顯示所有功能單元FU皆具有一功能性向量區間210，其中某些單元並未具有一控制區間230或純量區間

220。

| 功能單元 | 控制 | 純量 | 向量 |
|-------------------------|-------|-------|--------------------|
| | | | |
| 指令分散單元 | 循序,迴圈 | | 指令分配 |
| 向量記憶體單元 | 位址計算 | 純量i/o | 向量輸入/輸出(i/o) |
| 碼產生單元 | | | 產生碼向量 |
| 算術邏輯-乘及加 (ALU-MAC)單元 | 索引 | 廣播 | 向量間:ALU,MAC,mul,.. |
| | | 區段 | 向量內: add, Max |
| 弄亂樣式(Shuffle)單元 | | | 向量弄亂樣式 |
| 向左位移(Shift-Left)單元 | | 純量i/o | 向量位移 |
| 向右位移(Shift-Right)單元 | | 純量i/o | 向量位移 |

如本發明之該純量/向量處理器以二主要方式應用指令階層平行：

向量處理，其中一單一指令係在(純量)資料的數個向量上運算。此方法也稱為單一指令流，多重資料流或SIMD。

多重功能單元的平行處理，每一個皆在向量上運算。可視為一(限制的)種形式的VLIW指令階層平行。

請注意此二種形式的指令階層平行皆為獨立的，並且其效果為累加的。

FU內通信

該等功能單元(FU)平行運算。每一FU能接收和傳送向量資料。許多FU也能接收和傳送純量資料。

| FU | 來源 | vmu | cgu | amu | sfu | slu | sru |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 目標 | #輸入 | | | | | | |
| vmu | 1 | ! | ! | ! | ! | ! | ! |
| cgu | 1 | ! | | ! | ! | ! | ! |
| amu | 2 | ! | ! | ! | ! | ! | ! |
| sfu | 1 | ! | ! | ! | ! | ! | ! |
| slu | 1 | ! | | ! | ! | ! | ! |
| sru | 1 | ! | | ! | ! | ! | ! |

所有功能單元皆平行運算。一旦接收一指令中該等功能單元的區段，他們即輸入、處理並輸出資料，該資料包括向量資料和，其中只要能應用，純量資料兩者。在FU內，該通信僅限於該等純量區或向量區間內(FU內通信)。亦即，除了該IDU外，所有FU的向量區間係由一管道化連接。在一較佳具體實例中，此管道化可根據指令基礎來建構。為達此目的，偏好該FU由一互連網路互連，原則上可讓每一向量區間在每一週期內從任何其它的向量區間接收一向量。此特色致能，除了其它功能外，該FU (除了該IDU外)的任意管道化的建立。該有助於向量通路的功能單元中的六個可輸出一向量，並在每一時脈週期內將該向量平行傳送給其它單元。這些功能單元也能從另一單元接收一向量。該網路接近完全連接。只有沒有意義的連結才會被省略。該AMU能同時接收二向量。如圖2所示，該網路最好由每一FU形成，每一單元皆被當成一單一來源(以一disc標示)

連接至一網路通路。該網路被連接至所有其它的通路，做為一單一汲(以三角形標示)。該FU的VLIW指令的區間標示它應從哪一條通路完成一向量執行。以此法，可根據一指令基礎建構該管道化。每一通路可移轉一完整向量，例如使用256條平行線。類似地，可由一單獨的管道化連接至少某些該FU的純量區間。最好，此管道化也可根據指令基礎建構。該FU純量區間內的互連網路可為部份的，亦即至少一FU的純量區間無法傳送或接收純量。因而，可指定較少的管道化次序。可獨立建構該純量和向量管道化。例如，透過在該相關的VLIW區間內標示要被該功能單元讀取的純量管道化和向量管道化二者。

該不同功能單元的控制區間內並未指定可連接性。這些控制區間從該IDU接收一區段的該VLIW指令，更新本身的狀態，並控制其各別的純量和向量區間。

FU間通信

在一FU內，這些區間之間(FU間通信)有嚴密的互動。該互動為該FU運算的整數部份。該SLU和SRU皆為實例，其中所產生的和/或所完成的純量被供應至/取自該FU對應的純量區間部份。

通常在單一週期內執行數個指令。在該向量記憶體의 擁塞會造成例外並出現停滯週期。

資料寬度

在一較佳具體實例中，該純量/向量處理器支援一複數個資料寬度和資料類型，如圖3所示。記憶體定址的基本單位

為一字組，也稱為單一字組。偏好資料寬度為單一字組 (W)、二個字組 (DW ，或 $2W=16$ 位元)、或四個字組 (QW 或 $4W=32$ 位元)。一字組的大小為 $W=8$ 位元。偏好三種大小的純量：(單一)字組，二個字組，和四個字組。一向量具有一固定大小的 P_Q 四個字組。最好能以下列三種格式之一建構字組：

P_Q 個四字組大小的元素，

$P_D = 2P_Q$ 個二字組大小的元素，

$P_S = 2P_D = 4P_Q$ 個(單一)字組大小的元素。

該向量-元素索引範圍為 $[0 \dots 4P_Q - 1]$ 。因而二個字組具有偶數個索引，並且四個字組的索引為四的倍數。圖3提供該資料大小的概觀。該架構的為具有完整可測的 P_Q ，並且由任何 $P_Q \geq 1$ 的向量大小來定義。然而，大部分的情況較偏好 P_Q 為2的乘冪。在該較佳具體實施例中， P_Q 為8，意指一資料通路寬度和記憶體寬度為32字組。

指令

一CVP指令可為一控制指令亦或一VLIW指令。控制指令可，例如，為零負擔迴圈起始/起動initialization。沒有分支、跳躍、或副程式。一VLIW指令被分割成數個區段，其中每一指令區段指示該對應的功能單元要執行的運算。該區段可進一步被細分成一部份給向量區間，及一部份給純量區間(若出現的話)。對此二部份而言，該區段也包括要使用哪一個網路部份來接收資料的資訊(該向量區間的一或多個向量以及該純量區間的一或多個純量)。

該純量/向量處理器的狀態

該CVP的狀態為其數種功能單元的結合狀態。在該較佳具體實施例中，它包括：

該向量記憶體(部份的VMU)；

該程式記憶體(部份的IDU)；

向量暫存器(所有功能單元)；

純量暫存器(大部份的功能單元)；

控制暫存器，包括該程式計數器，和位址補償暫存器。

除了該程式設計師可看見的暫存器之外，一CVP實現通常包括用於管道化和快取的額外暫存器(向量，純量，和控制)。這些暫存器並非該CVP指令集架構的一部份。

某些該(向量，純量，和控制)暫存器為所謂的架構暫存器。一架構暫存器的內容僅能從該向量記憶體載入；並無其它方式可改變其值。一架構暫存器支援該等功能單元的架構，並通常定義一功能參數。透過儲存這些在架構暫存器內的"半常數"功能參數而大量減少該指令寬度和記憶體流量。

下表顯示該CVP狀態的元件概要。

| FU | 控制通路 | | | | 純量通路 | | | | 向量通路 | | | |
|-----|------|---|----------|---|------|---|----|---|-----------|------|-----|---|
| | 資料 | | 架構 | | 資料 | | 架構 | | 資料 | | 架構 | |
| vmu | 補償 | 5 | 位址 cu | 8 | | | | | 資料 記憶體 | 2048 | | |
| cgu | | | | | 計數器 | 3 | 碼 | 3 | 狀態 | 6 | 遮罩 | 2 |
| | | | | | | | | | | | 多項式 | 2 |

| | | | | | | | | | | | | |
|-----|----|---|------|---|----|---|------|---|-------|------|------|---|
| amu | | | | 1 | 接收 | 1 | 區段大小 | 1 | 暫存器檔 | 16 | | |
| sfu | | | | | | | | | 暫存器 | 1 | 弄亂樣式 | 4 |
| slu | | | | | 接收 | 1 | | | 暫存器檔 | 2 | | |
| sru | | | | | 接收 | 1 | | | 暫存器檔 | 2 | | |
| idu | pc | 1 | 迴圈cu | 2 | | | | | 程式記憶體 | 2048 | | |

可從該向量記憶體載入所有程式設計師可看到的暫存器。除了該架構暫存器之外，所有的暫存器皆可存入該向量記憶體內。透過在一指派工作的尾端儲存該等CVP暫存器，並稍後將這些暫存器復原，該CVP可持續一特別任務，就好像同時並未執行其它的指派工作。這些儲存和復原運算為隨選的，可能為部份的，並必須明確加以程式化。

該記憶體單元和AGU

圖4圖示該記憶體單元(VMU 400)之方塊圖。在以下所述的較佳具體實例中，在一向量處理器內使用該記憶體單元來結合一實體記憶體，其寬度能儲存一整個向量。較偏好同樣的觀念也能應用至純量處理器，例如傳統的數位信號處理(DSP)。該VMU包括並控制該向量記憶體410，供應其它功能單元大量的資料頻寬。該實體向量記憶體410最好係根據一單埠的靜態隨機存取記憶體SRAM。由於通常無法取得內建為 $P_s * W$ 寬度的SRAM，該實體記憶體可由一或多排的寬隨機存取記憶體(RAM)平行裝置所形成。

該VMU 400包括至少一位址計算單元(ACU) 420，該等單

元支援自動位址產生。參考圖2的整體架構，假設該ACU位於VMU 260的控制區間230內。偏好該ACU不需實際位於VMU內但也能連接至該VMU。最好，該ACU支援如同那些在傳統數位信號處理(DSP)內的定址模式(位址產生演算法)。該ACU不需使用該處理器的主要資料通路即可對每個指令執行一或多個位址計算。例如，一純量的位址可為在每一純量讀取存取後的遞增後位址。這可讓位址計算與資料上的計算/數學運算平行發生，改良該處理器效能。取決於所支援的該組定址模式，此一ACU需存取至一些暫存器。例如，相對定址，亦即對一所謂的基底位址定址，需要：

一基底暫存器 base

一對該基底位址的補償，儲存在一補償暫存器內 offs

該補償的一前/後遞增，由一儲存在一遞增暫存器內的值 incr

對一位址的 modulo(取餘數)定址，儲存在一限制暫存器內 bound

有了此組定址模式，可支援以下情況。假設一補償暫存器 offs。在每一次記憶體存取(讀或寫)位址 $base + offs$ 之後，根據 $offs := (offs + incr) \text{ Modulo } bound$ 更新暫存器 offs。因而，offs 會經常改變(在每次存取後)，而儲存於 base，incr，和 bound 內的值則不會經常改變。通常會在一程式迴圈前激發後三個暫存器。該組暫存器的起始也被敘述為"該ACU架構"。為了僅可能避免過長的指令及避免將位址計算

上的指令分離，該VMU的控制區間最好包括一些位址計算單元。每一位址計算單元(ACU)可與一組(位址)暫存器產生關連，並與一位址計算("遞增")運算產生關連。該ACU和該組暫存器間的關連可為固定的("硬體線路")或可建構的，若在指令階層也偏好如此的話。

假設該向量記憶體包括 2^L 直列，一純量或向量位址需要 $L + 2 \log 4P_Q$ 位元。例如，若 $P_Q = 8$ 且 $L = 12$ ，則需要17位元。該ACU暫存器的大小與該位址相同。若偏好如此的話，某些暫存器可能較小。例如，遞增可能僅限於相當少的步驟，例如8位元。最好所有的暫存器大小皆相等。下表圖示一較佳具體實施例，其中一組ACU暫存器包括四個位址暫存器：

| 名稱 | #位元 | 說明 |
|-------|--------|---|
| Base | 24無正負號 | 位址基底暫存器 |
| Offs | 24無正負號 | 來自基底的位址補償 |
| Incr | 24正負號 | 遞增值($-\text{bound} < \text{incr} < \text{bound}$) |
| Bound | 24無正負號 | Upper bound上限 |

表內也標示所偏好的位址範圍和類型(正負號/無正負號)。在此架構中，每一個ACU暫存器為24位元。如將在以下詳細討論的，該ACU暫存器可被儲存至該記憶體410/從該記憶體410載入。一暫存器被儲存於該記憶體內時，為簡化該暫存器的改良，可選擇一與存取該記憶體的基本單位對準的暫存器寬度，亦即在8位元字組界限。因而，對該說明性的記憶體大小來說，使用24位元而非17位元的暫存器

才夠。應了解對某些位址計算來說，少於三個暫存器也夠用。因而可能使一或多組暫存器專用於此等計算，在暫存器內提供一儲存/節省。

在該較佳具體實施例中，該VMU包括八組ACU暫存器。此致能一功能單元的該向量區間和純量區間不同的位址計算。它也對一演算法內的多重資料流提供有效支援，其中每一資料流皆有自己的指示器(並且因而有計算和更新該指示器的位址計算)。傳統上，建構一獨立組的ACU暫存器每一組都要花一些時脈週期。如此，建構ACU暫存器所需的時間會成為瓶頸。為克服此一建構遲延，在單一運算中可建構屬於一組ACU暫存器的至少二暫存器。這可透過將所有那些ACU暫存器映成在單一記憶體字組上，例如一向量，並透過使用從該向量記憶體到該ACU記憶體的專用載入和儲存指令而實現。偏好在一時脈週期的單一運算內建構一組ACU暫存器的整組相關暫存器。若該記憶體寬度允許，偏好在一運算中建構一組以上ACU暫存器的數個暫存器。在該實例中，一組四個ACU暫存器需要 $4*24 = 96$ 位元。如前述，最好一向量為256位元寬。在此一實例中，透過將多組內的該等暫存器映成至一記憶體直列(向量)，可更進一步增加該ACU的建構速度。在該實例中，可將二組ACU暫存器映成至一向量。這也圖示於圖5內。圖示一向量500和該四個字組界限。也圖示二組ACU暫存器510和520。在該實例中，該ACU暫存器為24位元，並因為如此而未對應該向量的標準資料大小之一。為也能經由該向量記憶體輕易

存取該個別的ACU暫存器，用於將該等ACU暫存器載入/儲存至該記憶體的特別指令能確保該個別ACU暫存器能對準字組界限(在該實例中，該24位元暫存器係對準四個字組的界限上)。取決於該ACU暫存器大小和該向量大小，熟諳先前技藝者將能定義一最佳映成。例如，採用16位元的數個ACU暫存器和一256位元向量，使得將四組ACU暫存器映成至一向量成為可能。在該指令中，需標示要被儲存/載入的該等ACU暫存器組的數目。可使用單獨的或組合的指令來載入單一ACU暫存器組或一群ACU暫存器組。該群要被載入/儲存的ACU暫存器組可為固定的。例如，若該組係以0到7的數字標示，則可形成四個固定群組，其中該群組的數目為該組的二最重要位元。若是如此，例如透過允許在該載入/儲存指令內指定一組以上，也可以動態方式形成偏好的群組。

圖6圖示一具有四個ACU暫存器的組610，固定連接至一ACU 620。經由通路630和632將該暫存器的資料供應給該ACU。該ACU提供該所計算出的位址做為輸出640，並且在該輸出642提供該已更新的資料給該等暫存器。在具有八個獨立組的ACU暫存器的較佳具體實施例中，這可被複製八次(未顯示於圖中)。圖7圖示一替代裝置。此裝置圖示八組ACU暫存器，編號為710到717。可使用不同數目的ACU。在此實例中，係使用三個ACU，編號為720，721和722。該ACU可動態連接至該組暫存器之一。若偏好如此，可呈現一完全互連730，以致能將每一個該等ACU連接至該等暫存

器組中之一。當然，該互連730不需完全(例如，一ACU可僅連接三到四組)。該互連740確保該已更新的暫存器值可被供應回該所偏好的組。互連704鏡像互連730。

以下詳細敘述所偏好的VMU和ACU。此敘述假設每一ACU固定與一組ACU暫存器關連。該VMU可被分割為一控制，純量和向量區間。根據一摩爾機器(Moore machine)模型/式將這些區間模式化，包括五個元素：輸入，輸出，狀態，下一狀態的功能，和輸出功能。該Moore machine的狀態由所取得的記憶體和/或暫存器來決定。對每一功能單元而言，提供一定義所有許可的轉移的表列，包括該對應的保護。一保護必須為真，該轉移才能發生。該轉移定義該Moore machine的下一狀態功能和輸出功能。在該表列中，使用中括弧來選擇在一向量內的元素。例如： $v[p]$ 代表向量 v 的元素 p 。

在單一VMU指令中，該向量記憶體單元可支援多達四個同步"子運算"：

傳送一向量，或傳送一直列，或接收一直列來回於VM位置(向量子運算)；

從一VM位置傳送一純量(傳送純量子運算)；

接收一純量至一VM位置(接收純量子運算)；

修正一位址計算單元的狀態/輸出(ACU子運算)。

用於那四個同步指令中每一指令的參數皆在以下的VMU指令(VWU_cmd)中提供，其中由vopc，aid_v和ainc_v指定該向量子運算；由sopc，aid_s，ainc_s和size指定該傳送純

量子運算；由 `srcv`，`aid_r`，和 `ainc_r` 指定該第三接收純量子運算；而由 `aopc`，`aid_a`，和 `imm_addr` 指定該 ACU 子運算。該第四運算直接控制該等 ACU 之一，而當做副作用的另外三個運算也可控制一 ACU，將詳述於下。

```

VMU_cmd      = (vopc, aid_v, ainc_v, sopc, aid_s, ainc_s, size, srcv, aid_r, ainc_r,
                aopc, aid_a, imm_addr)

vopc         = NOP | SENDL | SENDV | RCVL_CGU | RCVL_AMU |
                RCVL_SFU | RCVL_SLU | RCVL_SRU

aid_v       = {0, ..., 7}
ainc_v     = NOP | INC

sopc       = NOP | SEND
aid_s     = {0, ..., 7}
ainc_s    = NOP | INC
size      = WORD | DWORD | QWORD

srcv      = NONE | VMU | AMU | SLU | SRU
aid_r     = {0, ..., 7}
ainc_r    = NOP | INC

aopc     = NOP | IMM | LDBASE | LDOFFS | LDINCR | LDBOUND
aid_a    = {0, ..., 7}
imm_addr = {0.0, ..., 524288.31} | {-262144.0, ..., 262143.31}

```

取決於該子運算的數目和該位址順序的連續性，該 VMU 指令可能要花上許多不同的時脈週期。

該 VMU 輸入/輸出為：

| 輸入 | 說明 |
|---------|-------------|
| Cmd | VMU 指令 |
| rcv_amu | AMU 向量接收匯流排 |

| | |
|-----------|------------|
| rcv_cgu | CGU向量接收匯流排 |
| rcv_sfu | SFU向量接收匯流排 |
| rcv_slu | SLU向量接收匯流排 |
| rcv_sru | SRU向量接收匯流排 |
| s_rcv_amu | AMU純量接收匯流排 |
| s_rcv_slu | SLU純量接收匯流排 |
| s_rcv_sru | SRU純量接收匯流排 |

| 輸出 | 說明 |
|-------|---------|
| Snd | VMU向量結果 |
| s_snd | VMU純量結果 |

此外，尚有二純量埠(一傳送，一接收)連接至該外部匯流排。該微控制器 130 的任務為使這些記憶體存取與 CVP 指令同步。

該 VMU 向量區間包括該實體向量記憶體 410：

| 名稱 | 說明 |
|---------------|--------------------------|
| mem[4096][32] | 向量記憶體：4096 列，每一列有 32 個字組 |

向量子運算

請注意向量子運算無法存取該純量記憶體。因而，向量子運算會忽略最重要的位址位元。該 VMU 的向量區間支援七個子運算，於該指令的 VOPC 欄中加密：向量傳送

(SENDV)，列傳送 (SENDL)，和五列接收子運算 (RCVL_CGU, RCVL_AMU, RCVL_SFU, RCVL_SLU, 和 RCVL_SRU)。無法同時執行此七個子運算。一次只能指定一子運算。身為該接收來源的功能單元在對應的列接收子運算內很清楚的被加密。一對應的位址計算單元指定每一子運算的讀取位址或寫入位址。所有向量子運算共享該 AINC_V 欄。該欄將被傳送至在該 AID_V 欄內加密的 ACU。以此法，所指定的 ACU 被當做為一向量子運算的副作用來控制。該 AINC_V 欄指定是否該被影響的位址計算單元應執行一遞增後運算。

| 保護 | 轉移 |
|-----------------|------------------------------------|
| vopc = NOP | None |
| vopc = SENDL | snd = mem.line[acu[aid_v].out] |
| vopc = SENDV | snd = mem.vector[acu[aid_v].out] |
| vopc = RCVL_CGU | mem.line[acu[aid_v].out] = rcv_cgu |
| vopc = RCVL_AMU | mem.line[acu[aid_v].out] = rcv_amu |
| vopc = RCVL_SFU | mem.line[acu[aid_v].out] = rcv_sfu |
| vopc = RCVL_SLU | mem.line[acu[aid_v].out] = rcv_slv |
| vopc = RCVL_SRU | mem.line[acu[aid_v].out] = rcv_sru |

請注意，該等運算被視為傳送(或接收)行為而非涉及目的地(或來源)的載入(或儲存)行為。該載入行為通常由其它功能

單元內的運算指定。一列傳送在功能上等於一具有相同位址的向量傳送。列傳送子運算通常用於建構功能單元，或復原在該各種暫存器內一任務的狀態。透過導入一用於列傳送的特別模式，後續向量傳送("向量流")的存取次數可經由有效使用快閃記憶體 caches 而達到最佳狀況。

該 VMU 的純量傳送子運算係在該指令的 SOPC 欄內加密。它僅支援一子運算：純量傳送 (SEND)。由該 AID_S 欄指定的位址計算單元會指定讀取位址。該指令的 AINC_S 欄指定是否此位址計算單元應執行一遞增後運算。以此法可間接控制一第二 ACU (第一個係由該向量子運算控制)。該指令的 SIZE 欄決定純量子運算的運算域大小 (WORD, DWORD 或 QWORD)。

| 保護 | 轉移 |
|-----------------------------|-----------------------------------|
| sopc = NOP | None |
| sopc = SEND && size = WORD | S_snd = mem.word[acu[aid_s].out] |
| sopc = SEND && size = DWORD | S_snd = mem.dword[acu[aid_s].out] |
| sopc = SEND && size = QWORD | S_snd = mem.qword[acu[aid_s].out] |

該 VMU 的純量接收子運算在該指令的 SRCV 欄內加密。若其值為 NONE，則不會執行純量接收。不然，該指令的 SRCV 欄會決定將使用哪一個功能單元來作為該純量接收的來源。該 AID_R 欄指定的位址計算單元會指定寫入位址。以此法可間接控制一第三 ACU。該指令的 AINC_R 欄指定是否

此位址計算單元應執行一遞增後運算。該來源純量的大小決定純量接收子運算的運算域大小 (WORD, DWORD 或 QWORD)。

| 保護 | 轉移 |
|-------------|--|
| Srcv = NONE | None |
| Srcv = VMU | Mem.scalar[acu[aid_r].out] = s_rcv_vmu |
| Srcv = AMU | Mem.scalar[acu[aid_r].out] = s_rcv_amu |
| Srcv = SLU | Mem.scalar[acu[aid_r].out] = s_rcv_slu |
| Srcv = SRU | Mem.scalar[acu[aid_r].out] = s_rcv_sru |

可將該傳送和接收子運算結合至一純量移動運算內，從一 VM 位置到另一個 VM 位置。一對應位址計算單元指定每一存取之位址。

如上述，以此法，可彈性指派該等 ACU 中的每一個給任一向量、純量傳送和純量接收的 VMU 子運算。以此法，三個 ACU 皆可在每一指令內運算。為避免衝突，一限制為每一 ACU 僅可用於那些 VMU 子運算中之一，亦即 $AID_V \neq AID_S \neq AID_R$ 。熟諳先前技藝者能調整該指令和背後的硬體以支援三個以上的 ACU，若偏好如此的話；或支援在該 ACU 和該組 ACU 暫存器間的其它架構。例如，若未固定指派一組暫存器給一 ACU，則該指令也可帶有一要被使用的組的識別。

該 ACU 子運算由該 VMU 控制區間提供，並在該 VMU 指令

的 AOPC 欄內加密。它支援一子運算，將一 ACU 的輸出設定至一立即位址值 (IMM)；以及四個子運算，將一立即位址載入該等 ACU-暫存器 (LDBASE, LDOFFS, LDINCR, 和 LDBOUND) 中之一。該立即位址的目的是要跳過該一般位址並 '迅速地' 從該指令直接提取一位址。這對載入單一字組特別有用。該對應的立即位址係在 IMM_ADDR 欄內加密。該 AID_A 欄指定該 AOPC 子運算將影響哪一個 ACU；來自該 VMU 指令的 AOPC 欄和 IMM_ADDR 欄將直接被傳送至此特別 ACU，而所有其它 ACU 的 AOPC 欄則會被設定成無運算 (NOP)。

一位址計算單元 (ACU) 在單一 ACU 運算期間可支援二 "子運算"：

- 一遞增後子運算；
- 一立即位址操控子運算。

```

ACU_cmd      = (ainc, aopc, imm_addr)
ainc         = NOP | INC
aopc         = NOP | IMM | LDBASE | LDOFFS | LDINCR | LDBOUND
imm_addr     = {0.0, ..., 524288.31} | {-262144.0, ..., 262143.31}

```

該遞增後子運算在該指令的 AINC 欄內加密。它僅支援一子運算：遞增後 (INC)。此子運算用於避免過渡的明確的位址計算指令。

| 保護 | 轉移 |
|------------|--------------------------------|
| ainc = NOP | None |
| ainc = INC | offs = (offs + incr) mod bound |

該立即位址操控子運算在該指令的AOPC欄內加密。它支援一子運算，輸出一立即位址(IMM)；以及四個子運算，將一立即位址載入該等ACU-暫存器(LDBASE, LDOFFS, LDINCR, 和LDBOUND)中之一。該立即位址在該指令的IMM_ADDR欄內加密。

| 保護 | 轉移 |
|----------------|--------------------------------------|
| aopc = NOP | out = base + offs |
| aopc = IMM | out = imm_addr |
| aopc = LDBASE | out = base + offs ; base = imm_addr |
| aopc = LDOFFS | out = base + offs ; offs = inun_addr |
| aopc = LDINCR | out = base + offs ; incr = imm_addr |
| aopc = LDBOUND | out = base + offs ; bound = imm_addr |

該等ACU輸入/輸出為：

| 輸入 | 說明 |
|-----|------------------|
| Cmd | ACU指令(細節請參考指令格式) |

| 輸出 | 說明 |
|-----|---------------------|
| Out | ACU位址(直列位址 + 直列內純量) |

在該較佳具體實施例中，一向量不需對準該記憶體內的向量界限。如此，一包括 P_S 字組的向量可具有一任意的記憶體位址。一記憶體直列大小相同，但其起始位址之定義為 P_S 的倍數(對直列存取而言，忽略該位址的最不重要的 $2 \log P_S$ 位元。)。透過允許向量任意對準(通常對準該最小的字組界限)，可更有效利用該記憶體，空位也愈少。可測量以讓該純量/向量處理器讀取/寫入個別向量，而該向量可被儲存在該實體記憶體的二連續直列內。最好，該純量資料被儲存在與儲存該向量資料相同的記憶體內。在此一系統中，純量可和與其對應的向量互混。對成本有效性和對該記憶體的最佳存取時間/次數而言，該記憶體最好僅允許完整向量直列的讀取和寫入。如此，邏輯上該實體記憶體包括數條直列，每一條的大小和一向量相同。為支援純量的讀取和寫入，會以一純量的方式使用額外硬體(用於一直列內純量選擇的直列快取記憶體430和支援440)以存取該向量寬實體記憶體。

應注意，上述數個具體實施例僅為說明而非限制本發明，並且那些熟諳此藝者將能設計許多替代具體實施例而不偏離所附申請專利範圍之範疇。在該申請專利範圍內，任何出現於括號內的參考記號不應被解釋為限制該申請專利範圍。該字眼"包括"和"包含"並不排除會出現那些未列在

一申請專利範圍內的其它元素或步驟。

【圖式簡單說明】

圖1圖示一較佳架構，其中可使用如本發明之該純量/向量處理器；

圖2圖示如本發明之該純量/向量處理器的主要結構；

圖3圖示所支援的資料寬度和資料類型；

圖4圖示該向量記憶體單元之方塊圖；

圖5圖示將二ACU暫存器組映成至一向量暫存器；

圖6圖示一組ACU暫存器和一ACU間的固定關係；及

圖7圖示該數組ACU暫存器和該等ACU間可建構的關係。

【圖式代表符號說明】

| | |
|-----|----------------|
| 110 | 匯流排 |
| 120 | 可程式化純量/向量處理器 |
| 130 | 微控制器或數位信號處理子系統 |
| 140 | 介面方塊 |
| 210 | 管道化式向量處理區間 |
| 220 | 純量處理區間 |
| 230 | 控制區間 |
| 240 | 純量區間 |
| 250 | 指令分散單元 |
| 252 | 程式記憶體 |
| 260 | 向量記憶體單元 |
| 262 | 碼產生單元 |
| 264 | 算術邏輯-乘及加單元 |

| | |
|---------------------|------------|
| 266 | 弄亂樣式單元 |
| 268 | 向左位移單元 |
| 270 | 向右位移單元 |
| 400 | 記憶體單元 |
| 410 | 實體向量記憶體 |
| 420、620、720-722 | 位址計算單元 |
| 430 | 直列快取記憶體 |
| 440 | 向量對準及純量選擇 |
| 450 | 任意及對準 |
| 500 | 向量 |
| 510、520、610、710-717 | 位址計算單元暫存器組 |
| 630、632 | 通路 |
| 640、642 | 輸出 |
| 730 | 完全互連 |

伍、中文發明摘要：

一種處理器，包括一存取一實體記憶體之受一位址控制之記憶體埠。一處理單元，執行儲存在該記憶體內的指令，及/或在儲存在該記憶體內的資料上運作。一位址產生單元("AGU")產生一控制該記憶體存取之位址；該AGU係與一複數個N暫存器有關，該暫存器在位址產生機制的控制下致能該AGU產生該位址。一記憶體單元被一運算激發，用於儲存/載入該N暫存器中的k個暫存器，其中 $2 \leq k \leq N$ 。為達此目的，該記憶體單元包括一連接器，用於將該k個暫存器連接至一記憶體字組，該字組經由該記憶體埠寫入該記憶體；以及一分離一字組的分離器，該字組經由該記憶體埠從該記憶體讀取至該k個暫存器內。

陸、英文發明摘要：

A processor includes a memory port for accessing a physical memory under control of an address. A processing unit executing instructions stored in the memory and/or operates on data stored in the memory. An address generation unit ("AGU") generates address for controlling access to the memory; the AGU being associated with a plurality of N registers enabling the AGU to generate the address under control of an address generation mechanism. A memory unit is operative to save/load k of the N registers, where $2 \leq k \leq N$, triggered by one operation. To this end, the memory unit includes a concatenator for concatenating the k registers to one memory word to be written to the memory through the memory port and a splitter for separating a word read from the memory through the memory port into the k registers.

拾、申請專利範圍：

1. 一種處理器，包括：

一存取一實體記憶體之記憶體埠，該記憶體埠受一位址控制；

至少一處理單元，執行儲存在該記憶體內的指令和/或於儲存在該記憶體內的資料上運作；

至少一位址產生單元(此後稱"AGU")，用於產生一位址以控制對該記憶體之存取；該AGU與至少一第一組一複數個N暫存器有關，其中該組暫存器在一位址產生機制的控制下致能該AGU產生該位址；及

一記憶體單元，被一運算激發，用於儲存/載入該N暫存器中的k個暫存器，其中 $2 \leq k \leq N$ ，其中該記憶體單元包括一連接器，用於將該k個暫存器連接至一記憶體字組，該字組經由該記憶體埠寫入該記憶體；以及一用於分離一字組的分離器，該字組經由該記憶體埠從該記憶體讀取至該k個暫存器內。

2. 如申請專利範圍第1項之處理器，其中該記憶體單元用於在該記憶體的一寫入/讀取週期內執行該儲存/載入。

3. 如申請專利範圍第1項之處理器，其中 $k = N$ 。

4. 如申請專利範圍第1，2，或3項之處理器，其中該處理器包括至少額外一組的複數個暫存器，該暫存器在一位址產生機制的控制下致能一AGU產生一位址；及

該連接器係操作以將該額外數組暫存器中至少一組的至少一額外暫存器連接至該一記憶體字組，該字組經

- 由該記憶體埠寫入該記憶體；並且該分離器正運轉，將一字組分離，該字組經由該記憶體埠從該記憶體讀取至第一組暫存器以及該額外數組暫存器中至少一組的至少一額外暫存器的各別的k個暫存器。
5. 如申請專利範圍第4項之處理器，其中該處理器包括至少二AGU，每一個用於產生一位址，用於控制對該記憶體之存取；該等AGU中的每一個皆與一複數個暫存器中該等組內各自的一組有關。
 6. 如申請專利範圍第4項之處理器，其中該AGU可選擇性地連接至至少二組暫存器，其中每一組暫存器在一位址產生機制的控制下致能該AGU產生一各別的位址。
 7. 如申請專利範圍第3和4項之處理器，其中該連接器係操作以將至少二組暫存器中所有各別相關的暫存器連接至該一記憶體字組，該字組經由該記憶體埠寫入該記憶體；並且該分離器運轉，將一字組分離，該字組經由該記憶體埠從該記憶體讀取至該等暫存器組中至少二組內所有該各別的相關暫存器。
 8. 如申請專利範圍第4或5項之處理器，其中該處理器致能各別暫存器組的選擇，該等暫存器組的各別暫存器需被儲存或存放在一運算內；並且該記憶體單元正運轉，以執行該被選定的暫存器組來回於一記憶體字組的連接或分離。
 9. 如申請專利範圍第1項之處理器，其中該處理器係操作以運算在運算域上具有最小大小一處理器字組，並且該

記憶體字組的寬度為該處理器字組大小的數倍；並且該連接器運轉，將該暫存器映成至該記憶體內的界限，該界限對應該處理器字組，使該處理器能直接改良一暫存器。

10. 如申請專利範圍第1項之處理器，其中該處理器係操作以平行處理一複數個M資料元素，並且該實體記憶體具有一能儲存該M個資料元素的寬度。

拾壹、圖式：

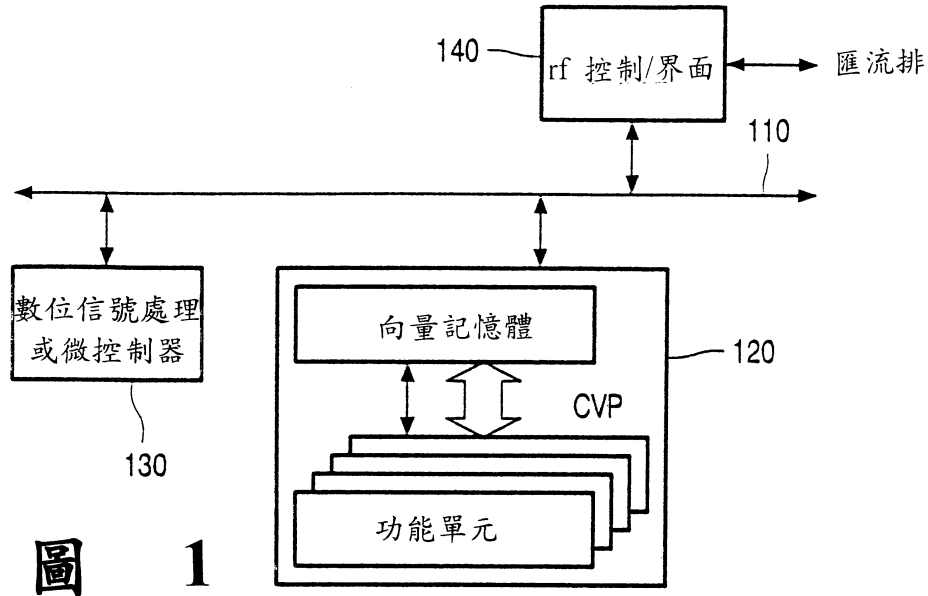


圖 1

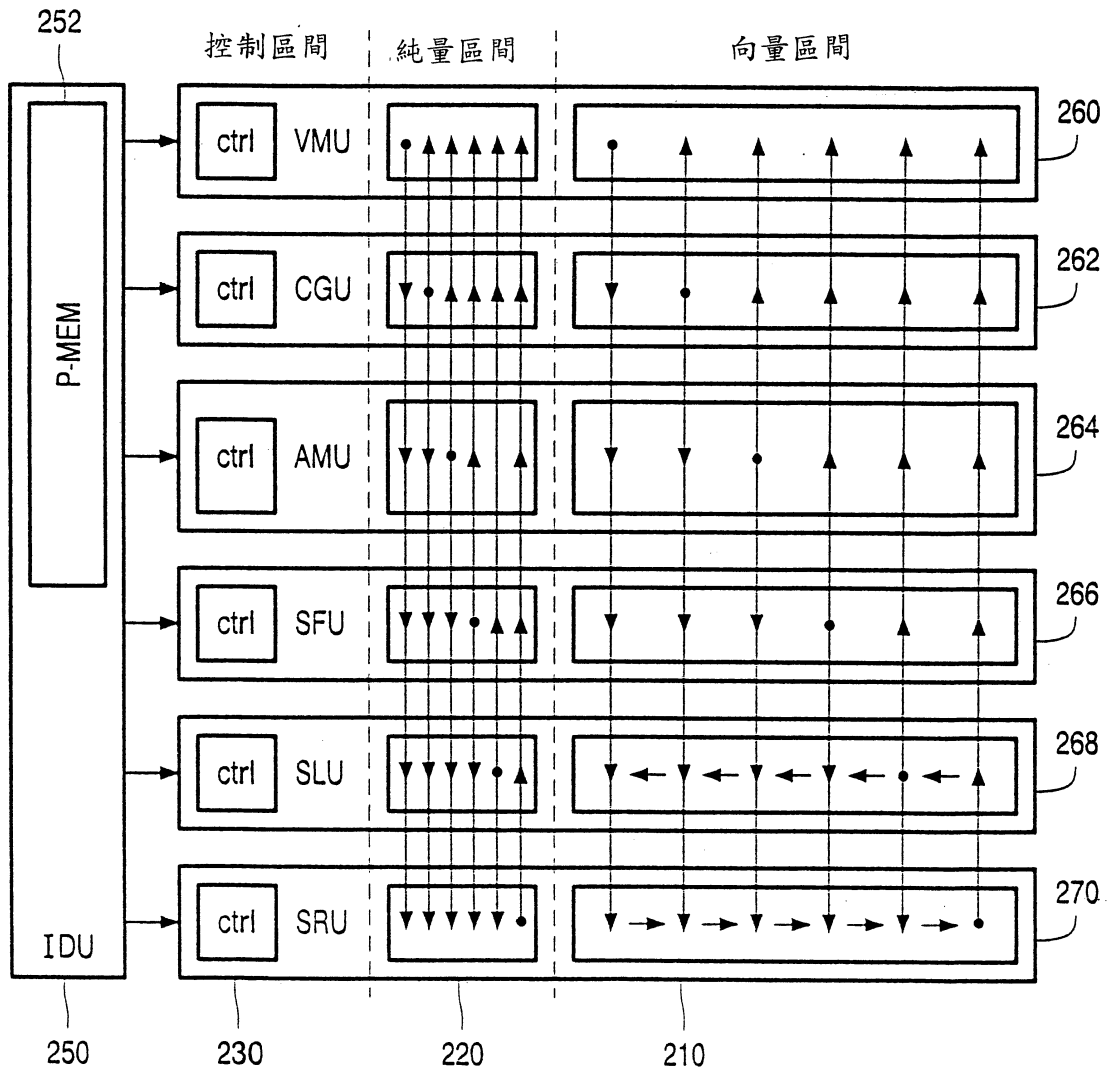


圖 2

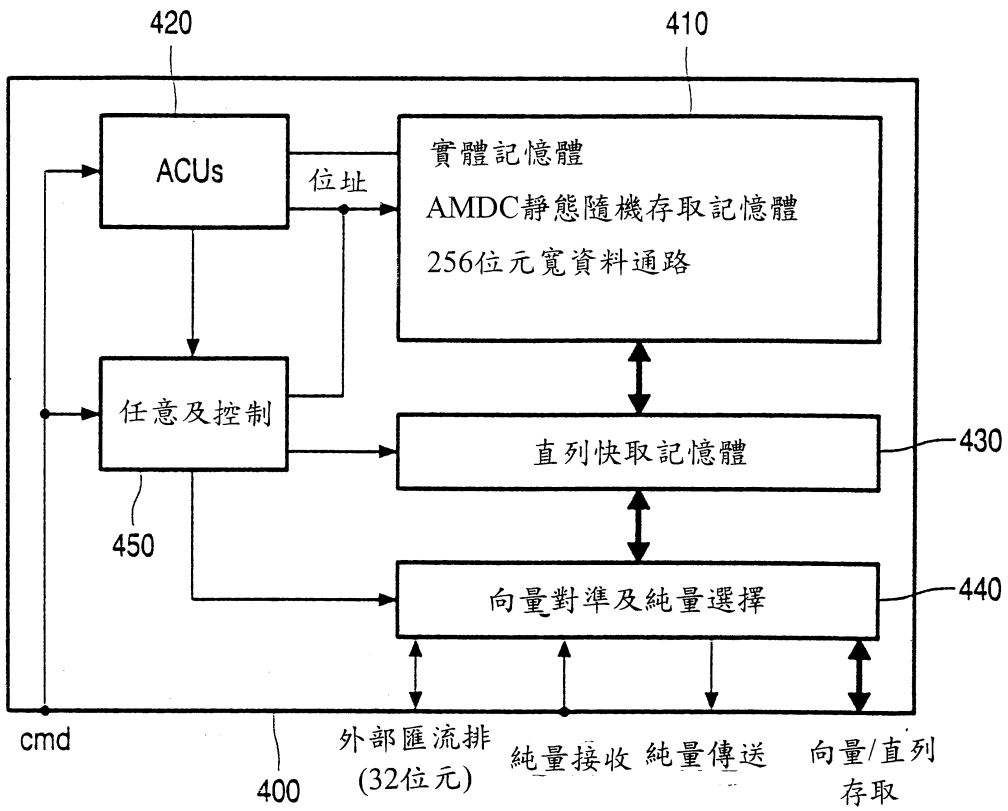


圖 4

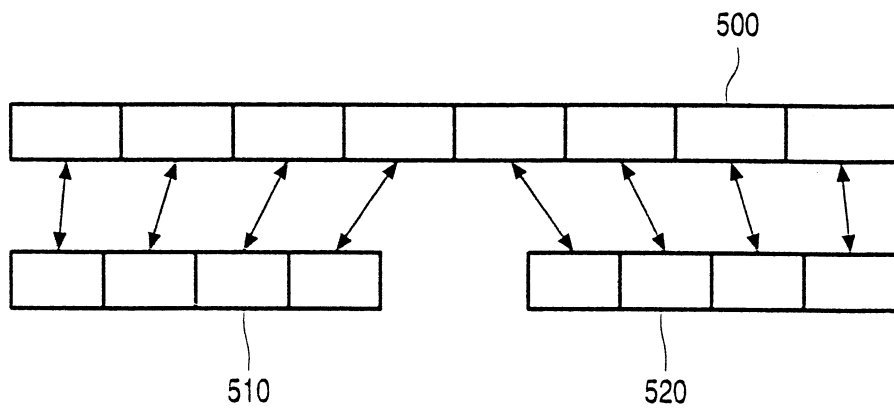


圖 5

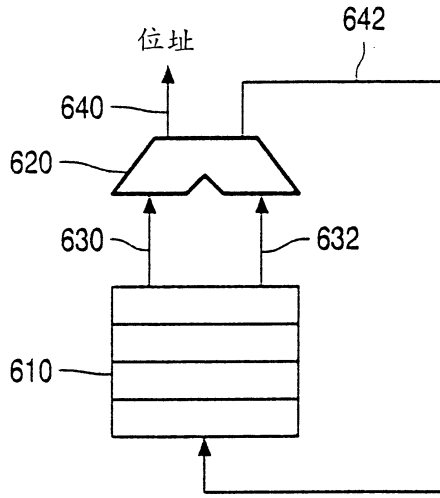


圖 6

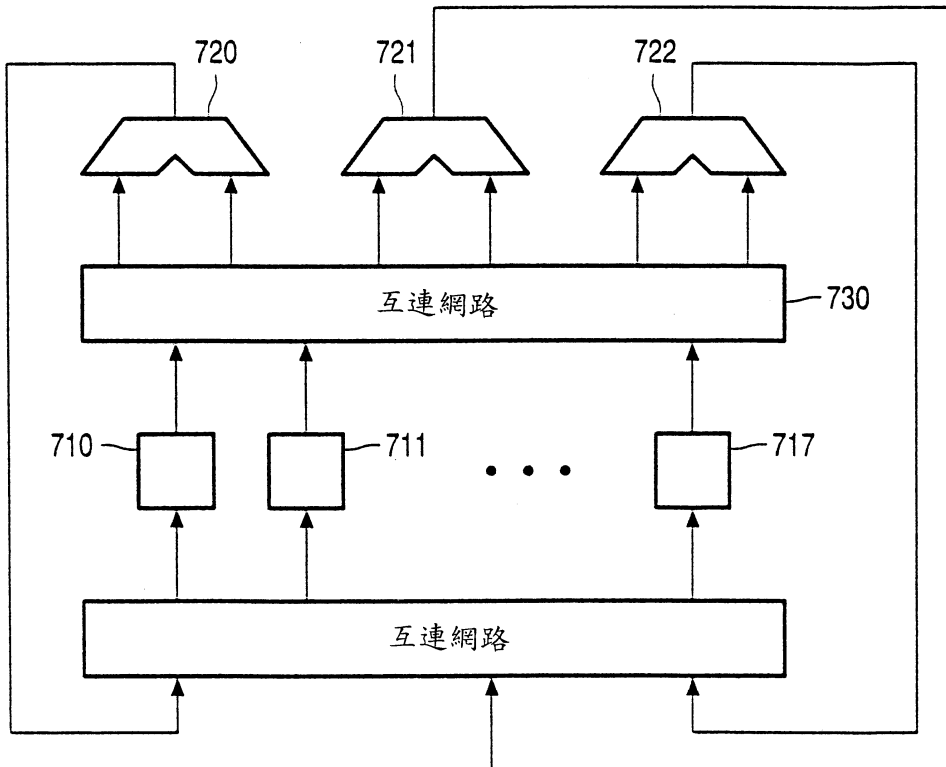


圖 7

柒、指定代表圖：

(一)本案指定代表圖為：第(2)圖。

(二)本代表圖之元件代表符號簡單說明：

- 210 管道化式向量處理區間
- 220 純量處理區間
- 230 控制區間
- 240 純量區間
- 250 指令分散單元
- 252 程式記憶體
- 260 向量記憶體單元
- 262 碼產生單元
- 264 算術邏輯-乘及加單元
- 266 弄亂樣式單元
- 268 向左位移單元
- 270 向右位移單元

捌、本案若有化學式時，請揭示最能顯示發明特徵的化學式：