

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 546 895**

21 Número de solicitud: 201430453

51 Int. Cl.:

G06F 7/38

(2006.01)

12

SOLICITUD DE PATENTE

A1

22 Fecha de presentación:

28.03.2014

43 Fecha de publicación de la solicitud:

29.09.2015

71 Solicitantes:

UNIVERSIDAD DE MÁLAGA (100.0%)
Plaza de El Ejido, s/n
29071 Málaga ES

72 Inventor/es:

HORMIGO AGUILAR, Francisco Javier y
VILLALBA MORENO, Julio

74 Agente/Representante:

ZEА CHECA, Bernabé

54 Título: **Multiplicadores coma flotante y conversores asociados**

57 Resumen:

Dispositivos para realizar una multiplicación de al menos dos números coma flotante pre-procesados y generar un tercer número coma flotante pre-procesado son propuestos. Un formato en coma fija pre-procesado es un formato en coma fija en el que el LSD de todos los números representados exactamente en dicho formato es igual a B/2 (es decir, 1 para base binaria), y el resto son redondeados a uno de estos números. Un formato en coma flotante pre-procesado es un formato en coma flotante en el que la mantisa es un número en coma fija pre-procesado. Para números teniendo una mantisa pre-procesada de m+2 dígitos, el dispositivo comprende un camino de datos del exponente y un camino de datos de la mantisa. El camino de datos de la mantisa comprende una primera entrada para recibir como mucho los m+1 Dígitos Más Significativos (MSDs) de la mantisa pre-procesada del primer número y una segunda entrada para recibir como mucho los m+1 MSDs de la mantisa pre-procesada del segundo número. El camino de datos de la mantisa está configurado para generar como mucho los m+1 MSDs de la mantisa pre-procesada del tercer número.

ES 2 546 895 A1

DESCRIPCIÓN

Multiplicadores coma flotante y conversores asociados

La presente invención se refiere al procesamiento de datos y más
5 concretamente a dispositivos para multiplicar números en coma flotante y los conversores asociados a los mismos.

ESTADO DE LA TÉCNICA

10 En los sistemas de procesamiento de información, la representación de los números se realiza mediante cadenas binarias. Los bits se pueden organizar en dígitos dependiendo del radix o base.

Los números pueden representarse en varios formatos. Los formatos más
15 utilizados son el formato en coma flotante (FP) y el formato de coma fija (FF). En formato de coma fija, el cual incluye los números enteros, el número de dígitos fraccionarios y dígitos enteros es fijo. En esta representación, los números negativos se representan típicamente en formato de complemento, respecto de la base. Por ejemplo para números binarios se utiliza un formato
20 de complemento a dos.

En coma flotante, el número se compone de la mantisa (Ma), la base (B) y el exponente (Ex). Por lo tanto, el valor (Va) representado sería $Va = B * Ma^{Ex}$. Entonces, solamente los números Ma y Ex necesitan almacenarse. El
25 formato estándar IEEE-754 es el más extendido. El estándar define cinco formatos básicos que llevan el nombre de su base numérica y el número de bits usados en su codificación de intercambio. La precisión típica de los formatos binarios básicos es un bit más que la anchura de su mantisa (o mantisa). El bit de precisión extra proviene de un bit a uno implícito (oculto)
30 en la parte más significativa. El número en coma flotante típico estará normalizado tal que el bit más significativo será un uno. Si conocemos que el

bit más significativo es uno, entonces no se necesita codificarlo en el formato de intercambio.

5 Los sistemas para realizar operaciones entre estos números pueden usar una pluralidad de unidades funcionales. Estas unidades pueden realizar transformaciones numéricas como operaciones aritméticas, conversiones de formato, evaluación de funciones, etc. El formato utilizado para representar los números con los que estos circuitos operan define completamente el diseño de estos circuitos y, por tanto, sus parámetros fundamentales de
10 eficiencia tales como precisión, rango, velocidad, área y consumo. En consecuencia, el formato utilizado en estos sistemas influye enormemente en su eficiencia.

15 Dos circuitos básicos que se requieren en la mayoría de tales unidades funcionales son los circuitos de redondeo y los circuitos para complemento a dos.

Los circuitos de redondeo se utilizan cuando es necesario reducir el número de dígitos significativos, tanto en números en formato de coma fija como en la
20 mantisa de números en formato de coma flotante. El circuito que realiza la función de complemento a dos se utiliza para cambiar el signo del número. Cualquier mejora en la eficiencia de estos dos circuitos afecta directamente a la eficiencia de la mayoría de las unidades funcionales que los incluyan.

25 Para realizar el complemento a la base de un número, primero se realiza el complemento a la base menos uno, una operación que se realiza sobre todos los dígitos en paralelo. Posteriormente se le suma al número una unidad-en-el-último lugar (ULP). En el caso binario, para que un circuito que lleva a cabo el complemento a dos de un número de N bits serían necesarios N inversores
30 y un sumador de N bits. En el caso de una operación de resta ($X - Y = X + (-Y)$), que en realidad consiste en una suma con el complemento a dos del sustraendo, el bit de entrada de acarreo del sumador se suele utilizar para

añadir el ULP. Sin embargo, esto no significa que cada vez que se requiere llevar a cabo el complemento a dos el motivo es una resta. Tales casos son la operación de valor absoluto o la suma/resta de números en representación signo-magnitud, una representación típicamente usada en coma flotante.

5

Con respecto a los circuitos de redondeo, se utilizan varias formas de redondeo. Una que demuestra importantes propiedades y es la más utilizada es el "redondeo al par más cercano". En este modo, el valor que se utiliza como valor final es el valor que está más cerca del valor real y, en caso de

10 empate, el valor par. Usando este tipo de redondeo, se obtiene un error inferior a $\pm 0.5\text{ULP}$ y no presenta ningún sesgo en los errores.

Dado un número de $D1$ dígitos, para realizar una operación de redondeo a $D2$ dígitos, asumiendo $D1 > D2$, $D1-D2$ dígitos deben desecharse. Para que el

15 redondeo sea al número más cercano, es importante examinar el valor del dígito más significativo de los que necesitan ser desechados (MD) y el dígito menos significativo de los que quedan (LD):

- Si $MD < (B/2)$ entonces simplemente dichos dígitos son descartados.
- Si $MD > (B/2)$ entonces dichos dígitos se descartan y se añade el valor

20 uno al dígito menos significativo que permanece.

- Si $MD = (B/2)$ entonces se debe verificar si alguno de los dígitos a descartarse no es cero (sticky bit). Si es así, entonces el redondeo se realiza según el segundo caso. Si todos son cero, entonces si el dígito LD es par entonces el redondeo se realiza según el primer caso y si es impar según el

25 segundo caso.

Por lo tanto, el circuito básico para implementar este tipo de redondeo requiere un sumador para sumar uno si es necesario y un circuito para calcular el sticky bit.

30

Los circuitos de complemento a la base y redondeo son necesarios en las unidades funcionales tales como sumadores, multiplicadores, divisores,

unidades FMAD, operadores de valor absoluto, conversores de formato o conversores de precisión etc. El coste adicional, por ejemplo en el área o retardo, que plantean dichos circuitos en las mencionadas unidades funcionales es generalmente substancial, sobre todo porque están
5 típicamente en la vía crítica.

En el estado de la técnica anterior se han hecho varios intentos para reducir los efectos de estos cálculos, es decir el complemento a dos, el cálculo del sticky bit y redondeo. En ciertos documentos del estado de la técnica se ha
10 propuesto pre-calcular el sticky bit o quitar estas operaciones de la vía crítica o reducir el número total de operaciones de redondeo necesarias o combinar redondeo y complemento a dos.

Sería deseable tener circuitos y métodos que reduzcan el coste en área, retardo y consumo de los circuitos de redondeo al más cercano y/o de
15 complemento a la base.

La presente invención se refiere a varios métodos y dispositivos para evitar o al menos reducir parcialmente este problema.

20

RESUMEN

La presente descripción se refiere a configuraciones y circuitos para operaciones en coma flotante que implementan técnicas para codificar números con objeto de realizar funciones de redondeo al más cercano y
25 complemento a la base sin la necesidad de realizar una suma. Por tanto, los sistemas que usen el tipo de codificación propuesto y que requieran estas operaciones podrían simultáneamente reducir área, retardo y consumo de potencia.

30

- Con este fin, la presente descripción se centra en el diseño de sistemas digitales de procesamiento de información más eficientes (más rápidos, menor coste, menor consumo de energía) mediante el uso de una nueva familia de formatos o una modificación de los formatos de codificación numérica, aplicable a la mayoría de los formatos actuales, lo que implica cambios en los circuitos que procesan dichos formatos. Estos formatos simplifican drásticamente los circuitos para el redondeo al más cercano y complemento a la base, sin afectar negativamente al resto del circuito.
- En un primer aspecto, se propone un dispositivo para realizar una multiplicación de al menos dos números en coma flotante pre-procesados y generar un tercer número en coma flotante pre-procesado. Cada número tiene una mantisa de $m+2$ dígitos. El dispositivo comprende un camino de datos del exponente y un camino de datos de la mantisa. El camino de datos de la mantisa comprende una primera entrada para recibir como mucho los $m+1$ Dígitos Más Significativos (MSDs) de la primera mantisa pre-procesada y una segunda entrada para recibir como mucho los $m+1$ MSDs de la segunda mantisa pre-procesada. El camino de datos de la mantisa está configurado para generar como mucho los $m+1$ MSDs de la mantisa del tercer número pre-procesado. El Dígito Menos Significativo (LSD) de todas las mantisas pre-procesadas es igual a $B/2$, siendo B la base del sistema de representación numérica utilizado. En el caso de que el sistema numérico sea binario, entonces $B=2$ y el LSD es igual a uno.
- Una ventaja del dispositivo es la capacidad de realizar las operaciones mencionadas sin usar explícitamente el LSD de la mantisa de los números en coma flotante. Para lograr esto, los números en coma flotante necesitan estar en un formato pre-procesado. El formato propuesto puede derivarse de cualquier formato no procesado, ya sea formato de coma fija o de coma flotante. En el caso de números en coma fija el formato pre-procesado puede obtenerse mediante la adición de un nuevo dígito como el dígito menos significativo (LSD). El valor de dicho dígito (KD) es igual a la base de

representación dividida entre dos. En el caso de números de coma flotante, se lleva a cabo el mismo proceso para la mantisa del número FP.

Por lo tanto, en principio, los números pre-procesados necesitan un dígito más que los no procesados con la misma precisión. Sin embargo, como este dígito KD (o LSD) es una constante, no tiene que ser almacenado ni transmitido de forma explícita. Solamente puede ser requerido representar este dígito en una forma explícita cuando existe la necesidad de realizar operaciones (aritmética, conversiones, o de otro tipo) con esos números. Por lo tanto, el almacenamiento y transmisión de números en formato pre-procesado (implícito) es equivalente al convencional.

Además, el número de valores representados en los dos formatos correspondientes (pre-procesado y no procesado) será el mismo. Sin embargo, los valores representados exactamente en cada formato, será diferente. Por ejemplo, en un formato binario de coma fija con sólo dos bits fraccionarios, cuatro valores son exactamente representables (0, 0.25, 0.5, 0.75), y en el formato pre-procesado correspondiente (es decir, tres bits fraccionarios), también cuatro valores son exactamente representables, pero unos diferentes (0.125, 0.375, 0.625, 0.875). Más específicamente, los valores exactamente representables en formato pre-procesado aparecerán exactamente en el punto intermedio entre la representación numérica exacta de los valores no procesados exactamente representables en el formato no procesado original. Esto significa que la precisión será equivalente en ambos formatos, pero la conversión entre ellos no puede ser exacta.

Un sistema digital que use el formato pre-procesado puede implementarse más eficientemente si el dígito KD está implícito. Dicho dígito KD puede añadirse a la entrada de un circuito de procesamiento o introducirse cuando una operación requiere su presencia. Por otro lado, si el número tiene que incluir explícitamente el dígito KD, por ejemplo para una operación posterior,

entonces el dígito KD puede añadirse a la salida de una operación anterior.

Resumiendo, un formato en coma fija pre-procesado es un formato en coma fija en el que el LSD de todos los números representados exactamente en dicho formato es igual a $B/2$ (es decir, 1 para base binaria), y el resto, son redondeados a uno de estos números. Por tanto, dicho LSB podría ser almacenado, transmitido o incluso operado, implícitamente. Un formato en coma flotante pre-procesado es un formato en coma flotante en el que la mantisa es un número en coma fija pre-procesado.

El uso números en formato pre-procesado simplifica enormemente la operación de redondeo “al más cercano” o “al par más cercano”. Esta es la principal ventaja del uso de este formato. Dado un número en coma fija o la mantisa de un número en coma flotante de $D1$ -dígitos, la operación de redondeo “al más cercano” a un formato pre-procesado de $D2+1$ dígitos siendo $D1$ y $D2$ números naturales tal que $D1 > D2$, se realiza descartando los $D1-D2$ dígitos menos significativos (truncado). En el caso del redondeo “al par más cercano”, antes de operar es necesario comprobar si los $D1-D2$ dígitos menos significativos son todos cero (lo cual suele realizarse, calculando el sticky bit). Si es así, mientras se eliminan los $D1-D2$ dígitos menos significativos, se realizaría el siguiente proceso sobre el siguiente dígito:

- Si el siguiente dígito es par, entonces se quedaría igual.
- Si el siguiente dígito es impar, entonces se le restaría uno a dicho dígito (lo que en ningún caso provocaría acarreo).

El uso de números en formato pre-procesado también simplifica la operación de complemento a la base. Debido al valor específico del LSD, la suma de 1 ULP después de complementar el número a la base menos uno simplemente devuelve el valor del LSD a $B/2$ y no se produce acarreo hacia el resto de los dígitos. Por ejemplo, en formato binario, después de complementar a uno un número binario pre-procesado, el LSB es igual a cero y la suma de un ULP no produce ningún acarreo sino simplemente establece el LSB a uno de

nuevo. Por lo tanto, la implementación del complemento de la base de un número pre-procesado sólo requiere complementar a la base menos uno todos los dígitos menos el LSD que permanece igual.

- 5 Las implementaciones según dicho aspecto tienen la ventaja de que no se necesita lógica para redondear por exceso (o hacia arriba). La eliminación de la lógica para redondear por exceso, que generalmente es un sumador independiente (o incrementador) o un sumador compuesto (sumador que devuelve $X + Y$ y $X + Y + 1$) junto con otra lógica de control se hace posible
- 10 porque el redondeo "al más cercano" para obtener un número pre-procesado se realiza, como se ha explicado antes, simplemente mediante truncado. Además, no hay ninguna necesidad de tener una lógica para calcular el sticky bit. La eliminación de la lógica para el cálculo del sticky bit es posible porque, en el producto de dos números pre-procesados, el sticky bit es
- 15 siempre uno, puesto que el último dígito del resultado siempre es necesariamente $B/2$ (dígito KD). Esto es una ventaja para el redondeo. Por último, otra ventaja es que no puede ocurrir desbordamiento después del redondeo, puesto que éste no se realiza por exceso.

- 20 En las siguientes descripciones de realizaciones se considera generalmente que el formato coma flotante usa mantisas sin signo y un bit de signo independiente, sin embargo, alguien experto en el estado de la técnica, podría aplicar las enseñanzas divulgadas aquí, también para mantisas con signo, de una forma directa.

- 25 En algunas realizaciones, el camino de datos del exponente podría estar configurado para calcular el exponente de salida y el signo de la salida. Además, podría estar configurado para detectar valores especiales de las entradas, como cero, infinito, "no es un número" o números des-normalizados, y dar la orden al multiplicador para producir el resultado
- 30 correspondiente. Además, podría estar configurado para detectar y resolver

excepciones, tales como desbordamiento o desbordamiento hacia cero, y valores especiales, como los anteriores, después de dicha operación.

5 En algunas realizaciones, dichas mantisas pre-procesadas podrían estar normalizadas. Normalización significa que, excepto para el número cero, un número real se representa con un dígito entero con un valor diferente de cero y una parte fraccionaria. En esas realizaciones dichas primera y segunda entradas podrían estar configuradas para recibir los m MSD de la parte fraccionaria de la primera y segunda mantisas pre-procesadas,
10 respectivamente.

En algunas realizaciones, el dispositivo puede comprender además una tercera entrada para recibir el LSD de dichas primera y segunda mantisas pre-procesadas. Alternativamente, la tercera entrada puede tener un valor de
15 $B/2$, ya que el LSD de las mantisas pre-procesadas es igual a $B/2$. Por lo tanto, la mantisa pre-procesada completa será usada en las operaciones siguientes, aunque no sería necesario transmitir la mantisa completa hasta la entrada del dispositivo.

20 Aunque las siguientes descripciones de realizaciones representan circuitos diseñados para lógica binaria, una persona experta en el estado de la técnica podría aplicar también las enseñanzas mostradas aquí, a circuitos no binarios de una forma directa.

25 En algunas realizaciones, el camino de datos de la mantisa podría comprender un módulo de multiplicación en coma fija preparado para recibir, en una primera y una segunda entrada, como mucho los $m+1$ MSBs de la mantisa del primer y segundo número pre-procesado, respectivamente. Si el número está normalizado entonces podría recibir solamente los m LSBs de
30 los $m+1$ MSBs ya que el MSB de un número normalizado es siempre 1 y no es necesario recibirlo. En otro caso, recibiría todos los $m+1$ MSBs. El módulo de multiplicación en coma fija podría estar configurado para generar los $m+2$

MSBs del valor que corresponde a la operación de multiplicación entre dichas mantisas pre-procesadas

- Implementaciones según realizaciones divulgadas aquí tienen la ventaja de que el LSB de las mantisas de los operados no se requiere explícitamente, solamente los $m+2$ MSBs del producto tienen que generarse y no hay necesidad de una lógica de redondeo, incluyendo el computo del sticky bit. En algunas implementaciones de dicho módulo de multiplicación en coma fija, podría usarse un multiplicador coma fija estándar con dos entradas de $m+2$ bits, fijando, el LSB de dichas dos entradas, a uno y los restantes bits igual a las entradas de dicho módulo de multiplicación, mientras en otras implementaciones, el LSB implícito se tiene en cuenta internamente al multiplicador.
- En algunas realizaciones el módulo de multiplicación en coma fija podría comprender un multiplicador redundante configurado para recibir, en una primera y una segunda entrada, como mucho los $m+1$ MSBs de la mantisa del primer y segundo número pre-procesado pre-procesa, respectivamente, y generar, en un formato de representación redundante, los $2m+3$ MSDs del valor correspondiente a la operación de multiplicación entre dichas mantisas pre-procesadas. El LSD de dicho valor es constante e igual a 1. Además, el módulo de multiplicación en coma fija podría comprender un módulo de conversión, conectado a la salida de dicho módulo de multiplicación, configurado para recibir los $m+2$ MSDs de la salida de dicho multiplicador redundante y un bit de acarreo, y generar una salida de $m+2$ bits correspondiente a la conversión del valor redundante recibido a formato de representación no redundante. Además, el módulo de multiplicación en coma fija podría comprender un módulo de red de acarreo configurado para recibir los $m+1$ LSDs de la salida de dicho multiplicador redundante y generar dicho bit de acarreo correspondiente al acarreo de salida de la conversión de los $m+1$ LSDs de la salida de dicho multiplicador redundante a representación no redundante.

Alguien experto en la materia podría apreciar que la longitud de palabra de los valores intermedios en las realizaciones divulgadas aquí, garantiza el menor error de redondeo. Sin embargo, si se permite un mayor error, esos tamaños podrían reducirse para simplificar el hardware de una forma directa.

- 5 Por ejemplo, el tamaño de la salida del multiplicador redundante podría ser menor de $2m+3$ dígitos, tal que la entrada al módulo de conversión se mantiene igual mientras que la entrada del módulo de red de acarreo sería reducido en consonancia.

- 10 En algunas realizaciones el multiplicador redundante podría comprender un generador de productos parciales configurado para recibir, en una primera y una segunda entrada, como mucho los $m+1$ MSBs de la mantisa del primer y segundo número pre-procesado pre-procesa, respectivamente, y generar sus productos parciales en una salida. Además, el multiplicador redundante
- 15 podría comprender un árbol de compresores, con una primera entrada conectada a la salida del generador de productos parciales y una segunda entrada configurada para recibir como mucho los $m+1$ MSBs de la primera y segunda mantisa pre-procesada, dicho árbol de compresores configurado para generar, en una representación redundante, los $2m+3$ MSDs de un valor
- 20 correspondiente a la operación de multiplicación entre dichas mantisas pre-procesadas en una salida.

- Como el LSB de las mantisas pre-procesadas es igual a uno, el generador de productos parciales no requiere generar productos parciales para dichos
- 25 LSBs y podría considerarse que ya están generados. Ellos se introducen directamente en el árbol de compresores (externamente o internamente) lo que resulta en menos operaciones y lógica para el generador de productos parciales.

- 30 En algunas realizaciones el módulo de multiplicación en coma fija podría comprender una tercera entrada con un valor uno.

- En algunas realizaciones el dispositivo podría comprender además un módulo de normalización con una entrada conectada a la salida del módulo de multiplicación en coma fija, donde el módulo de normalización está configurado para generar como mucho los $m+1$ MSBs de la tercera mantisa pre-procesada seleccionando los $m+1$ LSBs de su entrada si el MSB es igual a cero o los $m+1$ MSBs, si dicho bit es igual a uno.
- 10 En algunas realizaciones, el dispositivo podría comprender un conversor de números coma fija pre-procesados a números coma flotante pre-procesados para convertir un número coma fija de $N+2$ bits a un número coma flotante con una mantisa de $M+2$ bits. El conversor de números coma fija pre-procesados a números coma flotante pre-procesados podría comprender un
- 15 calculador de cantidad de desplazamiento, un módulo para calcular el exponente, con una primera entrada para recibir la primera cantidad de desplazamiento del calculador de cantidad de desplazamiento, y una salida para generar el exponente del número coma flotante pre-procesado, y un calculador de la mantisa. El calculador de la mantisa podría comprender un
- 20 módulo de normalización con una primera entrada para recibir los N MSBs de los $N+1$ LSBs del número en coma fija, y una segunda entrada para recibir también la primera cantidad de desplazamiento. El módulo de normalización podría estar configurado para desplazar a la izquierda dichos N MSBs de acuerdo con dicha cantidad de desplazamiento, completando el MSB de las
- 25 posiciones vacantes con cero y el resto con unos, o el MSB con uno y el resto con ceros, para generar como mucho los $M+1$ MSBs de la mantisa. El signo del número coma flotante pre-procesado podría corresponder con el MSB del número coma fija pre-procesado. Introduciendo un conversor de este tipo antes del módulo de multiplicación permite que un número en formato de
- 30 coma fija pre-procesado sea procesado por dispositivos de multiplicación de acuerdo a las realizaciones descritas aquí.

En algunas realizaciones, el módulo de normalización del calculador de la mantisa podría estar configurado para completar dichas posiciones vacantes, aleatoriamente, basándose en un bit seleccionado, o en una combinación de bits seleccionados. En algunas implementaciones, dicho bit (o bits) podrían seleccionarse del número coma fija pre-procesado. En otras implementaciones, una nueva entrada podría configurarse.

En algunas realizaciones, el módulo de normalización del calculador de la mantisa podría estar configurado además para generar selectivamente el complemento a uno del resultado de dicho desplazamiento.

En algunas realizaciones, el dispositivo podría comprender un conversor de números coma fija no procesados a números coma flotante pre-procesados, para convertir un número coma fija no procesado de R bits a un número coma flotante pre-procesado con una mantisa de M+2 bits. El conversor de números coma fija no procesados a números coma flotante pre-procesados podría comprender un calculador de cantidad de desplazamiento, un módulo de normalización configurado para recibir los R bits de un número no procesado en coma fija y generar como mucho los M+1 MSBs de la mantisa del número pre-procesado en coma flotante, y un calculador de exponentes con una primera entrada para recibir la segunda cantidad de desplazamiento proveniente de dicho calculador de cantidad de desplazamiento y una salida para generar el exponente del número pre-procesado en coma flotante. El signo del número pre-procesado en coma flotante podría corresponder con el MSB del número en coma fija no procesado. Introduciendo un conversor de este tipo antes del módulo de multiplicación permite que un número en formato de coma fija no-procesado sea procesable por dispositivos de multiplicación de acuerdo a las realizaciones descritas aquí.

En algunas realizaciones, el módulo de normalización del conversor de números coma fija no procesados a números coma flotante pre-procesados podría comprender una primera entrada para recibir los R bits del número no

procesado en coma fija y una segunda entrada para recibir la segunda cantidad de desplazamiento. El módulo de normalización podría estar configurado para generar un valor que corresponde como mucho a los M+1 MSB de la mantisa pre-procesada mediante el desplazamiento a la izquierda de los R-2 MSBs de los R-1 LSBs de la primera entrada seguida hacia la derecha por un bit a cero y rellenando las posiciones vacantes con el valor del LSB de la primera entrada.

En algunas realizaciones, el módulo de normalización del conversor de números coma fija no procesados a números coma flotante pre-procesados podría estar configurado además para generar selectivamente el complemento a uno de dicho valor si la entrada es negativa.

En algunas realizaciones, el módulo de normalización del conversor de números coma fija no procesados a números coma flotante pre-procesados podría comprender una primera entrada para recibir los R bits del número en coma fija no procesado y una segunda entrada para recibir la segunda cantidad de desplazamiento, donde el módulo de normalización está configurado para generar un valor que se corresponde como mucho con los M+1 MSBs de la mantisa pre-procesada mediante el desplazamiento a la izquierda de los R-1 LSBs de la primera entrada.

El módulo de normalización de acuerdo a varias realizaciones presentes aquí, podría comprender un desplazador variable a la izquierda especial, configurado para recibir un bit para rellenar las posiciones vacantes. En algunas realizaciones el desplazador variable a la izquierda especial podría comprender un número de sucesivos multiplexores que es igual al primer entero mayor o igual que el logaritmo en base 2 de la máxima cantidad de desplazamiento $\lceil \log_2(\text{máxima cantidad de desplazamiento}) \rceil$, con cada multiplexor configurado para efectuar una operación de desplazamiento a la izquierda de 2^i posiciones, $i \in [0, \text{número de multiplexores}-1]$, cada multiplexor configurado para completar las posiciones vacantes usando el valor de dicho

bit recibido.

Además, el módulo de normalización de acuerdo a varias realizaciones presentes aquí, podría estar además configurado para generar selectivamente el complemento a uno del resultado de dicha operación de desplazamiento.

En algunas realizaciones, el calculador de exponentes del conversor de números coma fija no procesados a números coma flotante pre-procesados podría estar configurado para decrementar, de acuerdo a la segunda cantidad de desplazamiento, un valor base para obtener el exponente.

En algunas realizaciones, el calculador de exponentes del conversor de números coma fija no procesados a números coma flotante pre-procesados podría estar configurado además para detectar desbordamientos o valores cero y dar instrucciones al conversor para generar la salida correspondiente.

En algunas realizaciones, el dispositivo podría comprender además un conversor de números coma flotante pre-procesados a números coma fija no procesados para convertir el tercer número en coma flotante pre-procesado a un número en coma fija no procesado. Cuando el número en coma fija no procesado tiene $H+1$ bits, el conversor podría comprender un conversor de números coma flotante pre-procesados a números coma fija pre-procesados con una salida de $H+2$ bits conectada a un módulo de redondeo.

En algunas realizaciones, el módulo de redondeo del conversor de números coma flotante pre-procesados a números coma fija no procesados podría comprender un sumador. Dicho sumador podría estar configurado para recibir, en una entrada, los $H+1$ MSBs de la salida del mencionado conversor de números coma flotante pre-procesados a números coma fija pre-procesados e incrementar dicho valor de entrada si el LSB de dicha salida es igual a 1. Introduciendo un conversor de este tipo después de los dispositivos de acuerdo a las realizaciones descritas aquí permite que el resultado de las

operaciones sea usado por circuitos que funcionan con formato coma fija no procesado.

5 En algunas realizaciones, el dispositivo podría comprender además un conversor de números coma flotante pre-procesados a números coma flotante pre-procesados para convertir un número inicial coma flotante de $J+2$ bits a un subsecuente número coma flotante. Dicho subsecuente número coma flotante podría tener, al menos, un tamaño de mantisa diferente. Esto podría ser útil, por ejemplo, cuando los dos operandos son proporcionados al
10 multiplicador desde diferentes fuentes y necesitan tener mantisas de igual tamaño para permitir las operaciones entre ellos. De la misma forma, también sería útil si el resultado de la operación debe ser convertido a un número coma flotante con una mantisa de diferente tamaño de forma que éste pueda ser utilizado por un circuito posterior. Por lo tanto, el conversor podría
15 colocarse antes o después del multiplicador coma flotante, de acuerdo con esto.

Cuando el subsecuente número en coma flotante pre-procesado tiene una mantisa con $J+2-P$ bits, $P < J+1$, entonces el conversor podría comprender una
20 unidad de redondeo para eliminar los $P+1$ LSBs de los $J+2$ bits de la mantisa inicial pre-procesada, para generar como mucho los $J+1-P$ MSBs de la mantisa del subsecuente número en coma flotante pre-procesado. El LSB de la mantisa del subsecuente número en coma flotante pre-procesado es igual a 1. El conversor podría comprender además un calculador de exponentes
25 para generar el exponente del subsecuente número en coma flotante pre-procesado.

Cuando el subsecuente número en coma flotante pre-procesado tiene una mantisa con $J+2+Q$ bits, entonces el conversor podría comprender un módulo
30 de rellenado, configurado para recibir como mucho los $J+1$ MSBs de la mantisa del número en coma flotante pre-procesado inicial y generar como mucho los $J+Q+1$ MSBs de la mantisa del subsecuente número en coma

flotante pre-procesado fijando el MSB de los Q LSBs a uno o a cero y los restantes Q-1 bits de dicho Q LSBs al complemento del mencionado MSB. Los como mucho J+1 MSBs de la mantisa del subsecuente número en coma flotante pre-procesado son los mismos que los como mucho J+1 MSBs de la mantisa del número en coma flotante pre-procesado inicial. El conversor podría comprender además un calculador de exponentes para generar el exponente del subsecuente número en coma flotante pre-procesado.

En algunas realizaciones, el módulo de rellenado del conversor de números coma flotante pre-procesados a números coma flotante pre-procesados podría estar configurado para fijar aleatoriamente dicho MSB basándose en el valor de un bit seleccionado, o de una combinación de bits seleccionados. En algunas implementaciones, dicho bit (o bits) podrían seleccionarse de la mantisa del número en coma flotante pre-procesado inicial.

En algunas realizaciones, el dispositivo podría comprender además un conversor de números coma flotante pre-procesados a números coma fija pre-procesados para convertir un número en coma flotante con una mantisa de F+2 bits en un número en coma fija. Introduciendo un conversor de este tipo después de los dispositivos de acuerdo a las realizaciones descritas aquí permite que el resultado de las operaciones sea usado por circuitos que funcionan con formato coma fija pre-procesado.

Cuando el número en coma fija pre-procesado comprende L bits, con $L < F+4$, el conversor de números coma flotante pre-procesados a números coma fija pre-procesados podría comprender un calculador de la cantidad de desplazamiento que recibe el exponente del número en coma flotante pre-procesado en una entrada y genera una tercera cantidad de desplazamiento en una salida. El conversor podría comprender además un módulo de desplazamiento con una primera entrada para recibir como mucho los L-1 MSBs de la mantisa del número en coma flotante pre-procesado y una segunda entrada conectada a la salida del calculador de cantidad de desplazamiento y una tercera entrada para recibir el signo del mencionado

número en coma flotante, para generar los $L-1$ MSBs del número en coma fija pre-procesado en una salida. El LSB de dicho número en coma fija pre-procesado es igual a $B/2$ y podría estar implícito.

5 En algunas realizaciones, el módulo de desplazamiento del conversor de números coma flotante pre-procesados a números coma fija pre-procesados podría comprender un desplazador aritmético a la derecha conectado a un inversor de bits condicional.

10 Cuando el número en coma fija pre-procesado comprende $F+C+3$ bits, $C>0$, el conversor de números coma flotante pre-procesados a números coma fija pre-procesados podría comprender un calculador de cantidad de desplazamiento que recibe el exponente del número en coma flotante pre-procesado en una entrada y que genera una tercera cantidad de desplazamiento en una salida, y un módulo de desplazamiento aritmético a la
15 derecha con una primera entrada conectada a la salida del calculador de desplazamiento, y configurado para generar los $F+C+2$ MSBs del número en coma fija pre-procesado mediante el desplazamiento aritmético a la derecha de un valor intermedio de $F+C+2$ bits. Dicho valor intermedio podría estar formado, de izquierda a derecha, por el bit de signo, los $F+1$ MSBs de la
20 mantisa del número en coma flotante pre-procesado, y el MSB de los C LSBs puesto a cero y el resto a uno, o el MSB de los C LSBs puesto a uno y el resto a cero.

25 En algunas realizaciones, el módulo de desplazamiento aritmético a la derecha podría estar configurado para poner aleatoriamente dicho MSB de los C LSBs del mencionado valor de $F+C+2$ bits en base al valor de un bit seleccionado, o de una combinación de bits seleccionados. En algunas implementaciones, dicho bit (o bits) podrían seleccionarse del número en coma flotante pre-procesado.

30

En algunas realizaciones, el módulo de desplazamiento aritmético a la derecha podría estar configurado además para generar selectivamente el

complemento a uno del resultado de la mencionada operación de desplazamiento.

5 En algunas realizaciones, el dispositivo podría comprender además un conversor de números en coma flotante no procesados a números en coma flotante pre-procesados para convertir un número en coma flotante no procesado con una mantisa de $E+2$ bits en un número en coma flotante pre-procesado. Introduciendo este conversor en alguna etapa anterior a un dispositivo de acuerdo a las realizaciones descritas aquí, permite que
10 números que no están en el formato pre-procesado sean procesables por los mencionados dispositivos.

Cuando el número coma flotante pre-procesado tiene una mantisa de $E+2-D$ bits, $D < E+1$ entonces el conversor de números en coma flotante no
15 procesados a números en coma flotante pre-procesados podría comprender una unidad de redondeo configurada para eliminar los $D+1$ LSBs de la mantisa del número en coma flotante no procesado, para generar como mucho los $E+1-D$ MSBs de la mantisa del número coma flotante pre-procesado. El LSB de la mantisa del número en coma flotante pre-procesado
20 es igual a uno y podría estar implícito. El conversor de números en coma flotante no procesados a números en coma flotante pre-procesados podría comprender además un calculador de exponentes para generar el exponente del número en coma flotante pre-procesado.

25 En algunas realizaciones, la unidad de redondeo del conversor de números en coma flotante no procesados a números en coma flotante pre-procesados podría estar configurada además para, selectivamente, poner a cero el segundo LSB de la mantisa del número en coma flotante pre-procesado si todos los $D+1$ LSBs de la mantisa del número en coma flotante no procesado
30 son iguales a cero.

Cuando el número en coma flotante pre-procesado tiene una mantisa de

E+2+G bits entonces el conversor de números en coma flotante no procesados a números en coma flotante pre-procesados podría comprender un módulo de rellenado, configurado para recibir como mucho los E+2 bits de la mantisa de un número en coma flotante no procesado y generar como
 5 mucho los E+G+1 MSBs de la mantisa del número en coma flotante pre-procesado fijando como mucho los E+2 MSBs del número en coma flotante pre-procesado al mismo valor que como mucho los E+2 bits de la mantisa del número en coma flotante no procesado y los restantes bits a cero. El LSB de la mantisa del número en coma flotante pre-procesado es igual a uno y
 10 podría estar implícito. El conversor de números en coma flotante no procesados a números en coma flotante pre-procesados podría comprender además un calculador de exponentes para generar el exponente del número en coma flotante pre-procesado.

15 En algunas realizaciones, el módulo de rellenado del conversor de números en coma flotante no procesados a números en coma flotante pre-procesados podría estar configurado además para generar selectivamente el valor correspondiente a restar uno del segundo LSB de la mencionada mantisa generada cuando un bit seleccionado, o una combinación de bit
 20 seleccionados, de la mantisa no procesada de entrada es igual a uno.

En algunas realizaciones, el dispositivo podría comprender además un conversor de números en coma flotante pre-procesados a números en coma flotante no procesados para convertir un número en coma flotante pre-
 25 procesados con una mantisa de U+2 bits a un número en coma flotante no procesado. Introduciendo un conversor de este tipo después de los dispositivos de acuerdo a las realizaciones descritas aquí, permite que el resultado de la operación sea procesable por circuitos coma flotante comunes.

30 Cuando el número en coma flotante no procesado tiene una mantisa de U+2-V bits, entonces el conversor podría comprender un módulo de redondeo,

configurado para recibir como mucho los $U+3-V$ MSBs de la mantisa del número en coma flotante pre-procesado y generar como mucho los $U+2-V$ bits de la mantisa del número en coma flotante no procesado y un calculador de exponentes configurado para generar el exponente del número en coma flotante no procesado.

En algunas realizaciones, el módulo de redondeo del conversor de números en coma flotante pre-procesados a números en coma flotante no procesados podría comprender un sumador. El sumador podría estar configurado para recibir, en una entrada, como mucho los $U+2-V$ MSBs de la mantisa del número en coma flotante pre-procesado e incrementar dicho valor de entrada si el $(U+3-V)$ -ésimo MSB de dicha mantisa es igual a 1, y generar una instrucción para el calculador de exponentes, si se produjera un desbordamiento.

En algunas realizaciones, el calculador de exponentes podría estar configurado además para incrementar el exponente de salida cuando se genera la mencionada instrucción desde el módulo de redondeo.

Cuando el número en coma flotante no procesado tiene una mantisa con $U+2+W$ bits entonces el conversor de números en coma flotante pre-procesados a números en coma flotante no procesados podría comprender un módulo de rellenado, configurado para recibir como mucho los $U+1$ MSBs de la mantisa del número en coma flotante pre-procesado y generar como mucho los $U+W+2$ bits de la mantisa del número en coma flotante no procesado poniendo el MSB de los $W+1$ LSBs a uno y los restantes bits a cero, y un calculador de exponentes configurado para generar el exponente del número en coma flotante pre-procesado.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

A continuación se describirán realizaciones particulares de la presente invención por medio de ejemplos no limitativos, con referencia a los dibujos

adjuntos, en los que:

Fig. 1 and 1b ilustran el camino de datos de la mantisa de un multiplicador en coma flotante (FP) de acuerdo a dos ejemplos;

5

Fig. 2 ilustra un ejemplo de implementación de un multiplicador en coma fija pre-procesado;

Fig. 3 and 3b ilustran ejemplos de implementación de un multiplicador en coma fija pre-procesado redundante;

10

Fig. 4 ilustra un ejemplo de implementación de un conversor de números coma fija pre-procesados a números coma flotante pre-procesados;

Fig. 4a ilustra un ejemplo de implementación de un desplazador a la izquierda pre-procesado;

15

Fig. 4b ilustra un ejemplo de implementación de un desplazador a la izquierda especial;

20

Fig. 5 ilustra un ejemplo de implementación de un conversor de números coma fija no procesados a números coma flotante pre-procesados;

Fig. 5a and 5b ilustran ejemplos de implementación de un módulo de normalización de un conversor de números coma fija no procesados a números coma flotante pre-procesados;

25

Fig. 6a, 6b and 6c ilustran ejemplos de implementación de un conversor de números coma flotante pre-procesados a números coma flotante pre-procesados;

30

Fig. 7, 8a and 8b ilustran ejemplos de implementación de un conversor de

números coma flotante pre-procesados a números coma fija pre-procesados;

Fig. 9, 10a, 10b ilustran ejemplos de implementación del camino de datos de la mantisa de un conversor de números en coma flotante no procesados a números en coma flotante pre-procesados;

Fig. 11 ilustra un ejemplo de implementación de un conversor de números coma flotante pre-procesados a números coma flotante no procesados;

10

Fig. 11a ilustra un ejemplo de implementación del módulo de redondeo de un conversor de números coma flotante pre-procesados a números coma flotante no procesados;

Fig. 12 ilustra un ejemplo de implementación de un conversor de números coma flotante pre-procesados a números coma fija no procesados;

DESCRIPCION DETALLADA DE LAS REALIZACIONES

Fig. 1 ilustra el camino de datos de la mantisa del multiplicador coma flotante (FP) de acuerdo a un ejemplo. El multiplicador FP 100M recibe m bits de una primera mantisa Mx y de una segunda mantisa My, respectivamente. Ambas mantisas pertenecen a números coma flotante pre-procesados. Las mantisas Mx y My ambas tienen m+1 bits. Sin embargo, como ambas mantisas pertenecen a números pre-procesados, el LSB de ambas mantisas es igual a uno (1) y no necesita ser introducido en el multiplicador FP en la entrada. Además, en el ejemplo de punto flotante Fig. 1 los dos números FP están normalizados. Sin embargo, para simplificar la descripción, el MSB del número normalizado, el bit entero, está incluido en los m bits que se introducen en el multiplicador FP 100M. En una implementación alternativa, este bit podría no recibirse e introducirse antes del multiplicador coma fija o internamente a dicho multiplicador coma fija. El multiplicador FP 100M

comprende un multiplicador coma fija 105 y un módulo de normalización 115. El módulo de normalización 115 puede ser un desplazador a la derecha de una posición. El multiplicador coma fija 105 recibe los m MSBs de las mantisas M_x y M_y . El multiplicador coma fija 105 multiplica las mantisas completas y genera los $m+1$ MSBs del resultado de dicha multiplicación. Entonces, el módulo de normalización 115, desplaza dicho resultado una posición a la derecha si el MSB de dicho resultado es igual a uno. La salida del módulo de normalización 115 es un número m bits que corresponde a los m MSBs de la mantisa de $m+1$ bits del resultado de la multiplicación de los números coma flotante de la entrada. En el ejemplo de la Fig. 1 el LSB de las mantisas de la entrada está implícito y se introduce dentro del multiplicador coma fija. Alternativamente, éste puede introducirse como una entrada separada del multiplicador coma fija, como se muestra en el multiplicador coma fija 105b de la Fig. 1b. En ambas figuras el LSB de la mantisa M_z está implícito y es igual a 1.

Debemos indicar que en esta implementación, los m MSBs de la mantisa incluyen el bit entero. En una implementación alternativa, el bit entero de la mantisa de la salida podría descartarse después de la normalización, ya que es siempre uno.

Fig. 2 ilustra un ejemplo de implementación de un multiplicador coma fija. El multiplicador coma fija 200M comprende un multiplicador redundante 205, un módulo de red de acarreo 207 y un módulo de conversión 209. El multiplicador redundante 205 recibe, en una primera y segunda entrada, los m MSBs de la primera y segunda mantisa pre-procesada y dos entradas adicionales conectadas a uno, tal que los m bits en la entrada de cada mantisa se aumentan un bit por la derecha. Sin embargo, en una implementación alternativa, la introducción del uno adicional podría realizarse internamente al módulo 205 sin necesitar una entrada especial. Esto es meramente ilustrado en el ejemplo de Fig. 2, y en otros ejemplos siguientes, para indicar la necesidad de la introducción funcional del LSB implícito. El

multiplicador redundante 205 genera, en un formato de representación redundante, los $2m+1$ MSDs del valor correspondiente a la operación de multiplicación entre dichas mantisas pre-procesadas. El LSD de dicho resultado es siempre uno y no se requiere explícitamente. El multiplicador redundante 205 mostrado en la Fig. 2 genera el resultado en formato de acarreo almacenado y entonces dicho resultado se entrega en una primera y una segunda salida de $2m+1$ bit cada una, correspondientes a la palabra de suma y a la palabra de acarreo, respectivamente. Sin embargo, un experto en la materia podría apreciar que, con modificaciones menores de los circuitos presentados, podrían usarse otros formatos de representación redundante, tal como representación de dígitos con signo.

El módulo de red de acarreo 207 recibe los m LSDs de la salida pre-procesa de dicho multiplicador redundante, los cuales no incluyen el LSD implícito del formato pre-procesado, y genera el bit de acarreo correspondiente a la conversión de dichos dígitos a una representación binaria no redundante. En este ejemplo particular, como se usa representación de acarreo almacenado, el módulo de red de acarreo 207 recibe los m LSBs de las palabras de suma y acarreo en una primera y segunda entrada respectivamente y genera el último bit de acarreo correspondiente a la suma de ambas entradas correctamente alineadas.

El módulo de conversión 209 recibe los $m+1$ MSDs de la salida del multiplicador redundante 205 y el bit de acarreo desde el módulo de red de acarreo 207, y genera los $m+1$ bits correspondientes a los $m+1$ MSBs del valor de la multiplicación de las mantisas de entrada, en una representación no redundante. En este ejemplo particular, como se usa representación de acarreo almacenado, el módulo de conversión 209 recibe los $m+1$ MSBs de las palabras de suma y acarreo en una primera y una segunda entrada, respectivamente, y el bit de acarreo en una tercera entrada y genera un valor correspondiente a la suma de ambas palabras de entrada correctamente alineadas y el bit de acarreo.

Fig. 3 y 3b ilustra las implementaciones de un multiplicador coma fija. El multiplicador coma fija representado en Fig. 3 recibe los m MSBs de dos números coma fija pre-procesados (mantisas M_x y M_y), ya que el LSB es constante e igual a uno, y genera los $2m+1$ MSBs del resultado de la multiplicación de ambos números de entrada, siendo el LSB de dicho resultado también implícito e igual a uno. Dicho de otra forma, si los m MSBs de M_x se representan por A y los m MSBs de M_y por B , entonces el valor a la salida de $2m+1$ bits es igual a $A \cdot B + 1/2A + 1/2B$. El multiplicador coma fija 300M comprende un multiplicador redundante 305 y un módulo de conversión 335. El multiplicador redundante 305 comprende un módulo generador de productos parciales 325 y un árbol de compresores 330. El módulo generador de productos parciales 325 recibe dichos m MSBs de los dos números coma fija pre-procesados, en una primera y una segunda entrada, respectivamente, y genera todos los productos parciales correspondientes al producto de la primera entrada por cada bit de la segunda entrada. En una implementación alternativa, la segunda entrada podría estar dividida en varios grupos de bits y los productos parciales generados podrían corresponder a los productos de la primera entrada por cada dicho grupo de bits.

El árbol de compresores 330 recibe la salida del módulo generador de productos parciales 325 y una copia de las dos entradas del módulo generador de productos parciales 325 y genera una salida de $2m+1$ dígitos redundantes correspondiente a la suma de todas sus entradas. En este ejemplo particular, como se usa representación de acarreo almacenado, se producen dos números de $2m+1$ bits correspondientes a las palabras de suma y acarreo. Si se desea una salida no redundante, el módulo de conversión 335 es usado para transformar la salida del árbol de compresores 330, a un número no redundante de $2m+1$ bit correspondiente a los $2m+1$ MSBs del producto de las entradas iniciales.

El multiplicador coma fija representado en Fig. 3b es similar al anterior, pero

la segunda entrada se recodifica (por ejemplo, mediante recodificación de Booth) antes de entrar en el generador de productos parciales 325b para producir menos productos parciales, mediante el uso del módulo de recodificación 320b. El valor uno se inserta también en la entrada del módulo de recodificación 320b, tal que los m bits de la segunda entrada son aumentados por la derecha con un bit correspondiente al LSB implícito. Sin embargo, en otras implementaciones, la introducción del uno adicional podría realizarse internamente al módulo de recodificación 320b sin necesidad de una entrada especial. Esto es meramente ilustrado en el ejemplo para indicar la necesidad de la introducción funcional del LSB implícito. De forma similar, el LSB de la otra entrada está también ilustrado en la primera entrada del generador de productos parciales 325.

El multiplicador FP descrito arriba requiere números FP que hayan sido pre-procesados de acuerdo a la invención como se describió arriba. Estos números pre-procesados podrían ser generados por circuitos, tales como los mencionados multiplicadores FP, que están diseñados para funcionar con números pre-procesados o podrían ser generados por conversores, diseñados para convertir número no procesados o números pre-procesados no FP en números pre-procesados. Además, los números pre-procesados generados por los multiplicadores descritos arriba podrían, en concordancia, requerir conversores tales que los números generados podrían ser usados por circuitos que no estén diseñados para operar números pre-procesados.

En los siguientes ejemplos, se considera que los números en coma flotante, tanto los no procesados como los pre-procesados, son representados por un bit de signo, un exponente y una mantisa normalizada sin signo de tal forma que el MSB es igual a uno y está explícitamente incluido en la representación de la mantisa. De la misma forma, los números en coma fija, tanto los no procesado como los procesados, son representados en representación en complemento a dos, siendo el MSB equivalente al bit de signo. Sin embargo, un experto en la técnica podría apreciar que otros formatos que tienen una

representación diferente podrían ser utilizados con modificaciones menores en los circuitos descritos. Algunas de estas variaciones podrían ser:

a) en FP

- representación implícita del MSB de la mantisa, o
- 5 - representación fusionada del signo y la mantisa mediante representación en complemento a dos o cualquier otra representación

b) en punto fijo: representación signo-magnitud, o representación sin signo.

10 Una categoría de tales conversores es la de conversores para convertir números en coma fija pre-procesados a números FP pre-procesados. La Fig. 4 ilustra un ejemplo de tal conversor para números en coma fija pre-procesados de $m+2$ bits y un número FP pre-procesado con una mantisa de $n+1$ bits. El conversor 600 comprende un módulo de normalización 630 que

15 tiene un inversor de bits condicional 605 en serie con un desplazador a la izquierda pre-procesado 610. El inversor de bits condicional 605 tiene una primera entrada para recibir los m LSBs de los $m+1$ MSBs del número en coma fija pre-procesado de $m+2$ bits. El MSB dicho número de $m+2$ bits es el signo y será el signo del número FP pre-procesado así como será usado

20 para controlar el inversor de bits condicional. La salida de m bits del inversor de bits condicional 605 es la entrada al desplazador a la izquierda pre-procesado 610. En implementaciones alternativas el desplazador a la izquierda pre-procesado precede al inversor de bits condicional 605. La función del desplazador a la izquierda pre-procesado 610 es descrito con

25 más detalle en la Fig. 4a. El desplazador a la izquierda pre-procesado 610 requiere un desplazador a la izquierda especial 610a con una nueva tercera entrada de un bit, el cual permite seleccionar el valor usado para rellenar las posiciones vacantes después del desplazamiento. Una implementación del desplazador a la izquierda especial 610a podría ser similar al del desplazador

30 a la izquierda especial 245 ilustrado en la Fig. 4b. El desplazador a la izquierda especial 245, mostrado en Fig. 4b, se implementa usando varios multiplexores dos a uno ($\log_2$ de la máxima cantidad de desplazamiento

requerida) conectados en serie, tal que la salida de un desplazador es usada en la entrada del siguiente. Las entradas de datos del primer multiplexor son conectadas a la primera entrada del desplazador a la izquierda especial, a la posición no desplazada y a la desplazada (2^0), respectivamente, mientras
 5 que el bit de control se acopla al LSB de la cantidad de desplazamiento (segunda entrada). Las entradas de datos del segundo multiplexor se acoplan a la salida de las posiciones primera, no desplazada y desplazada en 2 (2^1), respectivamente, mientras el bit de control se acopla al segundo LSB de la cantidad de desplazamiento (segunda entrada). El resto del multiplexor es
 10 conectado en concordancia. En desplazadores a la izquierda convencionales las posiciones vacantes son completadas con ceros. En esta propuesta las posiciones vacantes son completadas con la tercera entrada (nueva entrada L). En este ejemplo de la Fig. 4b, la máxima cantidad de desplazamiento es m-1.

15 Volviendo al ejemplo de la Fig. 4, en este caso, la máxima cantidad de desplazamiento es m o m+1. Si el número en coma fija es igual a cero y el bit R en la Fig. 4a es también igual a cero, requiere una máxima cantidad de desplazamiento que tiene un bit adicional (m+1) de manera que la mantisa
 20 está normalizada. Alternativamente, si cuando el número en coma fija es igual a cero, puede ser tratado como un caso especial y convertido a cero en FP. Entonces la máxima cantidad de desplazamiento podría ser igual a m.

Usando este desplazador a la izquierda especial 610a, el valor de entrada del
 25 desplazador a la izquierda pre-procesado 610 es aumentado con un LSB adicional fijado a cualquier bit aleatorio (por ejemplo, el LSB del valor de entrada inicial) y la tercera entrada del desplazador a la izquierda especial se pone al inverso de dicho valor aleatorio, para rellenar ambas, las posiciones vacantes requeridas para completar el tamaño n si $n > m+1$ y los posiciones
 30 vacantes producidas después del desplazamiento. La salida del desplazador a la izquierda pre-procesado 610 comprende los n MSBs de la mantisa Mz del número FP pre-procesado. Dicha salida se corresponde sólo con los n

MSBs del valor desplazado si $n < m$. El LSB de la mantisa M_z está implícito y es igual a 1.

En un camino paralelo, el conversor 600 comprende el módulo detector de uno de cabecera (LOD) 615 que tiene una entrada conectada a la salida del inversor de bits condicional 605 y una salida para la generación de la cantidad de desplazamiento del desplazador a la izquierda pre-procesado especial 610 que también se utiliza como entrada al módulo de cálculo de exponentes 620 para generar el exponente E_z del número FP pre-procesado. Alternativamente, la entrada del módulo LOD 615 podría estar conectada directamente a la entrada del conversor 600, pero en este caso debería detectar el primer cero, en lugar del uno, cuando el número es negativo.

En comparación con los conversores convencionales de números en coma fija a FP, cuando $M > N$, no hay redondeo hacia arriba después de la operación de desplazamiento y por lo tanto hay una reducción en los componentes y en el procesamiento. Cuando $M < N$, entonces no hay sesgo producido por el redondeo con la utilización del conversor propuesto.

Otra categoría de conversores son los conversores para convertir números en coma fija no procesados a números en coma flotante pre-procesados. La Fig. 5 ilustra un conversor de este tipo. El conversor 700 comprende un módulo de normalización 705 configurado para recibir los m LSBs de un número en coma fija de $m+1$ bits. El MSB del número en coma fija es el signo y se utiliza para controlar el módulo de normalización 705 y para poner el signo del número FP pre-procesado. El módulo de normalización 705 podría comprender un desplazador a la izquierda convencional, para desplazar el valor de entrada hasta eliminar los bits no significativos de la izquierda, seguido de un inversor de bit condicional, para calcular el complemento a uno de dicho valor desplazado si el número de entrada es negativo. Esta configuración evita el sesgo redondeando hacia arriba los números de entrada positivos y hacia abajo los negativos. Además, el módulo de

normalización podría ser implementado de acuerdo con los ejemplos descritos en la Fig. 5a y en la Fig. 5b. En la Fig. 5a, el módulo de normalización 705a comprende un desplazador a la izquierda especial 706a que es similar al desplazador a la izquierda especial 610 descrito en la Fig. 4a. En este caso el desplazador a la izquierda especial 706a recibe los m-1 MSBs de los m LSBs del número en coma fija no procesado, extendidos a la derecha con un bit con valor cero, mientras que el LSB del número en coma fija se utiliza como la tercera entrada del desplazador a la izquierda especial 706a. La salida del desplazador a la izquierda especial 706a corresponde a los n bits más significativos del valor desplazado y es la entrada a un inversor de bits condicional 708a que tiene una segunda entrada para recibir el bit de signo del número en coma fija. La salida del inversor de bits condicional 708a son los n bits más significativos de la mantisa Mz del número FP pre-procesado. El LSB de la mantisa está implícito y es igual a 1. En otras implementaciones, el MSB de la mantisa normalizada Mz podría no incluir el uno de cabecera. Por lo tanto, la salida del inversor de bits condicional podría tener un bit menos.

La Fig. 5b muestra una implementación alternativa del módulo de normalización 705. El módulo de normalización 705b comprende un primer inversor de bits condicional 706b para la recepción de los m LSBs del número en coma fija no procesado. La salida del inversor de bits condicional 706b se introduce en el desplazador a la izquierda especial 708b. Los m-1 MSBs de la salida del inversor de bits condicional se introducen en la primera entrada del desplazador a la izquierda especial 708b, mientras que el LSB se utiliza como la tercera entrada. Además, el bit de signo se introduce como el LSB de la primera entrada del desplazador a la izquierda especial 708b para aumentar los m-1 bits. La salida de n bits del desplazador a la izquierda especial son los n MSBs de la mantisa Mz del número FP pre-procesado. El LSB de la mantisa está implícito y es igual a 1.

Volviendo al conversor 700 de la Fig. 5, un camino paralelo comprende un

módulo LOD 710 que tiene una entrada que recibe el número en coma fija no procesado y una salida para la generación de la cantidad de desplazamiento para el módulo de normalización 705 que también se utiliza como entrada al módulo de computación del exponente 715 para generar el exponente E_z del número FP pre-procesado. En otras implementaciones que podrían utilizar el módulo de normalización 705b, la entrada del módulo LOD 710 podría recibir la salida del inversor de bits condicional 706b en su lugar.

Otra categoría de conversores son los conversores para convertir números FP pre-procesados a números FP pre-procesados de diferente tamaño de mantisa. La Fig. 6a es un ejemplo de un conversor de este tipo. El conversor 800a ilustra un conversor adaptado para convertir un número FP pre-procesado que tiene $n+m+1$ bits de mantisa a una mantisa de $n+1$ bits. El LSB de ambas mantisas es igual a 1 y por lo tanto no se representa. El signo ($sign_x$) del número FP pre-procesado original va a seguir siendo el mismo en el número FP pre-procesados objetivo (representado como $sign_z$). Los n MSBs de la mantisa original serán los n MSBs de la mantisa pre-procesada objetivo. Es decir, tiene lugar una simple función de truncamiento. Por lo tanto, no se genera un bit de desbordamiento, y un calculador de exponentes 801a podría generar el exponente objetivo E_z basándose simplemente en el exponente original E_x .

La Fig. 6b es otro ejemplo de un conversor de pre-procesados FP a pre-procesados FP. El conversor 800b ilustra un conversor adaptado para convertir un número FP pre-procesado con una mantisa de $m+1$ bits a una mantisa de $n+m+1$ bits. El conversor 800b es una versión con sesgo de un conversor de este tipo. Una vez más, el LSB de ambas mantisas es igual a 1 y por lo tanto no se representa. De acuerdo con el conversor 800b, el bit de signo sigue siendo el mismo, el calculador de exponentes 801b calcula el nuevo exponente, y un circuito para ampliar el tamaño mantisa añadiendo a la derecha un bit a uno y tantos ceros como sea necesario para completar el nuevo tamaño de la mantisa. Alternativamente, se podría usar un cero

seguido de unos.

La Fig. 6c es otro ejemplo de un conversor de pre-procesados FP a pre-procesados FP. El conversor 800c ilustra un conversor adaptado para
 5 convertir un número FP pre-procesado con $n+1$ bits de mantisa a una mantisa de $n+m+1$ bits. El conversor 800c es una versión sin sesgo de un conversor de este tipo. Una vez más, el LSB de ambas mantisas es igual a 1 y por lo tanto no se representa. De acuerdo con conversor 800c, el bit de signo sigue siendo el mismo, el calculador de exponentes 801c calcula el
 10 nuevo exponente, y un circuito para ampliar el tamaño de la mantisa añadiéndole a la derecha un bit con un valor aleatorio y tantos bits, con el inverso de dicho valor, como se requieran para completar el nuevo tamaño de la mantisa. El bit aleatorio podría ser cualquier bit de la mantisa inicial o una combinación de ellos, tal como el inverso del segundo LSB que se usa en la
 15 Fig. 6c.

Otra categoría de conversores son los conversores para convertir números FP pre-procesados a números en coma fija pre-procesados. La Fig. 7 ilustra
 20 un conversor de este tipo para la conversión de un número FP que tiene una mantisa de $n+m+1$ bits y un exponente de d bits en un número en coma fija de $n+2$ bits. Los n bits más significativos de la mantisa son de entrada al inversor de bits condicional 905. El LSB de la mantisa es igual a 1 y no se introduce. El signo del número FP pre-procesado se utiliza para controlar el
 25 inversor de bits condicional 905. La salida del inversor de bits condicional 905 junto con el signo (sign_x) se introducen en desplazador a la derecha 910. El desplazador a la derecha 910 tiene otra entrada para recibir la cantidad de desplazamiento del calculador de cantidad de desplazamiento 915. El calculador de cantidad de desplazamiento 915 recibe el exponente del
 30 número FP pre-procesado y genera la cantidad de desplazamiento. La salida del desplazador a la derecha 910 son los $n+1$ MSBs del número en coma fija pre-procesado. El LSB es, de manera similar, igual a 1 y no es ni generado ni

representado.

La Fig. 8a ilustra un conversor con sesgo para la conversión de un número FP pre-procesado que tiene $n+1$ bits de mantisa y un exponente de d bits a un número en coma fija pre-procesado de $n+m+2$ bits. Los n MSBs de la mantisa se introducen en el inversor de bits condicional 1005a. El LSB de la mantisa es igual a 1 y no se introduce. El signo del número FP pre-procesado se utiliza para controlar el inversor convencional 1005a. La salida del inversor de bits condicional 1005a junto con el signo (sign_x) son introducidos al desplazador a la derecha 1010a. La salida del inversor de bits condicional es expandida mediante la adición por la derecha de un bit a uno y tantos bits a cero como sean necesarios para completar el nuevo tamaño. En una implementación alternativa esta expansión se podría realizar con un bit a cero y tantos bits a uno como fuesen necesarios. Este número expandido entra al desplazador a la derecha 1010a. El desplazador a la derecha 1010a tiene otra entrada para recibir la cantidad de desplazamiento del calculador de cantidad de desplazamiento 1015a. El calculador de cantidad de desplazamiento 1015a recibe el exponente del número FP pre-procesado y genera la cantidad de desplazamiento. La salida del desplazador a la derecha 1010a son los $n+m+1$ MSBs del número en coma fija pre-procesado. El LSB es, similarmente, igual a 1 y no es ni generado ni representado.

La Fig. 8b ilustra un conversor sin sesgo para la conversión de un número FP pre-procesado, que tiene $n+1$ bits de mantisa y un exponente de d bits, a un número en coma fija pre-procesado de $n+m+2$ bits. Los n bits más significativos de la mantisa se introducen en el inversor de bits condicional 1005b. El LSB de la mantisa es igual a 1 y no se introduce. El signo del número FP pre-procesado se utiliza para controlar el inversor de bits condicional 1005b. La salida del inversor de bits condicional 1005b junto con el signo (sign_x) son introducidos al desplazador a la derecha 1010b. La salida del inversor de bits condicional es expandida mediante la adición por la derecha un bit seleccionado al azar y tantos bits con el valor inverso de dicho

bit aleatorio como sean necesarios para completar el nuevo tamaño. El bit aleatorio podría ser cualquiera de la mantisa inicial. Este número expandido entra al desplazador a la derecha 1010b. El desplazador a la derecha 1010b tiene otra entrada para recibir la cantidad de desplazamiento del calculador de cantidad de desplazamiento 1015b. El calculador de cantidad de desplazamiento 1015b recibe el exponente del número FP pre-procesado y genera la cantidad de desplazamiento. La salida del desplazador a la derecha 1010b son los $n+m+1$ MSBs del número en coma fija pre-procesado. El LSB es, similarmente, igual a 1 y no es ni generada ni representado.

10

En otras implementaciones de los ejemplos de las figuras Fig. 7, 8a y 8b, el MSB de la mantisa normalizada podría no incluir el bit 1 de cabecera. Por lo tanto, este bit a 1 debería ser introducido en el inversor de bit condicional.

15 Otra categoría de conversores son los conversores para convertir números FP no procesados a números FP pre-procesados. En un primer caso, la mantisa del número original FP es mayor que la mantisa del número FP objetivo. El conversor discutido con referencia a la Fig. 6 podría ser utilizado, pero introduce algo de sesgo. En caso de redondeo sin sesgo, la nueva mantisa se calcula con el circuito ilustrado en la Fig. 9. Para una mantisa de entrada de $n+m+1$ bits, los $n-1$ MSBs son los mismos en el original y en el número FP objetivo. El enésimo MSB de la nueva mantisa se pone a cero si los $m+1$ LSBs de la mantisa original son todos cero, o igual al enésimo MSB de la mantisa original en otro caso. El LSB de la nueva mantisa será 1 y está implícito, ya que el número FP es un número FP pre-procesado.

25

Cuando la mantisa del número FP pre-procesado tenga más bits ($n+m+1$) que la mantisa del número FP no procesado (n) entonces:

30 a) en el caso del redondeo con sesgo la mantisa del número no procesado se expande con tantos ceros como sea necesario. Esto se ilustra en la Fig. 10a. El LSB será igual a 1 y está implícito.

b) en el caso de redondeo sin sesgo, los $n-1$ MSB son los mismos. El enésimo bit se fuerza a cero. Los $m+1$ bits a la derecha se hacen igual al LSB de la mantisa no procesada. Esto se ilustra en la Fig. 10b. El LSB de la mantisa pre-procesada será 1, ya que el número FP es un número pre-procesado

Otra categoría de conversores son los conversores para convertir números FP pre-procesados a números FP no procesados. Cuando la mantisa del número FP pre-procesado tiene más bits ($n+m+1$) que la mantisa no procesada (n), entonces el circuito ilustrado en la Fig. 11 se podrían utilizar. El signo sigue siendo el mismo. Los $n+1$ MSB de la mantisa pre-procesada se redondean a n bits por medio del redondeador 1310. El redondeador 1310 también genera un bit de desbordamiento que utiliza el calculador de exponentes 1320, junto con el exponente de entrada, para generar el exponente del número FP no procesado. El redondeador 1310 se explica en la Fig. 11a. Un sumador 1310a se usa para incrementar en uno los n MSBs de la mantisa pre-procesada si el $(n+1)$ -ésimo MSB es uno. Cuando la mantisa del número FP pre-procesado tiene menos bits ($m+1$) que la mantisa no procesada ($m+n$), entonces se podría utilizar el circuito ilustrado en la Fig. 6b. En una implementación alternativa el redondeador podría realizar otro tipo de redondeo.

Aún, otra categoría de conversores son los conversores para convertir números FP pre-procesados a en coma fija no procesados. La Fig. 12 ilustra un conversor de este tipo en el que el número de bits de la mantisa de entrada es mayor que el número de bits del número en coma fija de salida. Se compone de un sub-conversor 1410, que corresponde a un conversor de pre-procesado FP a número en coma fija pre-procesado 900 como se discutió con referencia a la Fig. 7. El sub-conversor 1410 recibe el exponente Ex , el bit del signo del número FP ($sign_x$) y la mantisa Mx que comprende $n+m$ bits. Genera un número en coma fija pre-procesado de $n+2$ bits a la

salida. Conectada a la salida de dicho sub-conversor 1410 hay una unidad de redondeo 1415 que incluye un incrementador 1420 similar al sumador 1310a descrito con referencia a la Fig. 11a para incrementar los $n+1$ MSBs si el LSB es uno. La salida del sumador 1420 y por lo tanto de la unidad de redondeo 1415 es un número en coma fija no procesado de $n+1$ bits. En una implementación alternativa el redondeador podría realizar otro tipo de redondeo.

Si el número de bits de la mantisa de entrada es menor que el número de bits de número en coma fija de salida, un conversor de este tipo podría ser idéntico al conversor 1000a descrito en la Fig. 8a

A pesar de que se han descrito aquí sólo algunas realizaciones y ejemplos particulares de la invención, el experto en la materia comprenderá que son posibles otras realizaciones alternativas y/o usos de la invención, así como modificaciones obvias y elementos equivalentes. Además, la presente invención abarca todas las posibles combinaciones de las realizaciones concretas que se han descrito. El alcance de la presente invención no debe limitarse a realizaciones concretas, sino que debe ser determinado únicamente por una lectura apropiada de las reivindicaciones adjuntas.

Por otro lado, las realizaciones descritas de la invención con referencia a los dibujos comprenden sistemas informáticos y procesos realizados en sistemas informáticos, caracterizados a nivel funcional, e independientes del soporte o tecnología empleada para su implementación. Este medio de soporte podría ser, por ejemplo, un circuito integrado para aplicaciones específicas (ASIC, siglas en inglés), un circuito lógico programable (FPGA o CPLD, siglas en inglés) que incluyen una memoria, o cualquier otro dispositivo, estando dichos circuitos adaptados o configurados para realizar, o para usarse en la realización de, los procesos relevantes.

A pesar también de que las realizaciones descritas comprenden dispositivos

- informáticos, la invención también se extiende a programas informáticos, más particularmente a programas informáticos en unos medios portadores, adaptados para llevar a cabo la invención. El programa informático puede estar en forma de código fuente, código objeto o un código intermedio entre
- 5 código fuente y código objeto, tal como en una forma parcialmente compilada, o en cualquier otra forma adecuada para su uso en la implementación de los procesos de acuerdo con la invención. El medio portador puede ser cualquier entidad o dispositivo capaz de portar el programa.
- 10 Por ejemplo, el medio portador puede comprender un medio de almacenamiento, tal como una ROM, por ejemplo un CD ROM o una ROM semiconductora, o un medio de grabación magnético, por ejemplo un floppy disc o un disco duro. Además, el medio portador puede ser un medio portador transmisible tal como una señal eléctrica u óptica que puede transmitirse vía
- 15 cable eléctrico u óptico o mediante radio u otros medios.
- Cuando el programa informático está contenido en una señal que puede transmitirse directamente mediante un cable u otro dispositivo o medio, el medio portador puede estar constituido por dicho cable u otro dispositivo o
- 20 medio.

REIVINDICACIONES

1. Dispositivo para realizar una operación de multiplicación de al menos dos
5 números coma flotante pre-procesados y generar un tercer número coma
flotante pre-procesado, en el que cada número tiene una mantisa de $M+2$
dígitos, el dispositivo comprende:

un camino de datos del exponente y

un camino de datos de la mantisa, que comprende

10 una primera entrada para recibir como mucho los $M+1$ Dígitos
Más Significativos (MSDs) de la mantisa pre-procesada del primer número,

una segunda entrada para recibir como mucho los $M+1$ MSDs
de la mantisa pre-procesada del segundo número,

15 en el que el camino de datos de la mantisa está configurado
para generar como mucho los $M+1$ MSDs de la mantisa del tercer número
pre-procesado, donde el Dígitos Menos Significativo (LSD) de todas las
mantisas pre-procesadas es igual a $B/2$, siendo B la base del sistema de
representación numérica.

20

2. Dispositivo según reivindicación 1, en el que el camino de datos del
exponente está configurado para

calcular el exponente de salida;

calcular el signo de salida; y

25 detectar y resolver excepciones y valores especiales.

3. Dispositivo según reivindicación 1 ó 2, en el que dichas mantisas pre-procesadas están normalizadas y dichas primera y segunda entradas están configuradas para recibir los m MSDs fraccionarios de la primera y segunda mantisa pre-procesada, respectivamente.

5

4. Dispositivo según cualquiera de las reivindicaciones 1 a 3, en el que comprende además una tercera entrada para recibir el LSD de dicha primera y segunda mantisa pre-procesada.

10 5. Dispositivo según cualquiera de las reivindicaciones 1 a 3, en el que además comprende una tercera entrada con el valor $B/2$.

6. Dispositivo según cualquiera de las reivindicaciones 1 a 5, donde $B=2$ y los dígitos son bits.

15

7. Dispositivo según reivindicación 6, en el que el camino de datos de la mantisa comprende

un módulo de multiplicación en coma fija preparado para recibir, en una primera y una segunda entrada, como mucho los $M+1$ MSBs de la mantisa del primer y segundo número pre-procesado, respectivamente, y generar los $M+2$ MSBs del valor que corresponde a la operación de multiplicación entre dichas mantisas pre-procesadas.

20

8. Dispositivo según reivindicación 7, en el que el módulo de multiplicación en coma fija comprende

25

un multiplicador redundante configurado para recibir, en una primera y una segunda entrada, como mucho los $M+1$ MSBs de la mantisa del primer y segundo número pre-procesado, respectivamente, y generar, en un formato

de representación redundante, los $2M+3$ MSDs del valor correspondiente a la operación de multiplicación entre dichas mantisas pre-procesadas;

5 un módulo de conversión, conectado a la salida de dicho módulo de multiplicación, configurado para recibir los $M+2$ MSDs de la salida de dicho multiplicador redundante y un bit de acarreo, y generar una salida de $M+2$ bits correspondiente a la conversión del valor redundante recibido a formato de representación no redundante;

10 un módulo de red de acarreo configurado para recibir los $M+1$ LSDs de la salida de dicho multiplicador redundante y generar dicho bit de acarreo correspondiente al acarreo de salida de la conversión de los $M+1$ LSDs de la salida de dicho multiplicador redundante a representación no redundante.

9. Dispositivo según reivindicación 8, en el que el multiplicador redundante comprende

15 un generador de productos parciales configurado para recibir, en una primera y una segunda entrada, como mucho los $M+1$ MSBs de la mantisa del primer y segundo número pre-procesado, respectivamente, y generar sus productos parciales en una salida

20 un árbol de compresores, con una primera entrada conectada a la salida del generador de productos parciales y una segunda entrada configurada para recibir como mucho los $M+1$ MSBs de la mantisa del primer y segundo número pre-procesado pre-procesa, dicho árbol de compresores configurado para generar, en una representación redundante, los $2M+3$ MSDs de un valor correspondiente a la operación de multiplicación entre dichas
25 mantisas pre-procesadas en una salida.

10. Dispositivo según cualquiera de las reivindicaciones 7 a 9 en el que el módulo de multiplicación en punto fijo comprende además una tercera entrada con el valor uno.

5 11. Dispositivo según cualquiera de las reivindicaciones 7 a 10 en el que comprende además un módulo de normalización con una entrada conectada a la salida del módulo de multiplicación en coma fija, donde el módulo de normalización está configurado para generar como mucho los M+1 MSBs de la tercera mantisa pre-procesada seleccionando los M+1 LSBs de su entrada si el MSB es igual a cero o los M+1 MSBs si dicho bit es igual a uno.

10

12. Dispositivo según cualquiera de las reivindicaciones anteriores, en el que comprende además un conversor de números coma fija pre-procesados a números coma flotante pre-procesados para convertir un número coma fija de N+2 bits a un número coma flotante con una mantisa de M+2 bits.

15

13. Dispositivo según reivindicación 12 en el que dicho conversor de números coma fija pre-procesados a números coma flotante pre-procesados comprende:

un calculador de cantidad de desplazamiento,

20

un módulo para calcular el exponente, con una primera entrada para recibir la primera cantidad de desplazamiento del calculador de cantidad de desplazamiento, y una salida para generar el exponente del número coma flotante pre-procesado; y

un módulo de normalización con

25

una primera entrada para recibir los N MSBs de los N+1 LSBs del número coma fija pre-procesado y una segunda para recibir la primera cantidad de desplazamiento; dicho módulo de normalización configurado para desplazar a la izquierda dichos N MSBs de acuerdo con dicha cantidad de desplazamiento, completando el MSB de las posiciones vacantes con cero

y el resto con unos, o el MSB con uno y el resto con ceros, para generar como mucho los M+1 MSBs de la mantisa,

mientras que el signo del número coma flotante pre-procesado corresponde al MSB del número coma fija pre-procesado.

5

14. Dispositivo según reivindicación 13 en el que el módulo de normalización está configurado además para, completar dichas posiciones vacantes, aleatoriamente, basándose en un bit seleccionado, o en una combinación de bits seleccionados.

10

15. Dispositivo según reivindicación 13 ó 14, en el que dicho módulo de normalización está configurado además para generar selectivamente el complemento a uno del resultado de dicho desplazamiento.

15

16. Dispositivo según cualquiera de las reivindicaciones anteriores, en el que comprende además un conversor de números coma fija no procesados a números coma flotante pre-procesados, para convertir un número coma fija no procesado de R bits a un número coma flotante pre-procesado con una mantisa de M+2 bits. El conversor comprende:

20

un calculador de cantidad de desplazamiento

un módulo de normalización configurado para recibir los R bits del número en coma fija no procesado y generar como mucho los M+1 MSBs de la mantisa del número pre-procesado en coma flotante,

25

un calculador de exponentes con una primera entrada para recibir la segunda cantidad de desplazamiento proveniente del calculador de cantidad de desplazamiento y una salida para generar el exponente del número pre-procesado en coma flotante,

en el que el signo del número pre-procesado en coma flotante se corresponde con el MSB del número en coma fija no procesado.

17. Dispositivo según la reivindicación 16, en el que el módulo de normalización comprende una primera entrada para recibir los R bits del número no procesado en coma fija y una segunda entrada para recibir la segunda cantidad de desplazamiento, donde el módulo de normalización está configurado para generar un valor que corresponde a como mucho los M+1 MSB de la mantisa pre-procesada mediante el desplazamiento a la izquierda de los R-2 MSBs de los R-1 LSBs de la primera entrada seguida hacia la derecha por un bit a cero y rellenando las posiciones vacantes con el valor del LSB de la primera entrada.

18. Dispositivo según la reivindicación 17, en el que el módulo de normalización está configurado además para generar selectivamente el complemento a uno de dicho valor generado si la entrada es negativa.

19. Dispositivo según cualquiera de las reivindicaciones 14,15,17,ó 18, en el que el módulo de normalización comprende un desplazador variable configurado para recibir un bit para completar las posiciones vacantes.

20. Dispositivo según la reivindicación 19, en el que dicho desplazador variable comprende un número de sucesivos multiplexores que es igual al primer entero mayor o igual que el logaritmo en base 2 de la máxima cantidad de desplazamiento $\lceil \log_2(\text{máxima cantidad de desplazamiento}) \rceil$, con cada multiplexor configurado para efectuar una operación de desplazamiento a la izquierda de 2^i posiciones, $i \in [0, \text{número de multiplexores}-1]$, y cada multiplexor configurado para completar las posiciones vacantes usando el valor de dicho bit recibido.

21. Dispositivo según la reivindicación 16, en el que el módulo de normalización comprende una primera entrada para recibir los R bits del número en coma fija no procesado y una segunda entrada para recibir la
5 segunda cantidad de desplazamiento, donde el módulo de normalización está configurado para generar un valor que se corresponde con como mucho los M+1 MSBs de la mantisa pre-procesada mediante el desplazamiento a la izquierda de los R-1 LSBs de la primera entrada.
- 10 22. Dispositivo según la reivindicación 21, en el que el módulo de normalización está configurado además para generar selectivamente el complemento a uno del resultado de la mencionada operación de desplazamiento.
- 15 23. Dispositivo según a cualquiera de las reivindicaciones 13 a 22, en el que el calculador de exponentes está configurado para decrementar, de acuerdo a la segunda cantidad de desplazamiento, un valor base para obtener el exponente.
- 20 24. Dispositivo según la reivindicación 23, en el que el calculador de exponentes además está configurado para detectar desbordamientos o valores cero y provocar que el conversor genere la salida correspondiente.
- 25 25. Dispositivo según cualquiera de las reivindicaciones anteriores, en el que comprende además un conversor de números coma flotante pre-procesados a números coma fija no procesados para convertir el tercer número en coma flotante pre-procesado a un número en coma fija no procesado.

26. Dispositivo según la reivindicación 25, en el que cuando el número en coma fija no procesado tiene $H+1$ bits, el conversor comprende un conversor de números coma flotante pre-procesados a números coma fija pre-procesados con una salida de $H+2$ bits conectada a un módulo de redondeo.

5

27. Dispositivo según reivindicación 26, en el que el módulo de redondeo comprende un sumador; dicho sumador está configurado para recibir, en una entrada, los $H+1$ MSBs de la salida del mencionado conversor de números coma flotante pre-procesados a números coma fija pre-procesados e
10 incrementar dicha entrada si el LSB de dicha salida es igual a 1.

28. Dispositivo según cualquiera de las reivindicaciones anteriores, que comprende además un conversor de números coma flotante pre-procesados a números coma flotante pre-procesados para convertir un número inicial
15 coma flotante con una mantisa de $J+2$ bits a un subsecuente número coma flotante, donde dicho subsecuente número coma flotante tiene, al menos, un tamaño de mantisa diferente.

29. Dispositivo según la reivindicación 28, en el que cuando el subsecuente
20 número en coma flotante pre-procesado tiene una mantisa con $J+2-P$ bits, $P < J+1$, entonces el conversor comprende:

una unidad de redondeo para eliminar los $P+1$ LSBs de los $J+2$ bits de la mantisa inicial pre-procesada, para generar como mucho $J+1-P$ MSBs de la mantisa del subsecuente número en coma flotante pre-procesado,

25 donde el LSB de la mantisa del subsecuente número coma flotante pre-procesado es igual a 1,

y un calculador de exponentes para generar el exponente del subsecuente número en coma flotante pre-procesado.

30. Dispositivo según la reivindicación 28, en el que cuando el subsecuente número en coma flotante pre-procesado tiene una mantisa con $J+2+Q$ bits, entonces el conversor comprende:

- 5 un módulo de rellenado, configurado para recibir como mucho los $J+1$ MSBs de la mantisa del número en coma flotante pre-procesado inicial y generar como mucho los $J+Q+1$ MSBs de la mantisa del subsecuente número en coma flotante pre-procesado fijando el MSB de los Q LSBs a uno o a cero y los restante $Q-1$ bits de dichos Q LSBs al complemento del mencionado
- 10 MSB, mientras los como mucho $J+1$ MSBs de la mantisa del subsecuente número en coma flotante pre-procesado son los mismos que los como mucho $J+1$ MSBs de la mantisa del número en coma flotante pre-procesado inicial, y
- un calculador de exponentes para generar el exponente del
- 15 subsecuente número en coma flotante pre-procesado.

31. Dispositivo según la reivindicación 30, en el que el módulo de rellenado está configurado para fijar aleatoriamente dicho MSB basándose en el valor de un bit seleccionado, o de una combinación de bits seleccionados.

20

32. Dispositivo según cualquiera de las reivindicaciones anteriores, en el que comprende además un conversor de números coma flotante pre-procesados a números coma fija pre-procesados para convertir un número en coma flotante con una mantisa de $F+2$ bits en un número en coma fija.

25

33. Dispositivo según la reivindicación 32, en el que el número en coma fija pre-procesado tiene L bits, con $L < F+4$, el conversor comprende:

un calculador de la cantidad de desplazamiento que recibe el exponente del número en coma flotante pre-procesado en una entrada y que genera una cantidad de desplazamiento en una salida,

5 un módulo de desplazamiento con una primera entrada para recibir los L-1 MSBs de la mantisa del número en coma flotante pre-procesado y una segunda entrada acoplada a la salida del calculador de cantidad de desplazamiento y una tercera entrada para recibir el signo del mencionado número en coma flotante, para generar los L-1 MSBs del número en coma fija pre-procesado en una salida.

10

34. Dispositivo según la reivindicación 33, en el que el módulo de desplazamiento comprende un desplazador aritmético a la derecha acoplado a un inversor de bit condicional.

15 35. Dispositivo según reivindicación 32, en el que cuando el número en coma fija pre-procesado comprende $F+C+3$ bits, $C>0$, el conversor comprende:

un calculador de cantidad de desplazamiento que recibe el exponente del número en coma flotante pre-procesado en una entrada y que genera una cantidad de desplazamiento en una salida,

20 un módulo de desplazamiento aritmético a la derecha con una primera entrada conectada a la salida del calculador de desplazamiento, configurado para generar los $F+C+2$ MSBs del número en coma fija pre-procesado mediante el desplazamiento aritmético a la derecha de un valor intermedio de $F+C+2$ bits formado, de izquierda a derecha, por el bit de signo, los $F+1$ MSBs de la mantisa del número en coma flotante pre-procesado, y el MSB
25 de los C LSBs puesto a cero y el resto a uno, o el MSB de los C LSBs puesto a uno y el resto a cero.

36. Dispositivo según la reivindicación 35, en el que el módulo de desplazamiento aritmético a la derecha está configurado para poner aleatoriamente dicho MSB de los C LSBs del mencionado valor intermedio de $F+C+2$ bits en base al valor de un bit seleccionado, o de una combinación de bits seleccionados.

37. Dispositivo según las reivindicaciones 35 o 36, en el que el módulo de desplazamiento aritmético a la derecha está configurado para generar selectivamente el complemento a uno del resultado de la mencionada operación de desplazamiento.

38. Dispositivo según cualquiera de las reivindicaciones anteriores, comprende además un conversor de números en coma flotante no procesados a números en coma flotante pre-procesados para convertir un número en coma flotante no procesado con una mantisa de $E+2$ bits en un número en coma flotante pre-procesado.

39. Dispositivo según la reivindicación 38, en el que cuando el número en coma flotante pre-procesado tiene una mantisa de $E+2-D$ bits, $D < E+1$ entonces el conversor comprende:

una unidad de redondeo configurada para eliminar los $D+1$ LSBs de la mantisa del número en coma flotante no procesado, para generar como mucho los $E+1-D$ MSBs de la mantisa del número coma flotante pre-procesado, donde el LSB de la mantisa del número en coma flotante pre-procesado es igual a uno, y

un calculador de exponentes para generar el exponente del número en coma flotante pre-procesado.

40. Dispositivo según la reivindicación 39, en el que la unidad de redondeo está configurada además para, selectivamente, poner a cero el segundo LSB de la mantisa del número en coma flotante pre-procesado si todos los $D+1$ LSBs de la mantisa del número en coma flotante no procesado son iguales a
5 cero.

41. Dispositivo según la reivindicación 38, en el que cuando el número en coma flotante pre-procesado tiene una mantisa de $E+2+G$ bits entonces el conversor comprende:

10 un módulo de rellenado, configurado para recibir como mucho los $E+2$ bits de la mantisa del número en coma flotante no procesado y generar como mucho los $E+G+1$ MSBs de la mantisa del número en coma flotante pre-procesado fijando como mucho los $E+2$ MSBs del número en coma flotante pre-procesado al mismo valor que como mucho los $E+2$ bits de la mantisa del
15 número en coma flotante no procesado y los restantes bits a cero, donde el LSB de la mantisa del número en coma flotante pre-procesado es igual a uno, y

 un calculador de exponentes configurado para generar el exponente del número en coma flotante pre-procesado.

20

42. Dispositivo según la reivindicación 41, en el que el módulo de rellenado está configurado además para generar selectivamente el valor correspondiente a restar uno del segundo LSB de la mencionada mantisa generada, cuando un bit seleccionado, o una combinación de bit
25 seleccionados, de la mantisa no procesada de entrada es igual a uno.

43. Dispositivo según cualquiera de las reivindicaciones anteriores, comprende además un conversor de números coma flotante pre-procesados a números coma flotante no procesados para la conversión de un número en

coma flotante pre-procesado con una mantisa de $U+2$ bits en un número en coma flotante no procesado.

44. Dispositivo según la reivindicación 43, en el que cuando el número en coma flotante no procesado tiene una mantisa de $U+2-V$ bits, $V < U$, entonces el conversor comprende:

un módulo de redondeo, configurado para recibir como mucho los $U+3-V$ MSBs de la mantisa del número en coma flotante pre-procesado y generar como mucho los $U+2-V$ bits de la mantisa del número en coma flotante no procesado,

un calculador de exponentes configurado para generar el exponente del número en coma flotante no procesado.

45. Dispositivo según la reivindicación 44, en el que el módulo de redondeo comprende un sumador; dicho sumador está configurado para recibir, en una entrada, como mucho los $U+2-V$ MSBs de la mantisa del número en coma flotante pre-procesado e incrementar dicho valor de entrada si el $(U+3-V)$ -ésimo MSB de dicha mantisa es igual a 1, y generar una instrucción para el calculador de exponentes, si se produjera un desbordamiento.

20

46. Dispositivo según las reivindicaciones 44 ó 45, en el que el calculador de exponentes está configurado, además, para incrementar el exponente de salida cuando se genera la mencionada instrucción del módulo de redondeo.

47. Dispositivo según la reivindicación 43, en el que cuando el número en coma flotante no procesado tiene una mantisa de $U+2+W$ bits entonces el conversor comprende:

un módulo de rellenado, configurado para recibir como mucho los $U+1$ MSBs de la mantisa del número en coma flotante pre-procesado y generar como mucho los $U+W+2$ bits de la mantisa del número en coma flotante no procesado poniendo el MSB de los $W+1$ LSBs a uno y los restantes bits a
5 ceros, y

un calculador de exponentes configurado para generar el exponente del número en coma flotante pre-procesado.

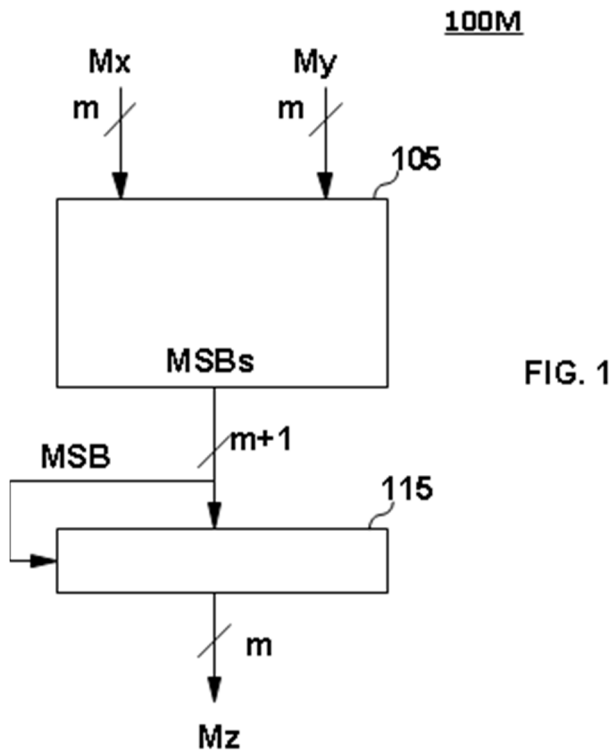


FIG. 1b

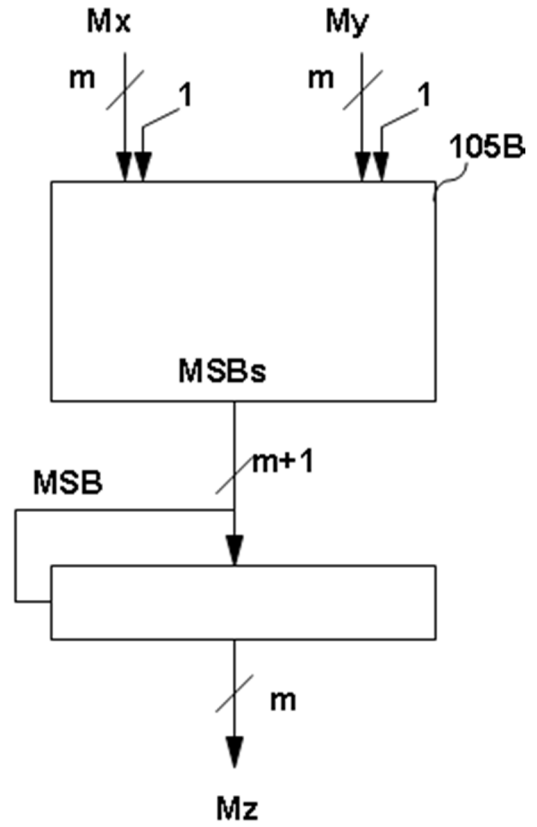
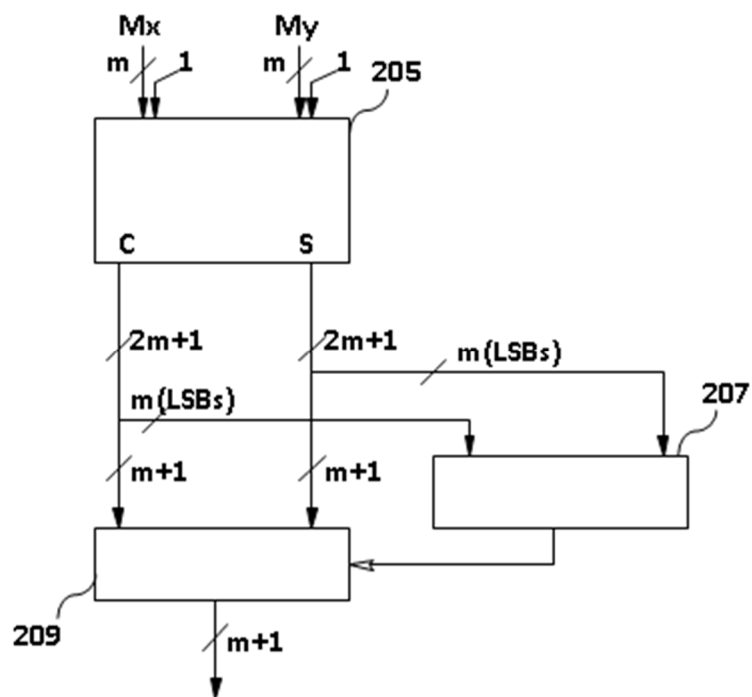


FIG. 2



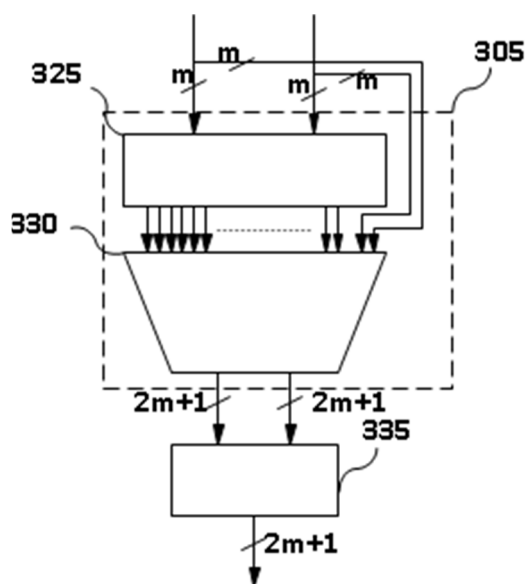


FIG. 3a

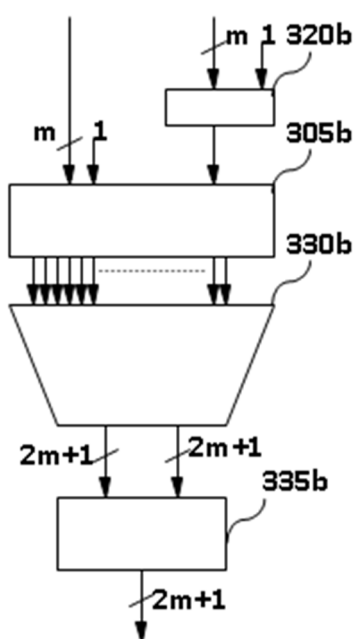
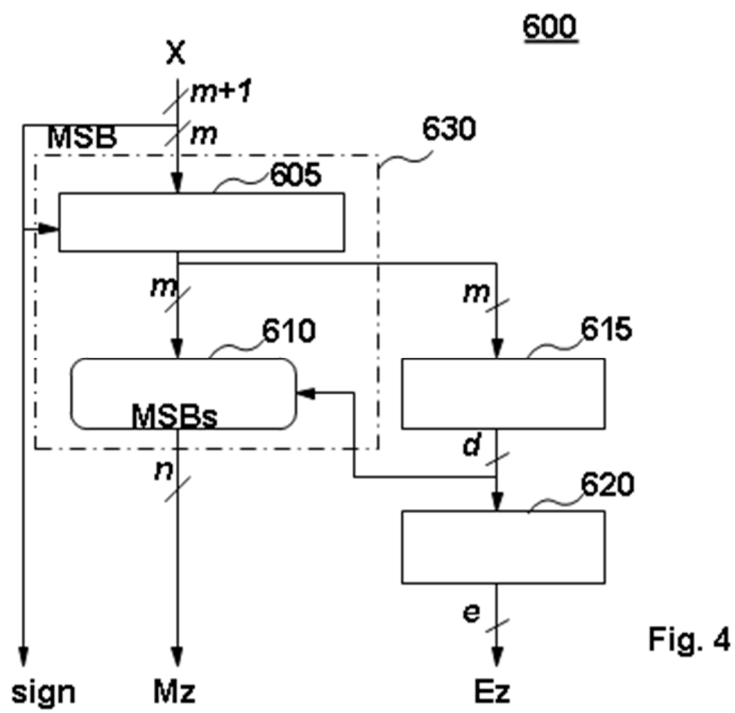


FIG. 3b



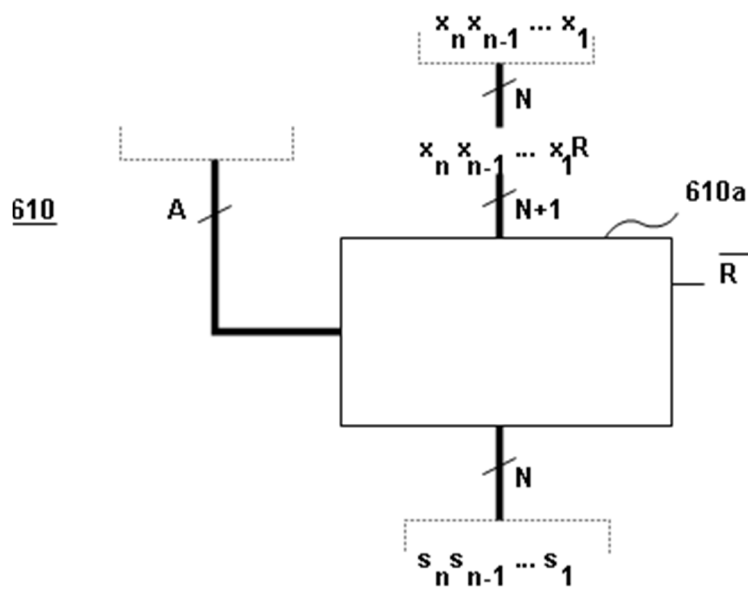
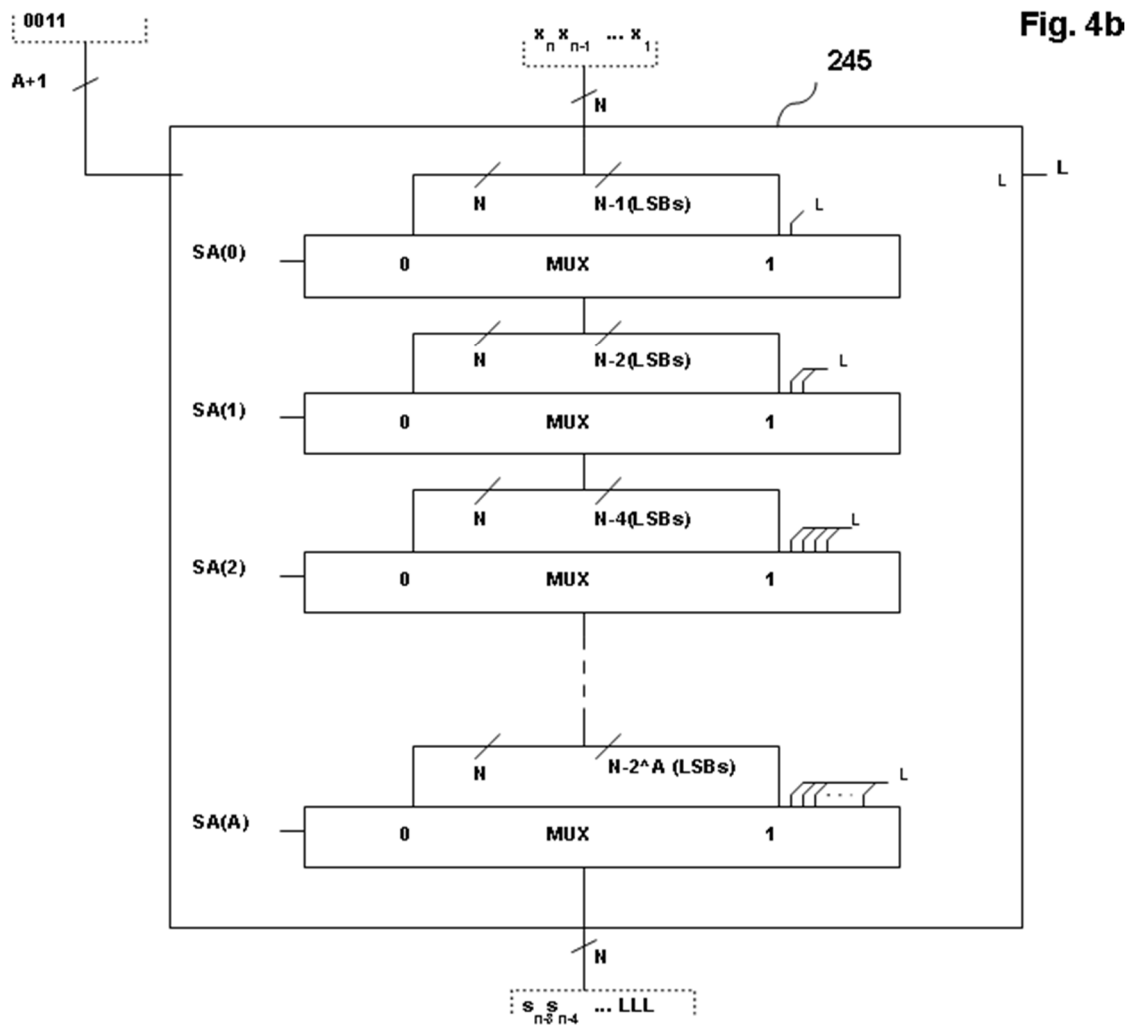
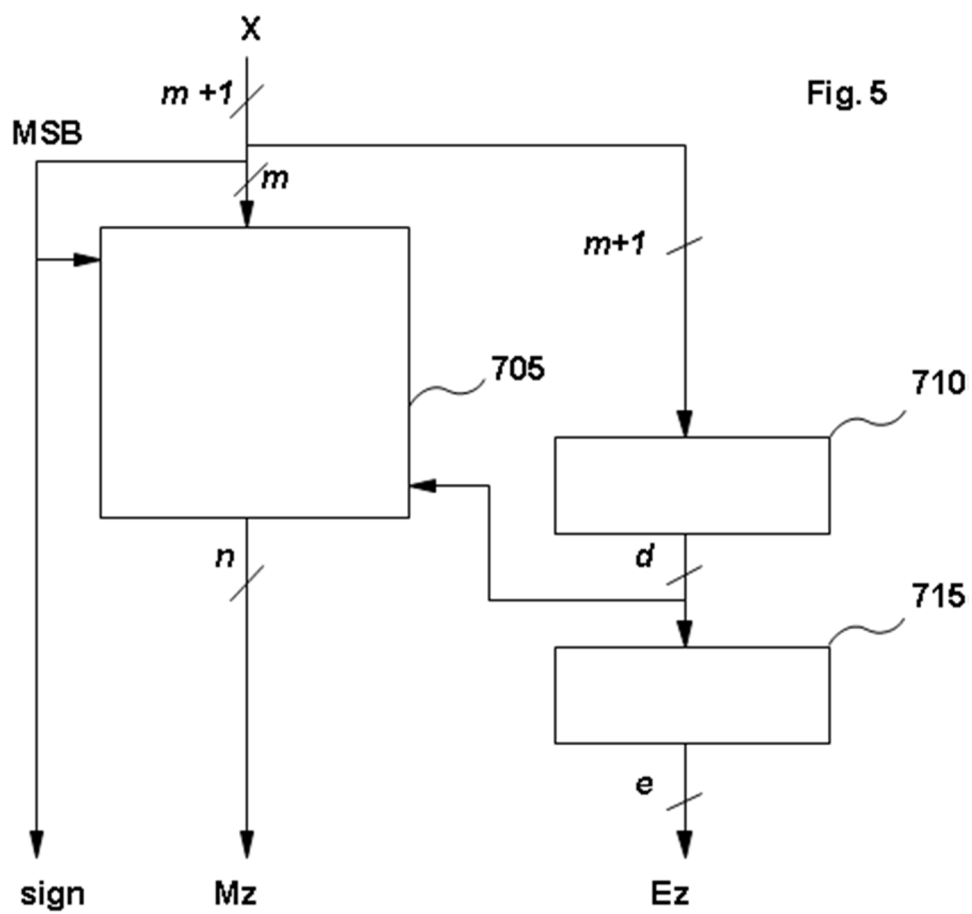
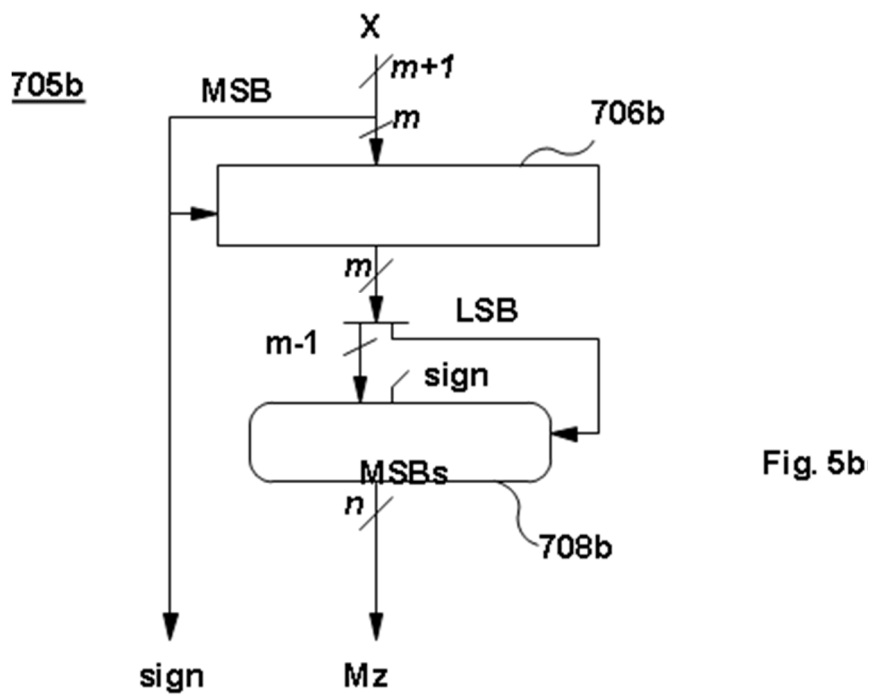
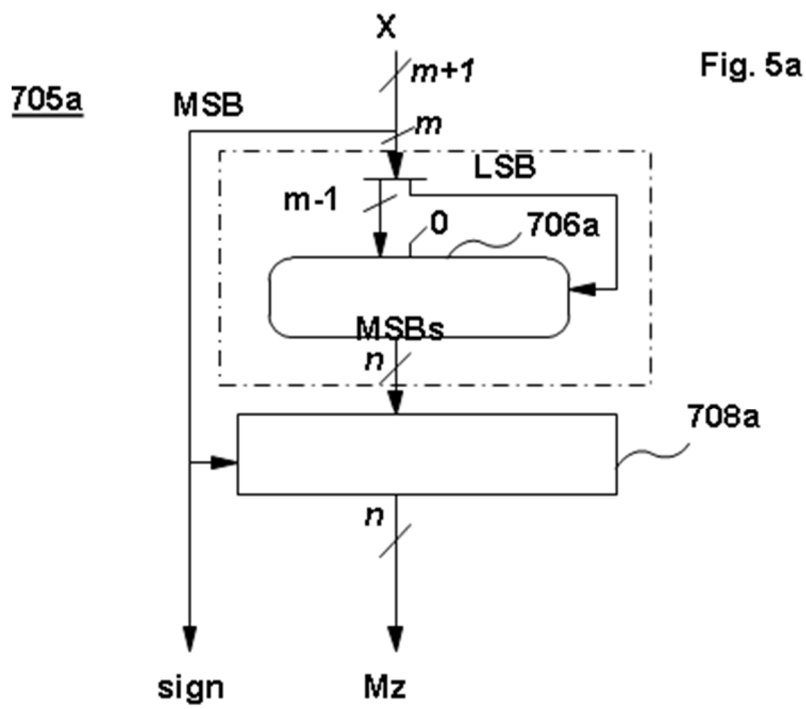


Fig. 4a

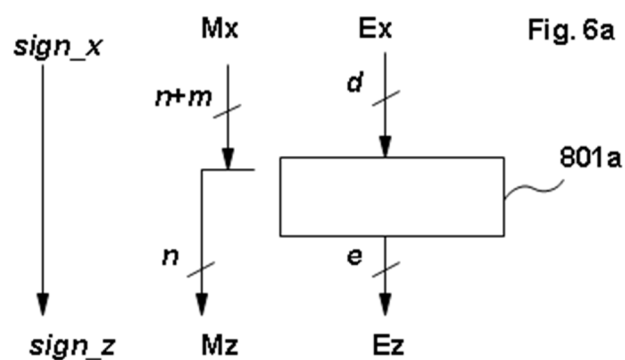


700

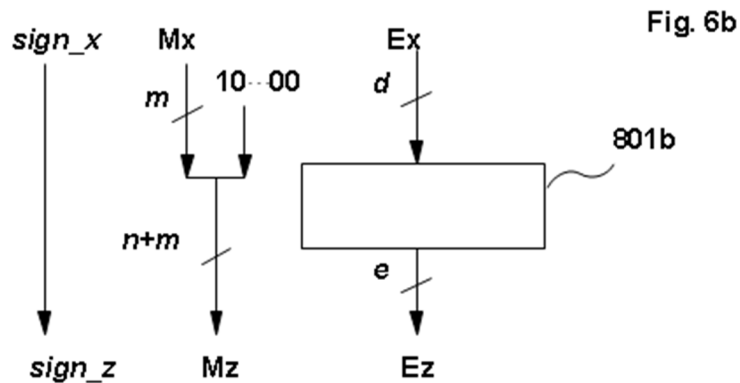




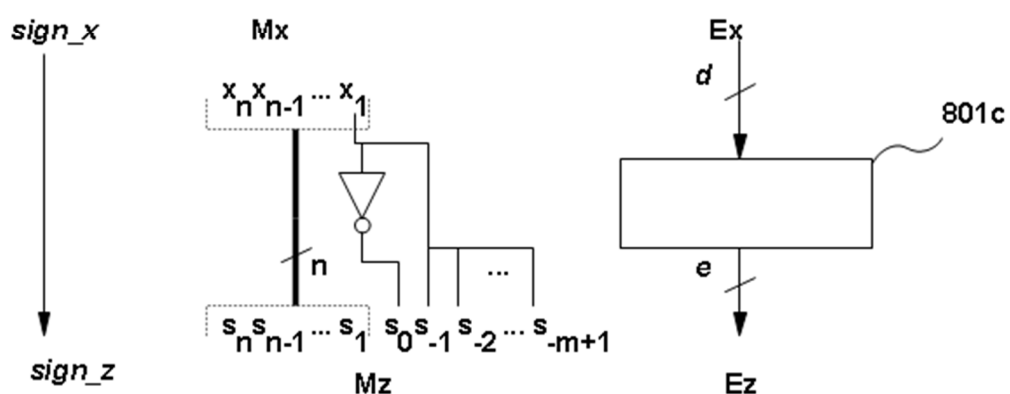
800a

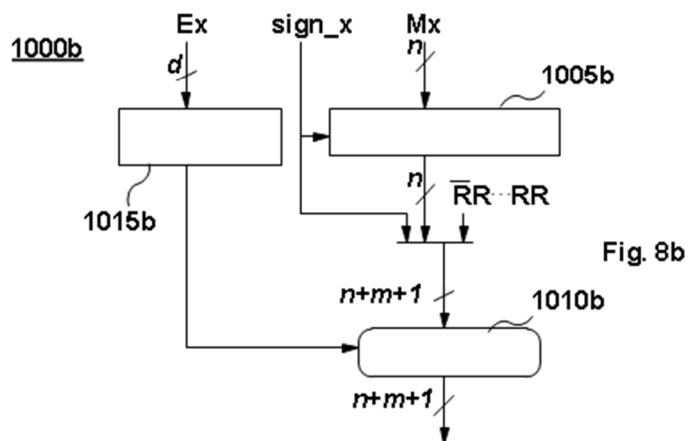
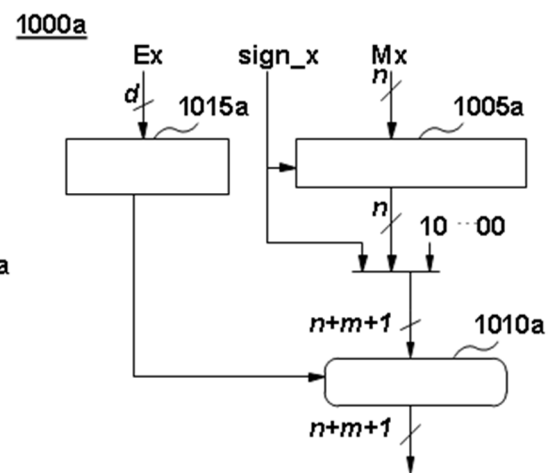
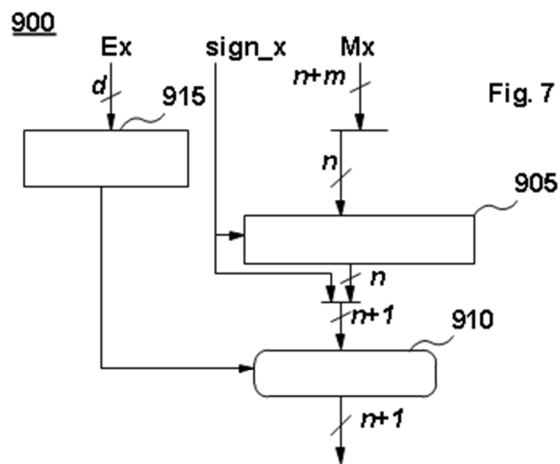


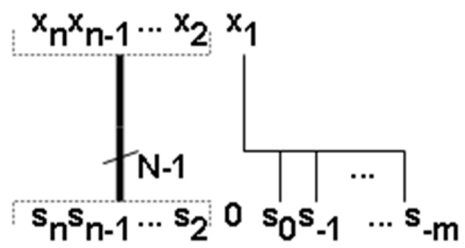
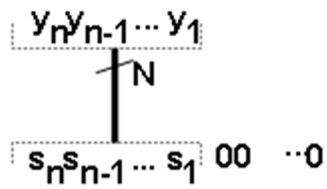
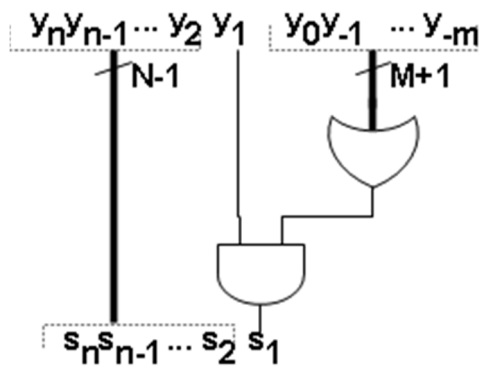
800b



800c







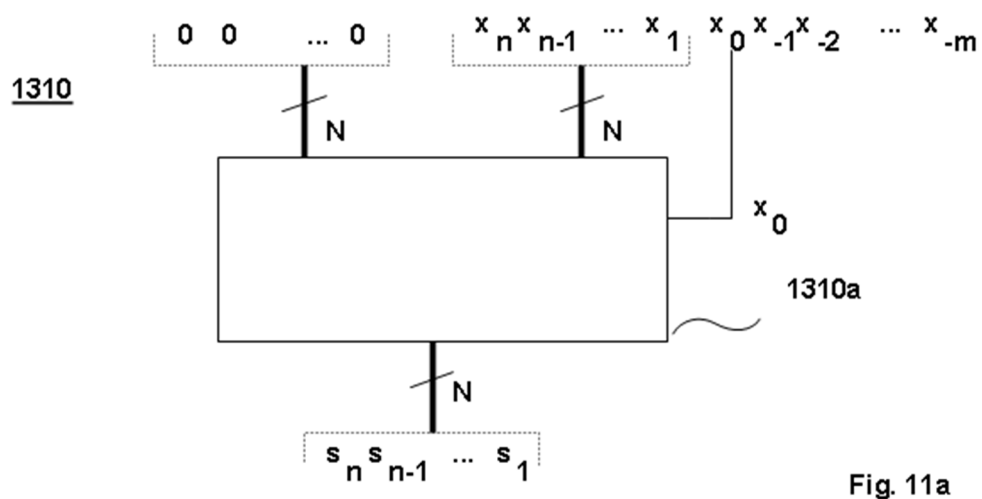
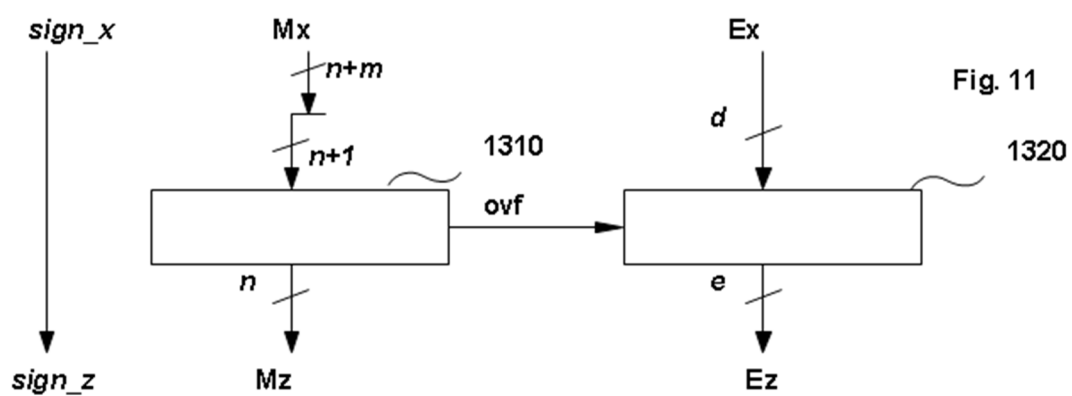
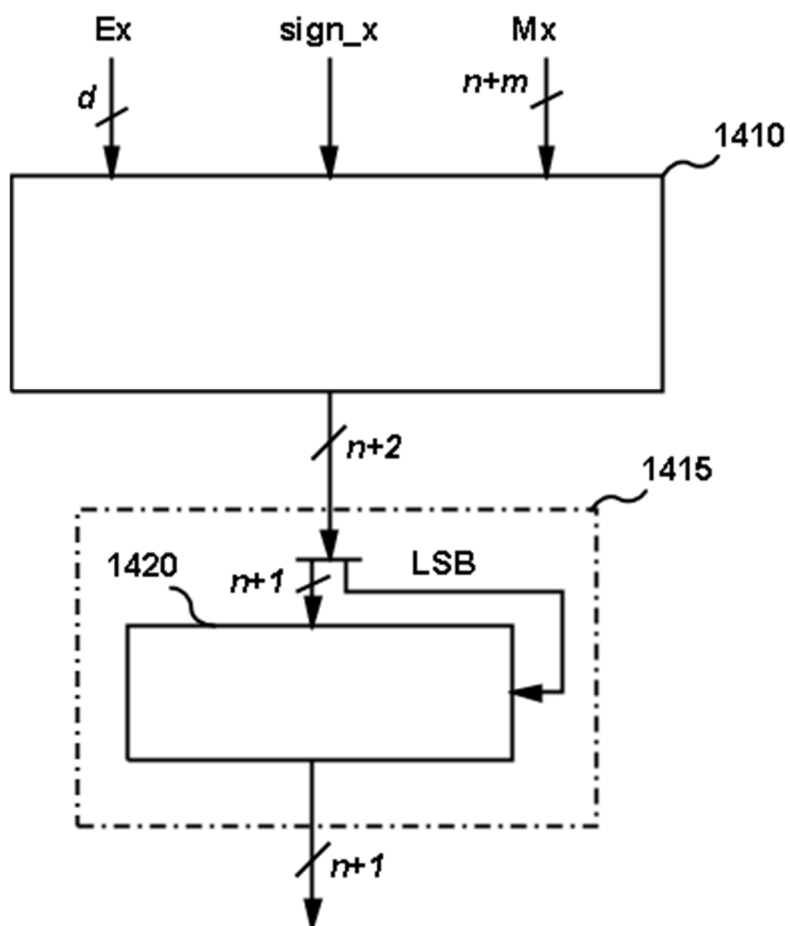


Fig. 12





- ②① N.º solicitud: 201430453
②② Fecha de presentación de la solicitud: 28.03.2014
③② Fecha de prioridad:

INFORME SOBRE EL ESTADO DE LA TECNICA

⑤① Int. Cl.: **G06F7/38** (2006.01)

DOCUMENTOS RELEVANTES

Categoría	⑤⑥ Documentos citados	Reivindicaciones afectadas
A	US 2010125621 A1 (OLIVER DAVID S et al.) 20.05.2010, todo el documento.	1
A	SOMSUBHRA GHOSH et al. FPGA based implementation of a double precision IEEE floating-point adder. Intelligent Systems and Control (ISCO), 2013 7th International Conference on, 20130104 IEEE 04.01.2013 VOL: Págs: 271-275 ISBN 978-1-4673-4359-6; ISBN 1-4673-4359-5 doi:10.1109/ISCO.2013.6481161. Todo el documento.	1
A	US 5408426 A (TAKEWA HIDEHITO et al.) 18.04.1995	1
A	CATANZARO B et al. Higher Radix Floating-Point Representations for FPGA-Based Arithmetic.Field-Programmable Custom Computing Machines, 2005. FCCM 2005. 13 th Annual IEEE Symposium on Napa, CA, USA 18-20 Abril 2005, 20050418; 20050418-20050420 Piscataway, NJ, USA, IEEE 18.04.2005 VOL: Págs: 161-170 ISBN 978-0-7695-2445-0; ISBN 0-7695-2445-1 Doi: doi:10.1109/FCCM.2005.43.	1
A	LIBO HUANG et al. A New Architecture For Multiple-Precision Floating-Point Multiply-Add Fused Unit Design.Computer Arithmetic, 2007. ARITH '07. 18th IEEE Symposium on, 20070601 IEEE, Pi 01.06.2007 VOL: Págs: 69-76 ISBN 978-0-7695-2854-0; ISBN 0-7695-2854-6 Anonymous.	1
A	PARHAMI B On producing exactly rounded results in digit-serial on-line arithmetic. Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference on Oct. 29 - Nov. 1, 2000, 20001029 Piscataway, NJ, USA,IEEE 29.10.2000 VOL: Págs: 889-893 vol. 2 ISBN 978-0-7803-6514-8; ISBN 0-7803-6514-3 Doi: doi:10.1109/ACSSC.2000.910641.	1

Categoría de los documentos citados

X: de particular relevancia

Y: de particular relevancia combinado con otro/s de la misma categoría

A: refleja el estado de la técnica

O: referido a divulgación no escrita

P: publicado entre la fecha de prioridad y la de presentación de la solicitud

E: documento anterior, pero publicado después de la fecha de presentación de la solicitud

El presente informe ha sido realizado

☒ para todas las reivindicaciones

☐ para las reivindicaciones nº:

Fecha de realización del informe
05.01.2015

Examinador
M. Muñoz Sánchez

Página
1/4

Documentación mínima buscada (sistema de clasificación seguido de los símbolos de clasificación)

G06F

Bases de datos electrónicas consultadas durante la búsqueda (nombre de la base de datos y, si es posible, términos de búsqueda utilizados)

INVENES, EPODOC, WPI, NPL, XPIEE, XPI3E

Fecha de Realización de la Opinión Escrita: 05.01.2015

Declaración

Novedad (Art. 6.1 LP 11/1986)	Reivindicaciones 1-47	SI
	Reivindicaciones	NO
Actividad inventiva (Art. 8.1 LP11/1986)	Reivindicaciones 1-47	SI
	Reivindicaciones	NO

Se considera que la solicitud cumple con el requisito de aplicación industrial. Este requisito fue evaluado durante la fase de examen formal y técnico de la solicitud (Artículo 31.2 Ley 11/1986).

Base de la Opinión.-

La presente opinión se ha realizado sobre la base de la solicitud de patente tal y como se publica.

1. Documentos considerados.-

A continuación se relacionan los documentos pertenecientes al estado de la técnica tomados en consideración para la realización de esta opinión.

Documento	Número Publicación o Identificación	Fecha Publicación
D01	US 2010125621 A1 (OLIVER DAVID S et al.)	20.05.2010
D02	SOMSUBHRA GHOSH et al. FPGA based implementation of a double precision IEEE floating-point adder. Intelligent Systems and Control (ISCO), 2013 7th International Conference on, 20130104 IEEE 04.01.2013 VOL: Págs: 271-275 ISBN 978-1-4673-4359-6; ISBN 1-4673-4359-5 doi:10.1109/ISCO.2013.6481161. Todo el documento.	04.01.2013
D03	US 5408426 A (TAKEWA HIDEHITO et al.)	18.04.1995
D04	CATANZARO B et al. Higher Radix Floating-Point Representations for FPGA-Based Arithmetic. Field-Programmable Custom Computing Machines, 2005. FCCM 2005. 13th Annual IEEE Symposium on Napa, CA, USA 18-20 Abril 2005, 20050418; 20050418-20050420 Piscataway, NJ, USA, IEEE 18.04.2005 VOL: Págs: 161-170 ISBN 978-0-7695-2445-0; ISBN 0-7695-2445-1 doi:10.1109/FCCM.2005.43.	18.04.2005
D05	LIBO HUANG et al. A New Architecture For Multiple-Precision Floating-Point Multiply-Add Fused Unit Design. Computer Arithmetic, 2007. ARITH '07. 18th IEEE Symposium on, 20070601 IEEE, Pi 01.06.2007 VOL: Págs: 69-76 ISBN 978-0-7695-2854-0; ISBN 0-7695-2854-6 Anonymous	01.06.2007
D06	PARHAMI B On producing exactly rounded results in digit-serial on-line arithmetic. Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference on Oct. 29 - Nov. 1, 2000, 20001029 Piscataway, NJ, USA, IEEE 29.10.2000 VOL: Págs: 889-893 vol. 2 ISBN 978-0-7803-6514-8; ISBN 0-7803-6514-3 doi:10.1109/ACSSC.2000.910641	29.10.2000

2. Declaración motivada según los artículos 29.6 y 29.7 del Reglamento de ejecución de la Ley 11/1986, de 20 de marzo, de Patentes sobre la novedad y la actividad inventiva; citas y explicaciones en apoyo de esta declaración

Se considera D01 el documento más próximo del estado de la técnica al objeto de la solicitud.

Reivindicaciones independientes

Reivindicación 1: El documento D01, divulga una unidad de cómputo aritmético para realizar operaciones de suma o multiplicación de coma flotante con caminos de datos para la mantisa y el exponente respectivo. Los operandos tienen un bit de valor implícito 1 (el más significativo). Los resultados se redondean y normalizan.

La diferencia entre el documento D01 y la reivindicación 1 es que el bit implícito es el menos significativo y su efecto técnico es la simplificación de los cálculos de redondeo y truncamiento. El problema técnico objetivo consistiría así en cómo simplificar los cálculos habituales que se realizan en las operaciones de suma o resta.

El documento D02 por su parte divulga una implementación de un sumador de coma flotante en el que el bit implícito es el más significativo. La suma se realiza tras el preprocesamiento de los operandos. En este documento tampoco se recoge la diferencia mencionada en el análisis del documento D01 por lo que la reivindicación 1 posee actividad inventiva según el art. 8.1 de la Ley de Patentes.

Reivindicaciones dependientes

Reivindicaciones 2-47: estas reivindicaciones poseen actividad inventiva según el art. 8.1 de la Ley de Patentes porque dependen de la reivindicación 1 que, como se ha mencionado, también la tiene.