(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 546 916**

(21) Número de solicitud: 201430451

(51) Int. CI.:

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:

ZEA CHECA, Bernabé

(54) Título: **Sumadores coma flotante y conversores**

(57) Resumen:

Dispositivos para realizar una operación deseada de suma o resta 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.

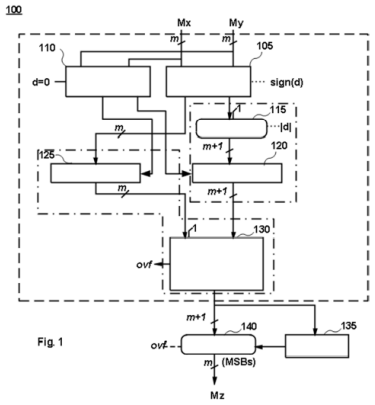


Fig. 1

ES 2 546 916 A1

DESCRIPCIÓN

Sumadores coma flotante y conversores

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

ESTADO DE LA TÉCNICA

En los sistemas de procesado 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 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 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 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) 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.

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 eficiencia tales como precisión, rango, velocidad, área y consumo. En consecuencia, el formato utilizado en estos sistemas influye enormemente en su eficiencia.

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 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.

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 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.

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 empate, el valor par. Usando este tipo de redondeo, se obtiene un error inferior a +-0.5ULP 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 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 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 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.

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 típicamente en la vía crítica.

En el estado dela 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 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 complemento a la base.

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

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

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.

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 suma o resta de al menos dos números en coma flotante pre-procesados y generar un tercer número en coma flotante pre-procesado. Cada número podría tener una mantisa de m+2 dígitos. El dispositivo podría comprender un camino de datos del exponente y un camino de datos de la mantisa. El camino de datos de la mantisa podría comprender una primera entrada para recibir como mucho los m+1 Dígitos Más Significativos (MSDs) de la mantisa del primer número pre-procesado y una segunda entrada para recibir como mucho los m+1 MSDs de la mantisa del segundo número pre-procesado. El camino de datos de la mantisa podría estar configurado para generar como mucho los m+1 MSDs de la mantisa del tercer número pre-procesado. 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 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 digito:

- 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.

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 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, si la alineación es necesaria antes de la suma, el sticky bit es siempre uno, puesto que el último dígito oculto de dicha suma siempre es necesariamente B/2 (dígito KD). Esto es una ventaja para el redondeo y para cuando la operación efectiva es una resta. Por último, otra ventaja es que no puede ocurrir desbordamiento después del redondeo.

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.

En algunas realizaciones, el camino de datos del exponente podría estar configurado para definir la operación efectiva entre las mantisas según la

operación de coma flotante deseada y los signos de las entradas. Además, puede configurarse para detectar el número coma flotante con el mayor exponente, y generar una primera cantidad de desplazamiento para alinear las mantisas de entrada. También se puede configurar para calcular el exponente de la salida y el signo de la salida. Finalmente, se puede configurar para detectar valores especiales de las entradas, como cero, infinito, "no es un número" o números des-normalizados, y dar la orden al sumador para producir el resultado correspondiente. Además, se puede configurar para detectar y resolver excepciones, tales como desbordamiento o desbordamiento hacia cero, y valores especiales, como los anteriores, después de dicha operación efectiva.

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 mantisa del primer y segundo número pre-procesado, respectivamente.

En algunas realizaciones, el dispositivo podría comprender además una tercera entrada para recibir el LSD de dichas mantisas del primer y segundo número pre-procesado. Alternativamente, la tercera entrada podría tener un valor de 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.

En la suma coma flotante, el funcionamiento del camino de datos de la mantisa se divide generalmente en varios casos. En algunas implementaciones puede dividirse en dos casos: el “camino corto”, cuando se calcula una resta efectiva, para una diferencia de exponentes $|d|\leq 1$, y el

“camino lejano” cuando se realiza una suma efectiva, o bien una resta efectiva para un diferencia de exponentes |d|>1. En algunas implementaciones dicho camino de datos de la mantisa, o cualquier parte del mismo, puede implementarse usando dos o más caminos paralelos para calcular por separado los casos, para lograr así un mejor rendimiento. Cada sub-camino realiza el cálculo suponiendo un caso diferente y un multiplexor final selecciona el resultado correcto para el caso presente. En las siguientes descripciones de realizaciones consideraremos generalmente una implementación unificada del camino de datos de la mantisa, sin embargo, alguien experto en el estado de la técnica podría apreciar que varios de los módulos descritos aquí podrían ser usados de una forma replicada o dividida, con pequeñas modificaciones, para implementarlos en caminos paralelos. Además, aunque las siguientes descripciones de las 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.

En algunas realizaciones, el camino de datos de la mantisa podría comprender al menos un módulo de suma configurado para recibir como mucho los m+1 MSBs de la mantisa del primer y segundo número pre-procesado. Si el número está normalizado entonces podría recibir solamente los m LSBs de 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 camino de datos de la mantisa podría estar configurado para recibir una instrucción desde el camino de datos del exponente sobre la mantisa correspondiente al número de mayor exponente, la primera cantidad de desplazamiento y la operación efectiva. Además el camino de datos podría estar configurado para generar un valor que corresponde a la suma o resta de dichas mantisas pre-procesadas después de alinearlas.

En algunas realizaciones, dicho módulo de suma está además configurado para generar un valor que corresponde al valor absoluto del resultado de la

operación efectiva entre dichas mantisas pre-procesadas.

En algunas realizaciones, el módulo de suma podría comprender un primer módulo de desplazamiento configurado para recibir como mucho los m+1 MSBs de la mantisa pre-procesada del número con el menor exponente, en una primera entrada, y la primera cantidad de desplazamiento, en una segunda entrada, y generar un valor de salida correspondiente al desplazamiento a la derecha de dicha mantisa pre-procesada del número con el menor exponente. El primer módulo de desplazamiento podría comprender además una tercera entrada con el valor uno para agregar explícitamente el LSB a dicha mantisa antes de desplazarla. Un módulo de intercambio podría ser usado para recibir una indicación sobre la mantisa del número con menor exponente y proporcionársela al primer módulo de desplazamiento. En el caso de que ambos exponentes sean iguales, cualquiera de las mantisas podría ser proporcionada como aquella correspondiente al menor exponente, sin cambiar la funcionalidad. Por claridad en la explicación, aunque ambos exponentes sean iguales, llamaremos “la mantisa correspondiente al número con el menor exponente” para referirnos a uno de las mantisas y lo contrario para referirnos a la otra. El primer módulo de desplazamiento podría estar preparado para negar selectivamente el valor de salida. Como la mantisa es un número pre-procesado, esta negación podría ser implementada simplemente invirtiendo todos los bits menos el LSB, y no se requiere ninguna suma. En algunas implementaciones, el bit de signo de la mantisa podría ser incluido al principio como el MSB de la mantisa, mientras que en otras, el bit de signo podría añadirse a la izquierda de la mantisa antes de invertirla. En otras implementaciones, el bit de signo podría incluirse después de la inversión, justo antes de operar con el número. En implementaciones alternativas, la mantisa del formato coma flotante podría ser con signo y la negación no sería necesaria.

En algunas realizaciones, el primer módulo de desplazamiento podría comprender un desplazador a la derecha conectado a un inversor de bits

condicional. En algunas implementaciones, el desplazador a la derecha está colocado delante del inversor de bits condicional y podría requerir una lógica adicional para poner a uno el LSB de la salida después de la inversión si los exponentes son iguales, ya que no se realiza ningún desplazamiento y el LSB de la mantisa se representa explícitamente. En otras implementaciones, el desplazador a la derecha, el cual debería implementarse con extensión de signo, se coloca después del inversor de bits condicional y podría no requerir lógica adicional, ya que el LSB de la mantisa se podría añadir después del circuito inversor.

En algunas realizaciones, el módulo de suma podría comprender además un sumador en coma fija, con una primera entrada conectada a la salida del primer módulo de desplazamiento, y una segunda entrada configurada para recibir como mucho los m+1 MSBs de la mantisa pre-procesada correspondiente al número con mayor exponente. El sumador en coma fija podría estar configurado para generar un valor que corresponde al resultado de la operación efectiva entre dichas mantisas pre-procesadas después de alinearlas. En algunas implementaciones el sumador en coma fija podría además estar configurado para generar una señal de desbordamiento como una salida independiente, mientras que en otras podría añadir un MSB extra a la salida. En algunas implementaciones, el bit de signo se puede distribuir como una salida independiente, mientras que en otras puede añadirse como el MSB de la salida.

En algunas implementaciones el sumador en coma fija podría estar configurado para incorporar explícitamente el LSB de la mantisa pre-procesada del número con mayor exponente, el cual es siempre uno, antes de que se realice la operación efectiva. En otras implementaciones, el sumador en coma fija podría estar configurado para tener en cuenta dicho LSB internamente cuando se realice la operación efectiva.

En algunas realizaciones, el sumador en coma fija podría estar configurado

para negar selectivamente la mantisa pre-procesada correspondiente al número con el mayor exponente. Esto podría usarse cuando la operación efectiva es una resta, se requiere un resultado positivo y los exponentes son iguales.

En algunas realizaciones, el sumador en coma fija podría comprender un inversor de bits condicional para negar selectivamente la mantisa pre-procesada correspondiente al número de mayor exponente. De nuevo, una ventaja de las realizaciones propuestas es que para negar solamente se necesita una inversión. En algunas implementaciones el bit de signo de la mantisa podría ser incluido al principio como el MSB de la mantisa, mientras que en otras, el bit de signo podría añadirse a la izquierda de la mantisa antes de invertirla.

En algunas realizaciones, el módulo de suma podría comprender además un circuito de control configurado para recibir la operación efectiva y controlar si el primer módulo de desplazamiento o el sumador de números en coma fija deben realizar dicha negación. El circuito de control podría ser diferente dependiendo de los requerimientos de la salida, por ejemplo cuando la salida se requiere en formato de valor absoluto, o cuando se permite la salida negativa.

En algunas realizaciones, el dispositivo podría comprender además un módulo de normalización. El módulo de normalización del sumador en coma flotante podría tener una primera entrada conectada a la salida del módulo de suma, y una segunda entrada para recibir una segunda cantidad de desplazamiento. El módulo de normalización podría estar configurado para generar como mucho los m+1 MSBs de la mantisa del tercer número pre-procesado, mediante el desplazamiento selectivo a izquierda, o derecha, de la salida del módulo de suma. Como la salida es un número pre-procesado, el redondeo al más cercano podría ser realizado mediante un simple truncado pero cierto sesgo puede aparecer después de redondear.

En algunas realizaciones, el módulo de normalización del sumador coma flotante podría estar configurado además para generar selectivamente el valor equivalente a restar uno del LSB del resultado de la operación de desplazamiento, cuando un bit seleccionado, o una combinación de bits seleccionados, de la salida del módulo de suma es igual a uno. Esta configuración permite al módulo de normalización eliminar el sesgo (hacia el par, en caso de empate) cuando d={1,0} y la operación efectiva es una resta, es decir el caso del “camino cercano”.

En algunas realizaciones, el módulo de normalización podría estar configurado además para generar selectivamente el complemento a uno del resultado de dicho desplazamiento, o dicha resta posterior. Esto permite una salida positiva cuando el sumador en coma fija proporciona una salida negativa y, además, elimina el sesgo del redondeo cuando d=0 y la operación efectiva es una resta.

En algunas realizaciones, el módulo de normalización podría estar configurado además para completar selectivamente las posiciones vacantes debidas al desplazamiento a la izquierda, con ceros, con un cero en el MSB de dichas posiciones y el resto unos, o con un uno en el MSB de dichas posiciones y el resto ceros.

En algunas realizaciones, el módulo de normalización podría estar configurado además para completar selectivamente dichas posiciones vacantes, aleatoriamente, basándose en el valor de un bit seleccionado, o en una combinación de bits seleccionados, de la primera entrada del módulo de normalización, cuando la diferencia de exponentes es igual a uno. En implementaciones alternativas, dicho valor podría ser cualquier bit, o combinación de bits, con las características estadísticas adecuadas. En otras implementaciones, una nueva entrada podría configurarse. Esto permite eliminar cualquier sesgo del redondeo cuando d=1.

En algunas realizaciones, el módulo de normalización podría estar configurado además para forzar a cero el segundo LSB del valor correspondiente a la mantisa del tercer número pre-procesado, cuando los operandos de entrada tienen el mismo exponente, los valores del segundo LSB de las mantisas pre-procesadas de dichos operandos son diferentes, y la operación efectiva es una suma. Esto permite eliminar el sesgo del redondeo para la suma alineada (hacia el par en caso de empate).

En algunas realizaciones, el dispositivo podría comprender además un circuito configurado para identificar la posición del primer bit significativo por la izquierda, de la salida del módulo de suma, y calcular la segunda cantidad de desplazamiento, que será usada, por el camino de datos del exponente, para calcular el exponente de salida, y, por el módulo de normalización, para normalizar la mantisa.

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 calculador de cantidad de desplazamiento, un módulo para calcular el exponente, con una primera entrada para recibir la tercera 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 módulo de normalización con una primera entrada para recibir los N MSBs de los N+1 LSBs del número coma fija y una segunda entrada para recibir también la tercera 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 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 suma permite que un número en formato de coma fija pre-procesado sea procesado por dispositivos de suma 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 del 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 cuarta cantidad de desplazamiento proveniente del 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 suma permite que un número en formato de coma fija no-procesado sea procesable por dispositivos de suma 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 cuarta 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 cuarta 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 [log2(máxima cantidad de desplazamiento)], con cada multiplexor configurado para efectuar una operación de desplazamiento a la izquierda de 2^i posiciones, iϵ[0, 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 cuarta 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 tercer 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 del sumador coma flotante de acuerdo a las realizaciones descritas aquí permite que el resultado de las operaciones sea usado por circuitos que funcionan con formato no procesado.

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 sumador 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 colocarse antes o después del sumador 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 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 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 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 quinta 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.

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.

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 quinta cantidad de desplazamiento, en una salida, y un módulo de desplazamiento aritmético a la 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 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.

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.

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.

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 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 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 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.

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 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 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 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 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.

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 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-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.

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 ilustra el camino de datos de la mantisa de un sumador en coma flotante (FP) de acuerdo con un ejemplo;

Fig. 1a muestra en detalle un ejemplo de un inversor de bit condicional especial;

Fig. 2 ilustra otro ejemplo de implementación del camino de datos de la mantisa de un sumador FP, el cual elimina algunas fuentes de sesgo;

Fig. 2a ilustra un ejemplo de implementación de un desplazador a la izquierda especial;

Fig. 3 ilustra otro ejemplo de implementación del camino de datos de la mantisa de un sumador FP, el cual elimina algunas fuentes de sesgo de un modo más simplificado;

Fig. 3a ilustra un ejemplo de implementación de un módulo de suma en complemento a dos;

Fig. 4 ilustra otro ejemplo de implementación de un sumador FP, el cual evita el sesgo debido al redondeo;

Fig. 4a ilustra un ejemplo de un módulo de redondeo cercano;

Fig. 4b ilustra un ejemplo de un módulo de redondeo lejano;

Fig. 5 ilustra el camino de datos de la mantisa de un sumador FP de “doble camino” de acuerdo con un ejemplo;

Fig. 6 ilustra un ejemplo de implementación de un conversor de números en coma fija pre-procesados a números en coma flotante pre-procesados;

Fig. 6a ilustra un ejemplo de implementación de un desplazador a la izquierda pre-procesado;

Fig. 6b ilustra un ejemplo de implementación de un desplazador a la izquierda especial;

Fig. 7 ilustra un ejemplo de implementación de un conversor de números coma fija no procesados a números coma flotante pre-procesados;

Fig. 7a and 7b ilustran ejemplos de implementación de un módulo de normalización de un conversor de números en coma fija no procesados a números en coma flotante pre-procesados;

Fig. 8a, 8b and 8c ilustran ejemplos de implementación de un conversor de números en coma flotante pre-procesados a números en coma flotante pre-procesados;

Fig. 9, 10a and 10b ilustran ejemplos de implementación de un conversor de números coma flotante pre-procesados a números coma fija pre-procesados;

Fig. 11, 12a, 12b 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. 13 ilustra un ejemplo de implementación de un conversor de números coma flotante pre-procesados a números en coma flotante no procesados;

Fig. 13a ilustra un ejemplo de implementación del módulo de redondeo de un conversor de números coma flotante pre-procesados a números en coma flotante no procesados;

Fig. 14 ilustra un ejemplo de implementación de un conversor de números coma flotante pre-procesados a números en coma fija no procesados;

DESCRIPCION DETALLADA DE LAS REALIZACIONES

La Fig. 1 muestra el camino de datos de la mantisa del sumador en coma flotante (FP) de acuerdo con un ejemplo. La salida del sumador en coma fija, en este ejemplo mostrado en la Fig. 1, es siempre positiva. El sumador FP 100 recibe m bits de una primera mantisa Mx y de una segunda mantisa My, respectivamente. Ambas mantisas pertenecen a números en coma flotante pre-procesados. Cada una de las mantisas Mx y My tiene m+1 dígitos. 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 sumador a la entrada. En el ejemplo de la Fig. 1, los dos números en coma flotante están normalizados. Sin embargo, para simplificar la descripción, tanto el MSB como el bit de signo de los dos números normalizados son incluidos en los m bits que se introducen en el sumador 100. En una implementación alternativa, estos bits pueden ser introducidos después del módulo de conmutación. El sumador FP 100 comprende un módulo de conmutación 105 y un comparador 110, teniendo ambos una primera y segunda entradas para recibir los m MSBs de las mantisas. El módulo de conmutación 105 tiene una primera y segunda salidas y está configurado para

que la mantisa del número con el menor exponente salga por la primera salida y la mantisa del número con el exponente más alto salga en la segunda salida. El módulo de conmutación 105 comprende además una tercera entrada para recibir el signo de la diferencia de exponentes. Esto será calculado por un comparador de exponentes (no mostrado). El módulo comparador 110 comprende además una tercera entrada para recibir una señal de control en caso de que los números tengan el mismo exponente y que la operación efectiva sea una resta. El módulo de comparación 110 genera una primera señal de control en la primera salida y una segunda señal de control en la segunda salida para ordenar una negación de una de las mantisas, cuando la operación efectiva es una resta. Como se mencionó anteriormente, esta negación puede ser implementada simplemente mediante la inversión de todos los bits menos el LSB. El sumador FP 100 comprende además un desplazador a la derecha 115 que tiene una primera entrada acoplada a la primera salida del módulo de conmutación 105 y una segunda entrada para recibir la cantidad de desplazamiento (dibujada en la Fig. 1 como el valor absoluto de la diferencia de exponentes). La primera salida del módulo de conmutación 105 porta los m MSBs de la mantisa del número con el exponente menor. El desplazador a la derecha 115 podría comprender además una tercera entrada acoplada a 1. Esto introduce el LSB de la mantisa en el desplazador a la derecha 115 de forma que éste recibe los m+1 bits de la mantisa. El desplazador a la derecha 115 desplazará a la derecha este número de m+1 bits de acuerdo a la cantidad de desplazamiento recibida y generará un número desplazado de m+1 bits. El desplazador a la derecha 115 está conectado a un inversor de bits condicional especial 120. El inversor de bit condicional especial 120 recibirá la primera señal de control del módulo comparador 110, para realizar una inversión bit a bit de todos los m+1 bits recibidos, excepto si los números tienen el mismo exponente. En tal caso el LSB de la salida es forzado a 1.

La Fig. 1a muestra en detalle el inversor de bits condicional especial. Comprende un inversor estándar 120a que recibe los m MSBs de la entrada y

efectúa una inversión bit a bit de los m bits. El LSB se introduce en un puerta XOR 122a junto con la salida de una puerta AND de dos entradas que recibe la operación efectiva en la primera entrada y una señal indicando si los exponentes son diferentes, en la segunda entrada. Por tanto, la salida del inversor especial comprende m+1 bits, donde el LSB de los m+1 bits es la salida de la puerta XOR 122a.

De acuerdo con lo anterior, el sumador FP 100 comprende además un inversor de bits condicional 125 que tiene una primera entrada conectada a una segunda salida del módulo de conmutación 105 y una segunda entrada conectada a la segunda salida del módulo comparador. El inversor de bits condicional 125 es un inversor de bits condicional convencional sin casos especiales, ya que el LSB de la mantisa no se introduce en su entrada. Ahora, el inversor de bits condicional 125 genera un número de m bits. Cuando la operación efectiva es una resta y d=0, el módulo comparador 110 compara las mantisas de entrada, e instruye, o al inversor de bits condicional 120, o al inversor de bits condicional 125, para negar la mantisa de valor más bajo. Si d<>0, el inversor de bits condicional 120 siempre niega su entrada para efectuar una resta efectiva. El sumador FP 100 comprende además un módulo sumador en complemento a dos 130 que tiene una primera entrada conectada a la salida del inversor de bits condicional 125, y una segunda entrada conectada a la salida del inversor de bits condicional especial 120. La primera entrada recibe m bits, mientras que la segunda recibe m+1 bits. Entonces, el módulo sumador en complemento a dos 130 comprende además una tercera entrada conectada a 1, de forma que los m bits a la salida del inversor de bits condicional 125 son aumentados en 1 bit a la derecha. Sin embargo, en implementaciones alternativas, la introducción del uno adicional podría ser efectuada internamente por el módulo 130 sin necesidad de una entrada especial. Dicho uno es mostrado de manera explícita en el ejemplo de la Fig. 1, y en los subsiguientes ejemplos, para indicar la necesidad de la introducción funcional del LSB implícito. El módulo sumador en complemento a dos 130 efectúa una suma de los dos números con signo, y genera un

resultado en una primera salida. El modulo sumador en complemento a dos 130 tiene además una segunda salida para generar un bit de desbordamiento. La primera salida del módulo sumador en complemento a dos 130 está conectada al detector de unos de cabecera (LOD) 135 y al desplazador 140. El módulo LOD 135 está configurado para calcular el número de bits a desplazar a la izquierda que realizará el desplazador 140. En otras implementaciones este módulo podría ser alternativamente un anticipador de ceros de cabecera (LZA), o un circuito similar. El desplazador 140 desplaza una posición a la derecha, si hay un desbordamiento. En otro caso, desplaza tantas posiciones a la izquierda como los indicados por el módulo LOD 135. El desplazador 140 genera los m MSBs de la mantisa Mz que es la suma, o diferencia, normalizada de las mantisas Mx y My después de su alineamiento. El LSB de la mantisa Mz es implícito e igual a 1. Por tanto, el redondeo al más cercano se efectúa mediante truncamiento. Sin embargo, este redondeo produce un sesgo en la suma alineada (de números con el mismo exponente), y en el caso del camino cercano, si se efectúa un desplazamiento a la izquierda.

Debe indicarse que en esta implementación los m MSBs de la mantisa incluyen el bit de signo y el bit entero. En una implementación alternativa, el bit de signo podría ser desechado después de la suma, ya que es siempre cero y, de manera similar, el bit entero podría ser desechado después de la normalización, ya que es siempre uno.

La Fig. 2 ilustra el camino de datos de la mantisa para un sumador en coma flotante (FP) de acuerdo a otro ejemplo. En este ejemplo, el sesgo se produce debido al redondeo, sólo en el caso del camino cercano, si d=1, o en la suma alineada. En el caso de d=0 y resta efectiva, se realiza un redondeo “tie to away”. En este ejemplo no hay módulo comparador como ocurría en el ejemplo de la Fig. 1. Por tanto, la salida del sumador en coma fija podría ser también negativa. El sumador FP 200 recibe m bits de una primera mantisa Mx y de una segunda mantisa My, respectivamente. Ambas mantisas

pertenecen a los números pre-procesados en coma flotante. Las dos mantisas Mx y My tienen m+1 bits. Sin embargo, de nuevo, como ambas mantisas pertenecen a los números pre-procesados, el LSB de ambas mantisas es igual a uno (1) y no necesita ser introducida en el sumador a la entrada. Por tanto, de nuevo, como en ejemplo de la Fig. 1, sólo los m MSBs de cada mantisa Mx y My son entradas al sumador FP 200. Además, de nuevo, los dos números en coma flotante están normalizados. De nuevo, para simplificar la descripción, el MSB de los dos números normalizados y el bit de signo de ambos son incluidos en los m bits que son introducidos en el sumador 200, aunque, en una implementación alternativa, podrían ser introducidos justo antes de que sean requeridos. El sumador FP 200 comprende un módulo de conmutación 205 que tiene una primera y segunda entradas para recibir los m MSBs de las mantisas. El módulo de conmutación 205, que tiene una función similar al módulo de conmutación 105 de la Fig. 1, comprende además una tercera entrada para recibir el signo de la diferencia de exponentes. Ésta será calculada por un comparador de exponentes (no mostrado). El sumador FP 200 comprende además un inversor de bits condicional 210 que tiene una primera entrada conectada a una primera salida del módulo de conmutación 205 para recibir los m MSBs de la mantisa del número con el menor exponente, y una segunda entrada para recibir un bit indicativo de la operación efectiva (op). El inversor de bits condicional 205 llevará a cabo una inversión bit a bit de los m bits, si la operación efectiva es una resta. El sumador FP 200 comprende además un desplazador a la derecha 215 que tiene una primera entrada conectada a la salida del inversor de bits condicional y una segunda entrada conectada a un 1 lógico. Esto introduce el LSB de la mantisa en el desplazador a la derecha 215 de forma que éste recibe m+1 bits. En una implementación alternativa, este LSB a uno, podría ser introducido internamente en el desplazador. El desplazador a la derecha 215 desplazará a la derecha este número de m+1 bits. El sumador FP 200 comprende además un módulo de suma en complemento a dos 220 teniendo una primera entrada conectada a la salida del desplazador a la derecha 215 y una segunda entrada conectada a una segunda salida del

módulo de conmutación 205. La primera entrada recibe m+1 bits, mientras que la segunda entrada recibe m bits. Por tanto, el módulo sumador en complemento a dos 220 comprende además una tercera entrada conectada a 1, de forma que los m bits de la segunda salida del módulo de conmutación 205 son aumentados en un LSB. De nuevo, en implementaciones alternativas, la introducción del uno adicional podría ser efectuada internamente en el módulo 220 sin la necesidad de una entrada especial. El módulo de suma en complemento a dos 220 efectúa una suma de dos números con signo, y genera un resultado de m+1 bits en una primera salida. El módulo de suma en complemento a dos 220 comprende además una segunda salida para generar un bit de desbordamiento. El módulo de suma en complemento a dos 220 está conectado al desplazador a la derecha de una posición 235, del módulo de normalización 230. Un entrada de control del desplazador a la derecha 235 está conectada a la segunda salida del módulo de suma en complemento a dos 220, y un desplazamiento a la derecha se efectúa si ocurre un desbordamiento. El sumador FP 200 comprende además un módulo de anticipación de ceros de cabecera (LZA) 225, que tiene una primera entrada conectada a la segunda salida del módulo de conmutación 205 y una segunda entrada conectada a la salida del desplazador a la derecha 215. El valor 1 también se inserta en la entrada del módulo LZA 225 de forma que los m bits en la segunda salida del módulo de conmutación 205 son aumentados con un bit a la derecha correspondiéndose con el LSB implícito. Sin embargo, en otras implementaciones la introducción del uno adicional podría realizarse internamente en el módulo LZA 225, sin la necesidad de una entrada especial. El módulo de normalización 230 comprende además un inversor de bits condicional 240 que tiene una entrada conectada con la primera salida del módulo de suma en complemento a dos 220 y un desplazador a la izquierda especial 245, que tiene una primera entrada conectada a la salida del inversor de bits condicional 240. Una segunda entrada del desplazador a la izquierda especial 245 se acopla a la salida del módulo LZA 225. El número de bits a desplazar por el desplazador a la izquierda especial 245 es proporcionado por el módulo LZA 225. Este

desplazador 245 es un desplazador especial de tal manera que en un desplazamiento a la izquierda, las posiciones vacantes son completadas con un bit que viene de una tercera entrada del desplazador especial, es cual está conectado al signo del resultado del módulo de suma en complemento a dos 220. Una implementación del desplazador a la izquierda especial 245 basada en la implementación del desplazador variable clásico es ilustrada en la Fig. 2a.

El desplazador a la izquierda especial 245, mostrado en Fig. 2a, se implementa usando varios multiplexores dos a uno (log2 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, a la posición no desplazada, y a la desplazada (2^0), respectivamente, mientras 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 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, la máxima cantidad de desplazamiento es m-1. La salida del desplazador a la izquierda especial 245 comprende los m MSBs del valor desplazado. El módulo de normalización 230 comprende además un multiplexor 250 que tiene una primera entrada conectada a la salida del desplazador a la derecha 235 y una segunda entrada conectada a la salida del desplazador a la izquierda especial 245. La salida del multiplexor es, o la salida del desplazador a la derecha 235, o la salida del desplazador a la izquierda especial 245, y comprende los m MSBs de la mantisa Mz, que es la suma o resta normalizada de las mantisas Mx y My después de alinearlas. Por tanto,

la mantisa es normalizada por el módulo de normalización 230. De nuevo, el LSB de la mantisa Mz es implícito e igual a uno.

Se debe indicar que en esta implementación, los m MSBs de la mantisa incluyen el bit de signo y el bit entero. En una implementación alternativa, el bit de signo podría ser extraído después de la suma y, similarmente, el bit entero podría ser desechado.

La Fig. 3 ilustra el camino de datos de la mantisa de un sumador en coma flotante (FP) de acuerdo a otro ejemplo. El ejemplo de acuerdo con la Fig. 3 tiene un módulo LZA diferente, un módulo de suma en complemento a dos diferente y un módulo de normalización más simple comparado con el ejemplo de acuerdo a la Fig. 2. El sumador FP 300 recibe los MSBs de una primera mantisa Mx, y de una segunda mantisa My, respectivamente. Ambas mantisas pertenecen a números pre-procesados en coma flotante. Las dos mantisas Mx y My tienen ambas m+1 dígitos. De nuevo, 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 sumador FP 300 a la entrada. Además, los dos números en coma flotante están también normalizados. De nuevo, para simplificar la descripción, tanto el MSB del número normalizado, como el bit de signo, están ambos incluidos en los m bits que son introducidos en el sumador FP 300. El sumador FP 300 comprende un módulo de conmutación 305, similar a los módulos de conmutación 105 y 205, teniendo una primera y segunda entradas para recibir los m MSBs de las mantisas. El módulo de conmutación 305 comprende además una tercera entrada para recibir el signo de la diferencia de exponentes. Ésta será calculada por un comparador de exponentes (no mostrado). El sumador FP 300 comprende además un inversor de bits condicional 310 que tiene una primera entrada conectada a una primera salida del módulo de conmutación 305, para recibir los m MSBs de la mantisa del número con el exponente menor. El inversor de bits condicional 310 llevará a cabo una inversión bit a bit de los m bits, si la operación efectiva es

una resta. El sumador FP 300 también, como en el sumador FP 200 de la Fig. 2, comprende además un desplazador a la derecha 315 que tiene, una primera entrada conectada a una salida de un inversor de bits condicional, y una segunda entrada conectada a un 1 lógico. El sumador FP 300 también comprende además un módulo sumador en complemento a dos 320, que tiene una primera entrada conectada a la salida del desplazador a la derecha 315, y una segunda entrada conectada a la segunda salida de módulo de conmutación 305. Similarmente al sumador FP 200 de la Fig. 2, la primera entrada recibe m+1 bits, mientras que la segunda entrada recibe m bits. Sin embargo, en este ejemplo el módulo de suma en complemento a dos 320 podría sumar internamente el LSB implícito de la segunda entrada. El módulo de suma en complemento a dos 320 efectúa una suma de los dos números con signo, y genera un resultado de m+1 bits en una primera salida. El módulo de suma en complemento a dos 320 comprende una segunda salida para generar el bit de desbordamiento. En la Fig. 3a se ilustra una implementación del módulo de suma en complemento a dos 320 considerando el LSB, a uno, de la segunda entrada de manera implícita. Para generar los m MSBs de la primera salida y el bit de desbordamiento, se usa un sumador estándar 320b de m bits, mientras que el LSB de la primera entrada se acopla al acarreo de entrada del mencionado sumador estándar, y se genera el LSB de la primera salida por inversión del mismo.

La primera salida del módulo de suma en complemento a dos 320 se acopla a una primera entrada del desplazador 335 del módulo de normalización 330. Una segunda entrada del desplazador 335 se acopla a la salida del módulo LZA 325. El sumador FP 300 comprende además el módulo LZA 325 teniendo una primera, y segunda, entrada conectadas a la primera y segunda salida del módulo de conmutación 305, respectivamente, y una tercera entrada conectada al LSB de la diferencia de exponentes. Al igual que el módulo LZA de la Fig. 2, el valor 1 es insertado también en la entrada del módulo LZA 325. De nuevo, como en otras implementaciones, la introducción del uno adicional podría ser efectuada internamente al LZA 325 sin la

necesidad de una entrada especial. Ahora, el módulo de normalización 330 comprende además un inversor de bits condicional 340 teniendo una entrada conectada con la salida del desplazador 335. La salida del inversor de bits condicional 340 comprende los m MSBs de la mantisa Mz, que es la suma normalizada de las mantisas Mx y My después de alinearlas. De nuevo, el LSB de la mantisa Mz es implícito, de la misma manera que se discutió en referencia a las Fig.1 y Fig. 2, ya que es siempre igual a 1. Congruentemente, la mantisa se normaliza con el módulo de normalización 330.

La Fig. 4 ilustra un sumador coma flotante (FP) de acuerdo con un ejemplo. El ejemplo mostrado en la Fig.4 evita cualquier fuente que pueda producir sesgo durante el redondeo. El sumador FP 400 comprende un camino de datos de mantisa 400m y un camino de datos de exponente 400e. El camino de datos de mantisa 400m recibe m bits de una primera Mantisa Mx y de una segunda Mantisa My, respectivamente. Ambas mantisas pertenecen a números en coma flotante pre-procesados. Las mantisas Mx y My tienen ambas m+1 dígitos. De nuevo, ya que ambas mantisas pertenecen a números pre-procesados, el LSB de ambas mantisas es igual a uno (1) y no necesita ser introducido en el sumador en la entrada. Entonces, de nuevo, como en los ejemplos de las Fig. 1 y Fig. 2, solo los m MSBs de cada mantisa Mx y My son entradas al camino de datos de mantisa 400m. Además, los dos números en coma flotante están normalizados también. Además, para simplificar la descripción, tanto el MSB del número normalizado como el bit de signo están incluidos en los m bits que son introducidos en el sumador 400. El camino de datos de mantisa 400m comprende un módulo de conmutación 405, similar a los módulos de conmutación 105, 205 y 305, teniendo una primera, y segunda, entrada para recibir los m MSBs de las mantisas. El módulo de conmutación 405 comprende además una tercera entrada para recibir el signo de la diferencia de exponentes. Ésta será calculada por el camino de datos de exponente 400e. El camino de datos de mantisa 400m comprende además un inversor de bits condicional 410, teniendo una primera entrada conectada a la primera salida del módulo de conmutación 405 para recibir los

m MSBs de la mantisa del número con exponente menor. El inversor de bits condicional 410 llevará a cabo una inversión bit a bit de los m bits, si la operación efectiva es una resta. El inversor de bits condicional 410 tiene una segunda entrada para recibir un bit de control indicativo de la operación efectiva. El camino de datos de mantisa 400m comprende además un desplazador a la derecha 415, que tiene una primera entrada conectada a la salida del inversor de bits condicional 410, y una segunda entrada, para recibir la cantidad de desplazamiento (|d|). El desplazador a la derecha 415 comprende además una tercera entrada conectada a un 1 lógico para introducir explícitamente el LSB. El desplazador a la derecha 415 desplazará a la derecha este número de m+1 bits de acuerdo a la cantidad de desplazamiento recibida, y genera un número desplazado de m+1 bits. El camino de datos de mantisa 400m también comprende, además, un módulo de suma en complemento a dos 420, teniendo una primera entrada conectada a la salida del desplazador a la derecha 415, y una segunda entrada conectada a una segunda salida del módulo de conmutación 405. Similarmente a los sumadores de las Fig. 1, Fig. 2 y Fig. 3, la primera entrada recibe m+1 bits mientras que la segunda entrada recibe m bits. Entonces el módulo de suma en complemento a dos 420 comprende además una tercera entrada conectada a 1, de manera que los m bits a la salida del módulo de conmutación 405 son ampliados en un LSB. El módulo de suma en complemento a dos 420 efectúa la suma de los dos números con signo y genera un resultado de m+1 bits en una primera salida. El módulo de suma en complemento a dos 420 comprende además una segunda salida para generar un bit de desbordamiento.

La primera salida del módulo de suma en complemento a dos 420 se acopla a la primera entrada del módulo de redondeo cercano 425 del módulo de normalización 430. El módulo de normalización 430 comprende además un desplazador especial 435 teniendo una primera entrada conectada a una primera salida del módulo de redondeo cercano 425 para recibir m+2 bits. El desplazador 435 es un desplazador especial de tal manera que en un

desplazamiento a la izquierda de la primera entrada, las posiciones vacantes son completadas con una tercera entrada, la cual, en este ejemplo, está conectada a la segunda salida del módulo de redondeo cercano 425, para recibir un bit. El módulo de redondeo cercano 425 proporciona los valores adecuados al módulo de desplazamiento especial 435 para obtener correctamente el resultado redondeado, y sin sesgo, después de la normalización, si la operación efectiva es una resta y la diferencia de exponentes es menor o igual que uno (op=1,d={0,1}, es decir, el caso del camino cercano). La Fig. 4a muestra el módulo de redondeo cercano 425 en detalle. El inversor de bits condicional 425a efectúa la inversión bit a bit de los m+1 bits de entrada si la salida del módulo de suma 420 es negativa, es decir, el MSB de la entrada es igual a uno (sign(c)=1). De otra forma, la salida del inversor de bits condicional 425a, que produce los m+1 MSBs de la primera salida del módulo de redondeo cercano 425, es igual a la entrada. Además, el módulo de redondeo cercano 425 comprende cierta lógica configurada de forma que, si los operandos tienen el mismo exponente (d=0), entonces el LSB de la primera salida, y la segunda salida, del módulo de redondeo cercano 425 son iguales al signo de la salida del módulo sumador 420. Si los exponentes son diferentes, este LSB de la primera salida es igual al LSB de la salida del módulo sumador 420, y la segunda salida, igual a su inversa. Debemos indicar que, cuando no estamos en el caso del camino cercano, entonces estos dos bits no afectan a la salida del módulo de normalización 430, ya que no tiene lugar ningún desplazamiento a la izquierda mayor de 1 posición. En implementaciones alternativas, el LSB de la primera salida podría ser cualquier bit o combinación de bits con las adecuadas características de aleatoriedad, y la segunda salida, su inverso.

El desplazador especial 435 proporciona una salida de m+1 bits que corresponde al MSB de la primera entrada (m+2 bits) después de desplazarla un bit a la derecha (desbordamiento) o desplazarlo a la izquierda de acuerdo a la segunda entrada, que está conectada con la salida del módulo LZA 445. El sumador FP 400 comprende además un módulo LZA 445 que tiene una

primera entrada conectada a la segunda salida del módulo de conmutación 405, y una segunda entrada conectada a la salida del desplazador a la derecha 415. Similar al módulo LZA 225 de la Fig. 2, el valor 1 se inserta también en la entrada del módulo LZA 445, para aumentar el valor de la segunda salida del módulo de conmutación 405 en un LSB. De nuevo, como en otras implementaciones, la introducción del uno adicional podría ser efectuada internamente en el módulo LZA 445 sin la necesidad de una entrada especial.

El camino de datos de mantisa 400m comprende además un módulo de redondeo lejano 440 que tiene una entrada conectada a la salida del desplazador especial 435. El módulo de redondeo lejano 440 evita redondeos con sesgo en la suma alineada. El módulo de redondeo lejano 440 proporciona un bus de m bits a la salida a partir de m+1 bits a la entrada. Fig. 4b ilustra en detalle el módulo de redondeo lejano 440. La salida es igual a los m MSB de la entrada, excepto si la operación efectiva es una suma (op=0), los exponentes son iguales (d=0) y el LSB de la entrada es cero. En este caso, el LSB de la salida se pone a cero. La salida del módulo de redondeo lejano 440 comprende los m MSBs de la mantisa Mz que es la suma o diferencia normalizada de las mantisas Mx y My después de alinearlas. El LSB de la mantisa Mz está implícito, de la misma forma que el que discutimos con referencia a la Fig. 1, 2 y 3, ya que es siempre igual a 1. Concordantemente, la mantisa es normalizada por el módulo de normalización 430.

El camino de datos de exponente comprende un módulo de diferencia de exponentes 450 que tiene una primera entrada para recibir el primer exponente Ex y una segunda entrada para recibir el segundo exponente Ey y generar un valor a la salida que representa la diferencia de exponentes d. Este valor incluye información relevante al signo de la diferencia y la magnitud de la diferencia. Un multiplexor 455 recibe el exponente en la primera y segunda entradas, respectivamente, y el signo de la diferencia de exponentes

en una tercera entrada. El camino de datos de exponente comprende además un módulo de actualización de exponentes 460 que tiene una primera entrada que recibe la salida del multiplexor 455, una segunda entrada que recibe la salida del módulo LZA 445 y una tercera entrada que recibe el bit de desbordamiento del sumador en complemento a dos 420. El módulo actualizador de exponentes genera el exponente Ez del resultado de la operación coma flotante. Además, un módulo de signo 465 recibe los bits de signo Sx y Sy de los operandos, el signo de la diferencia de exponentes (sign(d)) y el signo (sign(c)) de la diferencia de las mantisas, y genera el bit indicativo de la operación efectiva (op) y el bit de signo Sz del resultado de la operación coma flotante.

La Fig. 5 ilustra el camino de datos de la mantisa de un sumador FP con un camino doble de acuerdo a un ejemplo. El ejemplo mostrado en la Fig. 5 evita toda fuente que pueda producir sesgo durante el redondeo. El sumador FP 500 recibe m bits de una primera mantisa Mx y de una segunda Mantisa My, respectivamente. Ambas mantisas pertenecen a números pre-procesados en coma flotante. Las dos mantisas Mx y My tienen ambas m+1 bits. Sin embardo, de nuevo, 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 sumador a la entrada. Entonces, de nuevo, como en el ejemplo de la Fig. 1, solo los m MSBs de cada mantisa Mx y My son las entradas al sumador FP 500. Además, los dos números en coma flotante son de nuevo normalizados. De nuevo, para simplificar la descripción, el MSB de ambos números normalizados y el bit de signo se incluyen en los m bits que son introducidos en el sumador 500. El sumador FP 500 comprende un módulo de conmutación 505 que tiene una primera, y segunda, entrada para recibir los m MSBs de las mantisas. El módulo de conmutación 505 comprende además una tercera entrada para recibir el signo de la diferencia de exponentes.

El sumador 500 comprende además un inversor de bits condicional 510 que

tiene una primera entrada conectada a una primera salida del módulo de conmutación 505, para recibir los m MSBs de la mantisa del número de exponente menor. El inversor de bits condicional 510 llevará a cabo una inversión bit a bit de los m bits si la operación efectiva es una resta. El inversor de bits condicional 510 tiene una segunda entrada para recibir un bit de control indicativo de la operación efectiva. El sumador FP 500 comprende además un desplazador a la derecha 515 que tiene una primera entrada conectada a la salida del inversor de bits condicional 510 y una segunda entrada para recibir la cantidad de desplazamiento (|d|). El desplazador a la derecha 515 podría comprender además una tercera entrada conectada a 1, para recibir el LSB. El desplazador a la derecha 515 desplazará a la derecha este número de m+1 bits de acuerdo a la cantidad de desplazamiento recibida, y genera un número desplazado de m+1 bits. El sumador FP 500 también comprende además un módulo de suma en complemento a dos 520 que tiene una primera entrada conectada a la salida del desplazador a la derecha 515 y una segunda entrada conectada a una segunda salida del módulo de conmutación 505. De manera similar a los módulos de suma en complemento a dos de las Fig. 1, 2, 3 y 4, la primera entrada recibe m+1 bits mientras que la segunda entrada recibe m bits. Entonces, el módulo de suma en complemento a dos 520 comprende además una tercera entrada conectada a 1, de forma que los m bits en la segunda entrada del módulo de conmutación 505 son ampliados en un LSB. El módulo de suma en complemento a dos 520 efectúa la suma de los dos números con signo, y genera un resultado de m+1 bits en una primera salida. El módulo de suma en complemento a dos 520 comprende además una segunda salida para generar un bit de desbordamiento.

El sumador 500 comprende además un segundo desplazador a la derecha 525 que tiene una primera entrada conectada a la salida del inversor de bits condicional 510. El segundo desplazador a la derecha 525 comprende además una segunda entrada conectada a 1, de forma que los m bits a la salida del inversor de bits condicional 510 son ampliados en un LSB. El

segundo desplazador a la derecha 525 desplazará a la derecha como mucho una posición de este número de m+1 bits, generando un número desplazado de m+1 bits.

El sumador FP 500 comprende además un módulo de suma en complemento a dos 530 que tiene una primera entrada conectada a la salida del segundo desplazador a la derecha 525 y una segunda entrada conectada a la segunda salida del módulo de conmutación 505. Similarmente al módulo de suma 520, la primera entrada recibe m+1 bits mientras que la segunda entrada recibe m bits. Entonces, el segundo módulo de suma en complemento a dos 530 comprende además una tercera entrada conectada a 1, de forma que los m bits a la salida del módulo de conmutación 505 son ampliados en un LSB. El módulo de suma en complemento a dos 530 efectúa una suma de dos números con signo, y genera un resultado de m+1 bits en una salida.

La salida del módulo de suma en complemento a dos 530 está conectada a la primera entrada del módulo de redondeo cercano 550 del módulo de normalización 540.

El módulo de normalización 540 comprende además un desplazador a la izquierda especial 555. El desplazador a la izquierda especial es igual al descrito con referencia a la Fig. 2. Una primera y tercera entradas del desplazador a la izquierda 555 están conectadas a la primera y segunda salida del módulo de redondeo cercano 550, respectivamente, mientras que una segunda entrada del desplazador a la izquierda 555 está conectada a la salida del módulo LZA 535. El módulo de redondeo cercano 550 proporciona los valores adecuados al desplazador a la izquierda especial 555 para obtener el resultado correctamente redondeado, y sin sesgo, después de la normalización, si la operación efectiva es una resta y la diferencia de exponentes es menor o igual que uno (op=1,d={0,1}, es decir el caso del camino cercano). Además, el módulo de redondeo cercano 550 comprende cierta lógica que está configurada tal que, si los operandos tienen el mismo

exponente (d=0), entonces el LSB de la primera salida, y la segunda salida, del módulo de redondeo cercano 550 son iguales al signo de la salida del módulo sumador 530. Si los exponentes son diferentes, este LSB de la primera salida es igual al LSB de la salida del módulo sumador 530, y la segunda salida, igual a su inversa. El sumador FP 500 comprende además un módulo LZA 535 que tiene, una primera entrada conectada a la salida del desplazador a la derecha 525, y una segunda entrada conectada a la segunda salida del módulo de conmutación 505. De forma similar a los módulos LZA previos, el valor 1 es también insertado en la entrada del módulo LZA 535 para ampliar el valor de salida del módulo de conmutación 505 en un LSB. De nuevo, como en otras implementaciones, la introducción del uno adicional podría ser efectuada internamente en el módulo LZA 535 sin la necesidad de una entrada especial.

La salida de m bits del desplazador a la izquierda especial 555, que es la salida del módulo de normalización 540, es introducida como primera entrada en el multiplexor 565. La segunda entrada del multiplexor 565 está conectada a la salida del módulo de redondeo lejano 560. La unidad de redondeo lejano 560 está conectada a la salida de m+1 bits del módulo de desplazamiento 545 que, a su vez, tiene una entrada conectada con la salida del módulo de suma en complemento a dos 520. El módulo de desplazamiento 545 produce un desplazamiento a la derecha o a la izquierda de un máximo de una posición para normalizar el resultado del camino lejano. La unidad de redondeo lejano 560 es igual al descrito y referenciado en la Fig. 4.

El multiplexor 565 recibe la operación efectiva y la diferencia de exponentes, y genera los m MSBs de la mantisa Mz, que es la suma o resta normalizada de las mantisas Mx y My después de alinearlas. El LSB de la mantisa Mz está implícito, de la misma manera que se discutió con referencia a las Fig. 1, 2, 3 y 4, ya que es siempre igual a 1. En concordancia, la mantisa es normalizada por el módulo de normalización 540. El multiplexor 565 selecciona o el camino cercano, si la operación efectiva es una resta y la diferencia de

exponentes es menor que 2 (op=1, d<2), o el camino lejano, en el resto de casos.

Los sumadores FP descritos arriba requieren números FP que hayan sido pre-procesados de acuerdo a la invención como se describió también arriba. Estos números pre-procesados podrían ser generados por circuitos, tales como los mencionados sumadores 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 sumadores 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
   - representación fusionada del signo y la mantisa mediante representación en complemento a dos o cualquier otra representación
b) en coma fija: representación signo-magnitud, o representación sin signo

Una categoría de tales conversores es la de conversores para convertir

números en coma fija pre-procesados a números FP pre-procesados. La Fig. 6 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 tiene un inversor de bits condicional 605 en serie con un desplazador a la izquierda pre-procesado especial 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 del número de m+2 bits es el signo y será el signo del número FP pre-procesado, así como es usado para controlar el inversor de bits condicional 605. 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 610 precede al inversor de bits condicional 605. La función del desplazador a la izquierda pre-procesado 610 es descrita con más detalle en la Fig. 6a. El desplazador a la izquierda pre-procesado 610 requiere un desplazador a la izquierda especial 610a con una nueva entrada, la tercera, de un bit, la 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 a la izquierda especial 245 ilustrado en la Fig. 2a. En este ejemplo de la Fig. 6a, 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. 6a es también igual a cero, requiere una máxima cantidad de desplazamiento que tiene un bit adicional (m+1) de manera que la mantisa está normalizada. Alternativamente, si cuando el número en coma fija es igual a cero, éste es 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 desplazador a la izquierda pre-procesado especial 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 vacantes producidas después del desplazamiento. La salida del desplazador a la izquierda pre-procesado especial 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 Mz 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 Ez 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 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. 7 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 m+1 bits. El MSB del número en coma fija es el signo del número en coma fija 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 ser similar a los módulos de normalización 230 y 330 discutidos con

referencia a las Fig. 2 y 3. Además, el módulo de normalización podría ser implementado de acuerdo a los ejemplos descritos en la Fig. 7a y en la Fig. 7b. En la Fig. 7a, 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. 6a. 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 y 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. 7b 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 bits menos significativos 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 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 bits más significativos 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. 7, un camino paralelo comprende 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 Ez 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. 8a 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 bits más significativos de la mantisa original serán los n bits más significativos 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 Ez basándose simplemente en el exponente original Ex.

La Fig. 8b 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. 8c es otro ejemplo de un conversor de pre-procesados FP a pre-procesados FP. El conversor 800c ilustra un conversor adaptado para 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 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, como se muestra en al Fig. 8c.

Otra categoría de conversores son los conversores para convertir números FP pre-procesados a números en coma fija pre-procesados. La Fig. 9 ilustra 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 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 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. 10a 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 condicional 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 1005a 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. 10b 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.

En otras implementaciones de los ejemplos de las figuras Fig. 9, 10a y 10b, el MSB de la mantisa normalizada podría no incluir el bit 1 de cabecera. Por lo tanto, este bit a 1 podría ser introducido en el inversor de bit condicional.

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. 8 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. 11. 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.

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

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. 12a. El LSB será igual a 1, y está implícito.

b) en el caso de redondeo sin sesgo, los n-1 MSBs 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. 12b. 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. 13 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. 13a. 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. En implementaciones alternativas diferentes unidades de redondeo que realizan diferentes modos de redondeo podrían ser usadas. 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. 8b.

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 coma fija no procesados. La Fig. 14 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. 9. 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. 13a, para incrementar los n+1 MSBs de dicha salida, 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 del número en coma fija de salida, un conversor de este tipo podría ser idéntico al conversor 1000a descrito en la Fig. 10a.

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 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.

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 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 medio.

REIVINDICACIONES

1. Dispositivo para realizar una suma o resta de al menos dos números coma flotante pre-procesados y generar un tercer número coma flotante pre-procesado, tal que cada número tiene una mantisa de M+2 dígitos, que comprende:

un camino de datos del exponente y

un camino de datos de la mantisa, que comprende

una primera entrada para recibir como mucho los M+1 Digitos 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,

en el que 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, 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 utilizado.

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

definir la operación efectiva entre las mantisas, teniendo en cuenta la operación coma flotante deseada y el signo de los números de entrada;

detectar cuál es el número coma flotante de mayor exponente y generar una primera cantidad de desplazamiento correspondiente al número de posiciones a desplazar para alinear las mantisas;

calcular el exponente del número coma flotante de salida;

calcular el signo del número coma flotante de salida; y

detectar y resolver excepciones y valores especiales en los números de entrada o salida.

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 mantisa del primer y segundo número pre-procesado, respectivamente.

4. Dispositivo según cualquiera de las reivindicaciones 1 a 3, en el que comprende además una tercera entrada para recibir el LSD de dicha mantisa del primer y segundo número pre-procesado.

5. Dispositivo según cualquiera de las reivindicaciones 1 a 3, que comprende además 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.

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

al menos un módulo de suma configurado para recibir como mucho los M+1 MSBs de la mantisa del primer y segundo número pre-procesado y configurado para recibir, desde el camino de datos del exponente, una instrucción sobre la mantisa correspondiente al número de mayor exponente, la primera cantidad de desplazamiento y la operación efectiva, y generar un valor que corresponde a la suma o resta de dichas mantisas pre-procesadas y alineadas.

8. Dispositivo según reivindicación 7, en el que dicho módulo de suma está configurado para generar un valor correspondiente al valor absoluto del resultado de la operación efectiva entre dichas mantisas pre-procesadas.

9. Dispositivo según reivindicación 7 u 8, en el que dicho módulo de suma comprende un primer módulo de desplazamiento configurado para recibir, en una primera entrada, como mucho los M+1 MSBs de la mantisa pre-procesada del número con el menor exponente, y, en una segunda, la primera cantidad de desplazamiento, y para generar un valor de salida correspondiente al desplazamiento a la derecha de dicha mantisa pre-procesada del número con el menor exponente.

10. Dispositivo según reivindicación 9, en el que el primer módulo de desplazamiento está configurado para negar selectivamente el valor de salida.

11. Dispositivo según reivindicación 9 ó 10, en el que el primer módulo de desplazamiento comprende una tercera entrada con el valor lógico “uno”.

12. Dispositivo según cualquiera de las reivindicaciones 9 a 11, en el que el primer módulo de desplazamiento comprende un desplazador variable conectado a un inversor de bits condicional.

13. Dispositivo según cualquiera de las reivindicaciones 7 a 12, en el que el módulo de suma comprende además un sumador para números en coma fija, el cual tiene una primera entrada conectada a la salida del primer módulo de

desplazamiento y una segunda entrada configurada para recibir como mucho los M+1 MSBs de la mantisa pre-procesada correspondiente al número con mayor exponente; dicho sumador configurado para generar el valor correspondiente a la operación efectiva entre dichas mantisas pre-procesadas.

14. Dispositivo según reivindicación 13, en el que el sumador para números en coma fija está configurado para negar selectivamente la mantisa pre-procesada del número con mayor exponente.

15. Dispositivo según reivindicación 14, en el que el sumador para números en coma fija comprende un inversor de bits condicional para negar selectivamente la mantisa pre-procesada del número con mayor exponente.

16. Dispositivo según cualquiera de las reivindicaciones 13 a 15, en el que el módulo de suma comprende además un circuito de control configurado para recibir la operación efectiva y controlar si, el primer módulo de desplazamiento, o el sumador de números en coma fija, deben realizar dicha negación.

17. Dispositivo según cualquiera de las reivindicaciones 7 a 16, en el que dicho dispositivo comprende además un módulo de normalización con una primera entrada conectada a la salida del módulo de suma y una segunda entrada para recibir una segunda cantidad de desplazamiento; tal que dicho módulo de normalización está configurado para generar como mucho los M+1 MSBs de la mantisa del tercer número pre-procesado mediante el desplazamiento selectivo a izquierda, o derecha, de la salida del módulo de suma.

18. Dispositivo según reivindicación 17, en el que el módulo de normalización está configurado además para generar selectivamente el valor equivalente a restar uno del LSB del resultado de la operación de desplazamiento cuando un bit seleccionado, o una combinación de bits seleccionados, de la salida del módulo de suma, es igual a uno.

19. Dispositivo según reivindicación 17 ó 18, en el que el módulo de normalización está configurado además para generar selectivamente el complemento a uno del resultado de dicha operación.

20. Dispositivo según reivindicación 17, 18 ó 19, en el que el módulo de normalización está configurado además para completar selectivamente las posiciones vacantes debidas al desplazamiento a la izquierda, fijándolas a cero, o fijando el MSB de la posiciones vacantes a cero, y el resto a uno, o fijando el MSB de la posiciones vacantes uno, y el resto a cero.

21. Dispositivo según reivindicación 20, en el que el módulo de normalización está configurado para, selectivamente, completar dichas posiciones vacantes, aleatoriamente, basándose en el valor de un bit seleccionado, o de una combinación de bits seleccionados, cuando la diferencia de exponentes es igual a uno.

22. Dispositivo según cualquiera de las reivindicaciones 17 a 21, en el que el módulo de normalización está configurado además para forzar a cero el segundo LSB del valor correspondiente a la tercera mantisa pre-procesada, cuando los operandos de entrada tienen el mismo exponente, los valores del

segundo LSB de las mantisas pre-procesada de dichos operandos son diferentes, y la operación efectiva es una suma.

23. Dispositivo según cualquiera de las reivindicaciones 17 a 22, en el que comprende además un circuito para identificar el primer bit significativo por la izquierda de la salida del módulo de suma, y calcular la segunda cantidad de desplazamiento, que será usada, por el camino de datos del exponente, para calcular el exponente de salida, y, por el módulo de normalización, para normalizar la mantisa.

24. Dispositivo según cualquiera de las reivindicaciones 6 a 23, 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.

25. Dispositivo según reivindicación 24 en el que dicho conversor de números coma fija pre-procesados a números coma flotante pre-procesados comprende:

un calculador de cantidad de desplazamiento,

un módulo para calcular el exponente, con una primera entrada para recibir la tercera 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

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 tercera 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.

26. Dispositivo según reivindicación 25 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.

27. Dispositivo según reivindicación 25 ó 26, en el que dicho módulo de normalización está configurado además para generar selectivamente el complemento a uno del resultado de dicho desplazamiento.

28. Dispositivo según cualquiera de las reivindicaciones 6 a 27, 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:

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 mantisa del número pre-procesado en coma flotante,

un calculador de exponentes con una primera entrada para recibir la cuarta 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.

29. Dispositivo según la reivindicación 28, 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 cuarta 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.

30. Dispositivo según la reivindicación 29, en el que el módulo de normalización está configurado además para generar selectivamente el complemento a uno de dicho valor si la entrada es negativa.

31. Dispositivo según cualquiera de las reivindicaciones 18, 19, 21, 26, 27, 29 ó 30, en el que el módulo de normalización comprende un desplazador variable configurado para recibir un bit para completar las posiciones vacantes.

32. Dispositivo según la reivindicación 31, 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 [log2(máxima cantidad de desplazamiento)], con cada multiplexor configurado para efectuar una operación de desplazamiento a la izquierda de 2^i posiciones, i□[0, número de multiplexores-1], y cada

multiplexor configurado para completar las posiciones vacantes usando el valor de dicho bit recibido.

33. Dispositivo según la reivindicación 28, 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 cuarta 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.

34. Dispositivo según la reivindicación 33, 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.

35. Dispositivo según a cualquiera de las reivindicaciones 25 a 34, en el que el calculador de exponentes está configurado para decrementar, de acuerdo a la cuarta cantidad de desplazamiento, un valor base para obtener el exponente.

36. Dispositivo según la reivindicación 35, 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.

37. Dispositivo según cualquiera de las reivindicaciones 6 a 36, 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 tercer número en coma fija no procesado.

38. Dispositivo según la reivindicación 37, 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.

39. Dispositivo según reivindicación 38, 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 incrementar dicha entrada si el LSB de dicha salida es igual a 1.

40. Dispositivo según cualquiera de las reivindicaciones 6 a 36, que comprende además un conversor de números coma flotante pre-procesados a números coma flotante pre-procesados para convertir un número inicial 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.

41. Dispositivo según la reivindicación 40, en el que cuando el subsecuente 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,

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.

42. Dispositivo según la reivindicación 40, en el que cuando el subsecuente número en coma flotante pre-procesado tiene una mantisa con J+2+Q bits, entonces el conversor comprende:

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 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 subsecuente número en coma flotante pre-procesado.

43. Dispositivo según la reivindicación 42, 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.

44. Dispositivo según cualquiera de las reivindicaciones 6 a 43, 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.

45. Dispositivo según la reivindicación 44, 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,

un segundo 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.

46. Dispositivo según la reivindicación 45, en el que el segundo módulo de desplazamiento comprende un desplazador aritmético a la derecha acoplado a un inversor de bit condicional.

47. Dispositivo según reivindicación 44, 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,

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 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.

48. Dispositivo según la reivindicación 47, 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.

49. Dispositivo según las reivindicaciones 47 o 48, 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.

50. Dispositivo según cualquiera de las reivindicaciones 6 a 49, 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.

51. Dispositivo según la reivindicación 50, 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.

52. Dispositivo según la reivindicación 51, 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 cero.

53. Dispositivo según la reivindicación 50, en el que cuando el número en coma flotante pre-procesado tiene una mantisa de E+2+G bits entonces el conversor comprende:

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 los como mucho E+G+1 MSBs de la mantisa del número en coma flotante pre-procesado fijando los como mucho E+2 MSBs del número en coma flotante pre-procesado al mismo valor que los como mucho E+2 bits de la mantisa del 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.

54. Dispositivo según la reivindicación 53, 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 seleccionados, de la mantisa no procesada de entrada es igual a uno.

55. Dispositivo según cualquiera de las reivindicaciones 6 a 54, 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.

56. Dispositivo según la reivindicación 55, 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 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.

57. Dispositivo según la reivindicación 56, 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.

58. Dispositivo según las reivindicaciones 56 ó 57, 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.

59. Dispositivo según la reivindicación 55, 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 cero, y

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

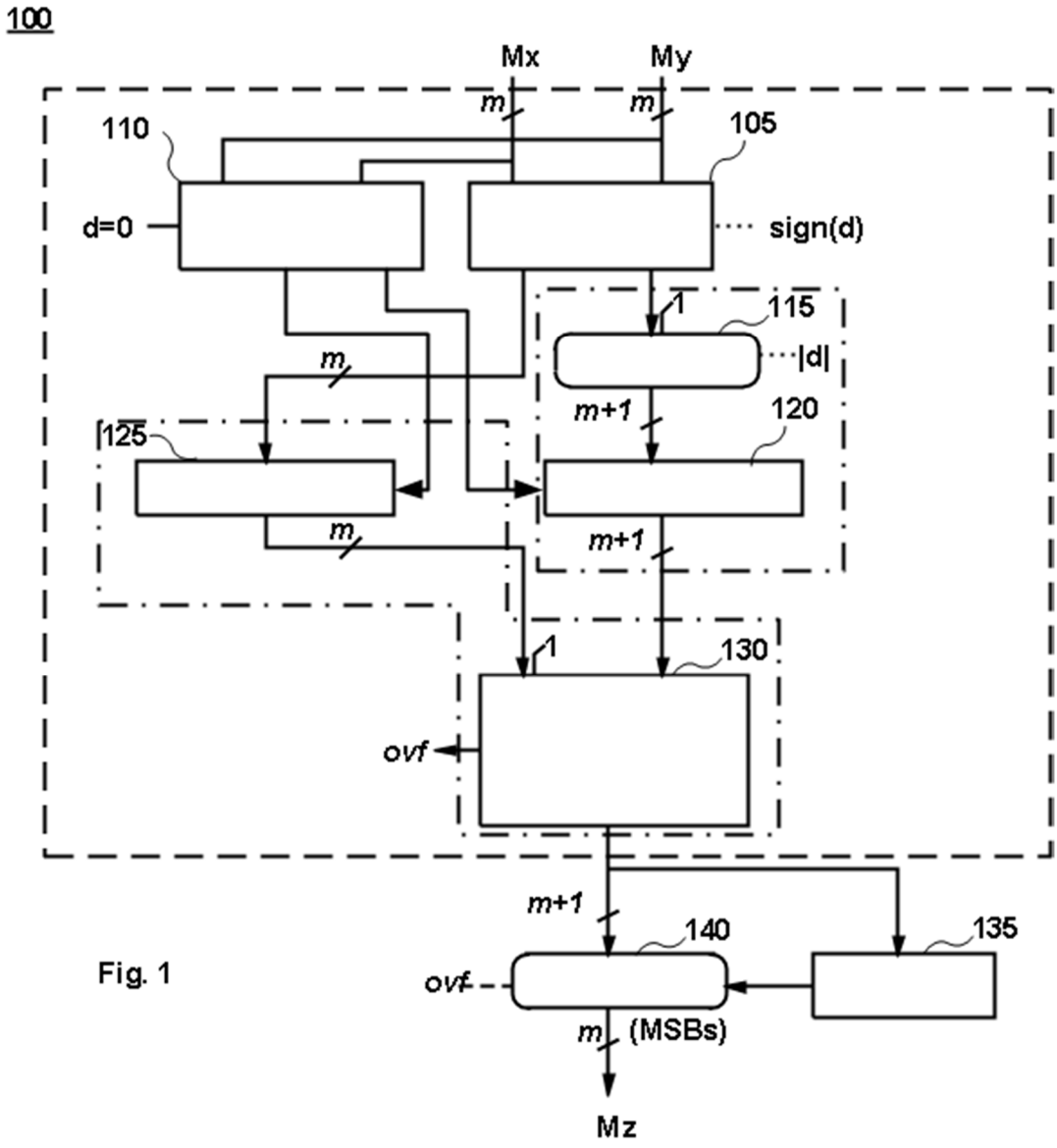

100
Mx
My
m
m
110
105
d=0
sign(d)
1
115
|d|
m
m+1
120
125
m
m+1
1
130
ovf
m+1
140
135
ovf
m (MSBs)
Mz

Fig. 1

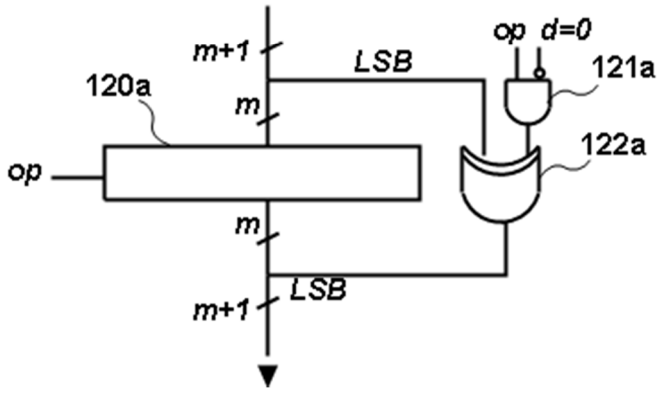

m+1
LSB
op
d=0
121a
120a
m
122a
op
m
m+1
LSB

Fig. 1a

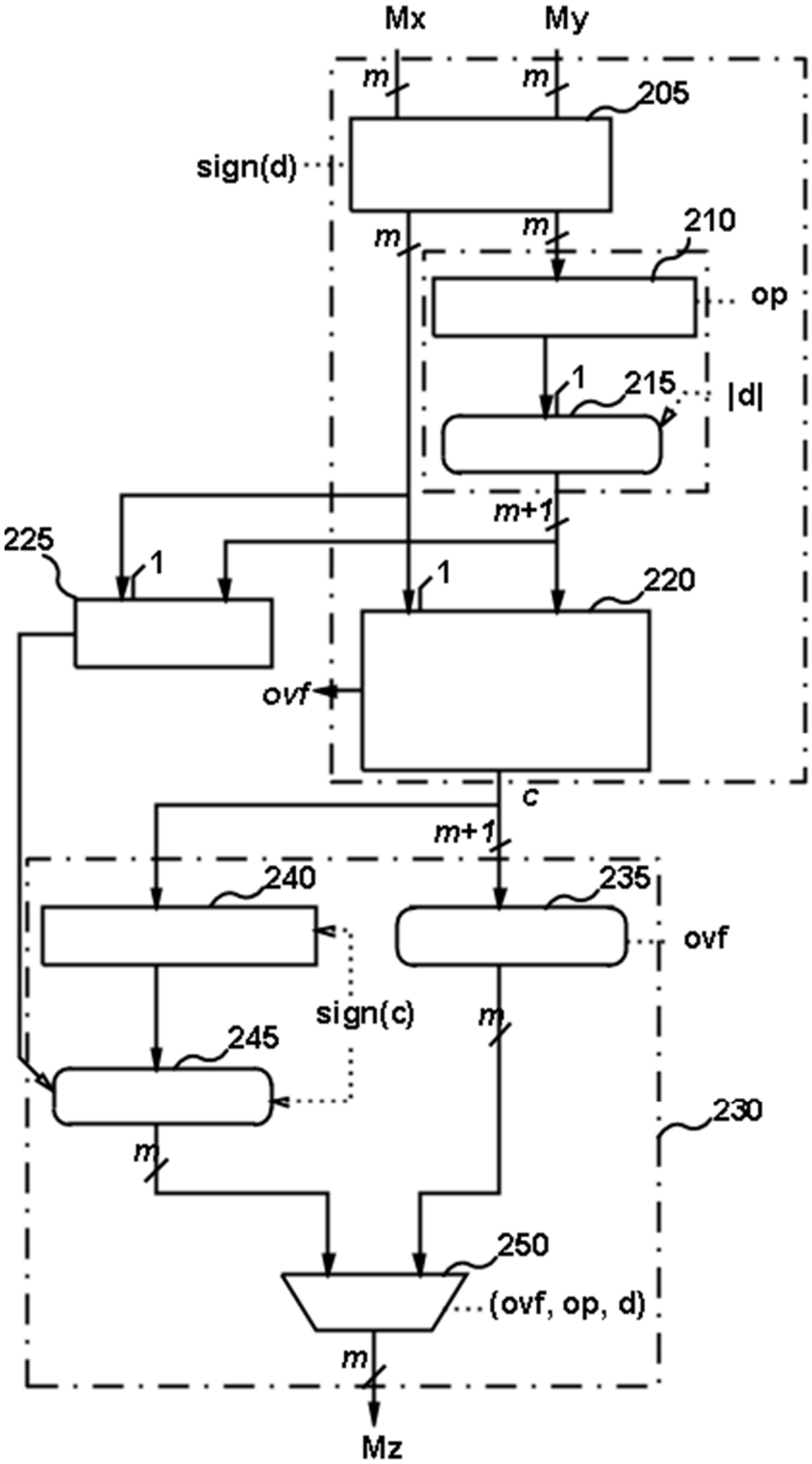

Mx
My
m
m
205
sign(d)
m
m
210
op
1
215
|d|
m+1
225
1
1
220
ovf
c
m+1
240
235
ovf
sign(c)
m
245
230
m
250
(ovf, op, d)
m
Mz

Fig. 2

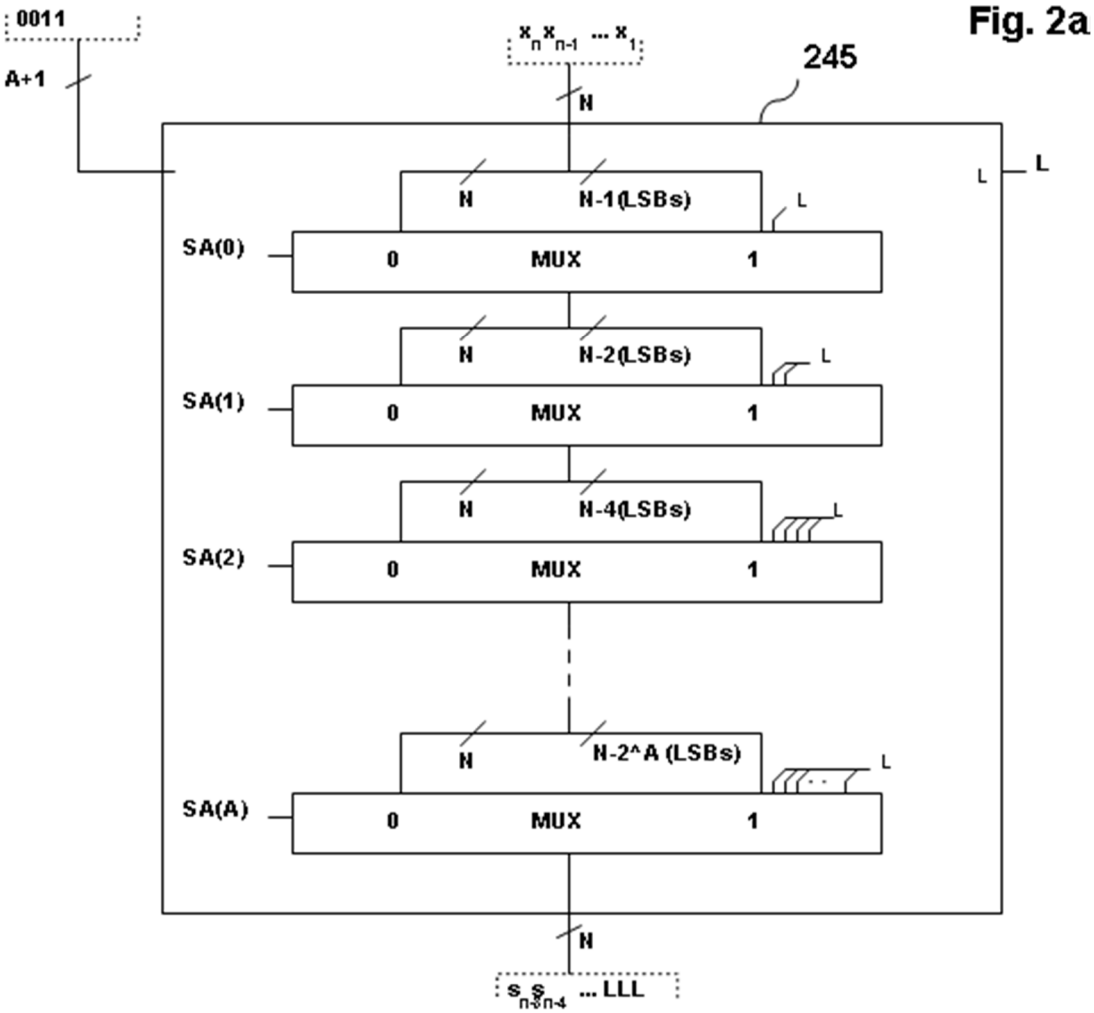

0011
A+1
$x_n x_{n-1} \ldots x_1$
245
N
L
L
N
N-1(LSBs)
L
SA(0)
0
MUX
1
N
N-2(LSBs)
L
SA(1)
0
MUX
1
N
N-4(LSBs)
L
SA(2)
0
MUX
1
N
N-2^A (LSBs)
L
SA(A)
0
MUX
1
N
$s_n s_{n-1} \ldots s_{n-4}$ ... LLL

Fig. 2a

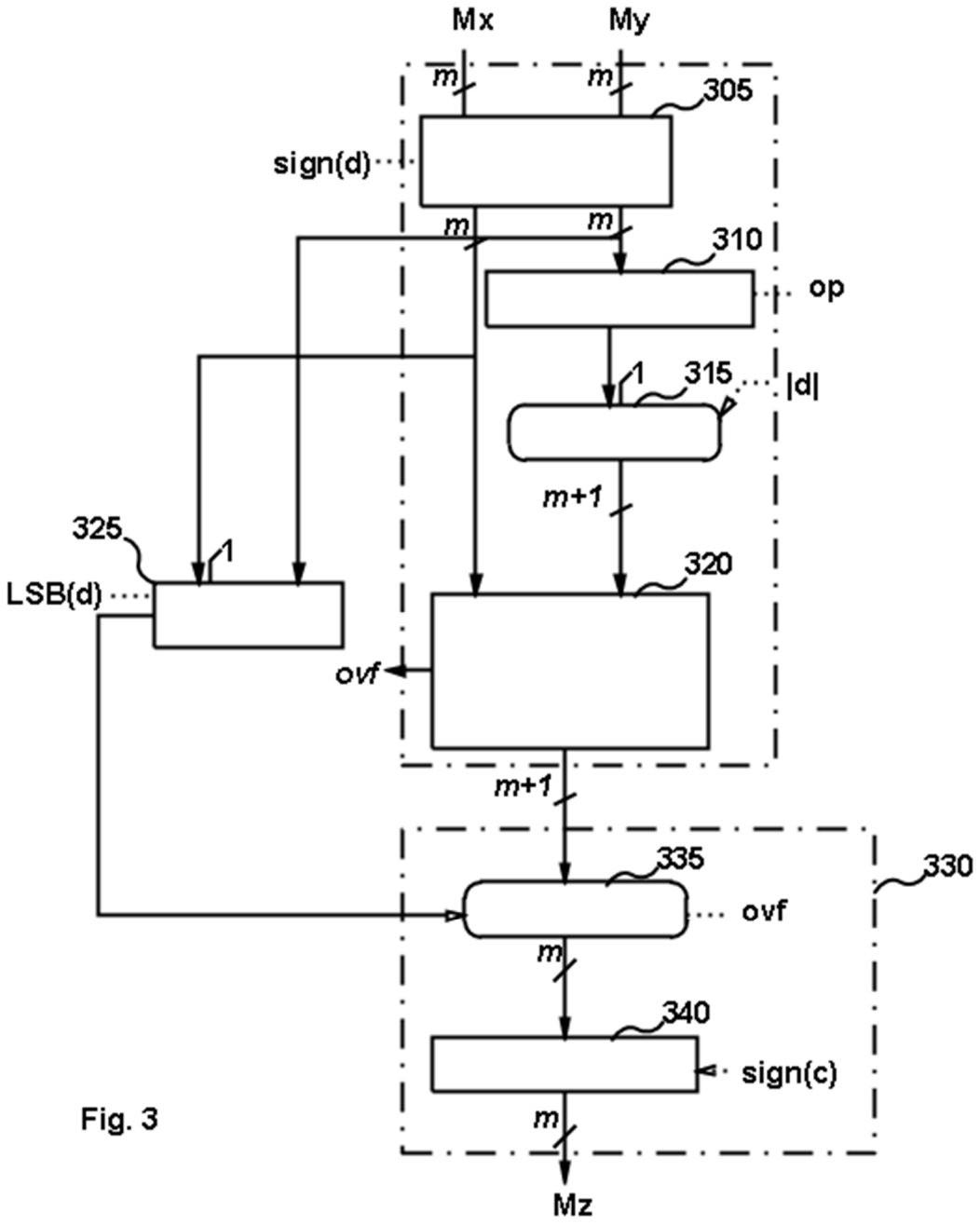

Mx
My
m
m
305
sign(d)
m
m
310
op
1
315
|d|
m+1
325
1
LSB(d)
320
ovf
m+1
330
335
ovf
m
340
sign(c)
m
Mz

Fig. 3

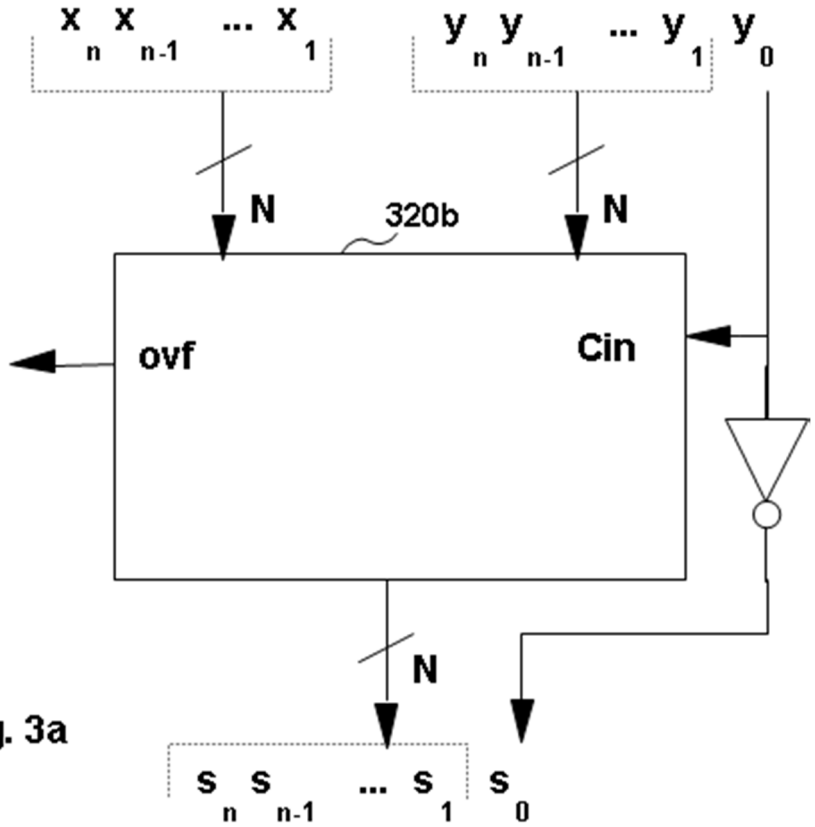


Fig. 3a

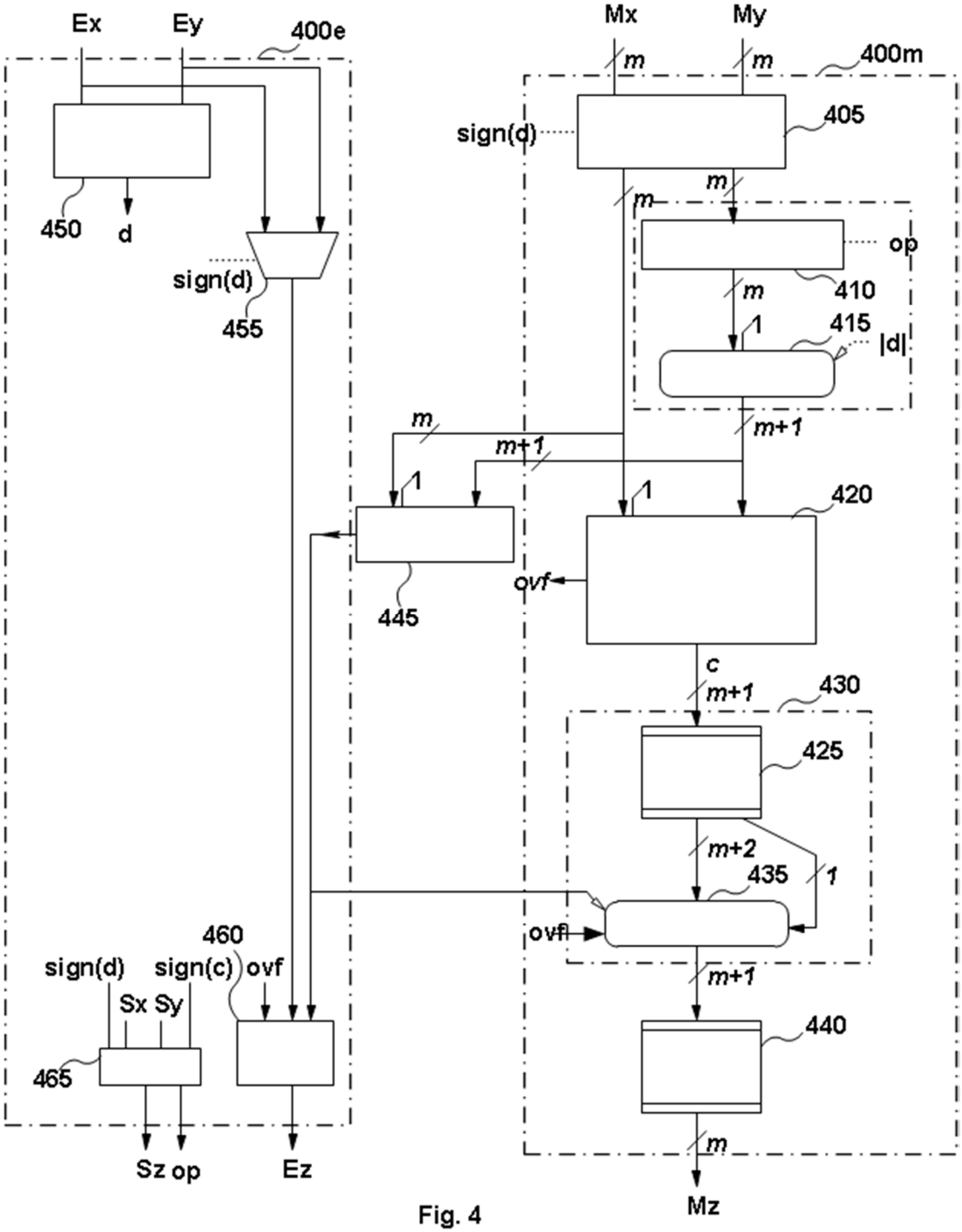
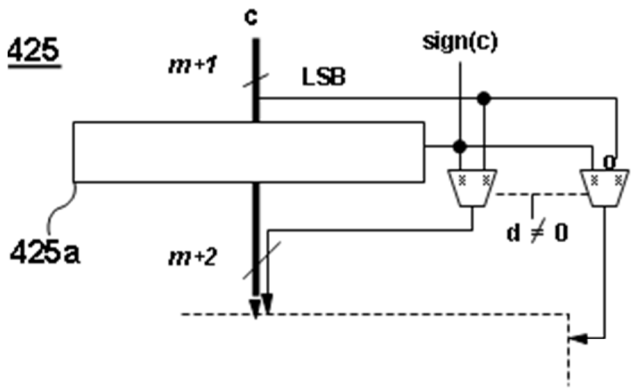

Ex
Ey
400e
450
d
sign(d)
455
Mx
My
400m
m
sign(d)
405
op
410
415
|d|
1
m+1
420
ovf
445
c
430
425
m+2
435
ovfl
460
sign(d)
sign(c)
ovf
Sx
Sy
465
Sz
op
Ez
440
Mz

Fig. 4

Fig. 4a



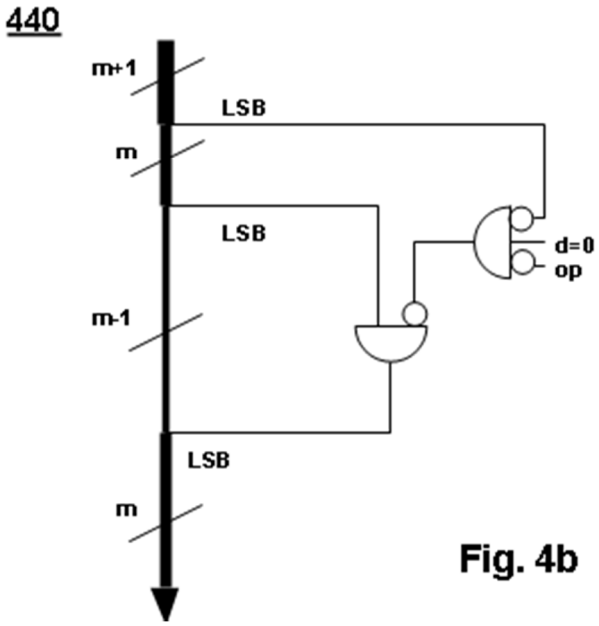


Fig. 4b

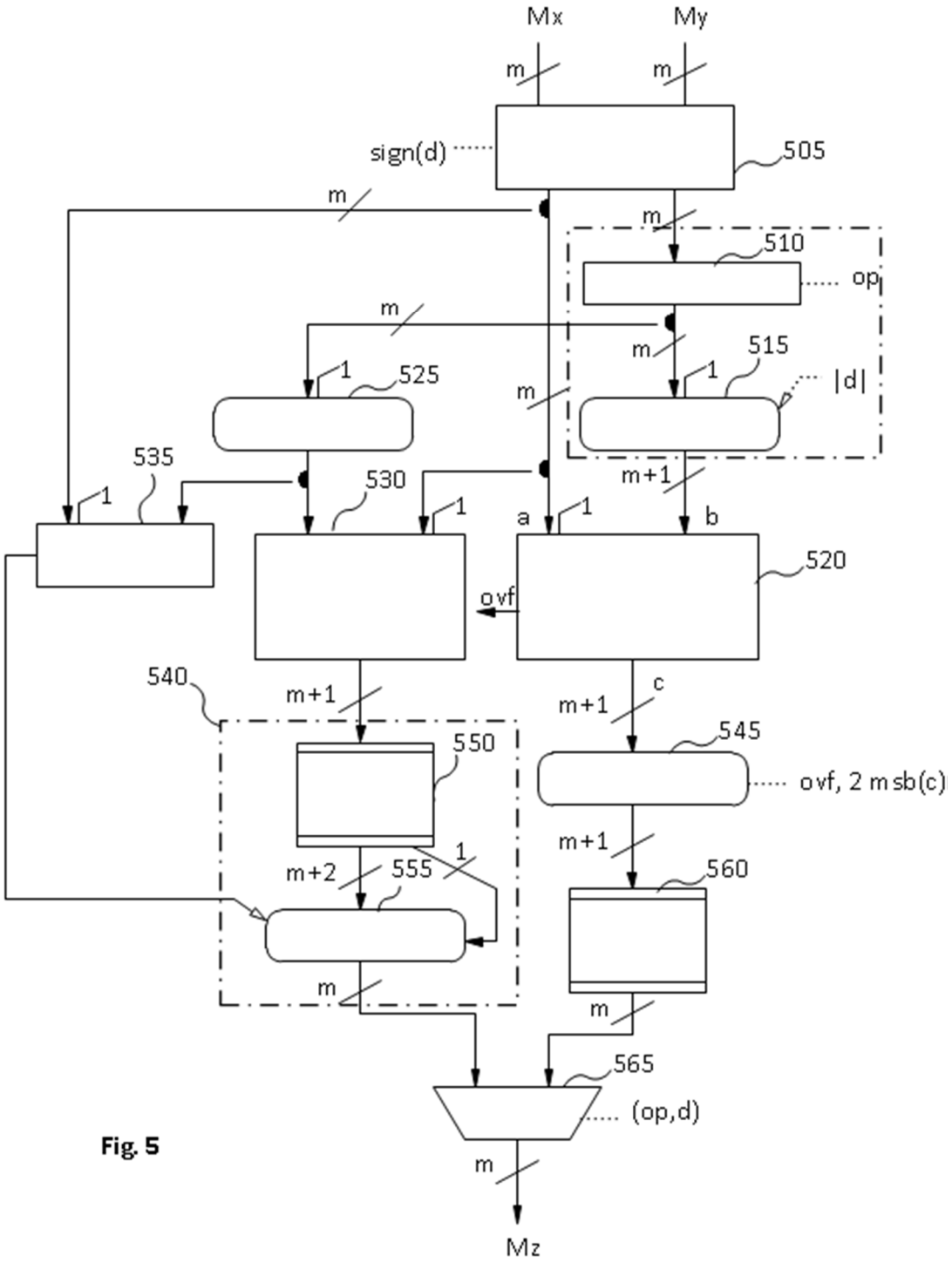

Mx
My
m
m
sign(d)
505
m
m
510
op
m
m
515
1
525
1
|d|
m
535
530
m+1
1
1
a
1
b
520
ovf
540
m+1
m+1
c
550
545
ovf, 2 msb(c)
m+1
1
m+2
555
560
m
m
565
(op,d)
m
Mz

Fig. 5

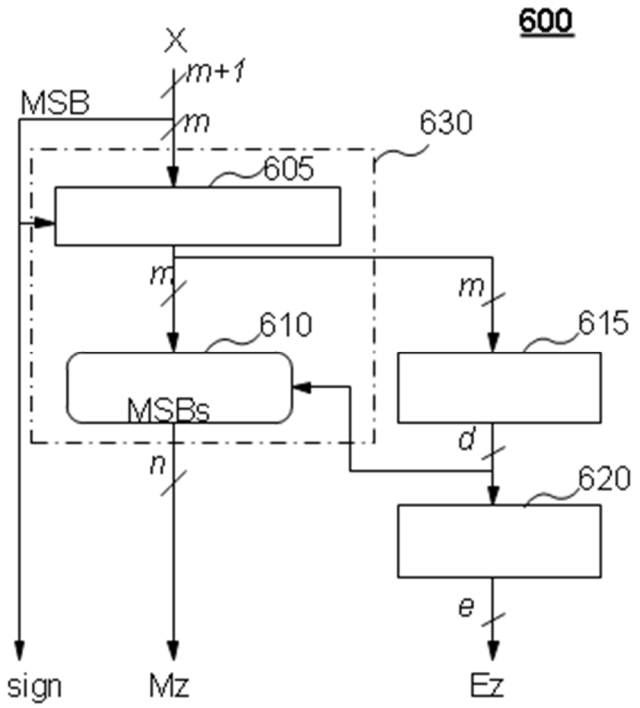


Fig. 6

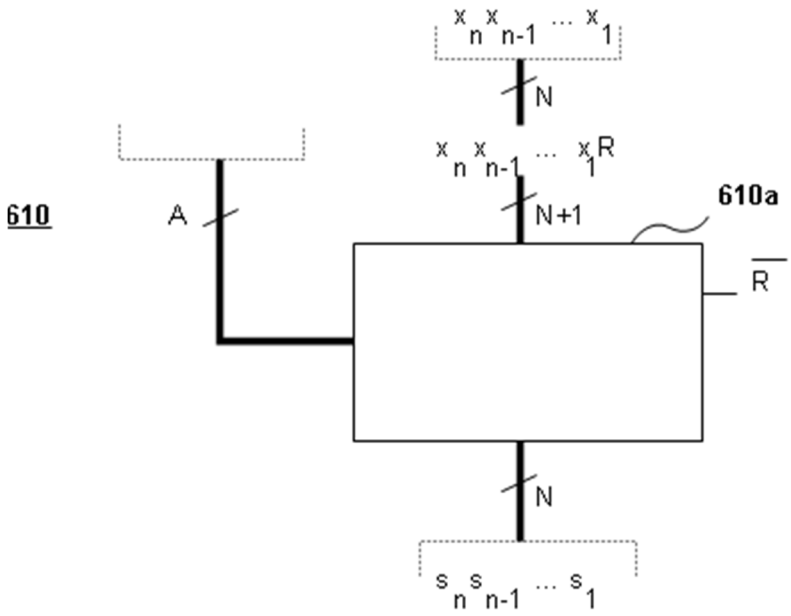


Fig. 6a

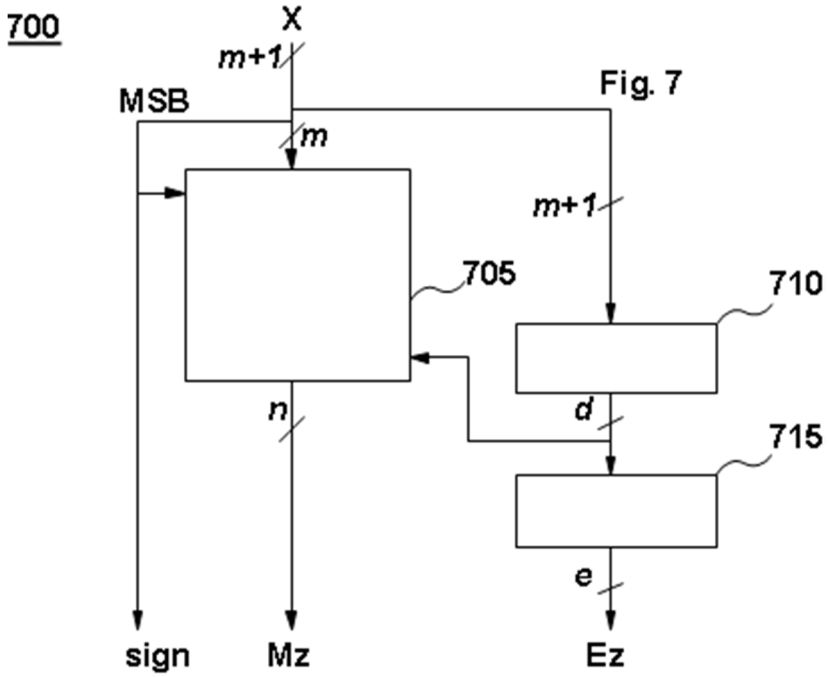


Fig. 7

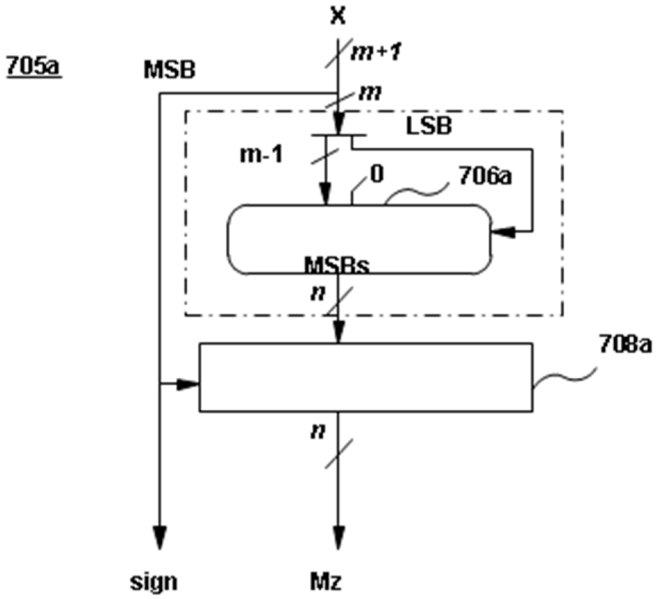
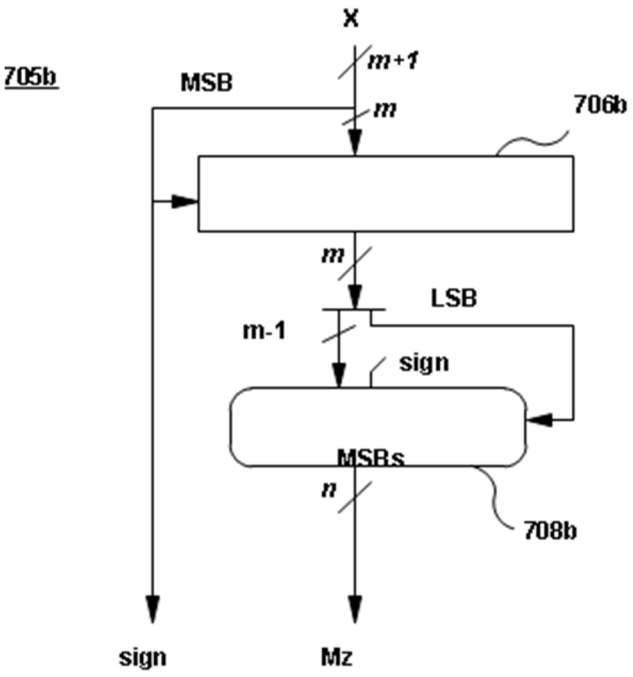

X
m+1
705a
MSB
m
LSB
m-1
0
706a
MSBs
n
708a
n
sign
Mz

Fig. 7a

Fig. 7b


X
m+1
705b
MSB
m
706b
m
LSB
m-1
sign
MSBs
n
708b
sign
Mz

800a

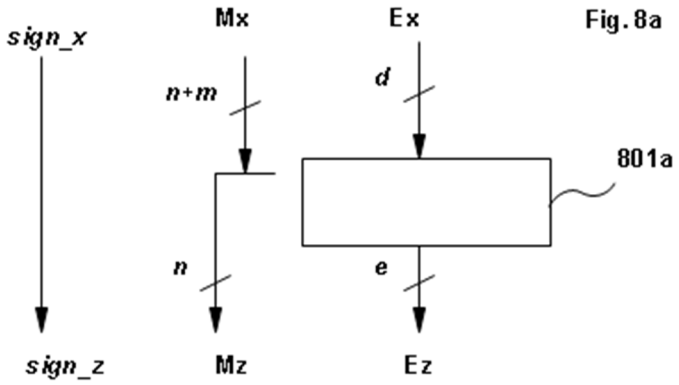


Fig. 8a

800b

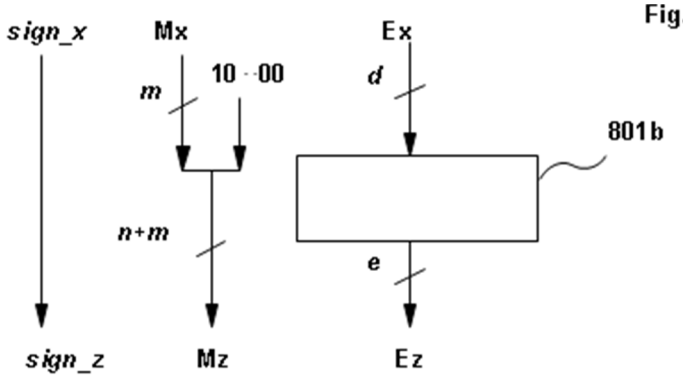


Fig. 8b

800c

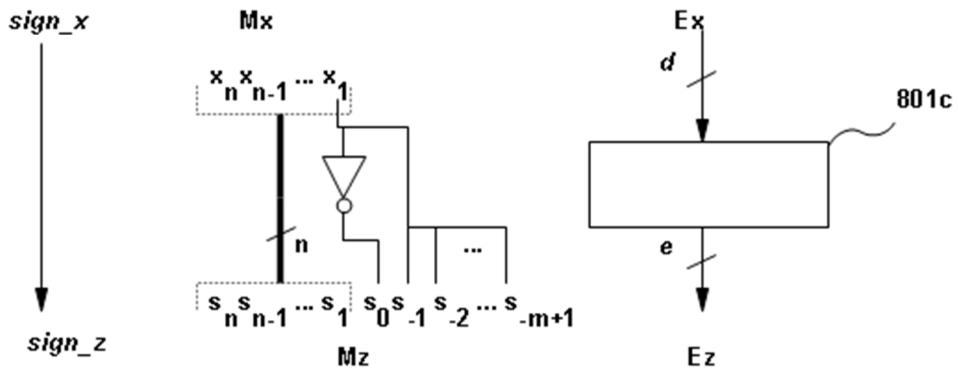


Fig. 8c

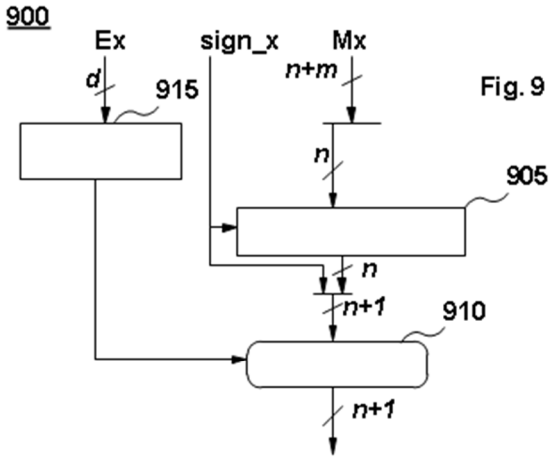


Fig. 9

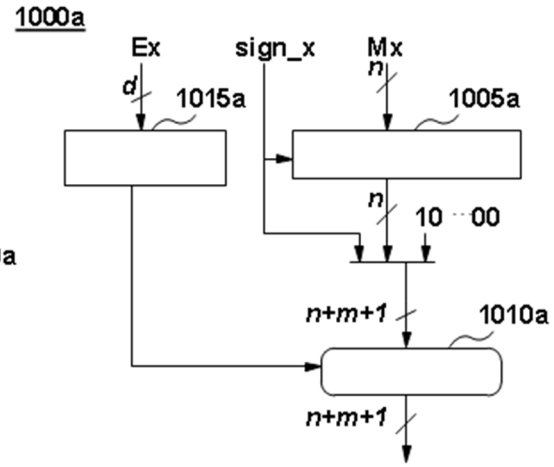


Fig. 10a

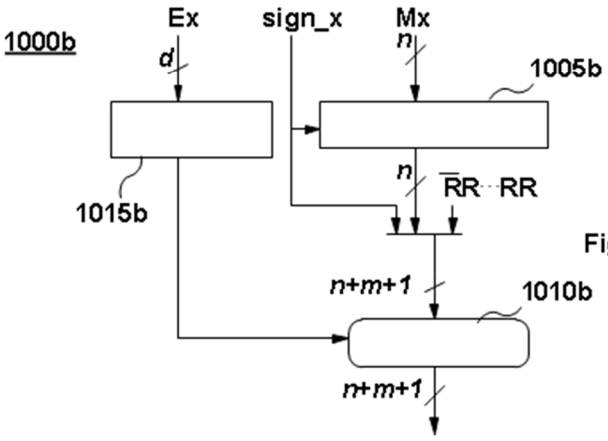


Fig. 10b

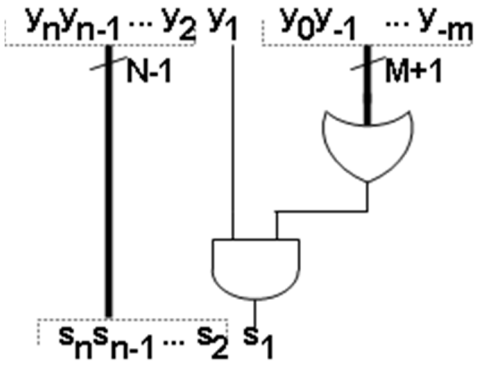


Fig. 11

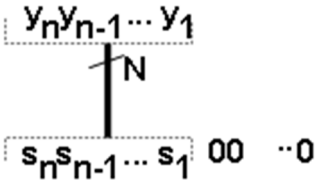


Fig. 12a

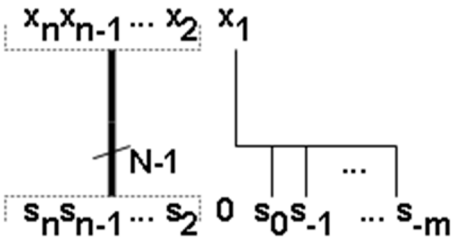


Fig. 12b

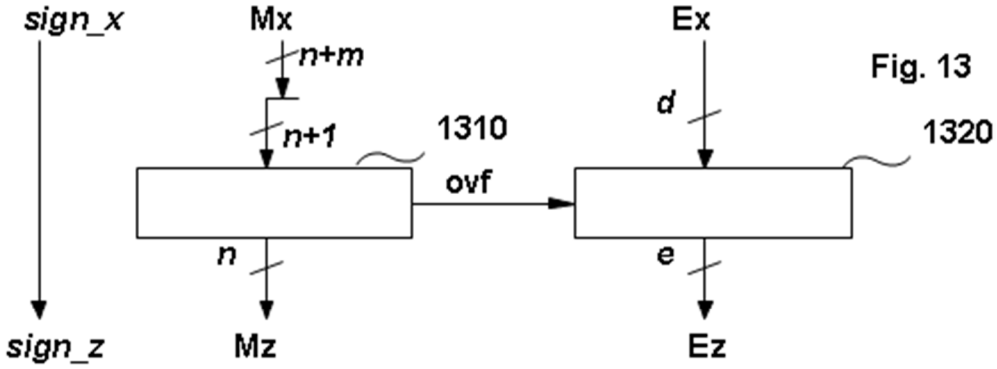


Fig. 13

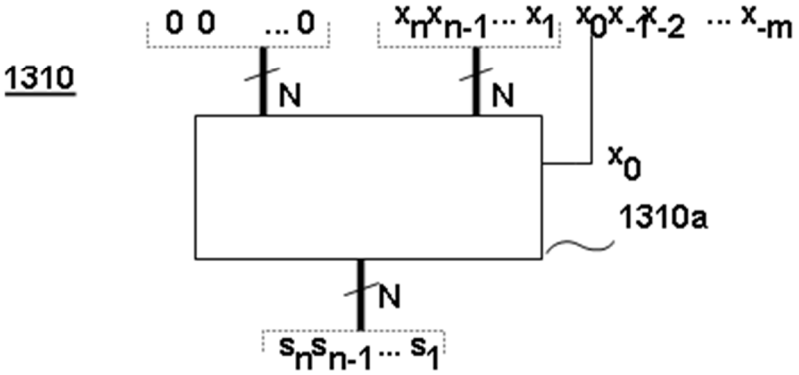
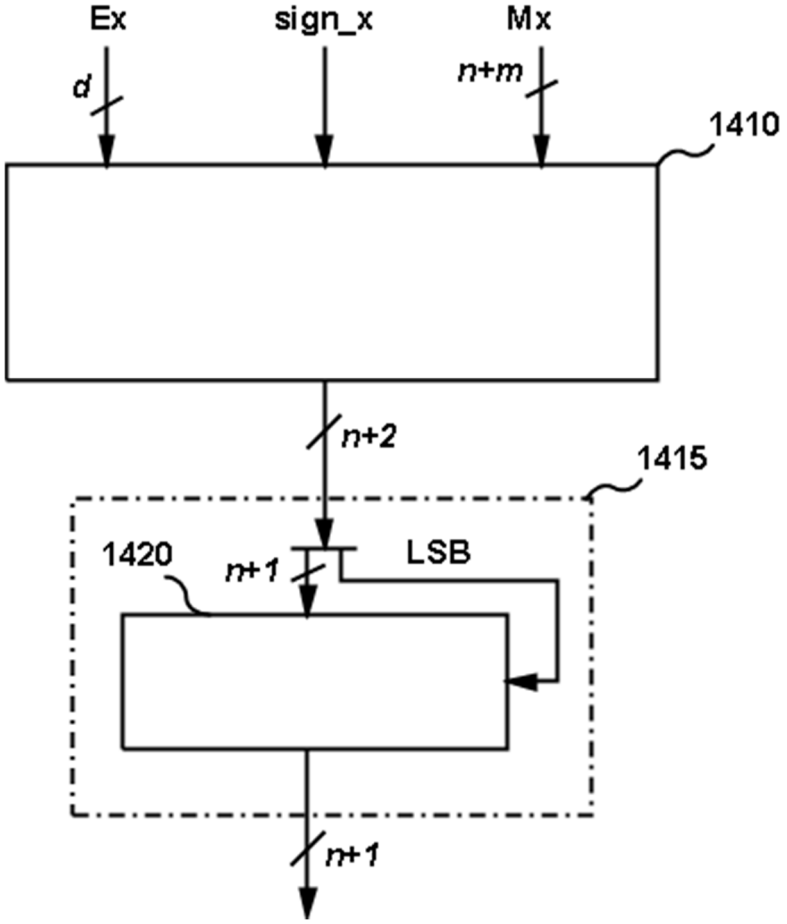


Fig. 13a

Fig. 14



OFICINA ESPAÑOLA
DE PATENTES Y MARCAS

ESPAÑA

(21) N.º solicitud: 201430451

(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. 13th An nual 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	**Examinador**	**Página**
23.12.2014	M. Muñoz Sánchez	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, XPIEE, XPI3E, NPL

Fecha de Realización de la Opinión Escrita: 23.12.2014

Declaración

Novedad (Art. 6.1 LP 11/1986)	Reivindicaciones	1-59	**SI**
	Reivindicaciones		**NO**
Actividad inventiva (Art. 8.1 LP11/1986)	Reivindicaciones	1-59	**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: 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: 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: 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 2-59: 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.