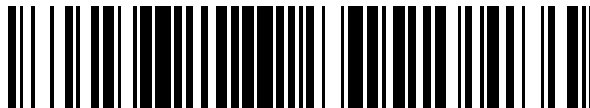


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 391 690**

51 Int. Cl.:
G06F 15/80 (2006.01)
G06F 9/34 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 96 Número de solicitud europea: **08774325 .8**
96 Fecha de presentación: **26.06.2008**
97 Número de publicación de la solicitud: **2162829**
97 Fecha de publicación de la solicitud: **17.03.2010**

54 Título: **Dispositivo de direccionamiento para procesador paralelo**

30 Prioridad:
26.06.2007 FR 0704574

45 Fecha de publicación de la mención BOPI:
29.11.2012

45 Fecha de la publicación del folleto de la patente:
29.11.2012

73 Titular/es:
THALES (100.0%)
45, RUE DE VILLIERS
92200 NEUILLY-SUR-SEINE, FR

72 Inventor/es:
GAILLAT, GÉRARD

74 Agente/Representante:
CARPINTERO LÓPEZ, Mario

ES 2 391 690 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Dispositivo de direccionamiento para procesador paralelo

La presente invención se refiere a un dispositivo de direccionamiento para procesador paralelo, aplicable en particular a ordenadores paralelos de tipo SIMD.

5 El término SIMD se refiere a la clasificación de los ordenadores paralelos introducida por Flynn. Designa un ordenador paralelo de tipo Single Instruction Multiple Data Stream (SIMD).

En el pasado, una o más tarjetas electrónicas eran necesarias para implementar un SIMD. Actualmente, una SIMD se puede implantar en un circuito integrado único, por ejemplo un PFGA o un ASIC.

Un SIMD descrito en relación con la figura 1, incluye

10 un solo secuenciador 1,
una sola memoria de instrucciones 2,
un gran número de procesadores elementales 3 o Processing Elements (PEs).

Cada PE 3 incluye

15 una Unidad Aritmética y Lógica (ALU) 31,
un conjunto de memorias y/o de registros de datos 32.

En cada ciclo, el secuenciador

20 determina la siguiente instrucción y la lee en la memoria de instrucciones
transmite esta instrucción a todos los PEs.

La mayoría de las instrucciones definen:

una operación a ejecutar
la dirección Ad donde ir a buscar cada operando de la operación, la o las direcciones Ad donde almacenar el resultado.

25 La operación a ejecutar comprende uno o más operandos. Puede limitarse a una operación elemental (por ejemplo, una inversión de signo o una adición o un si o un cálculo máximo) que comprende uno o más operandos o ser la combinación de varias operaciones elementales que comprenden cada una uno o más operandos (por ejemplo; una multiplicación seguida de una adición). Asimismo, el resultado puede almacenarse en una o más direcciones.

30 Cada PE ejecuta esta misma instrucción al mismo tiempo pero sobre datos diferentes, en su caso sobre el conjunto de memorias y/o de registros que le pertenecen.

Evidentemente, cuando se dice en cada ciclo, se habla de un ciclo lógico cuya ejecución puede extenderse sobre varios ciclos físicos según la técnica clásica del conducto. Lo importante es que una nueva instrucción sea puesta en marcha en cada ciclo y que una nueva instrucción se acabe en cada ciclo. Independientemente de si la ejecución de la instrucción se extiende sobre varios ciclos.

35 Se considera por ejemplo la instrucción $R(23) = R(8) + R(3) * R(19)$. Significa leer el contenido del registro 3, multiplicarlo por el contenido del registro 19, añadir el resultado de esta multiplicación al contenido del registro 8 y escribir el resultado de esta suma en el registro 23. Esta misma instrucción va a ser ejecutada en paralelo por todos los PEs. Pero ninguno la ejecutará en el conjunto de registros que le pertenece. En total, si hay 129 PEs, la ejecución de la instrucción tendrá como consecuencia que 128 registros 3 serán leídos, 128 registros 19 serán leídos, 128 registros 8 serán leídos, 128 multiplicaciones serán realizadas en 2 veces 128 valores diferentes, 128 sumas realizadas en 2 veces 128 valores diferentes y 128 resultados diferentes serán escritos en 128 registros 23.

La eficacia de un SIMD procede de este alto nivel de paralelismo.

45 En general, en un SIMD, cada PE tiene la posibilidad de leer, es decir ir a buscar los operandos de una instrucción, no solo en el conjunto de las memorias y/o registros que le pertenecen sino también en todo o parte del conjunto de las memorias y/o registros que pertenecen a uno de sus vecinos, véase por ejemplo el documento GB 2 417 105. Una variante que puede ser un complemento o una alternativa es que cada PE tenga la posibilidad de escribir el resultado de una instrucción no solo en memorias y/o registros que le pertenecen sino también en memorias y/o registros que pertenecen a uno de sus vecinos.

50 La noción de vecindad entre PEs está definida por la arquitectura del SIMD y más concretamente por la topología de interconexión entre PEs. Varias topologías de interconexión entre PEs han sido propuestas. Las más complejas son verdaderas redes de interconexión entre PEs. La más sencilla es una columna vertical de PEs. Otra topología sencilla es una rejilla bidimensional de PEs. Se pueden imaginar también rejillas multidimensionales de PEs. En

5 general, en estos 3 últimos casos, los extremos están “pegado”. De este modo, en la topología monodimensional la columna vertical de PE se convierte en un círculo (a menudo se denomina anillo de PE. Gracias a este repegado, el vecino Sur del PE más al Sur es el PE más al Norte y el vecino Norte del PE más al Norte es el PE más al Sur. Asimismo, en la topología bidimensional, la rejilla de PEs se convierte en un toro. Finalmente, en la topología multidimensional, la rejilla multidimensional se convierte en un toro multidimensional.

10 En el caso de una topología monodimensional (columna vertical o anillo de PEs), cada PE dispone de 2 vecinos que se denominarán N y S (por Norte y Sur). En el caso de una topología bidimensional (rejilla bidimensional o toro de PEs), cada PE dispone de 4 vecinos que se denominarán N, S, E, W (por Norte, Sur, este, Oeste). En el caso de una topología multidimensional (rejilla multidimensional o toro multidimensional de PEs), cada PE dispone de 2^n vecinos, si n es la dimensión del espacio.

En el resto de este documento, se considerará en primer lugar el caso de las topologías monodimensionales, y a continuación el caso de topologías bidimensionales, y después el caso de las topologías multidimensionales.

15 Una instrucción típica en un SIMD moderno requiere que todos los PEs (o algunos de los mismos) efectúen una operación, la misma para todos, sobre varios operandos y almacenar el resultado en uno o varios emplazamientos. Para retomar el ejemplo anterior, la instrucción $R(23) = (R8) + R(3) * R(19)$ requiere que todos los PEs efectúen una operación sobre 3 operandos y almacenar el resultado en un emplazamiento.

20 En la mayoría de los SIMD, la instrucción define para cada operando y para cada resultado, la dirección donde los diferentes PEs van a leer el operando (o almacenar el resultado). En consecuencia, para un operando dado, como para un emplazamiento de resultado dado, la dirección es necesariamente la misma para todos los PEs.

25 Tal enfoque plantea un problema, en particular en aplicaciones típicas de los SIMD como el procesamiento de imágenes, la compresión de imágenes, la generación sintética de imágenes, el procesamiento de la señal o algunas técnicas de resolución de ecuaciones en derivadas parciales. Por ejemplo, cuando estas aplicaciones utilizan algoritmos en los cuales un pixel es calculado a partir de sus vecinos, siendo la topología de los PE, bien mono, bi, tri o multidimensional.

Para ilustrar esta dificultad, se toma un ejemplo que aplica el procesamiento de imágenes y se analiza considerando un SIMD de 16 PEs organizado en anillo.

30 En todo el resto de este documento y para simplificar las fórmulas matemáticas, los procesadores son numerados a partir de 0; al igual que para las líneas y las columnas de una estructura de datos (señal, imagen o volumen).

35 Una manera clásica de guardar los píxeles de la imagen en las memorias de los diferentes PEs es utilizar el PE0 (PE nº 0) para almacenar y procesar la línea 0 de la imagen, el PE1 (PE nº 1) para almacenar y procesar la línea 1 y así sucesivamente hasta PE15 (PE nº 15) para almacenar y procesar la línea 15. Como una imagen incluye en general más de 16 líneas, el usuario utilizará el PE0 para almacenar y procesar las líneas 0, 16, 32, 48, 64, 80, ... de la imagen, el PE1 para almacenar y procesar las líneas 1, 17, 33, 49, 65, 81, .. de la imagen y así sucesivamente hasta el PE 15 para almacenar y procesar las líneas 15, 31, 47, 63, 79, 95, ... de la imagen.

En esta disposición, si N_p designa el número de PEs y si L_x designa el ancho de la imagen a procesar, un pixel de coordenadas (x, y) está dispuesto en el procesador p en la dirección A_d según las siguientes fórmulas:

$$p = y \text{ Mod } N_p$$

40 $A_d = x + L_x * \text{Int}(y / N_p)$, (designando $\text{Int}(z)$ la parte entera de z).

45 Esta disposición está representada en la figura 2. Esta figura considera el caso de una imagen de ancho de 64 píxeles y muestra una manera habitual de disponerla en un SIMD de 16 PEs organizado en una topología de anillo. En esta figura, las columnas representan las columnas de la imagen, las líneas representan las líneas de la imagen, el número del PE está indicado en el extremo izquierdo de cada línea y la dirección donde está guardado cada pixel está indicada en el interior de cada casilla. En esta figura, como en las dos siguientes, las fronteras entre zonas de direcciones homogéneas (zona con la misma dirección en todos los PEs) se materializan mediante trazos negros.

50 Por convención, se ha elegido una representación visual en la cual la izquierda de una imagen está constituida por píxeles de abscisa débil mientras que la parte superior de una imagen está constituida por píxeles de ordenada débil. De este modo, la esquina superior izquierda de un cuadrado cuyos bordes son paralelos a los ejes de coordenadas es el punto de abscisa más débil y de ordenada más débil mientras que la esquina inferior derecha es el punto de abscisa más fuerte y de ordenada más fuerte. Esta convención se conserva en el resto del documento.

55 Este almacenamiento significa considerar el SIMD como un anillo vertical de PEs. Se podría evidentemente hacer la elección inversa consistente en considerar el SIMD no como un anillo vertical de PEs sino como un anillo horizontal y utilizar los PEs para almacenar ya no las líneas sino las columnas de la imagen. La figura 3 ilustra este almacenamiento. En realidad, solo se trata de una elección de representación visual para el programador que no

cambia casi nada al nivel de la máquina. Lo que se expone a continuación relativo a una representación en líneas se generaliza evidentemente a una representación en columnas.

5 El tipo de asignación de las líneas de la imagen a los PEs descritos anteriormente no plantea ningún problema cuando se trata de operaciones sencillas tales como la suma de 2 imágenes, la multiplicación de 2 imágenes, la multiplicación de una imagen por una constante, la convolución de una imagen por una máscara de línea, De hecho, cuando se trata de operaciones en las cuales los píxeles que sirven de operandos y el pixel de resultado son almacenados por el mismo PE.

10 Las cosas se complican cuando se trata de operaciones en las cuales los píxeles que sirven de operando y el pixel de resultado son almacenados por PEs diferentes. Por ejemplo, en el caso de la convolución de una imagen por una máscara de 2 dimensiones, el PE que está encargado de almacenar el pixel de resultados debe ir a buscar los operandos que necesita para efectuar sus cálculos no solo en memorias y/o registros que le pertenecen sino también en memorias y/o registros que pertenecen a los PEs vecinos, incluso a los vecinos de los vecinos, incluso más lejos.

15 Si se quiere efectuar una convolución de 3x3 en un cuadrado de 32x32 cuya esquina superior izquierda tiene por coordenadas x=8, y=4 y cuya esquina inferior derecha tiene por coordenadas x=39, y=35 y almacenar el resultado en un cuadrado de 32x32 cuya esquina superior izquierda tiene por coordenadas x=8, y=68 y cuya esquina inferior derecha tiene por coordenadas x=39, y=99. Se recuerda como se ha indicado anteriormente que en la referencia xy considerada, el eje de las x es creciente de izquierda a derecha, el eje de las y de arriba abajo.

20 Siendo a(-1, -1), a(0, -1), a(1, -1), a(-1, 0), a(0, 0), a(1, 0), a(-1, 1), a(0, 1), a(1, 1), los 9 coeficientes del filtro de convolución

Siendo I(x,y) los píxeles de la imagen de origen.

Siendo R(x,y+64) los píxeles de la imagen de resultado

El cálculo a efectuar, para cada valor de x comprendido entre 8 y 39 y para cada valor de y comprendido entre 4 y 35 es el siguiente:

$$\begin{aligned}
 R(x,y+64) = & a(-1, -1) * I(x-1, y-1) + a(0, -1) * I(x, y-1) + a(1, -1) * I(x+1, y-1) \\
 & + a(-1, 0) * I(x-1, y) + a(0, 0) * I(x, y) + a(1, 0) * I(x+1, y) \\
 25 & + a(-1, 1) * I(x-1, y+1) + a(0, 1) * I(x, y+1) + a(1, 1) * I(x+1, y+1).
 \end{aligned}$$

30 La primera operación es efectuar esta convolución en un segmento vertical 16x1 cuyo extremo superior tiene por coordenadas x=8, y=4 y cuyo extremo inferior tiene por coordenadas x=8, y=19. Este segmento aparece en gris oscuro en la figura 2. La primera etapa de esta primera operación es leer un segmento vertical 16x1 cuyo extremo superior tiene por coordenadas x=7, y=3 y cuyo extremo inferior tiene por coordenadas x=7, y=18. Este segmento aparece en gris claro en la figura 2. Cada uno de los 16 píxeles leídos es multiplicado por el primer coeficiente de la máscara de convolución. La siguiente etapa es leer un segmento vertical 16x1 cuyo extremo superior tiene por coordenadas x=8, y=3 y cuyo extremo inferior tiene por coordenadas x=8, y=18. Cada uno de los 16 píxeles leídos es multiplicado por el segundo coeficiente de la máscara de convolución y el resultado de esta segunda multiplicación es añadido al resultado de la primera multiplicación. Y así sucesivamente para los 9 coeficientes de la máscara de convolución. (9 coeficientes ya que se ha tomado el ejemplo de una convolución 3x3). El resultado es entonces almacenado en un segmento vertical 16x1 cuyo extremo superior tiene por coordenadas x=8, y=68 y cuyo extremo inferior tiene por coordenadas x=8, y=83. La primera operación está entonces concluida. A continuación se llevan a cabo 31 operaciones similares para procesar las 31 columnas restantes de la mitad superior del cuadrado a procesar y a continuación 32 operaciones similares para procesar las 32 columnas de la mitad inferior de este cuadrado.

40 La dificultad aparece desde la primera etapa de la primera operación. A continuación es la misma para todas las etapas de todas las operaciones. Para esta primera etapa, hay que leer un segmento vertical de 16x1 cuyo extremo superior tiene por coordenadas x=7, y=3 y cuyo extremo inferior tiene por coordenadas x=7, y=18. Como se muestra en la figura, los píxeles a leer para los PEs 3 a 15 se guardan en la dirección 7 mientras que los píxeles a leer para los PEs 0 a 2 se guardan en la dirección 71.

Ahora bien, en un SIMD convencional el secuenciador envía la misma dirección a todos los PEs.

50 Se puede objetar que el ejemplo elegido se refiere a un rectángulo cuyos bordes no están alineados con las fronteras entre zonas de direcciones homogéneas, es decir un rectángulo cuyo borde superior tiene un ordenada que no es un múltiplo del número de PEs. Esta objeción no resiste el análisis. Por una parte, el caso de tener que procesar tales rectángulos es un caso frecuente en el procesamiento de imágenes. Por otra parte, incluso si el rectángulo que contiene la imagen de resultado tuviese bordes alineados con las fronteras entre zonas de direcciones homogéneas, se tendría, sin embargo, para procesar las diferentes etapas de una convolución de 3x3 ir

a buscar como operandos segmentos verticales de 16x1 cuyos extremos no estuviesen todos alineados con las fronteras entre zonas de direcciones homogéneas.

Varias técnicas han sido propuestas para solucionar esta dificultad.

5 Una primera técnica es una carga en dos tiempos con una activación condicional de los PEs. En un primer tiempo, solo los PEs que deben leer en la dirección 7 realizan efectivamente la operación de lectura. En un segundo tiempo, solo los PEs que deben leer en la dirección 71 realizan efectivamente la operación de lectura. Por lo tanto es el programador de la aplicación el que debe gestionar el hecho de que, para realizar esta primera etapa, ciertos PEs deben cargar como operando un pixel que han guardado en la dirección 7 mientras que otros PEs deben cargar como operando un pixel que han guardado en la dirección 71. De ahí una dificultad importante para programar este tipo de máquina, pero también una menor eficacia de ejecución debida al exceso de instrucciones a ejecutar.

10 Una segunda técnica es la aplicación de un registro de indirección en el seno de cada PE. Un registro de indirección es un registro que permite acceder a una palabra cuya dirección está contenida en este registro. Gracias a este registro de indirección, cada PE puede calcular una dirección que le es propia, cargarla en el registro de indirección y servirse de ella para acceder a su propia memoria. Pero, como para la primera técnica, se encuentra por una parte una dificultad adicional de programación ya que es el programador de aplicación el que debe gestionar la dificultad y por otra parte una menor eficacia de ejecución debida al exceso de instrucciones a ejecutar.

Otras técnicas han sido imaginadas. Todas tienen en común el hecho de que conducen por una parte a una dificultad adicional de programación y por otra parte a una menor eficacia de ejecución debida al exceso de instrucciones a ejecutar.

20 La dificultad es aún mayor si se utiliza un SIMD organizado según una topología bidimensional. Las figuras 4 retoman el mismo ejemplo de una imagen de ancho de 64 píxeles pero utiliza un SIMD de 16 PEs organizados en una topología de toro de 4x4. Para la primera etapa de la primera operación, hay que leer un cuadrado de 4x4 cuya esquina superior izquierda tiene por coordenadas $x=7$, $y=3$ y cuya esquina inferior derecha tiene por coordenadas $x=10$, $y=6$. Este cuadrado aparece en gris claro en la figura 4a. Como se muestra en la figura 4b, los píxeles a leer están guardados ya no en 2 direcciones diferentes como anteriormente sino en 4 direcciones diferentes: dirección 1 para el PE15, dirección 2 para los PE 12, 13, 14, dirección 17 para los PE 3, 7, 11, dirección 18 para los otros PEs.

25 La dificultad crece aun si se utiliza un SIMD organizado según una topología multidimensional. Por ejemplo, los píxeles a leer están guardados en 8 direcciones diferentes en una topología tridimensional y en 16 direcciones diferentes en una topología cuatridimensional. Salvo en casos particulares (posición particular de la zona a leer o escribir) donde el número de direcciones diferentes es inferior.

30 El objetivo de la invención es solucionar estos inconvenientes.

La invención está definida en las reivindicaciones adjuntas.

35 La invención está basada en el hecho de que la instrucción, en lugar de definir el conjunto de las lecturas y/o el conjunto de las escrituras de los procesadores elementales indicando la dirección en la cual cada procesador elemental lee o escribe en su propia memoria, define el conjunto de las lecturas y/o el conjunto de las escrituras de los procesadores elementales cada uno en su propia memoria como una lectura y/o una escritura en un campo de acción.

40 Desde el punto de vista del usuario, el conjunto de las palabras de memoria del conjunto de los procesadores elementales es visto como una estructura de datos única de tipo tabla de dimensión N, pudiendo esta tabla representar una señal (dimensión 1) o una imagen (dimensión 2) o un volumen (dimensión 3) o El campo de acción es definido por la instrucción, por referencia a esta estructura de datos y como un subconjunto de esta estructura de datos.

45 Por campo de acción se entiende un conjunto de palabras de memoria a razón de una palabra de memoria por procesador elemental, estando esta palabra de memoria situada en la memoria del procesador elemental, siendo un campo de acción de un operando el conjunto de las palabras de memoria leídas por los procesadores elementales, cada uno en su propia memoria para adquirir este operando, siendo un campo de acción de un resultado el conjunto de las palabras de memoria escritas por los procesadores elementales, cada uno en su propia memoria para almacenar este resultado.

50 La instrucción comprende para cada operando y/o cada resultado, informaciones relativas a la posición de este campo de acción, estando esta posición definida como una posición en el seno de una estructura de datos única de tipo tabla de dimensión N (señal o imagen o volumen o ...), repartida en las memorias de los diferente procesadores elementales. Siendo N un número entero igual o superior a 1. En lo sucesivo, se designa por estructura de datos, esta estructura de datos única de tipo tabla de dimensión N.

La posición del campo de acción permite calcular la dirección en la cual cada procesador elemental debe leer en su propia memoria la palabra de memoria que le concierne en el operando y/o escribir en su propia memoria la palabra de memoria que le concierne en el resultado.

5 Una instrucción típica en un SIMD moderno requiere que todos los PEs o que algunos de los mismos efectúen una operación, la misma para todos, sobre varios operandos y almacenar el resultado en uno o más emplazamientos.

10 Retomando el ejemplo de la figura 2, la primera etapa de la primera operación requiere leer un segmento vertical de 16x1 cuyo extremo superior tiene por coordenadas $x=7, y=3$ et cuyo extremo inferior tiene por coordenadas $x=7, y=18$. El campo de acción del operando $I(x-1, y-1)$ está compuesto por píxeles de coordenadas $(x=7, y=3), (x=7, y=14), (x=7, y=5), \dots(x=7, y=18)$; como se muestra en la figura, los píxeles a leer para los PEs 3 a 15 están guardados en la dirección 7 mientras que los píxeles a leer para los PEs 0 a 2 están guardados en la dirección 71.

15 Ahora bien, en un ordenador SIMD convencional, la instrucción define para cada operando y para cada resultado, la dirección donde los diferentes PE van a leer el operando o almacenar el resultado. En consecuencia, un SIMD convencional solo puede tomar en cuenta campos de acción que corresponden a zonas de direccionamiento homogéneo. Por ejemplo, en un SIMD convencional, es posible tomar en cuenta un campo de acción cuyo extremo superior tiene por coordenadas $x=7, y=0$ y cuyo extremo inferior tiene por coordenadas $x=7, y=15$, ya que todos estos píxeles están guardados en la dirección 7, o un campo de acción cuyo extremo superior tiene por coordenadas $x=7, y=16$ y cuyo extremo inferior tiene por coordenadas $x=7, y=31$ ya que todos estos píxeles están guardados en la dirección 71, pero no un campo de acción cuyo extremo superior tiene por coordenadas $x=7, y=3$ y cuyo extremo inferior tiene por coordenadas $x=7, y=18$ ya que ciertos píxeles están guardados en la dirección 7 y otros en la
20 dirección 71. Dicho de otro modo, en una SIMD convencional, no es posible tener en cuenta un campo de acción cuando el mismo superpone zonas de direcciones no homogéneas.

Un procesador paralelo diseñado según la invención permite que los diferentes operando o resultados de una instrucción tomen en cuenta campos de acción que no corresponden necesariamente a zonas de direcciones homogéneas.

25 Más concretamente la invención tiene por objeto un procesador paralelo que comprende procesadores elementales PE que comprenden cada uno al menos una unidad de cálculo y al menos una memoria que incluye palabras de memoria, y dispuestos según una topología, con una posición determinada en el seno de esta topología y capaces de ejecutar simultáneamente una misma instrucción sobre datos diferentes, incluyendo la instrucción la lectura de al menos un operando y/o la escritura de al menos un resultado. Está principalmente caracterizado porque la
30 instrucción define el conjunto de las lecturas y/o el conjunto de las escrituras de los procesadores elementales, permite la colonne verticale de PEs devient un cercle de PEs (on dit souvent un anneau de PEs). Grâce à ce recollage, le voisin Sud du PE le plus au Sud est le PE le plus au Nord et le voisin Nord du PE le plus au Nord est le PE le plus au Sud. cada uno en su propia memoria como una lectura y/o escritura en un campo de acción, siendo un campo de acción un conjunto de palabras de memoria a razón de una palabra de memoria por procesador elemental, estando esta palabra de memoria situada en la memoria del procesador elemental, siendo un campo de acción de un operando el conjunto de las palabras de memoria leídas por los procesadores elementales, cada uno en su propia memoria para adquirir este operando, siendo un campo de acción de un resultado el conjunto de las palabras de memoria escritas por los procesadores elementales, cada una en su propia memoria para almacenar este resultado, porque la instrucción comprende para cada operando y/o cada resultado, informaciones relativas a la
40 posición de este campo de acción, porque la posición del campo de acción está definida como una posición en el seno de una estructura de datos única de tipo tabla de dimensión N, siendo N un número entero igual o superior a 1, porque esta tabla está repartida en las memorias de los diferentes procesadores elementales.

45 El procesador paralelo propuesto aplica el siguiente principio: para cada operando y para cada elemento de la estructura de datos donde almacenar el resultado, la instrucción define no la dirección en la que el PE debe leer el operando o escribir el resultado, sino informaciones que permiten definir la posición del campo de acción del operando o del resultado en el seno de la estructura de datos. Gracias a estas informaciones y a un valor ligado a la posición de cada PE, medios de cálculo permiten que cada PE reconozca la dirección en la que debe leer el operando o escribir el resultado.

50 De esta manera, la dirección en la que cada PE lee o escribe directamente en su propia memoria es obtenida por un cálculo y este cálculo es efectuado en función de la posición del campo de acción y de la posición del PE en el seno de la topología.

La forma de estos campos de acción (segmento horizontal o vertical, cuadrado, rectángulo o paralelepípedo) sigue siendo el mismo que en un SIMD convencional ya que está calcado de la topología de los PEs.

55 Por el contrario, la posición del campo de acción está definida de manera mucho más flexible que en un SIMD convencional. En particular, tal procesador paralelo permite tomar en cuenta campos de acción tales como los evocados en el ejemplo del procesamiento de imagen e ilustrado por las figuras 2, 3 y 4; y esto aunque las celdas que componen estos campos de acción estén situados en direcciones diferentes según los PEs.

5 Pero su interés va más allá. A título de ejemplo, todas las aplicaciones que utilizan algoritmos en los que un píxel es calculado a partir de sus vecinos pueden beneficiarse. Esto engloba el procesamiento de la señal, el procesamiento de imágenes, la compresión de imágenes, la generación sintética de imágenes e incluso ciertas técnicas de resolución de ecuaciones en derivadas parciales. Un ejemplo típico es la resolución de ecuaciones en derivadas parciales en un espacio de 3 dimensiones por el procedimiento de las diferencias finitas.

Para estas aplicaciones, el beneficio es doble:

- se simplifica la programación de un algoritmo ya que el programador ya no tiene que preocuparse de la noción de fronteras entre zonas de direcciones homogéneas,
- se acelera la ejecución ya que hay menos instrucciones a ejecutar.

10 Según una característica de la invención, el procesador paralelo comprende medios de cálculo de la posición del campo de acción en función de dichas informaciones, vinculados a los medios de cálculo de la dirección.

Los medios de cálculo de la dirección en el seno de cada procesador elemental en función de la posición del campo de acción pueden estar:

15 bien enteramente localizados al nivel de cada procesador elemental,
bien parcialmente localizados al nivel de cada procesador elemental y parcialmente agrupados entre varios procesadores elementales.

El procesador paralelo según la invención comprende eventualmente medios de cálculo de la posición del campo de acción en función de dichas informaciones. Estos medios de cálculo de la posición del campo de acción pueden estar:

20 - bien enteramente agrupados entre todos los PE.
- bien enteramente localizados al nivel de cada procesador elemental,
- bien parcialmente agrupados entre varios PE y parcialmente localizados al nivel de otros procesadores elementales

25 De manera general, los procesadores elementales están dispuestos según una topología de P dimensiones, siendo P un número entero superior o igual a 1 y/o los datos a tratar pertenecen a una estructura en un espacio de N dimensiones, siendo N un número entero superior o igual a 1.

Según una característica de la invención, para cada dimensión el cálculo de la dirección del elemento de la estructura de datos que se refiere a un procesador elemental de abscisa p según dicha dimensión, es obtenido por combinación de un valor ligado a la posición del campo y de un valor ligado a la posición del procesador elemental.

30 Según un modo de realización de la invención, el campo de acción que presenta extremos definidos por sus coordenadas, la posición del campo de acción está por ejemplo definida por la posición del extremo cuyas coordenadas son mínimas.

35 Según una característica de la invención, para cada dimensión los medios de cálculo de la dirección del elemento de la estructura de datos que se refiere a un procesador elemental de abscisa p según dicha dimensión en función de la posición del campo de acción comprenden medios de cálculo de:

$$Ad = \text{Int}((W_{\text{min}} + Np - 1 - p) / Np),$$

siendo Np el número de procesadores elementales considerado en dicha dimensión, estando Wmin definido en dicha dimensión en función de las coordenadas de dicha posición.

El procesador paralelo es por ejemplo de tipo SIMD.

40 Otras características y ventajas de la invención se pondrán de manifiesto en la siguiente descripción detallada, realizada a título de ejemplo no limitativo y con referencia a los dibujos adjuntos en los cuales:

la figura 1 ya descrita representa esquemáticamente un ejemplo de ordenador SIMD según el estado de la técnica,
45 la figura 2 ya descrita ilustra un almacenamiento de los 64 x 64 píxeles de una imagen en un ordenador SIMD de 16 PEs organizado en anillo,
la figura 3 ya descrita ilustra otra disposición de los 64 x 64 píxeles de una imagen en un ordenador SIMD de 16 PEs organizado en anillo,
las figuras 4 ya descritas ilustran un almacenamiento de los 64 x 64 píxeles de una imagen en un ordenador SIMD de 16 PEs organizado en toro 4 x 4, representando la figura 4a los PEs concernidos, la
50 figura 4b, las direcciones en el interior de los PEs,
la figura 5 representa esquemáticamente un ejemplo de procesador paralelo según la invención,

las figuras 6 representan esquemáticamente ejemplos de realización de hardware de los medios de cálculo de dirección según la invención en los siguientes casos:

- 5 - procesador paralelo monodimensional y estructura de datos monodimensional (figura 6a),
- procesador paralelo monodimensional y estructura de datos bidimensional (figura 6b),
- procesador paralelo monodimensional y estructura de datos multidimensional (figura 6c),
- procesador paralelo bidimensional y estructura de datos bidimensional (figura 6d y 6e).

De una figura a otra, los mismos elementos están indicados por las mismas referencias.

10 Según la invención, la instrucción define no la dirección en la cual cada PE debe leer un operando o escribir un resultado sino informaciones sobre la posición del campo de acción en el seno de la estructura de datos procesada. Esta posición puede estar definida bien en valor absoluto, bien en valor relativo respecto de una posición anterior. Un cierto número de etapas de cálculo son por lo tanto necesarios para deducir una dirección propia de cada PE.

15 Una primera etapa consiste en calcular la posición del campo de acción a partir de la instrucción en proceso de ejecución si la misma no está definida en valor absoluto por la instrucción. Tras esta primera etapa, el resultado es independiente del PE.

Una segunda etapa consiste en calcular la dirección en la cual cada PE debe leer un operando o escribir un resultado a partir de la posición del campo de acción. Como se verá en los ejemplos más adelante, la dirección es obtenida por suma o sustracción de un valor ligado a la posición del campo y de un valor ligado a la posición del PE. Tras esta segunda etapa, el resultado depende del PE.

20 Según lo que se busca a optimizar (por ejemplo: más bien el número de hilos o más bien el número de puertas), estos cálculos pueden estar bien realizados al máximo en modo central por lo tanto de una vez por todas y compartidos entre varios PEs, bien por el contrario realizados al máximo en modo local al nivel de cada PE o cualquier solución intermedia de agrupamiento de ciertos cálculos comunes. Al ser muy numerosas las variantes, no es posible describirlas todas y se limitará a algunos ejemplos.

25 De este modo, una implementación ilustrada en la figura 5 puede consistir en calcular la posición del campo de acción de manera central por los medios de cálculo 50 mientras que cada PE utiliza esta información para deducir una dirección Ad en su conjunto de memorias y/o de registros; esta dirección es obtenido por los medios de cálculo de dirección 41, 42, 43 de cada PE. La ventaja de este principio es minimizar el volumen de cálculos a realizar en cada PE y por lo tanto el volumen de hardware.

30 Si se lleva la deslocalización al extremo, otra implementación puede consistir en calcular todo al nivel de cada PE: la posición del campo de acción y la dirección correspondiente. El inconveniente de esta variante es aumentar el volumen de cálculos a realizar en cada PE y por lo tanto el volumen de hardware. Sin embargo, puede permitir reducir el volumen de informaciones a transmitir a todos los PEs y por lo tanto el número de hilos en el circuito, lo que puede ser interesante en ciertos casos.

35 Contrariamente, se puede desear reducir al mínimo el volumen de cálculos a realizar en cada PE y por lo tanto el volumen de hardware. Otra implementación puede entonces consistir en mutualizar ciertos cálculos comunes con varios PEs al nivel de un grupo de PEs, incluso de manera central. Se verá un ejemplo en el cuarto ejemplo de realización.

40 La realización de los medios de cálculo de dirección 41, 42, 43 de cada PE puede realizarse mediante un hardware por cable o un hardware más flexible, por ejemplo un procesador programable. Los cálculos pueden estar asimismo realizados como software puro por los ALU de los PEs, por lo tanto sin añadido de hardware adicional, siempre que cada PE dispone de un registro de indirección. Tal realización, aún más económica de hardware, puede por lo tanto ser interesante en ciertas aplicaciones.

45 Son posibles diferentes elecciones en cuanto a las informaciones que definen la posición del campo de acción. Por ejemplo, el centro del campo o un extremo en una dirección definida por adelantado. Como todas estas elecciones son equivalentes a aproximadamente una traslación y que el paso de uno al otro es trivial, no se insistirá más en este punto. De hecho, estas elecciones tienen en común el hecho de que la posición del campo de acción está definida por la posición de un baricentro obtenido por combinación lineal de los extremo, siendo la suma de los pesos de los extremos igual a 1. Se ha elegido para describir los ejemplos de realización el caso donde la información elegida para definir la posición de un campo de acción es el punto del campo en el que todas las coordenadas son mínimas. Dicho de otro modo, el extremo del campo en la dirección del punto al infinito vale $-\infty$ en cada coordenada. Siendo en las figuras 2, 3 y 4, el punto de coordenadas (7,3) para la primera etapa de la primera operación.

La invención se puede aplicar a procesadores paralelos organizados en topologías mono, bi, tri o multidimensionales, y en cada una de estas topologías, en estructuras de datos monodimensionales (por ejemplo: señal), bidimensionales (por ejemplo: imagen), tridimensionales (por ejemplo volumen 3D para la resolución de

ecuación en derivadas parciales) o multidimensionales, pero también para diferentes modos de almacenamiento de cada estructura de datos en cada tipo de procesador paralelo.

- 5 La invención se puede aplicar también a arquitecturas de tratamiento que hacen intervenir un gran número de procesadores elementales que ejecutan el mismo procesamiento sobre datos diferentes, incluido si esta arquitectura no forma parte de la categoría de los SIMD. Por ejemplo, si la estructura de control no es un secuenciador o no es programable o si la programación se realiza a un nivel más elevado que una instrucción.

Ahora se detallará algunos ejemplos de realización.

El primer ejemplo de realización se refiere al caso de un procesador paralelo monodimensional por ejemplo un anillo y de una estructura de datos monodimensional por ejemplo una señal.

- 10 Se considera el caso en que el usuario ha guardado sus datos en los procesadores paralelos de la siguiente manera: el elemento x de la señal está guardado en el PE $n^{\circ} p$ en la dirección Ad con:

$$p = x \text{ Mod } Np$$

$$Ad = \text{Int}(x / Np)$$

Siendo Np el número de PEs.

- 15 En este caso, el conjunto de los PEs es visto como un segmento del ancho Np y el campo de acción es un segmento de ancho Np : contiene Np elementos.

Se considera el caso en que la información utilizada para definir la posición del campo es la abscisa del extremo izquierdo del campo, aquella cuya abscisa es mínima. Siendo $Xmin$ esta abscisa, el campo de acción se extiende por lo tanto de $Xmin$ a $Xmin+Np-1$. Cada PE está concernido por uno solo de estos elementos.

- 20 La ecuación que permite calcular la dirección del elemento que concierne el PE $n^{\circ} p$ en el seno del campo es la siguiente:

$$Ad = \text{Int}((Xmin + Np - 1 - p) / Np)$$

Esta ecuación puede dar lugar a numerosas realizaciones, que van del hardware puro al software puro.

- 25 Cabe sin embargo mencionar una realización de hardware particularmente simple en el caso donde Np es una potencia de 2. En este caso en efecto, si LNp designa el logaritmo de base 2 de Np , el modo de cálculo de Ad a partir de $Xmin$ es el siguiente:

Sumar $Xmin$ y $(Np-1-p)$. Se señala por otra parte que, para cada PE, $(Np-1-p)$ es una constante igual al complemento de 2 de p (que se obtiene invirtiendo todos los bits de p) y que esta constante puede conservarse en memoria en el PE $n^{\circ} p$.

Suprimir los LNp últimos bits del resultado.

- 30 En total, hay una sola suma por PE.

Esta realización está ilustrada en la figura 6a para $Np=16$, $LNp=4$ y $Xmin$ está en 12 bits. Se caracteriza por que la dirección en la que un PE debe leer un operando o escribir un resultado es obtenido por suma (o sustracción) de un valor ligado a la posición del campo y de un valor ligado a la posición del PE.

- 35 Este tipo de aplicación particularmente sencillo puede trasladarse a todos los ejemplos de realización que se van a describir. Más en general, se puede trasladar a todo tipo de procesador paralelo organizado en topologías mono, bi, tri o multidimensionales y, en cada una de estas topología, en estructuras de datos mono (por ejemplo: señal), bi (por ejemplo: imagen), tri (por ejemplo: volumen 3D) o multidimensionales. Basta de hecho con retomar el cálculo en función del tipo de procesador paralelo, de la estructura de datos y del almacenamiento elegido y a continuación adaptar en consecuencia la aplicación.

- 40 El segundo ejemplo de realización se refiere al caso de un procesador paralelo monodimensional, por ejemplo un anillo y de una estructura de datos bidimensionales, por ejemplo una imagen.

Se puede retomar el cálculo realizado para una estructura de datos monodimensional trasladándolo a una estructura de datos bidimensional, deducir ecuaciones y adaptar en consecuencia la aplicación.

- 45 Se puede utilizar también un procedimiento que permite procesar una estructura de datos bidimensional volviendo al caso que se acaba de analizar para una estructura de datos monodimensionales.

Siendo N_p el número de PEs, L_x el ancho de la imagen a tratar, es decir el número de píxeles en cada línea y L_y su altura, es decir el número de píxeles en cada columna. Se trata más exactamente de los límites superiores para los números de PEs, de píxeles en cada línea y cada columna. En efecto, en la mayoría de los casos y por motivos de simplificación del hardware, se prefiere sumergir la imagen a procesar en una imagen mayor y por lo tanto elegir valores L_x y L_y superiores a las dimensiones de la imagen a procesar. Por una parte, si se deben procesar varias imágenes de dimensiones diferentes, se puede preferir realizar una elección única que permita procesar la más ancha de las mismas. Por otra parte, para simplificar la aplicación cuando la misma es de hardware, se elige a menudo una potencia de 2.

Una manera de volver a una estructura de datos monodimensional (dicho de otro modo, una manera de guardar los píxeles en los PEs) está descrita, para cada pixel (x, y) :

$$w = x + L_x * y$$

Este tipo de almacenamiento utilizado en la figura 3, vuelve a considerar el conjunto de los PEs como un bloque horizontal de PEs, siendo cada PE responsable de una columna o de varias columnas. Difiere del analizado anteriormente y representado por la figura 2 que vuelve a considerar el conjunto de los PEs como un bloque vertical de PEs, siendo cada PE responsable de una línea o de varias líneas. En este caso, hay que describir, para cada pixel (x, y) :

$$w = y + L_y * x$$

Comparando estas dos fórmulas, se entiende fácilmente como pasar de un almacenamiento a otro. Se indicará por otra parte posteriormente una manera general para pasar de un almacenamiento a otro. Por lo tanto se limitará por ahora al primer almacenamiento, utilizado en la figura 3 y caracterizado por la transformación:

$$w = x + L_x * y$$

Gracias a esta transformación, se pueden guardar los píxeles en el procesador paralelo como se ha hecho para la primera realización: el elemento w de la señal está guardado en el PE $n^\circ p$ en la dirección Ad con:

$$p = w \text{ Mod } N_p$$

$$Ad = \text{Int}(w / N_p)$$

En este caso, el campo de acción es un segmento horizontal de ancho N_p .

Se considera, a título de ejemplo, el caso en que la información utilizada para definir la posición del campo es la posición del extremo del campo cuyas coordenadas son mínimas.

Siendo X_{min} , Y_{min} las coordenadas de este punto.

Se formula:

$$W_{min} = X_{min} + L_x * Y_{min}$$

La ecuación que permite que el PE $n^\circ p$ calcule la dirección del pixel que le concierne en el seno del campo es la siguiente:

$$Ad = \text{Int}((W_{min} + N_p - 1 - p) / N_p)$$

Es la misma ecuación que en la primera realización. Simplemente, W_{min} sustituye a X_{min} . De este modo se tiene un medio simple de trasladar, a una estructura de datos bidimensionales, cualquier aplicación utilizable para una estructura de datos monodimensional y en particular la que se ha detallado. Si N_p y L_x son potencia de 2, solo hay una sola suma por PE. Se designa por LN_p y LL_x sus logaritmos respectivos. Para simplificar más aun la realización, en lugar de efectuar esta suma sobre W_{min} , se observa que solo se refiere a X_{min} es decir a solo L_x bits. Las otra operaciones son concatenaciones de bits, mucho más simples que sumas. Así simplificado, el modo de cálculo de Ad a partir de X_{min} , Y_{min} es el siguiente:

Sumar X_{min} y (N_p-1-p) .

Suprimir los LN_p últimos bits del resultado.

Concatenar en pesos bajos la cantidad anterior representada en $LL_x - LN_p$ bits y en pesos fuertes Y_{min} .

Esta aplicación está ilustrada por la figura 6b para $N_p=16$, $LN_p=4$, X_{min} y Y_{min} en 6 bits.

El tercer ejemplo de realización es el caso de un procesador paralelo monodimensional (por ejemplo un anillo) y de una estructura de datos de dimensión superior a 2.

Se considera en primer lugar el caso de una estructura de datos tridimensional (por ejemplo un volumen 3D).

5 Siendo L_x el ancho del volumen a procesar y L_y su profundidad y L_z su altura, o más exactamente, por razones que sean visto anteriormente, límites superiores de estas tres cantidades.

Una manera de volver a una estructura de datos monodimensionales (dicho de otro modo, una manera de guardar los píxeles en los PEs) es escribir, para cada píxel (x, y, z) :

$$w = x + L_x * (y + L_y * z)$$

Se considera para lo sucesivo el almacenamiento $W = x + L_x * (y + L_y * z)$.

10 Se puede trasladar fácilmente el razonamiento a otros almacenamientos tales como:

$$w = x + L_x * (z + L_z * y)$$

$$w = y + L_y * (x + L_x * z)$$

$$w = y + L_y * (z + L_z * x)$$

$$w = z + L_z * (x + L_x * y)$$

$$w = z + L_z * (y + L_y * x)$$

Gracias a esta transformación, se puede guardar los píxeles en el procesador paralelo como se ha hecho para la primera realización: el elemento w de la señal está guardado en el PE $n^o p$ en la dirección Ad con:

$$p = w \text{ Mod } N_p$$

15
$$Ad = \text{Int}(w / N_p)$$

En este caso, el conjunto de los PEs es visto como un segmento horizontal de ancho N_p y el campo de acción es un segmento horizontal de ancho N_p .

20 Se considera a título de ejemplo, el caso en que la información utilizada para definir la posición del campo es la posición del extremo del campo cuyas coordenadas son mínimas. Siendo X_{min} , Y_{min} , Z_{min} las coordenadas de este punto:

$$\text{Se formula: } W_{min} = X_{min} + L_x * (Y_{min} + L_y * Z_{min})$$

La ecuación que permite que el PE $n^o p$ calcule la dirección del píxel que le concierne en el seno del campo es la siguiente:

$$Ad = \text{Int}((W_{min} + N_p - 1 - p) / N_p)$$

25 Es la misma ecuación que en la primera realización. Simplemente, W_{min} sustituye a X_{min} . De este modo se tiene un medio simple de trasladar, a una estructura de datos tridimensionales, cualquier aplicación utilizable para una estructura de datos monodimensional y en particular la que se ha detallado. Si N_p , L_x y L_y son potencia de 2, solo hay una sola suma por PE. Se designa por LN_p , LL_x y LL_y sus logaritmos respectivos. Para simplificar más aun la realización, en lugar de efectuar esta suma sobre W_{min} , se observa que solo se refiere a X_{min} es decir a solo L_x bits. Las otra operaciones son concatenaciones de bits, mucho más simples que sumas. Así simplificado, el modo de cálculo de Ad a partir de X_{min} , Y_{min} es el siguiente:

- 30 Sumar X_{min} y $(N_p - 1 - p)$.
- Suprimir los LN_p últimos bits del resultado.
- Concatenar en pesos bajos la cantidad anterior representada en $L_x - LN_p$ bits y en pesos fuertes Y_{min} representado en L_y bits y en pesos fuerte Z_{min} representado en L_z bits.
- 35

El mismo principio puede ser aplicado a estructuras de datos de cualquier dimensión. De este modo, en dimensión 4 basta con:

escribir para cada pixel (x, y, z, t):

$$w = x + Lx * (y + Ly * (z + Lz * t)),$$

definir la posición del campo de acción gracias a Xmin, Ymin, Zmin, Tmin, formular: Wmin = Xmin + Lx + (Ymin + Ly + (Zmin + Lz + Tmin)), y finalmente calcular para cada PE:

$$Ad = \text{Int}((Wmin + Np - 1 - p) / Np).$$

El cuarto ejemplo de realización es el caso de un procesador paralelo bidimensional por ejemplo un toro y de una estructura de datos bidimensional por ejemplo una imagen.

Siendo Nx el número de PEs en el eje de los x, Ny el número de PEs en el eje de los y, y por lo tanto Nx*Ny es el número total de PEs. Siendo Lx el ancho de la imagen a procesar y Ly su altura, o como se ha visto anteriormente, límites superiores de estas dos cantidades. Preferiblemente, se elige Lx y Ly respectivamente entre los múltiplos de Nx y de Ny. Se designa por (px, py) el PE de coordenadas (px, py) en el toro de los PEs.

Se considera a título de ejemplo, el caso en que el usuario ha guardado sus datos en el procesador paralelo de la siguiente manera: el pixel (x, y) de la imagen está guardado en el PE (px, py) en la dirección Ad con:

$$px = x \text{ Mod } Nx$$

$$py = y \text{ Mod } Ny$$

$$Adx = \text{Int}(x / Nx)$$

$$Ady = \text{Int}(y / Ny)$$

$$Ad = Adx + (Lx / Nx) * Ady$$

20 En este caso de almacenamiento utilizado en las figuras 4, el conjunto de los PEs es visto como un rectángulo de ancho Nx y de altura Ny y el campo de acción es un rectángulo de ancho Nx y de altura Ny.

Se observa que este almacenamiento está calcado del almacenamiento utilizado para una estructura de datos monodimensional. De hecho, reproduce este almacenamiento en cada una de las 2 direcciones del espacio. El dispositivo de tratamiento será por lo tanto exactamente el mismo, pero deberá aplicarse en cada una de las 2 direcciones del espacio.

Se considera siempre a título de ejemplo, el caso en que la información utilizada para definir la posición del campo es la esquina del campo cuyas coordenadas son más débiles, es decir el extremo del campo según la dirección del punto al infinito que vale $-\infty$ en cada coordenada. Siendo Xmin, Ymin las coordenadas de este punto.

30 Las ecuaciones que permiten al PE (px, py) calcular la dirección del elemento que le concierne en el seno del campo son las siguientes:

$$Adx = \text{Int}((Xmin + Nx - 1 - px) / Nx)$$

$$Ady = \text{Int}((Ymin + Ny - 1 - py) / Ny)$$

$$Ad = Adx + (Lx / Nx) * Ady$$

Se encuentra por lo tanto dos veces, una vez en el eje de las x, una vez en el eje de las y, la ecuación encontrada en la primera realización y característica del dispositivo propuesto. Como se ha dicho, la misma puede dar lugar a numerosas aplicaciones, que van del hardware puro al software puro.

Se hace mención sin embargo, como se ha hecho anteriormente para las realizaciones precedentes, una aplicación de hardware particularmente simple en el caso en que N_x , N_y , L_x y L_y son potencias de 2. Si LN_x designa el logaritmo de base 2 de N_x , LN_y el logaritmo de base 2 de N_y , LL_x el logaritmo de base 2 de L_x , LL_y el logaritmo de base 2 de L_y , el modo de cálculo de A_d a partir de X_{min} , Y_{min} es el siguiente:

5 Sumar X_{min} y (N_x-1-p_x) . Se observa que X_{min} , N_x y p_x son valores comunes de una columna de PEs. Este cálculo se tiene que hacer de este modo una sola vez por columna de PEs. Además, la cantidad (N_x-1-p_x) es una constante para una columna de PEs. Puede ser conservada en una memoria única para una columna de PEs.
Suprimir los LN_x últimos bits del resultado \rightarrow A_{dx} en LL_x-LN_x bits.

10 Sumar Y_{min} y (N_y-1-p_y) . Se observa que Y_{min} , N_y y p_y son valores comunes de una línea de PEs. Este cálculo se tiene que hacer de este modo una sola vez por columna de PEs. Además, la cantidad (N_y-1-p_y) es una constante para una línea de PEs. Puede ser conservada en una memoria única para una línea de PEs.

Suprimir los LN_y últimos bits del resultado \rightarrow A_{dy} en LL_y-LN_y bits.

15 $A_d = A_{dx} + (L_x / N_x) * A_{dy}$, lo que significa concatenar en peso débil A_{dx} representado en LL_x-LN_x bits y en peso fuerte A_{dy} representado en LL_y-LN_y bits.

Se verifica que el tipo de aplicación anterior se ha podido trasladar. En lo sucesivo hay una sola suma por línea de PEs y una sola suma por columna de PEs. Además, estas sumas se refieren a muy pocos bits. De ahí una aplicación aun más ergonómica que la necesaria para las realizaciones anteriores. Esta aplicación está ilustrada en la figura 6c para $N_x=4$, $LN_x=4$, $LN_x=2$, $LN_y=2$, X_{min} e Y_{min} en 6 bits.

20 Se han analizado sucesivamente 4 realizaciones elegidas a título de ejemplo:

- procesador paralelo monodimensional y estructura de datos monodimensional
- procesador paralelo monodimensional y estructura de datos bidimensional
- procesador paralelo monodimensional y estructura de datos multidimensional
- procesador paralelo bidimensional y estructura de datos bidimensional.

25 Sin embargo, el principio propuesto puede aplicarse a procesadores paralelos organizados en topologías mono, bi, tri o multidimensionales y, en cada una de estas topologías, en estructuras de datos mono, bi, tri o multidimensionales.

Se puede generalizar de la siguiente manera. Siendo n la dimensión de la estructura de datos y p la dimensión de la topología del procesador paralelo.

30 Si $n=p$, se aplica el principio en cada una de las p direcciones, exactamente como se ha visto para $n=2$ (cuarta realización).

Si $n>p$, como se ha visto en el caso de la segunda y la tercera realización, se reduce la dimensión de la estructura, eligiendo dos ejes (y a continuación otros dos, y más tardes otros dos, ... hasta obtener una estructura de dimensión p) y se reducen estos dos ejes a uno solo aplicando una fórmula del tipo:

35
$$w = x + L_x * y.$$

Si $n<p$, se aumenta la dimensión de la estructura, eligiendo un eje, (y a continuación otro, y más tarde otro, ..., hasta obtener una estructura de dimensión p) y se recorta este eje en secciones para desdoblado. De este modo, para pasar de una coordenada x a dos coordenadas w_1 y w_2 , se elige un ancho de sección L_x y se aplica una fórmula del tipo:

$$w_1 = x \text{ Mod } L_x$$

40
$$w_2 = \text{Int} (x / L_x).$$

Se han analizado hasta ahora realizaciones que aplican un modo de almacenamiento en el cual dos elementos sucesivos en un eje (el eje de las x por ejemplo) están conservados y procesados por dos PEs diferentes. Típicamente, en el caso monodimensional (primera realización), se ha considerado, a título de ejemplo, el caso en que el usuario ha guardado el elemento x de la señal en el PE $n^\circ p$ en la dirección A_d con:

$$p = x \text{ Mod } Np$$

$$Ad = \text{Int}(x / Np).$$

Asimismo, en el caso bidimensional (segunda realización), se ha tomado por ejemplo, el caso en que el usuario ha guardado el pixel (x, y) en el PE nº p en la dirección Ad con:

$$w = x + Lx * y$$

$$p = w \text{ Mod } Np$$

$$Ad = \text{Int}(w / Np).$$

5 Solo son ejemplos. El procedimiento puede aplicarse a muchos otros modos de almacenamiento.

De este modo, para seguir con el mismo caso monodimensional, el usuario puede desear definir una dimensión Lx y guardar el elemento x de la señal en el PE nº p en la dirección Ad con:

$$x1 = x \text{ Mod } Lx$$

$$x2 = \text{Int}(x / Lx)$$

$$p = x2 \text{ Mod } Np$$

$$Ad = x1 + Lx * \text{Int}(x2 / Np).$$

10 Se trata de un modo de almacenamiento en el cual un segmento de Lx elementos sucesivos en el eje de las x es conservado y tratado por el mismo PE, mientras que el siguiente PE conserva y trata el siguiente segmento.

La ecuación que permite que el PE nº p calcule a partir de Xmin, la dirección del elemento que le concierne en el seno del campo es la siguiente:

$$Xmin1 = Xmin \text{ Mod } Lx$$

$$Xmin2 = \text{Int}(Xmin / Lx)$$

$$Ad = Xmin1 + Lx * \text{Int}((Xmin2 + Np - 1 - p) / Np).$$

Como se ve, esto significa aplicar el procedimiento en Xmin2 y a no tocar Xminº1.

15 Se considera el caso en que Lx y Np son potencia de 2 para el cual una aplicación de hardware particularmente simple es posible. Se designa por LNp y LLx sus logaritmos respectivos. Las ecuaciones anteriores pueden interpretarse de este modo:

Recortar Xmin en dos partes, constituyendo los pesos débiles en LLx bits, constituyendo los pesos fuertes Xmin2.

20 Aplicar a Xmin2 el dispositivo propuesto en la primera realización. El resultado es : $\text{Int}((Xmin2 + Np - 1 - p) / Np)$.

Concatenar en pesos débiles Xmin representado en Lx bits y en pesos fuertes el resultado del dispositivo.

Dicho de otro modo, en el caso en que Lx y Np son potencias de 2, es preciso

25 aparatar los LLx bits de peso débil de Xmin,
 aplicar el dispositivo a otros bits de Xmin,
 concatenar en peso débil los LLx bits apartados con el resultado del dispositivo.

Esta extensión es totalmente independiente de la extensión mencionada en el párrafo anterior; puede acoplarse a la misma. Se aplica por lo tanto al caso multidimensional, tanto para la topología del procesador paralelo como para la estructura de datos. Basta con aplicarla en cada una de las dimensiones. Consecuentemente, el procedimiento se aplica a cualquier modo de almacenamiento en el cual un paralelepípedo de dimensión L_x sobre el eje de los X, L_y en el eje de los y, ... es conservado y tratado por el mismo PE, mientras que los PEs vecinos (para la topología del procesador paralelo) conservan y tratan los paralelepípedos vecinos (para la topología de la estructura de datos).

Se puede generalizar a otros almacenamientos. Se va a mostrar en el caso multidimensional disponiéndose en el caso en que las dimensiones de la estructura de datos (L_x, L_y, L_z, \dots) así como el número de procesadores (N_x, N_y, N_z) en cada dirección son potencias de 2. El hecho de elegir potencias de 2 permite explicar las operaciones a realizar en las representaciones binarias, lo que es más simple, pero se puede generalizar a cualesquiera valores de $L_x, L_y, L_z, \dots, N_x, N_y, N_z, \dots$ sustituyendo las operaciones descritas en las representaciones binarias por fórmulas matemáticas que utilizan multiplicaciones, divisiones, módulos y partes enteras como se indica en los ejemplos anteriores. Para guardar la estructura de datos en el procesador y a continuación aplicar el procedimiento a este almacenamiento, se puede proceder de siguiente modo:

Concatenar en una palabra única de $LL_x + LL_y + LL_z \dots$ bits las diferentes coordenadas del punto utilizado para definir la posición del campo, esto significa formar lo que se ha denominado anteriormente w . Permutar el orden de los bits de w (hay tantos almacenamientos posibles como permutaciones posibles); siendo w' la palabra obtenido de este modo.
Recortar w' en palabras de dimensión LL_x, LL_y, LL_z, \dots bits; siendo w_x', w_y', w_z', \dots las palabras obtenidas de este modo.

Aplicar el procedimiento en cada una de las dimensiones del procesador paralelo, es decir:

- añadir $(N_x - 1 - p_x)$ a w_x' ,
- suprimir los LN_x últimos bits del resultado; se obtiene Ad_x' ,
- añadir $(N_y - 1 - p_y)$ a w_y' ,
- suprimir los LN_y últimos bits del resultado; se obtiene Ad_y' ,
- y así sucesivamente en cada dirección del procesador paralelo.

Permutar de nuevo el orden de los bits $Ad_x', Ad_y', Ad_z', \dots$ según una permutación que no está necesariamente relacionada con la presente permutación. Se obtienen entonces las direcciones de almacenamiento Ad_x, Ad_y, Ad_z en cada una de las direcciones del procesador paralelo.

Concatenar estas direcciones en una palabra única Ad que representa la dirección concernida por el campo en el seno de cada PE.

Para ilustrar esto último, se vuelve al ejemplo ilustrado por la figura 2. Se recuerda que este ejemplo se refiere a un procesador paralelo monodimensional de 16 PEs organizados en anillo, que la estructura de datos es una imagen de dimensión $L_x = 64$ y $L_y = 64$ y que los píxeles de la imagen son guardados en línea. Se tiene entonces

$$\begin{aligned} N_p = 16 &\rightarrow LN_p = 4 \\ N_p = 64 &\rightarrow LN_p = 6 \\ N_p = 64 &\rightarrow LN_p = 6 \end{aligned}$$

La operación para pasar de X_{min}, Y_{min} a Ad es la siguiente:

Concatenar X_{min}, Y_{min} en una palabra única de $6 + 6 = 12$ bits. Siendo w la palabra así obtenida. Permutar los 6 bits de peso débil y los 6 bits de peso fuerte de w . Siendo w' la palabra así obtenida. Aplicar el procedimiento en el caso de un procesador paralelo monodimensional, es decir:

- Sumar $(16 - 1 - p)$ a w' ,
- Suprimir los 4 últimos bits del resultado. Se obtiene Ad' .
- Permutar los 2 bits de peso débil y los 6 bits de peso fuerte de Ad' . La palabra así obtenida es la dirección Ad .

Esto queda bien enmarcado en la generalización que se quiere ilustrar. Esta aplicación está ilustrada en la figura 6 con $N_x=4, N_y=4, LN_x=2, LN_y=2, X_{min}$ e Y_{min} en 6 bits: es una variante de la realización de la figura 6 con una modificación del modo de almacenamiento.

Cabe señalar que se puede facilitar la lectura de la figura 6d reduciendo los hilos sin modificar en modo alguno el dispositivo. Se obtiene entonces la figura 6e.

Muchos modos distintos de almacenamiento son posibles. Las variantes más simples consisten, por ejemplo, en cambiar los signos en las fórmulas o cambiar el orden de los PEs. Es imposible mencionarlas todas.

REIVINDICACIONES

- 1.- Procesador paralelo que comprende procesadores elementales (3) que comprenden cada uno al menos una unidad de cálculo y al menos una memoria que incluye palabras de memoria, y dispuestos según una topología, con una posición determinada en el seno de esta topología y capaces de ejecutar simultáneamente una misma instrucción sobre datos diferentes, incluyendo la instrucción la lectura de al menos un operando y/o incluyendo la escritura de al menos un resultado, **caracterizado porque** la instrucción define el conjunto de las lecturas y/o el conjunto de las escrituras de los procesadores elementales, cada uno en su propia memoria como una lectura y/o escritura en un campo de acción, siendo un campo de acción un conjunto de palabras de memoria a razón de una palabra de memoria por procesador elemental, estando esta palabra de memoria situada en la memoria del procesador elemental, siendo un campo de acción de un operando el conjunto de las palabras de memoria leídas por los procesadores elementales, cada uno en su propia memoria para adquirir este operando, siendo un campo de acción de un resultado el conjunto de las palabras de memoria escritas por los procesadores elementales, cada una en su propia memoria para almacenar este resultado, **porque** la instrucción comprende para cada operando y/o cada resultado, informaciones relativas a la posición de este campo de acción, **porque** la posición del campo de acción está definida como una posición en el seno de una estructura de datos única de tipo tabla de dimensión N, siendo N un número entero igual o superior a 1, **porque** esta tabla está repartida en las memorias de los diferentes procesadores elementales, **porque** el procesador paralelo comprende medios de cálculo de la dirección de cada operando y/o cada resultado en el seno de cada procesador elemental, en función de la posición del campo de acción y de la posición del procesador elemental en el seno de la topología.
- 2.- Procesador paralelo según la reivindicación anterior, **caracterizado porque** comprende medios (50) de cálculo de la posición del campo de acción en función de dichas informaciones, vinculados a los medios de cálculo de la dirección.
- 3.- Procesador paralelo según una de las reivindicaciones anteriores, **caracterizado porque** los medios (41, 42, 43) de cálculo de la dirección en el seno de cada procesador elemental están localizados al nivel de cada procesador elemental.
- 4.- Procesador paralelo según la reivindicación anterior, **caracterizado porque** los medios (41, 42, 43) de cálculo de la dirección en el seno de cada procesador elemental están localizados al nivel de la unidad de cada procesador elemental.
- 5.- Procesador paralelo según una de las reivindicaciones anteriores, **caracterizado porque** los medios de cálculo de la dirección en el seno de cada procesador elemental están localizados al nivel de cada procesador elemental y parcialmente agrupados entre varios procesadores elementales.
- 6.- Procesador paralelo según una de las reivindicaciones anteriores tomada en combinación con la reivindicación 2, **caracterizado porque** los medios de cálculo de la posición del campo de acción son:
- comunes a todos los procesadores elementales o
 - localizados en totalidad o en parte al nivel de los medios de cálculo de la dirección de los datos.
- 7.- Procesador paralelo según una de las reivindicaciones anteriores, **caracterizado porque** los procesadores elementales están dispuestos según una topología de P dimensiones, siendo P un número entero superior o igual a 1.
- 8.- Procesador paralelo según una de las reivindicaciones anteriores, **caracterizado porque** los procesadores elementales están dispuestos según una topología de al menos una dimensión, para cada dimensión los medios de cálculo de la dirección del elemento de la estructura de datos que concierne un procesador elemental de abscisa p según dicha dimensión son obtenidos por combinación de un valor ligado a la posición del campo y de un valor ligado a la posición del procesador elemental.
- 9.- Procesador paralelo según una de las reivindicaciones anteriores, **caracterizado porque** el campo de acción que presenta extremos definidos por sus coordenadas, la posición del campo de acción está definida por la posición de un baricentro obtenido por combinación lineal de estos extremos, siendo la suma de los pesos de los extremos igual a 1.
- 10.- Procesador paralelo según una de las reivindicaciones anteriores, **caracterizado porque** el campo de acción que presenta extremos definidos por sus coordenadas, la posición del campo de acción está definida por la posición del extremo cuyas coordenadas son mínimas.
- 11.- Procesador paralelo la reivindicación anterior, **caracterizado porque** los procesadores que están dispuestos según una topología de al menos una dimensión, para cada dimensión los medios de cálculo de la dirección del elemento de la estructura de datos que concierne un PE de abscisa p según dicha dimensión comprenden medios de cálculo de

$$Ad = \text{Int}((W_{\min} + N_p - 1 - p) / N_p),$$

siendo N_p el número de procesadores elementales considerado en dicha dimensión, estando W_{\min} definido en dicha dimensión en función de las coordenadas de dicha posición.

5 12.- Procesador paralelo según la reivindicación anterior, **caracterizado porque** los procesadores están dispuestos según una topología de una dimensión y la estructura de datos es monodimensional y **porque** la posición del campo de acción está definida por X_{\min} y $W_{\min}=X_{\min}$.

13.- Procesador paralelo según la reivindicación anterior, **caracterizado porque** siendo N_p una potencia de 2, los medios de cálculo de Ad incluyen medios para:

- 10
- Sumar X_{\min} y (N_p-1-p) .
 - Suprimir los LNp últimos bits del resultado, siendo LNp el logaritmo de base 2 de N_p .

14.- Procesador paralelo según la reivindicación 10, **caracterizado porque** los procesadores están dispuestos según una topología de una dimensión y la estructura de datos es bidimensional, definida en función de X e Y , siendo la dimensión de la estructura según X inferior a un valor predeterminado L_x y **porque** la posición del campo de acción está definida por X_{\min} , Y_{\min} y $W_{\min}=X_{\min} + L_x * Y_{\min}$.

15 15.- Procesador paralelo según la reivindicación anterior, **caracterizado porque** siendo N_p y L_x potencias de 2, los medios de cálculo de Ad , incluyen medios para:

- 20
- Sumar X_{\min} y (N_p-1-p) .
 - Suprimir los LNp últimos bits del resultado, siendo LNp el logaritmo de base 2 de N_p ,
 - Concatenar en peso débil la cantidad anterior representada en $\text{LLx} - \text{LNp}$ bits y en peso fuerte Y_{\min} siendo LLx el logaritmo de base 2 de L_x .

16.- Procesador paralelo según la reivindicación 10, **caracterizado porque** los procesadores están dispuestos según una topología de una dimensión, y la estructura de datos es tridimensional, definida en función de X , Y y Z , siendo las dimensiones de la estructura según X e Y respectivamente inferiores a valores predeterminados L_x y L_y y **porque** la posición del campo de acción está definida por X_{\min} , Y_{\min} , Z_{\min} y $W_{\min}=X_{\min} + L_x * (Y_{\min} + L_y * Z_{\min})$.

25 17.- Procesador paralelo según la reivindicación anterior, **caracterizado porque** siendo N_p , L_x y L_y potencias de 2, los medios de cálculo de Ad , incluyen medios para:

- 30
- Sumar X_{\min} y (N_p-1-p) .
 - Suprimir los LNp últimos bits del resultado, siendo LNp el logaritmo de base 2 de N_p ,
 - Concatenar en peso débil la cantidad anterior representada en $\text{LLx} - \text{LNp}$ bits, en peso medio Y_{\min} representado en L_y bits y en peso fuerte Z_{\min} representado en L_z bits, siendo LLx el logaritmo de base 2 de L_x .

35 18.- Procesador paralelo según la reivindicación 10, **caracterizado porque** los procesadores están dispuestos según una topología de una dimensión y la estructura de datos es de cuatro dimensiones, definida en función de X , Y , Z y T , siendo la dimensión de la estructura según X , Y y Z respectivamente inferiores a valores predeterminados L_x , L_y y L_z y **porque** la posición del campo de acción está definida por X_{\min} , Y_{\min} , Z_{\min} , T_{\min} y $W_{\min}=X_{\min} + L_x * (Y_{\min} + L_y * (Z_{\min} + L_z * T_{\min}))$.

40 19.- Procesador paralelo según la reivindicación 10, **caracterizado porque** los procesadores están dispuestos según una topología de dos dimensiones y, comprende respectivamente N_x y N_y procesadores según una y otra dimensión, la estructura de datos bidimensional, definida en función de X e Y , siendo la dimensión de la estructura según X inferior a un valor predeterminado L_x y **porque** la posición del campo de acción está definida por X_{\min} , Y_{\min} , y

- la dirección Ad_x del elemento que concierne al PE n° p_x considerado según X en el seno del campo es obtenida por la fórmula:

45

$$Ad_x = \text{Int}((X_{\min} + N_x - 1 - p_x) / N_x),$$

- y la dirección Ad_y del elemento que concierne al PE n° p_y considerado según Y en el seno del campo es obtenida por la fórmula:

$$Ad_y = \text{Int}((Y_{\min} + N_y - 1 - p_y) / N_y).$$

20.- Procesador paralelo según la reivindicación anterior, **caracterizado porque** siendo N_x , N_y , L_x y L_y potencias de 2, siendo LL_x y LL_y respectivamente el logaritmo de base 2 de L_x y L_y , los medios de cálculo de Ad incluyen medios para:

- 5 – sumar X_{min} y (N_x-1-p_x) .
- suprimir los LN_x últimos bits del resultado, siendo LN_x el logaritmo de base 2 de N_x , se obtiene Ad_x en LL_x-LN_x bits,
- sumar Y_{min} y (N_y-1-p_y) ,
- suprimir los LN_y últimos bits del resultado, siendo LN_y el logaritmo de base 2 de N_y , se obtiene Ad_y en LL_y-LN_y bits,
- 10 – concatenar en peso débil Ad_x la cantidad anterior representada en $LL_x - LN_x$ bits, y en peso fuerte LL_y-LN_y bits.

21.- Procesador paralelo según la reivindicación 10, **caracterizado porque** los procesadores están dispuestos según una topología de al menos tres dimensiones, y comprende respectivamente N_x , N_y , N_z , ... procesadores según cada dimensión, la estructura de datos está definida en al menos tres dimensiones en función de X , Y y Z , ..., la dimensión de la estructura según cada una de las dimensiones siendo inferior a un valor predeterminado L_x , L_y , L_z , ... y siendo N_x , N_y , N_z , ..., L_x , L_y , L_z , ..., potencias de 2, siendo LL_x , LL_y , LL_z ... los logaritmos de base 2 de L_x , L_y , L_z , ... los medios de cálculo de Ad incluyen medios para:

- concatenar en una palabra única de $LL_x + LL_y + LL_z + \dots$ bits las diferentes coordenadas del punto utilizado para definir la posición del campo; se obtiene w ,
- 20 – permutar el orden de los bits de w ; siendo w' la palabra obtenida,
- recortar w' en palabras de dimensión LL_x , LL_y , LL_z , ... bits; siendo w_x' , w_y' , w_z' , ... las palabras obtenidas,
- aplicar en cada una de las dimensiones:
 - 25 i. añadir $(N_x - 1 - p_x)$ a w_x' ,
 - ii. suprimir los LN_x últimos bits del resultado; se obtiene Ad_x' ,
 - iii. añadir $(N_y - 1 - p_y)$ a w_y' ,
 - iv. suprimir los LN_y últimos bits del resultado; se obtiene Ad_y' ,
 - v. y así sucesivamente en cada dirección,
- 30 – permutar de nuevo el orden de los bits Ad_x' , Ad_y' , Ad_z' , ...; se obtienen entonces las direcciones de almacenamiento Ad_x , Ad_y , Ad_z en cada una de las direcciones,
- concatenar estas direcciones en una palabra única Ad que representa la dirección concernida por el campo en el seno de cada PE.

22.- Procesador paralelo según una de las reivindicaciones anteriores, **caracterizado porque** el procesador paralelo es un ordenador de tipo SIMD.

- 35

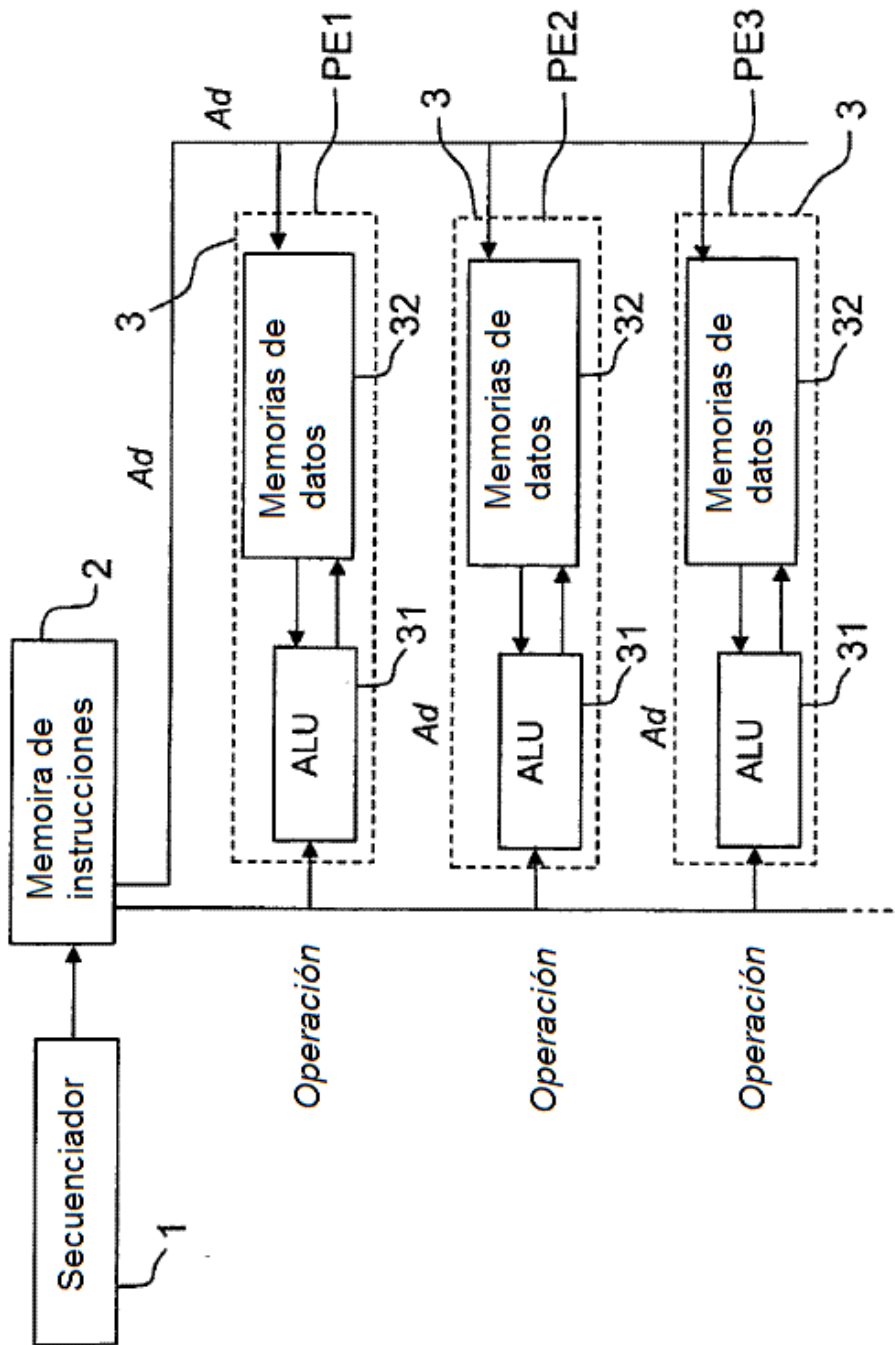


FIG.1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63	
P0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P1	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P2	2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P3	3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P4	4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P5	5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P6	6	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P7	7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P8	8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P9	9	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P10	10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P11	11	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P12	12	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P13	13	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P14	14	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P15	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	63
P0	16	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	...	127
P1	17	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	...	127
P2	18	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	...	127
P3	19	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	...	127
P4	20	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	...	127
P5	21	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	...	127
P6	22	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	...	127
P7	23	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	...	127
...

FIG.2

	... P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P0	P1	P2	P3	P4	P5	P6	P7	P8	... P15	
	... 6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	... 63	
0	... 0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	... 3
1	... 4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	... 7
2	... 8	8	8	8	8	8	8	8	8	8	9	9	9	9	9	9	9	9	9	9	... 11
3	... 12	12	12	12	12	12	12	12	12	12	13	13	13	13	13	13	13	13	13	13	... 15
4	... 16	16	16	16	16	16	16	16	16	16	17	17	17	17	17	17	17	17	17	17	... 19
5	... 20	20	20	20	20	20	20	20	20	20	21	21	21	21	21	21	21	21	21	21	... 23
6	... 24	24	24	24	24	24	24	24	24	24	25	25	25	25	25	25	25	25	25	25	... 27
7	... 28	28	28	28	28	28	28	28	28	28	29	29	29	29	29	29	29	29	29	29	... 31
8	... 32	32	32	32	32	32	32	32	32	32	33	33	33	33	33	33	33	33	33	33	... 35
9	... 36	36	36	36	36	36	36	36	36	36	37	37	37	37	37	37	37	37	37	37	... 39
10	... 40	40	40	40	40	40	40	40	40	40	41	41	41	41	41	41	41	41	41	41	... 43
11	... 44	44	44	44	44	44	44	44	44	44	45	45	45	45	45	45	45	45	45	45	... 47
12	... 48	48	48	48	48	48	48	48	48	48	49	49	49	49	49	49	49	49	49	49	... 51
13	... 52	52	52	52	52	52	52	52	52	52	53	53	53	53	53	53	53	53	53	53	... 55
14	... 56	56	56	56	56	56	56	56	56	56	57	57	57	57	57	57	57	57	57	57	... 59
15	... 60	60	60	60	60	60	60	60	60	60	61	61	61	61	61	61	61	61	61	61	... 63
16	... 64	64	64	64	64	64	64	64	64	64	65	65	65	65	65	65	65	65	65	65	... 67
17	... 68	68	68	68	68	68	68	68	68	68	69	69	69	69	69	69	69	69	69	69	... 71
18	... 72	72	72	72	72	72	72	72	72	72	73	73	73	73	73	73	73	73	73	73	... 75
19	... 76	76	76	76	76	76	76	76	76	76	77	77	77	77	77	77	77	77	77	77	... 79
...

FIG.3

Procesadores concernidos

FIG.4a

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	60	61	62	63
0	P0	P1	P2	P3	P0	P1	P2	P3	P0	P1	P2	P3	P0	P1	P2	P3	...	P0	P1	P2	P3
1	P4	P5	P6	P7	P4	P5	P6	P7	P4	P5	P6	P7	P4	P5	P6	P7	...	P4	P5	P6	P7
2	P8	P9	P10	P11	P8	P9	P10	P11	P8	P9	P10	P11	P8	P9	P10	P11	...	P8	P9	P10	P11
3	P12	P13	P14	P15	P12	P13	P14	P15	P12	P13	P14	P15	P12	P13	P14	P15	...	P12	P13	P14	P15
4	P0	P1	P2	P3	P0	P1	P2	P3	P0	P1	P2	P3	P0	P1	P2	P3	...	P0	P1	P2	P3
5	P4	P5	P6	P7	P4	P5	P6	P7	P4	P5	P6	P7	P4	P5	P6	P7	...	P4	P5	P6	P7
6	P8	P9	P10	P11	P8	P9	P10	P11	P8	P9	P10	P11	P8	P9	P10	P11	...	P8	P9	P10	P11
7	P12	P13	P14	P15	P12	P13	P14	P15	P12	P13	P14	P15	P12	P13	P14	P15	...	P12	P13	P14	P15
8	P0	P1	P2	P3	P0	P1	P2	P3	P0	P1	P2	P3	P0	P1	P2	P3	...	P0	P1	P2	P3
9	P4	P5	P6	P7	P4	P5	P6	P7	P4	P5	P6	P7	P4	P5	P6	P7	...	P4	P5	P6	P7
10	P8	P9	P10	P11	P8	P9	P10	P11	P8	P9	P10	P11	P8	P9	P10	P11	...	P8	P9	P10	P11
11	P12	P13	P14	P15	P12	P13	P14	P15	P12	P13	P14	P15	P12	P13	P14	P15	...	P12	P13	P14	P15

Direcciones en el interior de los procesadores

FIG.4b

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	60	61	62	63
0	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	...	15	15	15	15
1	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	...	15	15	15	15
2	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	...	15	15	15	15
3	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	...	15	15	15	15
4	16	16	16	16	17	17	17	17	18	18	18	18	19	19	19	19	...	31	31	31	31
5	16	16	16	16	17	17	17	17	18	18	18	18	19	19	19	19	...	31	31	31	31
6	16	16	16	16	17	17	17	17	18	18	18	18	19	19	19	19	...	31	31	31	31
7	16	16	16	16	17	17	17	17	18	18	18	18	19	19	19	19	...	31	31	31	31
8	32	32	32	32	33	33	33	33	34	34	34	34	35	35	35	35	...	47	47	47	47
9	32	32	32	32	33	33	33	33	34	34	34	34	35	35	35	35	...	47	47	47	47
10	32	32	32	32	33	33	33	33	34	34	34	34	35	35	35	35	...	47	47	47	47
11	32	32	32	32	33	33	33	33	34	34	34	34	35	35	35	35	...	47	47	47	47

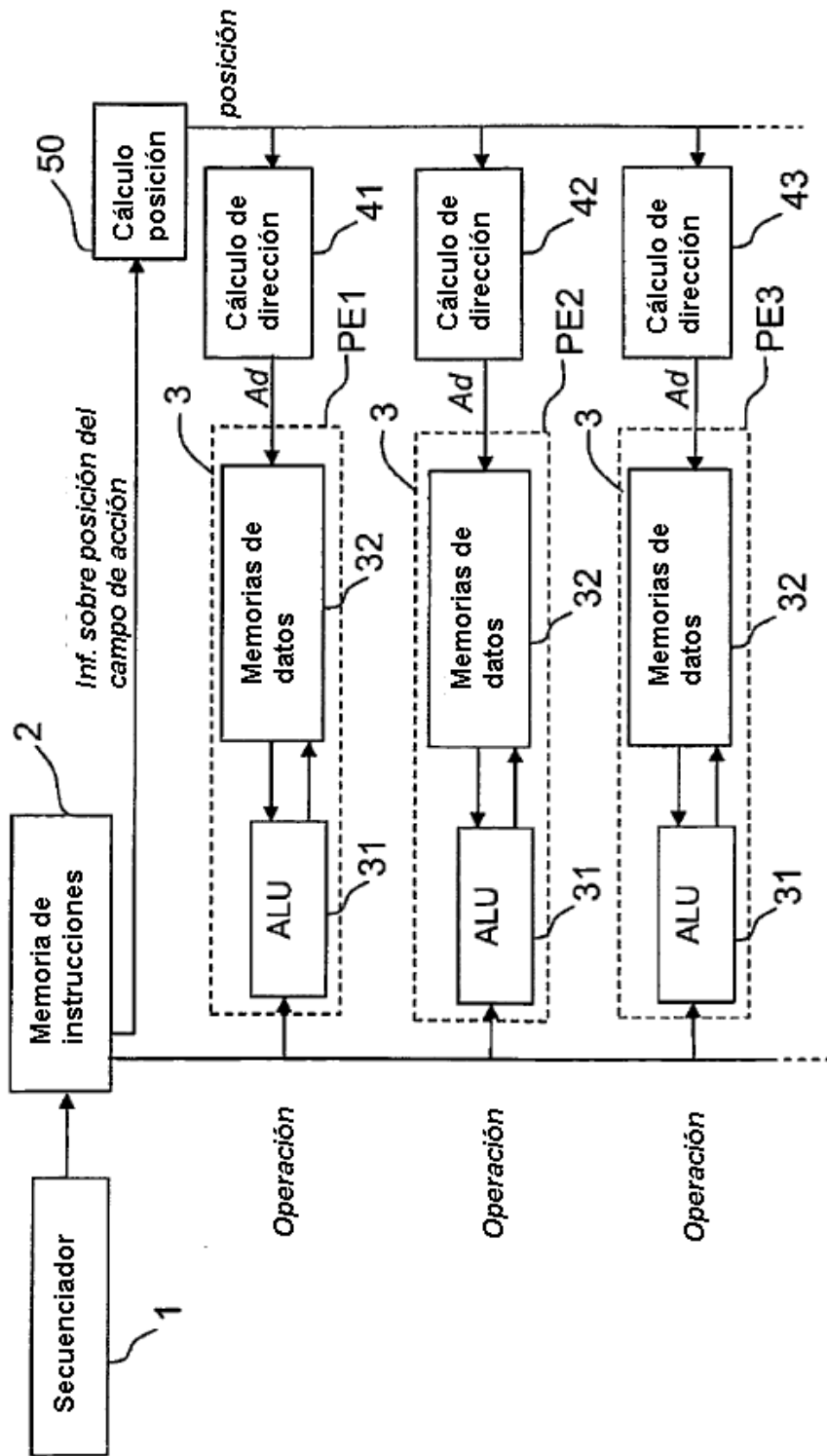


FIG.5

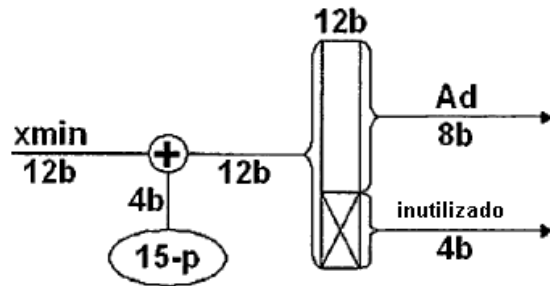


FIG.6a

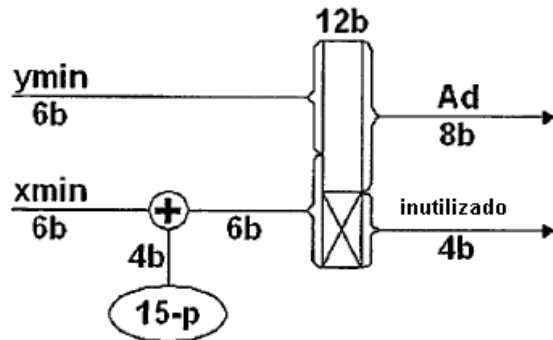


FIG.6b

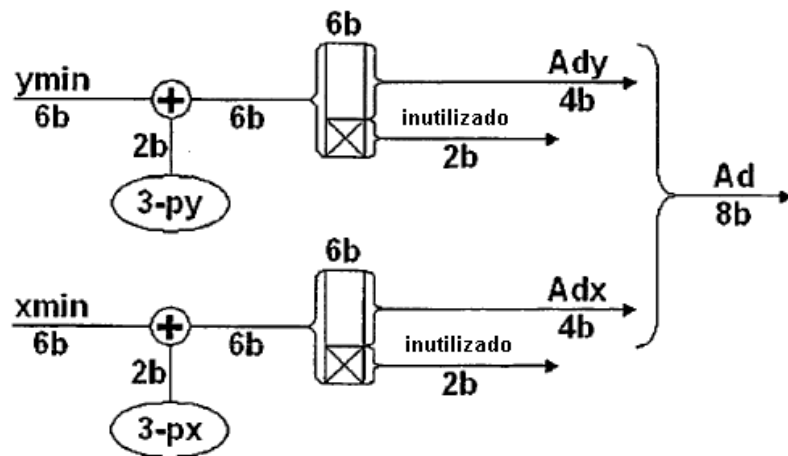


FIG.6c

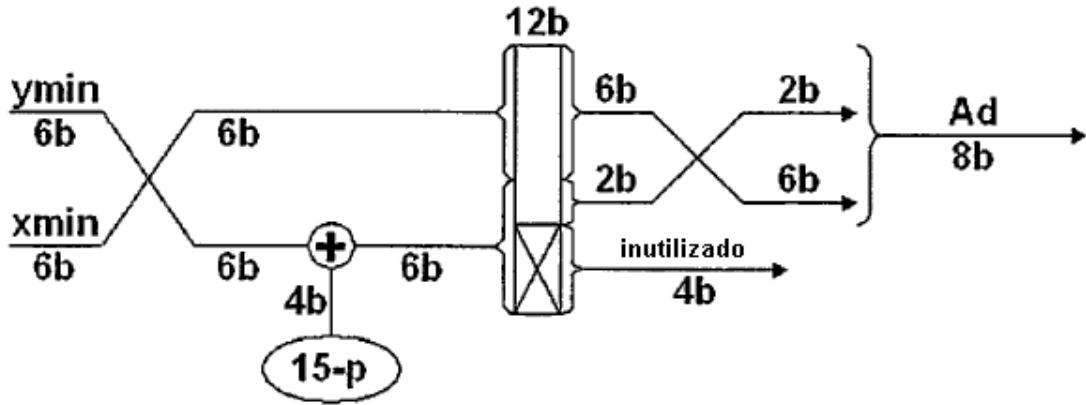


FIG.6d

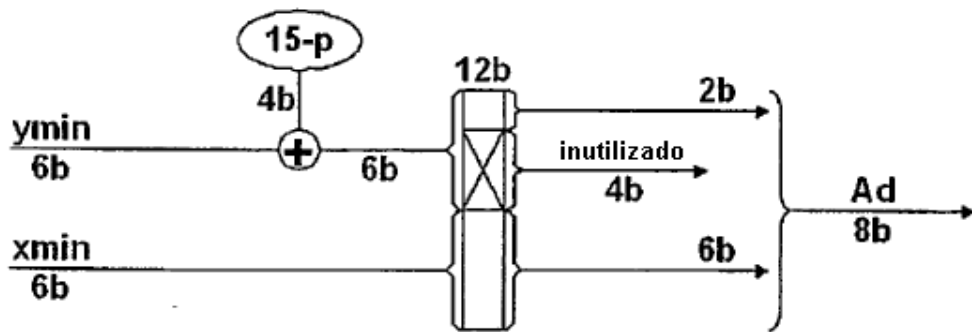


FIG.6e