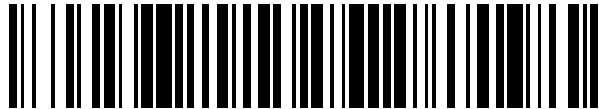


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 458 215**

21 Número de solicitud: 201200608

51 Int. Cl.:

**G06F 17/10** (2006.01)

12

SOLICITUD DE PATENTE

A1

22 Fecha de presentación:

**04.06.2012**

43 Fecha de publicación de la solicitud:

**30.04.2014**

56 Se remite a la solicitud internacional:

**PCT/ES2013/000134**

71 Solicitantes:

**UNIVERSIDAD DE SEVILLA (50.0%)**  
**OTRI-Pabellón de Brasil, Paseo de las Delicias**  
**s/n**  
**41013 Sevilla ES y**  
**CONSEJO SUPERIOR DE INVESTIGACIONES**  
**CIENTÍFICAS (50.0%)**

72 Inventor/es:

**ACOSTA JIMÉNEZ, Antonio José;**  
**BATURONE CASTILLO, María Iluminada;**  
**CASTRO RAMÍREZ, Javier;**  
**JIMÉNEZ FERNÁNDEZ, Carlos Jesús;**  
**BROX JIMÉNEZ, Piedad y**  
**MARTÍNEZ RODRÍGUEZ, Macarena Cristina**

74 Agente/Representante:

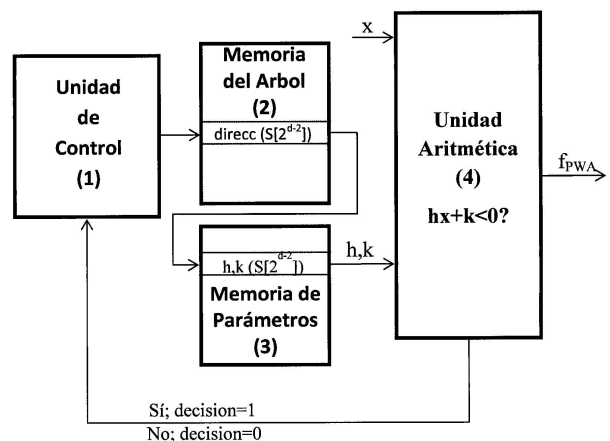
**GONZÁLEZ CARVAJAL, Ramón**

54 Título: **Dispositivo para generar funciones multivariadas afines a tramos con computación on-line del árbol de búsqueda**

57 Resumen:

Método para generar funciones multivariadas afines a tramos, en donde se realiza la computación on-line del árbol de búsqueda para la localización del valor de entrada en los polítopos de la partición, y la posterior generación de la correspondiente función afín. Asimismo se propone un dispositivo configurable y programable para generar funciones multivariadas afines a tramos compuesto por una arquitectura con cuatro bloques funcionales siendo éstos: un bloque unidad de control (1), un bloque de memoria del árbol, un bloque de memoria de parámetros y un bloque de unidad aritmética; y presenta al menos tres modos de operación seleccionables mediante distintos valores de un bus (config): escritura de la memoria del árbol, escritura de la memoria de parámetros y evaluación de la función afín. Puede incluir un cuarto modo de operación, que es el modo de test.

FIG. 2



ES 2 458 215 A1

**DESCRIPCIÓN****DISPOSITIVO PARA GENERAR FUNCIONES MULTIVARIABLES AFINES A TRAMOS CON COMPUTACIÓN ON-LINE DEL ÁRBOL DE BÚSQUEDA**

5 La presente invención se refiere a un dispositivo que permite la generación de funciones multivariables afines a tramos con computación on-line del árbol de búsqueda. Se enmarca en el ámbito de la automática y el control permitiendo implementar procesamiento de señal no lineal y/o controladores para cualquier planta cuya superficie de control sea definible mediante funciones afines a tramos.

10 **ESTADO DE LA TÉCNICA ANTERIOR**

15 La necesidad de implementar funciones afines a tramos surge de forma natural en muchos problemas de ingeniería como son el control adaptativo, el control difuso, la identificación de sistemas no lineales dinámicos o las redes de sensores. La realización de las funciones lineales a tramos mediante circuitos VLSI tiene muchas aplicaciones prácticas, principalmente en el ámbito de sistemas de control empotrados en tiempo real o siempre que no es viable o conveniente recurrir a procesadores, microcontroladores o dispositivos de propósito general, como por ejemplo las placas que incorporan elementos DSP (Digital Signal Processing).

20 Una función afín a tramos (PWA),  $f_{PWA}: D \rightarrow R$ , definida sobre un dominio compacto  $D \subset R^n$ , verifica que:

$$f_{PWA}(x) = f_i^T x + g_i, x \in \Omega_i, i=1, \dots, N_P$$

25 donde  $f_i \in R^n$ ,  $g_i \in R$  y  $\Omega_i$  son  $N_P$  regiones poliédricas, denominadas politopos, que no se solapan y cuya unión da lugar a  $D$ , es decir, inducen una partición poliédrica en el dominio  $D$ ; donde cada politopo es una región cerrada delimitada por hiperplanos ( $h_j^T x + k_j = 0$ , donde  $h_j \in R^n$ ,  $k_j \in R$ ); y donde en todo el dominio se podrán distinguir  $N_E$  hiperplanos que delimitan los politopos, en donde cada hiperplano divide el dominio en dos partes, y la función  $f_{PWA}$  es afín sobre cada politopo,  $\Omega_i$ ; de tal forma que el problema de localización del punto tiene como objetivo encontrar el índice  $i$  tal que  $x \in$   
30  $\Omega_i$  para así proporcionar la función afín en ese punto.

35 Del estado de la técnica se conocen diversos trabajos sobre la implementación electrónica de funciones afines a tramos dependientes de múltiples entradas. Entre los circuitos conocidos, las implementaciones analógicas no son lo suficientemente robustas por lo que en los últimos tiempos se han propuesto implementaciones digitales, orientadas a realizaciones sobre hardware reconfigurable como las FPGAs (Field Programmable Gate Arrays), especialmente interesantes para la realización de prototipos, u orientadas a realizaciones de circuitos integrados (ICs) dedicados, también conocidos como ASICs (Application Specific Integrated Circuits). Por lo  
40 general, aunque las implementaciones sobre FPGAs son factibles, las soluciones ASIC sobre tecnologías VLSI (Very Large Scale Integration) proporcionan mejor rendimiento y son más adecuadas para el diseño de sistemas empotrados.

45 La implementación electrónica de funciones PWA así como otras funciones no lineales para controladores difusos ya se conoce del estado de la técnica. El problema de realizar funciones multivariables PWA se resuelve en el estado de la técnica [1 y 2] empleando un dispositivo DSP comercial. Otros documentos del estado de la técnica

analizan cómo incluir controladores difusos no lineales (en particular PWA) en aplicaciones empotradas. A este respecto el documento [3] describe cómo implementar el controlador difuso como un módulo a medida IP que acelera las tareas de inferencia de un procesador de propósito general. La publicación en [4] describe una metodología de diseño basada en herramientas de CAD que permiten el rápido diseño de controladores difusos complejos que son implementados en un DSP de punto fijo. Los dos ASICs digitales descritos en [5] implementan funciones no lineales típicas del control difuso. Por último, la solución FPGA descrita en [6] también está orientada a la implementación de un controlador difuso.

Asimismo se han estudiado distintas formas canónicas para la implementación de funciones PWA multivariantes. Las representaciones canónicas realizan una descripción de una función PWA utilizando el menor número de parámetros posibles. La elección de una determinada forma canónica es fundamental, sobre todo en aplicaciones donde se trabaja con funciones definidas sobre un dominio de muchas dimensiones o definidas sobre un dominio de pocas dimensiones pero muy particionado.

La forma canónica denominada PieceWise-Affine Simplicial (PWAS) realiza una partición del dominio de entrada en regiones denominadas 'simplices'. En el estado de la técnica se propone también una arquitectura para la realización de circuitos integrados de señal mixta de funciones PWAS. Se conoce un ASIC digital que implementa controladores PWAS, y se conoce un diseño de un circuito integrado mixto que implementa funciones PWAS de tres entradas en una tecnología CMOS 0.5  $\mu\text{m}$ . Más recientemente, varias arquitecturas han sido presentadas para la realización de circuitos digitales PWAS. Entre ellas cabe destacar, la arquitectura propuesta en [7] de la que se describen dos versiones (una paralela y otra serie) implementadas sobre FPGAs. Las implementaciones PWAS presentan dos importantes limitaciones. La primera de ellas es la denominada "maldición de la dimensionalidad", que consiste en que el número de parámetros necesarios para definir las crece exponencialmente con el número de dimensiones del dominio. La segunda limitación de la forma canónica PWAS es que no es capaz de generar cualquier tipo de función PWA sino sólo un subconjunto de ellas. Las realizaciones microelectrónicas reportadas reducen aún más este subconjunto porque emplean un número máximo de particiones por dimensión que no es muy elevado.

Otra forma canónica es la basada en la representación 'lattice' descrita en [8], que ha sido implementada sobre FPGAs utilizando un flujo de diseño basado en las herramientas ISE y System Generator de Xilinx. Una limitación de esta forma canónica es que tan solo permite la realización de funciones afines continuas.

Otra solución consiste en realizar la implementación de una arquitectura jerárquica, donde la función PWA multivariable se descompone en módulos, que implementan funciones PWA uni-dimensionales, conectados en cascada. Esta arquitectura ha sido explorada para distintos casos de estudio, implementándose sobre FPGAs en [9]. Como ocurre con la forma canónica PWAS, esta solución no es capaz de generar cualquier tipo de función PWA sino sólo un subconjunto de ellas.

La única representación canónica que permite implementar cualquier función PWA es la denominada PWA genérica (PWAG). Solo existen dos circuitos digitales propuestos en la literatura que implementan funciones PWAG [10, 11]. Ambos circuitos emplean

arquitecturas destinadas a explorar un árbol de búsqueda binario para resolver el problema de localización del punto. El árbol se construye para minimizar su profundidad (máxima distancia entre la raíz y las hojas del árbol) y para obtener la mayor simetría posible, mediante el procedimiento descrito en [12]. En [10] no se propone una arquitectura específica para implementar funciones PWAG, sino que se realiza una síntesis hardware directa a partir de una descripción en alto nivel (lenguaje C) de la función PWAG que emplea el algoritmo basado en el árbol de búsqueda binario. La síntesis hardware emplea la herramienta PICO-NPA. Cualquier modificación de la función PWA que implique un cambio en el árbol de búsqueda binario, requiere repetir todo el procedimiento de síntesis para obtener un circuito digital nuevo. En [11], se recorre el árbol de búsqueda binario utilizando una máquina de estados finitos. Del mismo modo que ocurre en [10], una modificación del árbol implica una nueva descripción HDL (Hardware Description Language) que debe programarse nuevamente en la FPGA. Esto significa que la FPGA debe extraerse de su contexto de operación para ser de nuevo configurada o que su entorno de operación debe complicarse considerablemente para poder ser reconfigurada 'in situ'.

#### Referencias:

- [1] R. Rovatti, A. Ferrari, M. Borgatti, "Automatic implementation of piecewise-linear fuzzy systems addressing memory-performance tradeoff," in *Fuzzy hardware: architectures and applications*, A. Kandel and G. Langholz, Eds. Norwell, MA, USA: Kluwer Academic Publishers, 1998, pp. 159–179.
- [2] R. Rovatti, C. Fantuzzi, S. Simani, "High-speed DSP-based implementation of piecewise-affine and piecewise-quadratic fuzzy systems," *Signal Processing*, Vol. 80, pp. 951–963, 2000.
- [3] S. Sánchez-Solano, A. Cabrera, I. Baturone, F. J. Moreno-Velo, M. Brox, "FPGA implementation of embedded fuzzy controllers for robotic applications", *IEEE Trans. on Industrial Electronics*, Vol. 54 (4), pp. 1937- 1945, Aug. 2007,
- [4] I. Baturone, F. J. Moreno-Velo, V. Blanco, J. Ferruz, "Design of embedded DSP-based fuzzy controllers for autonomous robots", *IEEE Trans. on Industrial Electronics*, Vol. 55, pp. 928-936, Feb. 2008,
- [5] I. Baturone, A. Barriga, S. Sánchez Solano, C.J. Jiménez Fernández, D.R. López, "Microelectronic design of fuzzy logic-based systems", CRC Press, March 2000, ISBN: 0849300916.
- [6] P. Echevarría, M. Martínez, J. Echanobe, I. del Campo, J. Tarela, "Digital hardware implementation of high dimensional fuzzy systems," in *Applications of Fuzzy Sets Theory*, ser. Lecture Notes in Computer Science. Berlin: Springer, 2007, pp. 245–252.
- [7] M. Storace, T. Poggi, "Digital architectures realizing piecewise-linear multi-variate functions: two FPGA implementations", *Int. Journal of Circuit Theory and Applications*, Vol. 39 (1), pp. 1-15, 2011.

- [8] M. C. Martínez-Rodríguez, I. Baturone, P. Brox, "Design Methodology for FPGA Implementation of Lattice Piecewise-Affine Functions", in Proc. 2011 Int. Conf. on Field-Programmable Technology (FPT), New Delhi (India), 12-14 Dec., 2011, pp. 1-4.
- 5 [9] M. C. Martínez-Rodríguez, I. Baturone, P. Brox, "Circuit Implementation of Piecewise-Affine Functions Based on Lattice Representation", in Proceedings of the 2011 European Conference on Circuit Theory and Design (ECCTD), Linköping, Sweden, 29-31 August, 2011, pp. 644-647.
- 10 [10] I. Baturone, M. C. Martínez-Rodríguez, P. Brox, A. Gersnoviez, S. Sánchez-Solano "Digital implementation of hierarchical piecewise-affine controllers", in Proc. of the 21th IEEE International Symposium on Industrial Electronics (ISIE), Gdańsk, Poland, 27-30 June, 2011, pp. 1497-1502.
- 15 [11] T. Johansen, W. Jackson, R. Schreiber, and P. Tøndel, "Hardware synthesis of explicit model predictive controllers," IEEE Trans. on Control Systems Technology, Vol. 15 (1), pp. 191–197, Jan. 2007.
- [12] A. Oliveri, T. Poggi, M. Storace, "Circuit implementation of piecewise-affine functions based on a binary search tree", in Proc. European Conference on Circuit Theory and Design (ECCTD), Antalya (Turkey), 23-27 Aug., 2009, pp.145-148.
- 20 [13] P. Tøndel, T. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," Automatica, Vol. 39 (5), pp. 945–950, May 2003.

25

#### EXPLICACIÓN DE LA INVENCION

La presente invención propone un dispositivo para generar funciones multivariadas afines a tramos, en donde se realiza la computación on-line del árbol de búsqueda para la localización del valor de entrada en los polítopos de la partición, y la posterior generación de la correspondiente función afín.

30

El dispositivo para generar funciones multivariadas afines a tramos de la presente invención comprende una arquitectura para circuitos digitales que implementa funciones PWA en su forma genérica (PWAG). La arquitectura permite implementar cualquier función PWA, incluidas aquellas que no pueden ser implementadas por las soluciones dirigidas a formas PWAS, "lattice" o jerárquicas. La arquitectura de la presente invención permite configurabilidad y programabilidad 'in situ', siendo posible modificar el árbol de búsqueda y la función PWA sin necesidad de generar un nuevo circuito. Este procedimiento permite que con un único circuito digital puedan implementarse distintas funciones PWA. Por lo tanto, se pueden abarcar distintas aplicaciones industriales configurando y programando adecuadamente el circuito.

35

40

La invención hace referencia a la implementación electrónica VLSI de funciones afines a tramos (PWA) en su solución genérica (PWAG). Estos circuitos son de inmediata aplicación en aplicaciones de control que puedan resolverse mediante control predictivo basado en modelo (MPC, Model Predictive Control). En las soluciones existentes en el estado de la técnica del tipo PWAG, el problema de localización del

45

punto dentro del politopo, se resuelve mediante la construcción del árbol de búsqueda binario de forma externa al circuito (off-line). En la presente invención, la construcción del árbol de búsqueda binario se realiza internamente, es decir on-line. Para ello se emplea un esquema basado en dos memorias digitales. La primera memoria (memoria del árbol) almacena los posibles estados del árbol, mientras que la segunda memoria (memoria de parámetros) almacena los parámetros que sirven para computar la función mediante una unidad aritmética.

La memoria de parámetros contiene los coeficientes y los offsets que emplea la unidad aritmética para computar las funciones afines asociadas a los politopos y a los hiperplanos de la partición poliédrica. La memoria del árbol almacena las direcciones de la memoria de parámetros.

Las principales ventajas de la presente invención son la programabilidad, ya que el usuario puede programar distintas particiones con distintas funciones afines, y la configurabilidad, ya que puede emplearse el mismo circuito para resolver problemas que requieran diferentes números de entradas o árboles de búsqueda con distinta profundidad y estructura. Al ser directamente integrable en un Circuito Integrado de Aplicaciones Específicas (ASIC) sus requisitos de área, prestaciones temporales y de consumo mejoran en al menos un orden de magnitud las soluciones programables off-line. En tal sentido se puede considerar un generador universal PWA de altas prestaciones.

#### BREVE DESCRIPCIÓN DE LOS DIBUJOS

25

FIG. 1 muestra el árbol de búsqueda que tiene una profundidad  $d$ .

FIG. 2 muestra el esquema de recorrido del árbol binario con el dispositivo, mostrando la arquitectura del dispositivo.

30

FIG. 3 muestra el diagrama de bloques de la arquitectura de la invención.

FIG. 4 muestra el esquema a nivel de bloques del ASIC que se ha diseñado como prueba del dispositivo.

35

#### EXPOSICIÓN DETALLADA DE MODOS DE REALIZACIÓN

La presente invención describe un dispositivo para generar funciones multivariadas afines a tramos, compuesto por una arquitectura con cuatro bloques funcionales siendo éstos:

40

- un bloque unidad de control (1)
- un bloque de memoria del árbol (2)
- un bloque de memoria de parámetros (3) y
- un bloque de unidad aritmética (4)

que presenta al menos tres modos de operación seleccionables mediante distintos valores de un bus (*config*): escritura de la memoria del árbol (2), escritura de la memoria de parámetros (3) y evaluación de la función afín.

- 5 En el dispositivo se realiza la computación on-line del árbol de búsqueda para la localización del valor de entrada en los politopos de la partición, y la posterior generación de la correspondiente función afín; y donde una función afín a tramos (PWA),  $f_{PWA}: D \rightarrow R$ , definida sobre un dominio compacto  $D \subset R^n$ , verifica que:

$$f_{PWA}(x) = f_i^T x + g_i, x \in \Omega_i, i=1, \dots, N_P \quad (1)$$

- 10 donde  $f_i \in R^n$ ,  $g_i \in R$  y  $\Omega_i$  son  $N_P$  regiones poliédricas, denominadas politopos, que no se solapan y cuya unión da lugar al dominio D, es decir, inducen una partición poliédrica en el dominio D; donde cada politopo es una región cerrada delimitada por hiperplanos ( $h_j^T x + k_j = 0$ , donde  $h_j \in R^n$ ,  $k_j \in R$ ); y donde en todo el dominio se podrán distinguir  $N_E$  hiperplanos que delimitan los politopos, en donde cada
- 15 hiperplano divide el dominio en dos partes, y la función  $f_{PWA}$  es afín sobre cada politopo,  $\Omega_i$ ; de tal forma que el problema de localización del punto tiene como objetivo encontrar el índice  $i$  tal que  $x \in \Omega_i$  para así proporcionar la función afín en ese punto; y donde la resolución de este problema parte de la construcción de un árbol de búsqueda y posteriormente su exploración. El procedimiento que se implementa en el
- 20 dispositivo objeto de la invención se caracteriza porque el árbol de búsqueda binario lo constituyen una única raíz, que es un primer estado S[1] que se bifurca en dos estados, que son un segundo estado S[2] y un tercer estado S[3] que a su vez y sucesivamente se van bifurcando en dos estados, tal que para una profundidad del árbol  $d$ , el número de estados es  $2^d - 1$ , siendo los  $2^{d-1}$  estados del nivel más profundo las hojas del árbol; y donde el problema de localización del valor de entrada en los
- 25 politopos equivale a la evaluación iterativa de funciones afines cuyos parámetros ( $h_j$ ,  $k_j$ ) son valores reales; y donde es posible localizar el politopo que contiene el vector de entrada realizando sucesivamente las comparaciones ( $h_j^T x + k_j < 0$ ); y donde la exploración del árbol de búsqueda on-line se realiza de tal forma que dado un vector  $x$ ,
- 30 se comienza desde el primer estado S[1] chequeando si  $h_1^T x + k_1 < 0$ : si es cierto, se selecciona la rama que conduce al segundo estado S[2], si no es cierto, se selecciona la que conduce al tercer estado S[3], repitiéndose el proceso hasta que se alcanza una hoja. En la figura 1 se observa el árbol binario. En la figura 2 está representada la exploración del árbol de búsqueda on-line mediante el dispositivo de la invención (ilustrando la exploración por el estado S[2<sup>d-2</sup>]). Esta exploración se realiza desde la
- 35 raíz a las hojas.

En la figura 3 se aprecia el diagrama de bloques de la arquitectura del dispositivo, los cuales ejecutan las siguientes operaciones:

- 40 - el bloque unidad de control (1) conduce la operación del sistema en todos los modos de operación, controlando la adquisición de la entrada de datos al circuito, generando las señales de habilitación de los demás bloques (escritura o lectura en las memorias, a través de señales  $w/rMA$  y  $w/rMP$ ), el bus de datos  $x$  a la unidad aritmética (4) y la señal de validación de salida correcta (*valid\_out*).

45

- el bloque de memoria del árbol (2) almacena el árbol de búsqueda binario y los valores que configuran el número de entradas y la profundidad del árbol. En una realización de la invención el bloque memoria de árbol emplea una memoria RAM. El usuario puede rediseñar el árbol reprogramando la memoria, permitiendo la
- 5 computación de la función PWA on-line. Este bloque de memoria del árbol (2) contiene  $2^{d_{\max}}$  palabras, por lo que permite programar árboles de profundidad máxima  $d_{\max}$ , donde el número de posibles estados es  $2^{d_{\max}}-1$ . De las  $2^{d_{\max}}$  palabras,  $2^{d_{\max}}-1$  son las direcciones de la memoria de parámetros (3) y la longitud de cada palabra es  $\log_2(N_{\max})$ , siendo  $N_{\max}$  el número máximo de politopos e hiperplanos que pueden
- 10 programarse para conformar una partición. La palabra restante de la memoria que no almacena direcciones de la memoria de parámetros, se usa para almacenar los bits que configuran el número de entradas y la profundidad del árbol de la aplicación concreta en la que se vaya a emplear el dispositivo de la invención.
- el bloque de memoria de parámetros (3) almacena los parámetros  $f$ ,  $g$ ,  $h$  y  $k$  necesarios para computar las funciones afines asociadas a los hiperplanos de la partición y a los politopos. Todos los parámetros que se necesitan para la computación de una función afín o de un hiperplano se deben almacenar en la misma dirección de memoria, pudiéndose conseguir toda la información con un único acceso. Este bloque
- 15 tiene tantas palabras como número máximo  $N_{\max}$  de politopos ( $N_p$ ) más hiperplanos ( $N_E$ ) describen la función PWA. La longitud de cada palabra del bloque de memoria de parámetros (3) es la suma de las longitudes de los parámetros necesarios para computar la expresión afín, tal que en un caso  $n$ -dimensional con parámetros de  $m_i$  bits la longitud de palabra es  $m_1 + m_2 + \dots + m_n + m_{n+1}$  bits.
- el bloque unidad aritmética (4) realiza la computación de la función afín como la que
- 20 aparece en la ecuación (1), a través de un conjunto de multiplicadores y sumadores. Este bloque puede operar con una arquitectura en serie o en paralelo.
- 25

El dispositivo objeto de la presente invención posee una arquitectura de doble memoria y es de inmediata realización en un circuito integrado de aplicaciones

30 específicas, de forma que sobrepasa en al menos un orden de magnitud las prestaciones temporales y de potencia de las soluciones existentes basadas en lógica programable. Es de aplicación en el ámbito de la Automática y el Control, ya que es capaz de implementar procesamiento de señal no lineal y/o controladores para cualquier planta cuya superficie de control sea definible mediante funciones afines a

35 tramos. Su configurabilidad y programabilidad permite adecuar la función PWA al problema sin necesidad de rediseñar el circuito.

El dispositivo propuesto presenta al menos tres modos de operación diferentes, siendo éstos: el modo de escritura de memoria del árbol (2), el modo de escritura de memoria

40 de parámetros (3) y el modo de evaluación de la función afín. En una realización preferente incorpora un cuarto modo de operación, que es el modo test.

En el modo de escritura de la memoria del árbol (2) el circuito de control del dispositivo activa las señales específicas de la escritura de datos en la memoria del árbol (2)

45 habilitando la escritura de los estados en dicha memoria del árbol (2), dependiendo el mecanismo de escritura de las señales de entrada de la memoria y dependiendo la temporización de dicha escritura del tipo de memoria empleada.



En el modo de escritura de memoria de parámetros (3) el circuito de control activa las señales específicas de la escritura de datos en la memoria de parámetros (3) habilitando la escritura de los parámetros  $f$ ,  $g$ ,  $h$  y  $k$  correspondientes a la partición y distintas funciones afines de cada politopo, dependiendo el mecanismo de escritura de las señales de entrada de la memoria de parámetros (3) y dependiendo la temporización de dicha escritura del tipo de memoria empleada.

En el modo de operación de evaluación de la función afín el circuito de control del dispositivo activa las señales específicas de la lectura de datos en el bloque de memoria del árbol (2) y en el bloque de memoria de parámetros (3). Se realiza la exploración del árbol para la localización del valor de la entrada dependiendo el mecanismo de lectura de las señales de entrada de la memoria de parámetros (3) y dependiendo la temporización de dicha lectura del tipo de memoria empleada. El circuito de control determina a partir del estado actual, (por ejemplo  $S[2^{d-2}]$ , como se aprecia en el ejemplo de la figura 2, la posición correspondiente en el bloque de memoria del árbol (2), que contiene la dirección de la memoria de parámetros (3) que almacena los valores  $(h, k)$  correspondientes a la evaluación de ese hiperplano. A su vez, la unidad aritmética (4) realiza la operación conducente a evaluar si  $h^T x + k < 0$ , tal que la señal (*decision*) se hace 1 y la unidad de control (1) selecciona una de las ramas del árbol binario (la hoja  $S[2^{d-1}]$ , en la figura 2). Si la evaluación no es cierta, (*decision*) se hace 0 y se selecciona la otra rama del árbol binario (la hoja  $S[2^{d-1}+1]$  en el ejemplo de la figura 2); la información sobre el estado próximo que se alcanza se comunica al bloque de unidad de control (1) a través de esta señal (*decision*); al ser conocida y programable por el usuario la profundidad del árbol, el bloque de unidad de control (1) reconoce cuándo se ha alcanzado una hoja.

En una realización preferente se trabaja con números con signo, de forma que el bit de signo se puede utilizar directamente como la señal (*decisión*) que controla por cuál de las ramas del árbol hay que seguir la búsqueda.

En una realización preferente de la invención, el dispositivo permite un modo de operación de test. En este caso el bloque unidad de control (1) del dispositivo activa las señales necesarias para almacenar en un registro de desplazamiento las direcciones y contenidos de las memorias, extrayéndose mediante un protocolo adecuado y permitiéndose la verificación del correcto funcionamiento del dispositivo o en caso de fallo permitiéndose detectar qué bloque o bloques son los causantes del fallo.

El dispositivo de la invención es a la vez programable y configurable. La configurabilidad permite cambiar la profundidad del árbol de búsqueda y el número de entradas, siendo capaz de trabajar con diferentes números de dimensiones y tamaños de memoria que determinan la complejidad y tamaño del árbol de búsqueda binario-y números de parámetros. El paralelismo o la serialidad de las soluciones dependen de la forma en que se implemente la unidad de control (1) y la unidad aritmética (4), siendo posible cualquier grado de paralelismo sin modificar la arquitectura. La programabilidad permite, una vez seleccionada la configuración deseada, trabajar con diferentes funciones PWA, reprogramando los contenidos de las memorias. Por tanto,

un dispositivo incorporando esta arquitectura puede ser usado para distintas aplicaciones simplemente reconfigurándolo y reprogramándolo.

Ejemplo 1:

5

La arquitectura descrita en la presente invención ha sido incorporada en un ASIC, con el fin de mostrar sus características y prestaciones. Se muestra en la figura 4. La implementación trabaja hasta con 4 dimensiones, pero puede ser configurado para trabajar con 1, 2 y 3 dimensiones y emplea una unidad aritmética (4) de punto fijo con arquitectura paralela. El ASIC opera bajo la acción de las señales (*reset*), (*clk*) y (*datain*). La señal (*config*) configura los bloques para fijar cuatro modos de operación: evaluación de la función afín, escritura de la memoria del árbol (2), escritura de la memoria de parámetros (3) y modo de test. Para escribir las memorias, la unidad de control (1) selecciona los parámetros de las memorias que permiten su escritura, mediante un protocolo de acceso que emplea las señales (*reset*), (*clk*) y (*datain*), completándose con la señal (*ready*), lo que habilita un proceso de escritura automático. En el modo test, los contenidos de las memorias e información sensible se almacenan temporalmente en un registro y se extraen serialmente a través de la señal (*test*).

20

En el modo de operación de evaluación de la función afín, la unidad de control (1) habilita el cálculo de las funciones afines, una vez que las memorias están correctamente programadas. La señal (*reset*) se usa para inicializar la operación, (*clk*) sincroniza las operaciones y (*datain*) representa el bus de señales de entrada (x) que se valida con la señal (*ready*). Un protocolo local se emplea para cargar el valor correcto de x para su computación, usando la señal (*valid\_in*) para habilitar la introducción de nuevos datos válidos de entrada. Después de que (*reset*) se hace inactivo, se adquiere una de las n coordenadas del punto de entrada por la Unidad de control (1) en cada flanco activo de (*clk*) y se almacena en un registro de acuerdo a ese protocolo local. Cuando todas las coordenadas se han leído, la Unidad de control (1) direcciona la Memoria del árbol (2), que a su vez direcciona la Memoria de parámetros (3), habilitando la Unidad aritmética (4) para comenzar a procesar la búsqueda del punto de forma iterativa. El dato de salida es válido cuando la señal (*valid\_out*) está activa. El tiempo que se necesita para calcular el valor de la función en un punto dado en el modo de operación normal es  $TCLK (4+2d)$  donde TCLK es el periodo de la señal (*clk*) y d es la profundidad que el usuario programa para el árbol de búsqueda binaria ( $d \leq d_{max}$ , siendo  $d_{max}$  la profundidad máxima del árbol). Dado el protocolo de acceso a memoria empleado, el proceso de lectura en ambas memorias se hace en un único ciclo, usando la señal (*clk*) y su complementada como señales de control. Las características del ASIC que se ha empleado en este ejemplo de realización son las siguientes:

45

- Número de entradas: configurable de 1 a 4
- Número de salidas: 1
- Resolución de entradas: 12 bits
- Resolución de salida: 26 bits
- Resolución de parámetros: 12 bits
- Número máximo de hiperplanos más politopos: 4096
- Profundidad del árbol binario: configurable de 1 a 13

- Integrado en una tecnología de 90nm, 9 metales de TSMC (*Taiwan Semiconductor Manufacturing Company*).

**REIVINDICACIONES**

1.- Dispositivo para generar funciones multivariadas afines a tramos, compuesto por una arquitectura con cuatro bloques funcionales siendo éstos:

- un bloque unidad de control (1)
- un bloque de memoria del árbol (2)
- un bloque de memoria de parámetros (3) y
- un bloque de unidad aritmética (4)

que presenta al menos tres modos de operación seleccionables mediante distintos valores de un bus (*config*): escritura de la memoria del árbol (2), escritura de la memoria de parámetros (3) y evaluación de la función afín;

realizándose en el dispositivo la computación on-line del árbol de búsqueda para la localización del valor de entrada en los politopos de la partición, y la posterior generación de la correspondiente función afín; y donde una función afín a tramos (PWA),  $f_{PWA}: D \rightarrow R$ , definida sobre un dominio compacto  $D \subset R^n$ , verifica que:

$$f_{PWA}(x) = f_i^T x + g_i, x \in \Omega_i, i=1, \dots, N_P$$

donde  $f_i \in R^n$ ,  $g_i \in R$  y  $\Omega_i$  son  $N_P$  regiones poliédricas, denominadas politopos, que no se solapan y cuya unión da lugar a  $D$ , es decir, inducen una partición poliédrica en el dominio  $D$ ; donde cada politopo es una región cerrada delimitada por hiperplanos ( $h_j^T x + k_j = 0$ , donde  $h_j \in R^n$ ,  $k_j \in R$ ); y donde en todo el dominio se podrán distinguir  $N_E$  hiperplanos que delimitan los politopos, en donde cada hiperplano divide el dominio en dos partes, y la función  $f_{PWA}$  es afín sobre cada politopo,  $\Omega_i$ , de tal forma que el problema de localización del punto tiene como objetivo encontrar el índice  $i$  tal que  $x \in \Omega_i$  para así proporcionar la función afín en ese punto; y donde la resolución de este problema parte de la construcción de un árbol de búsqueda y posteriormente su exploración,

estando dicho árbol de búsqueda constituido por una única raíz, que es un primer estado (S[1]) que se bifurca en dos estados, que son un segundo estado (S[2]) y un tercer estado (S[3]) que a su vez y sucesivamente se van bifurcando en dos estados, tal que para una profundidad del árbol  $d$ , el número de estados es  $2^d - 1$ , siendo los  $2^{d-1}$  estados del nivel más profundo las hojas del árbol; y donde el problema de localización del valor de entrada en los politopos equivale a la evaluación iterativa de funciones afines cuyos parámetros  $(h_j, k_j)$  son valores reales; y donde es posible localizar el politopo que contiene el vector de entrada realizando sucesivamente las comparaciones ( $h_j^T x + k_j < 0$ ); y donde la exploración del árbol de búsqueda on-line se realiza de tal forma que dado un vector  $x$ , se comienza desde el primer estado (S[1]) chequeando si  $h_1^T x + k_1 < 0$ : si es cierto, se selecciona la rama que conduce al segundo estado (S[2]), si no es cierto, se selecciona la que conduce al tercer estado (S[3]), repitiéndose el proceso hasta que se alcanza un estado asociado a una hoja del árbol,

**estando dicho dispositivo caracterizado porque**

- el bloque unidad de control (1) conduce la operación del sistema en todos los modos de operación, controlando la adquisición de la entrada de datos al circuito, generando

las señales de habilitación de los demás bloques, esto es, la escritura o lectura en las memorias, a través de señales *w/rMA* y *w/rMP*, el bus de datos *x* a la unidad aritmética (4) y la señal de validación de salida correcta (*valid\_out*);

- 5 - el bloque de memoria del árbol (2) almacena el árbol de búsqueda binario y los valores que configuran el número de entradas y la profundidad del árbol;
- el bloque de memoria de parámetros (3) almacena los parámetros *f*, *g*, *h* y *k* necesarios para computar las funciones afines asociadas a los hiperplanos de la partición y a los politopos, en donde todos los parámetros que se necesitan para la computación de una función afín o de un hiperplano se deben almacenar en la misma dirección de memoria, pudiéndose conseguir toda la información con un único acceso; y donde el bloque de memoria de parámetros (3) tiene tantas palabras como número máximo  $N_{max}$  de politopos ( $N_p$ ) más hiperplanos ( $N_E$ ) describen la función PWA; y
- 10 - el bloque unidad aritmética (4) realiza la computación de la función afín.

15

2.- Dispositivo según la reivindicación 1 en donde la longitud de cada palabra del bloque de memoria de parámetros (3) es la suma de las longitudes de los parámetros necesarios para computar la expresión afín, tal que en un caso *n*-dimensional con parámetros de  $m_i$  bits la longitud de palabra es  $m_1 + m_2 + \dots + m_n + m_{n+1}$  bits.

20

3.- Dispositivo según una cualquiera de las reivindicaciones 1 a 2 donde el bloque memoria del árbol emplea una memoria RAM, el usuario puede rediseñar el árbol reprogramando la memoria, permitiendo la computación de la función PWA on-line y donde dicho bloque de memoria contiene  $2^{d_{max}}$  palabras, en el caso de un árbol de profundidad máxima  $d_{max}$ , donde el número de posibles estados es  $2^{d_{max}}-1$ . De dichas  $2^{d_{max}}$  palabras,  $2^{d_{max}}-1$  son las direcciones de la memoria de parámetros (3) y la longitud de cada palabra es  $\log_2(N_{max})$ , siendo  $N_{max}$  el número máximo de politopos e hiperplanos que conforman la partición, y la palabra restante almacena los valores que configuran el número de entradas y la profundidad del árbol.

30

4.- Dispositivo según una cualquiera de las reivindicaciones 1 a 3 en el que, al configurar el modo de operación de escritura de la memoria del árbol (2), el circuito de control del dispositivo activa las señales específicas de la escritura de datos en la memoria del árbol (2) habilitando la escritura de los estados en dicha memoria del árbol (2), dependiendo el mecanismo de escritura de las señales de entrada de la memoria y dependiendo la temporización de dicha escritura del tipo de memoria empleada.

35

5.- Dispositivo según una cualquiera de las reivindicaciones 1 a 4 en el que, al configurar el modo de escritura de la memoria de parámetros (3), el circuito de control del dispositivo activa las señales específicas de la escritura de datos en la memoria de parámetros (3) habilitando la escritura de los parámetros *f*, *g*, *h* y *k* correspondientes a la partición y distintas funciones afines de cada politopo, dependiendo el mecanismo de escritura de las señales de entrada de la memoria de parámetros (3) y dependiendo la temporización de dicha escritura del tipo de memoria empleada.

45

6.- Dispositivo según una cualquiera de las reivindicaciones 1 a 5 en el que el dispositivo permite un cuarto modo de operación, que es el modo test; de forma que al

configurar el modo de operación de test, el circuito de control activa las señales necesarias para almacenar en un registro de desplazamiento las direcciones y contenidos de las memorias, extrayéndose mediante un protocolo adecuado y permite la verificación del correcto funcionamiento del circuito o en caso de fallo permite  
5 detectar qué bloque o bloques son los causantes del fallo.

7.- Dispositivo según una cualquiera de las reivindicaciones 1 a 6 en el que al configurar el modo de operación de evaluación de la función afin, el circuito de control activa las señales específicas de la lectura de datos en el bloque de memoria del árbol  
10 (2) y en el bloque de memoria de parámetros (3), realizándose la exploración del árbol para la localización del valor de la entrada dependiendo el mecanismo de lectura de las señales de entrada y de la memoria de parámetros (3) y dependiendo la temporización de dicha lectura del tipo de memoria empleada; el circuito de control determina a partir del estado actual la posición correspondiente en el bloque de  
15 memoria del árbol (2), que contiene la dirección de la memoria de parámetros (3) que almacena los valores  $(h, k)$  correspondientes a la evaluación de ese hiperplano; la unidad aritmética (4) realiza la operación conducente a evaluar si  $h^T x + k < 0$ , tal que si la evaluación es cierta, la señal (*decisión*) se hace 1 y la unidad de control (1) selecciona la rama (u hoja) del árbol binario que esté a su izquierda en el nivel  
20 inmediatamente inferior, si la evaluación no es cierta, (*decisión*) se hace 0 y se selecciona la rama del árbol binario que esté a su derecha en el nivel inmediatamente inferior; la información sobre el estado próximo que se alcanza se comunica al bloque de unidad de control (1) a través de esta señal (*decisión*); al ser conocida y programable por el usuario la profundidad del árbol, el bloque de unidad de control (1)  
25 reconoce cuándo se ha alcanzado una hoja.

FIG. 1

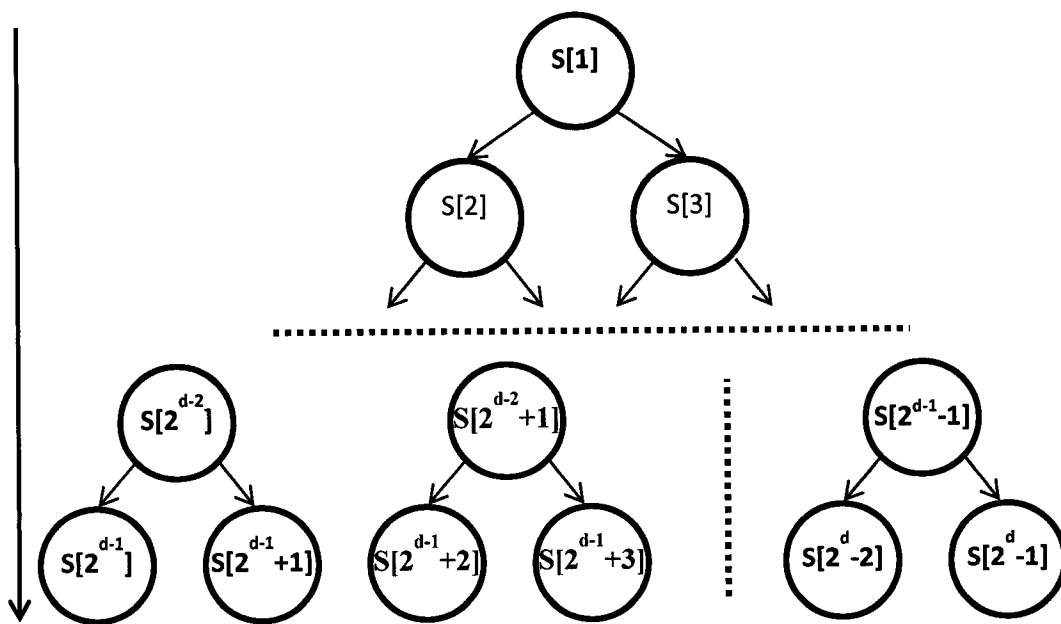
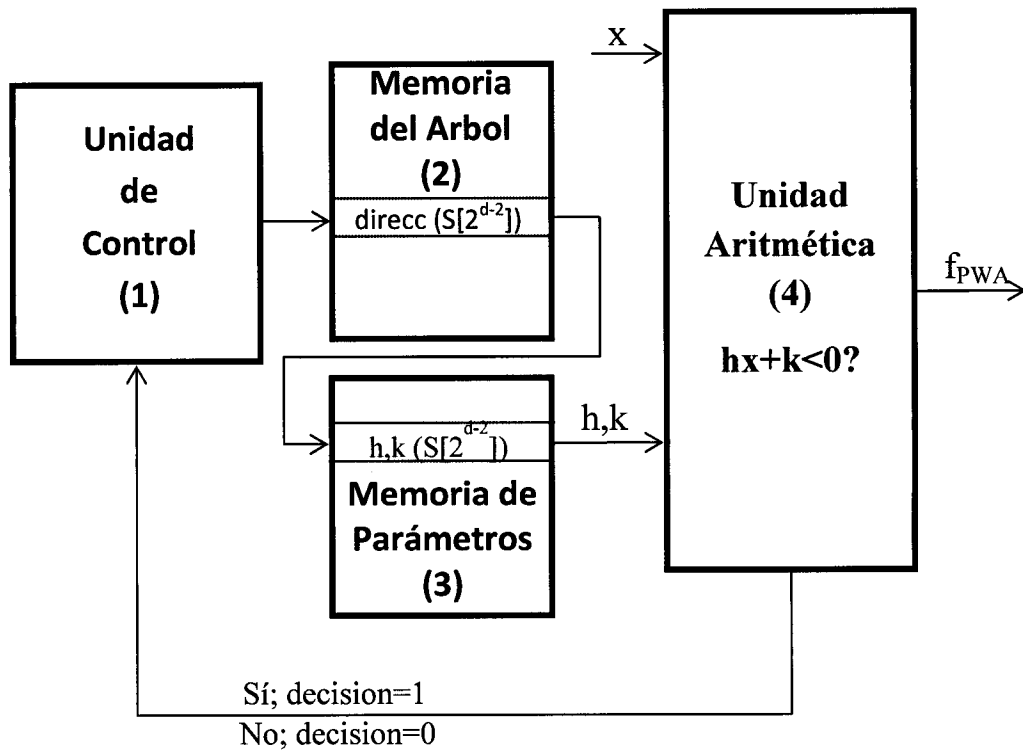


FIG. 2





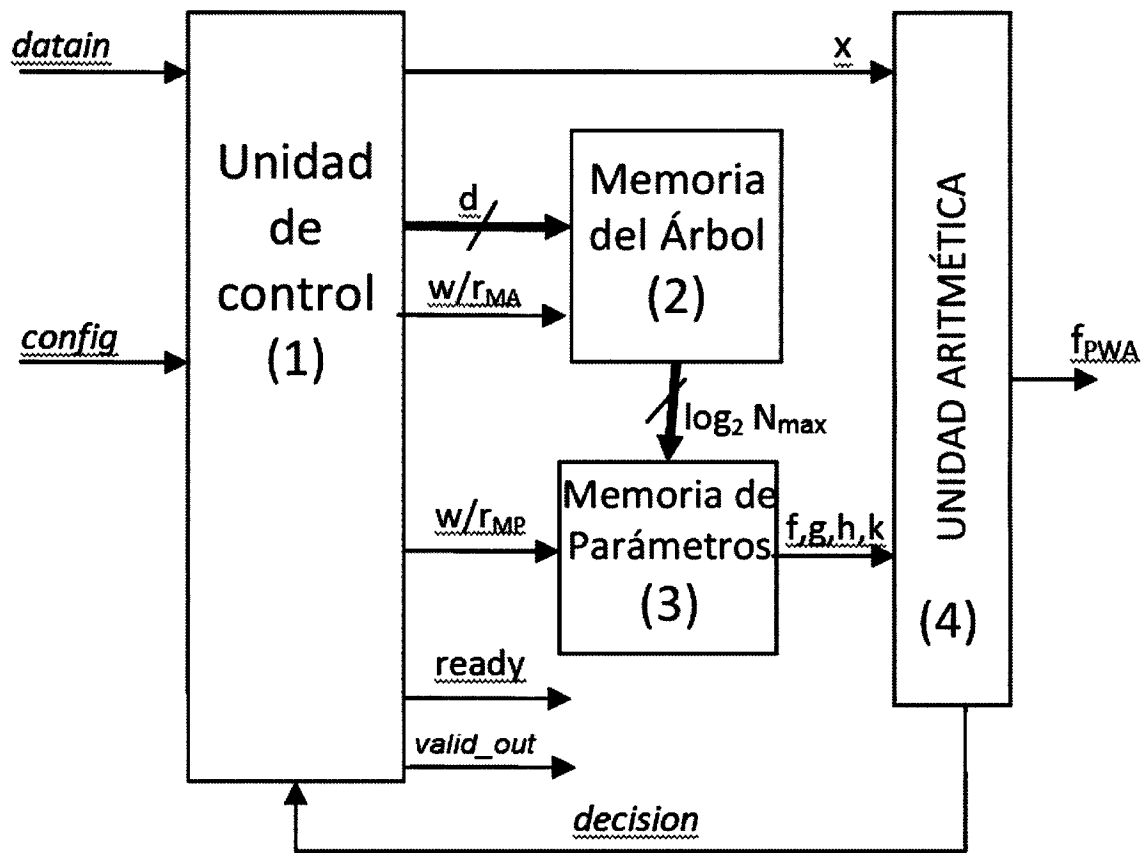


FIG. 3

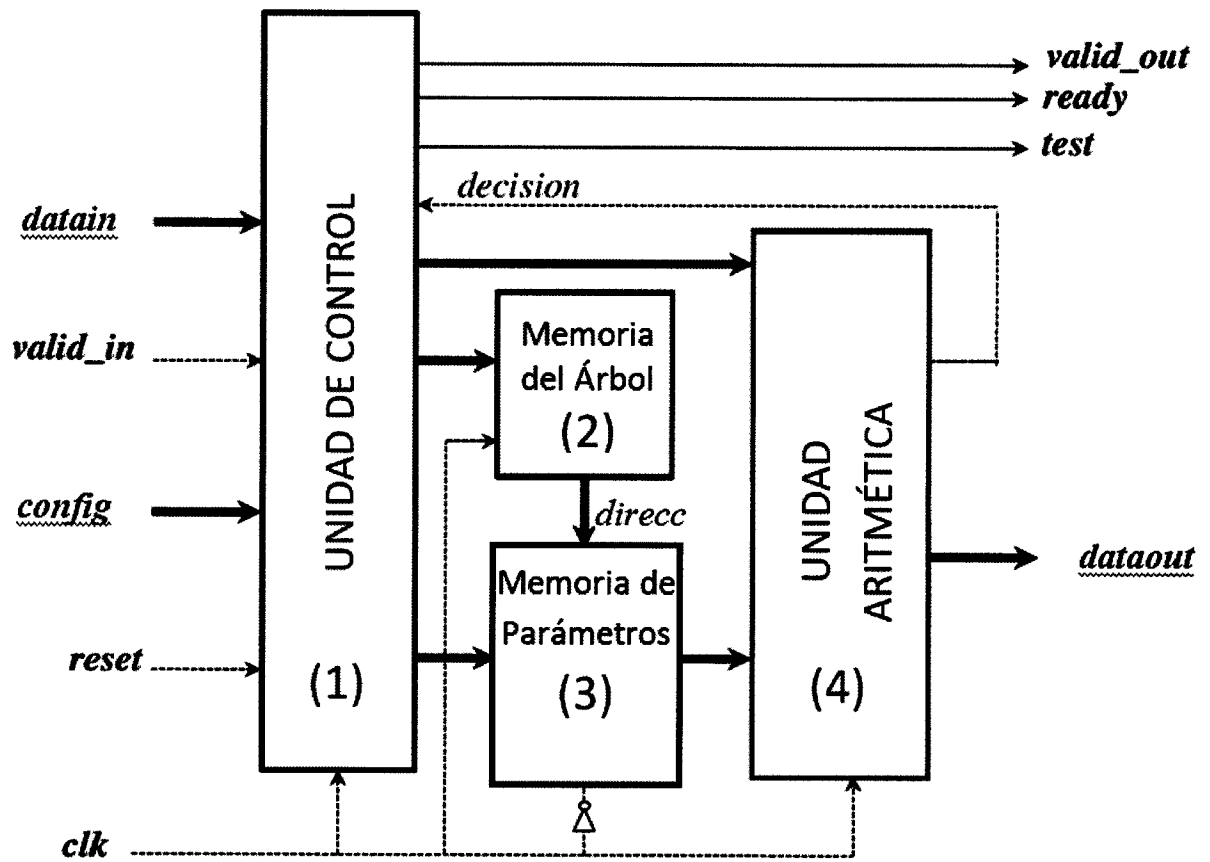


FIG. 4