(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 546 899**

(21) Número de solicitud: 201430454

(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: **Dispositivos para operaciones de multiplicación-suma fusionadas en coma flotante y conversores asociados**

ES 2 546 899 A1

(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 546 899**

(21) Número de solicitud: 201430454

(57) Resumen:

Dispositivos para realizar una operación de multiplicación-suma fusionada en coma flotante entre tres números coma flotante pre-procesados para generar un cuarto 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 configurado para recibir los exponentes de los tres números pre-procesados y generar el exponente del resultado de la operación de multiplicación-suma en coma flotante y un camino de datos de la mantisa. El camino de datos de la mantisa comprende un camino de multiplicación y un camino de suma. El camino de multiplicación comprende una primera entrada configurada 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 multiplicación está configurado para multiplicar dichas mantisas pre-procesadas del primer y segundo número y generar un resultado de la multiplicación en una salida. El camino de suma está configurado para recibir como mucho los m+1 MSDs de la mantisa pre-procesada del tercer número en una primera entrada y el resultado de la multiplicación en una segunda entrada y generar como mucho los m+1 MSDs de la mantisa del cuarto número pre-procesado.

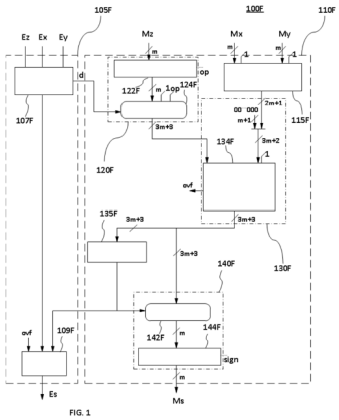


FIG. 1

ES 2 546 899 A1

DESCRIPCIÓN

Dispositivos para operaciones de multiplicación-suma fusionadas en coma flotante y conversores asociados.

La presente invención se refiere al procesamiento de datos y más concretamente a dispositivos para realizar una operación de multiplicación-suma fusionadas 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 de la técnica anterior se han hecho varios intentos para reducir los efectos de estos cálculos, es decir el complemento a dos, el cálculo del sticky bit y redondeo. En ciertos documentos del estado de la técnica se ha propuesto precalcular 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 operación de multiplicación-suma fusionadas en coma flotante entre tres números coma flotante pre-procesados para generar un cuarto número en coma flotante pre-procesado. Cada número tiene una mantisa pre-procesada de m+2 dígitos. El dispositivo comprende un camino de datos del exponente, configurado para recibir los exponentes de los tres números pre-procesados de entrada y generar el exponente del resultado de la operación de multiplicación-suma en coma flotante, y un camino de datos de la mantisa. El camino de datos de la mantisa comprende un camino de multiplicación y un camino de suma. El camino de multiplicación 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 multiplicación está configurado para multiplicar dichas mantisas pre-procesadas del primer y segundo número y generar un resultado de la multiplicación en una salida. El camino de suma está configurado para recibir como mucho los m+1 MSDs de la mantisa pre-procesada del tercer número en una primera entrada y el resultado de la multiplicación en una segunda entrada y generar como mucho los m+1 MSDs de la mantisa del cuarto 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. Cuando B=2, los dígitos son bits.
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 el sticky bit es siempre uno puesto que el último dígito oculto siempre es necesariamente B/2 (dígito KD) tanto en la mantisa del tercer número como en el resultado del producto. Por último, otra ventaja es que no puede ocurrir desbordamiento después del redondeo, puesto que éste no se realiza por exceso.

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 la tercera mantisa y el resultado de la multiplicación según los signos de las entradas; calcular el exponente de la salida; calcular el signo de la salida; y detectar y resolver excepciones, tal como desbordamientos y valores especiales, de las entradas o de dicha operación.

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, segunda y tercera entrada podrían estar configuradas para recibir los m MSD de la parte fraccionaria de la primera, segunda y tercera mantisa pre-procesada, respectivamente.

En algunas realizaciones, el dispositivo puede comprender además una cuarta entrada para recibir el LSD de dichas primera, segunda y tercera mantisa pre-procesadas. Alternativamente, la cuarta 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.

Aunque las siguientes descripciones de las realizaciones representan circuitos diseñados para lógica binaria, alguien experto en el estado de la técnica, podría aplicar las enseñanzas divulgadas aquí también para circuitos lógicos no binarios de una forma directa.

En algunas realizaciones, el camino de suma podría comprender un primer módulo de desplazamiento configurado para recibir como mucho los m+1 MSBs de la tercera mantisa pre-procesada en una primera entrada. 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 primer módulo de desplazamiento podría estar configurado además para recibir una instrucción desde al camino de datos del exponente sobre la primera cantidad de desplazamiento y la operación efectiva entre la tercera mantisa pre-procesada y la salida del camino de multiplicación, y alinearlos como corresponde. El camino de la suma podría comprender además un módulo de suma configurado para sumar la salida alineada del primer módulo de desplazamiento con la salida del camino de multiplicación. En estas realizaciones, el LSB de la tercera mantisa no es necesario recibirlo explícitamente para alinear la mantisa.

En algunas realizaciones, el camino de multiplicación podría comprender un módulo de multiplicación configurado para recibir, en una entrada, como mucho los m+1 MSBs de la mantisa del primer y segundo número pre-procesado, respectivamente, y generar los 2*m+3 MSBs del valor que corresponde a la operación de multiplicación entre dichas mantisas pre-procesadas en una salida. Si los números están normalizados entonces este podría recibir solamente los m LSBs de los m+1 MSBs, ya que el MSB de un número normalizado es siempre 1 y no necesita recibirse. En otro caso, este podría recibir todos los m+1 MSBs.

En algunas realizaciones, el camino de multiplicación podría comprender un multiplicador redundante configurado para recibir, en una primera y una segunda entrada, como mucho los m+1 MSBs de la mantisa del primer y segundo número pre-procesado pre-proc, respectivamente, y generar, en una representación redundante, los 2*m+3 MSDs del valor que

corresponde a la operación de multiplicación entre dichas mantisas pre-procesadas. De nuevo, si los números están normalizados entonces este podría recibir solamente los m LSBs de los m+1 MSBs, ya que el MSB de un número normalizado es siempre 1 y no necesita recibirse. En otro caso, este podría recibir todos los m+1 MSBs.

No solo las realizaciones con un módulo de multiplicación sino también las realizaciones con un multiplicador redundante tienen la ventaja de que el LSB de los operandos de entrada no se requieren explícitamente y el LSD (o LSB) de la salida no es necesario generarlo. En algunas implementaciones, un multiplicador coma fija estándar con dos entradas de m+2 bits podría usarse fijando el LSB de dichas dos entradas a uno y los bits restantes igual a las entradas de dicho módulo multiplicador, mientras, en otras implementaciones, el LSB implícito podría tenerse en cuenta internamente en el multiplicador. Un argumento similar es válido para el multiplicador redundante.

En algunas realizaciones el multiplicador redundante podría comprender un generador de productos parciales dispuesto para recibir, en una primera y una segunda entrada, como mucho los m+1 MSBs de la mantisa del primer y segundo número pre-procesado, respectivamente, y generar sus productos parciales en una salida. Además, el multiplicador redundante podría comprender un árbol de compresores, con una primera entrada conectada a la salida del generador de productos parciales y una segunda entrada dispuesta para recibir como mucho los m+1 MSBs de la mantisa del primer y segundo número pre-procesado, dicho árbol de compresores dispuesto para generar, en una representación redundante, los 2*m+3 MSDs de un valor correspondiente a la operación de multiplicación entre dichas mantisas pre-procesadas en una salida. Como el LSB de las mantisas pre-procesadas es siempre igual a uno, el generador de productos parciales no requiere generar productos parciales para los LSBs y podría considerarse que ya están generados. Ellos se introducen directamente en el árbol de compresores lo que resulta en menos operaciones y lógica para el generador de productos parciales.

En algunas realizaciones el módulo de multiplicación podría comprender además una tercera entrada con el valor 1.

En algunas realizaciones el primer módulo de desplazamiento podría estar configurado para recibir como mucho los m+1 MSBs de la mantisa de tercer número pre-procesado 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.

En algunas realizaciones, el primer módulo de desplazamiento podría estar configurado 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 además una tercera entrada con el valor uno para agregar explícitamente el LSB de la mantisa antes de desplazarla.

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, el cual debería implementarse con extensión de signo, se coloca después del inversor de bits condicional y no se requiere lógica adicional, ya que el LSB de la mantisa se añade después del circuito inversor. En otras implementaciones, el desplazador a la derecha está colocado delante del inversor de bits condicional y podría requerir una lógica adicional para sumar uno en el LSB de la salida después de la inversión, ya que dicha salida no es un número pre-procesado.

En algunas realizaciones, el módulo de suma podría comprender un sumador configurado para recibir la salida del camino de multiplicación en una primera entrada y la salida del primer módulo de desplazamiento en una segunda entrada y generar un valor correspondiente a la suma con signo del resultado de la multiplicación entre las mantisas del primer y segundo número pre-procesado, y la mantisa alineada del tercer número pre-procesado, en una salida.

En algunas realizaciones, dicho sumador podría estar configurado para recibir los 2*m+3 MSBs de la multiplicación de la mantisa del primer y segundo número pre-procesado, en una primera entrada, y la salida del primer módulo de desplazamiento, en una segunda entrada, y generar un valor correspondiente a la suma con signo de dicha multiplicación y el valor de la segunda entrada, en una salida. En otras realizaciones, dicho sumador podría estar configurado para recibir los 2*m+3 MSDs de la multiplicación de la mantisa del primer y segundo número pre-procesado, en un formato de representación redundante, en una primera entrada, y la salida del primer módulo de desplazamiento en una segunda entrada y generar un valor correspondiente a la suma con signo de dicha multiplicación y el valor de la segunda entrada, en una salida. Implementaciones de acuerdo a las realizaciones ilustradas aquí podrían tener la ventaja de que el LSB (o LSD) de dicho resultado de multiplicación no se recibe explícitamente. En algunas implementaciones el sumador podría estar dispuesto para incorporar explícitamente dicho LSB, el cual es siempre uno, antes que se realice la operación efectiva. En otras implementaciones, el sumador podría estar dispuesto para tener en cuenta dicho LSB internamente cuando se realice la operación efectiva.

En algunas realizaciones, dicha suma con signo podría comprender n bits, n>m, y dicho sumador podría estar configurado para generar como mucho los n-1 MSBs de dicha suma con signo, en una primera salida. El LSB podría estar implícito cuando es igual a uno, o podría ser no requerido en ciertos casos. En algunas realizaciones, dicho sumador podría estar además configurado para generar el LSB de dicha suma con signo, en una segunda salida. En algunas implementaciones, dichos n bits podrían estar alineados con el resultado de la multiplicación, es decir, el LSB de dichos n bits tiene el mismo peso que el LSB del resultado de la multiplicación. Sin embargo, en otras implementaciones, bits con menos peso podrían considerarse, aunque ellos no contribuirían a obtener un resultado final con más precisión. Similarmente, en otras implementaciones, el LSB de dichos n bits podría tener un mayor peso que el LSB del resultado de la multiplicación, pero el resultado final podría ser menos preciso en ciertos casos. En algunas implementaciones, n podría ser igual a 3*m+6 y una señal podría ser generada para detectar el desbordamiento. En otras implementaciones, n podría ser igual a 3*m+7, y el MSB podría ser el bit de signo y no se requeriría señal de desbordamiento.

En algunas realizaciones, el camino de datos de la mantisa podría comprender además un módulo de normalización, teniendo 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 dispuesto para generar como mucho los m+1 MSBs de la cuarta mantisa pre-procesada mediante el desplazamiento selectivo a la izquierda de la salida del módulo de suma. Como la salida es un número pre-procesado, el redondeo al más cercano puede realizarse mediante un simple truncado, pero cierto sesgo puede aparecer después de redondear.

En algunas realizaciones, el módulo de normalización 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, es igual a uno. En algunas implementaciones este bit o estos bits podrían seleccionarse de la primera entrada del módulo de normalización. En otras implementaciones, una nueva entrada podría configurarse. Esta configuración permite al módulo de normalización eliminar el sesgo del redondeo.

En algunas realizaciones, el módulo de normalización podría estar configurado además para completar selectivamente las posiciones vacantes debidas al desplazamiento a la izquierda, poniéndolas a cero, o poniendo a cero el MSB de dichas posiciones y el resto a uno, o poniendo a uno el MSB de dichas posiciones y el resto a cero. Esta configuración permite al módulo de normalización proveer el resultado correcto en ciertos casos, tal como cuando el LSB del resultado de la suma está implícito.

En algunas realizaciones, el módulo de normalización podría estar configurado además para completar selectivamente dichas posiciones vacantes, aleatoriamente, basándose en un bit concreto, o en una combinación de bits concretos, con las adecuadas características estadísticas. En algunas implementaciones este bit o estos bits podrían seleccionarse de la primera entrada del módulo de normalización. En otras implementaciones, una nueva entrada podría configurarse. Esta configuración permite al módulo de normalización eliminar el sesgo del redondeo.

Los módulos de normalización configurados de acuerdo a las realizaciones descritas aquí permiten realizar el redondeo al más cercano sin sesgo en ciertos casos. Uno de tales casos es después de una operación FMAD, cuando la normalización requiere un desplazamiento a la izquierda de más de 2*m+2 bits. Completar las posiciones vacantes a la derecha con ceros produce un redondeo efectivo hacia arriba y en consecuencia algún sesgo. Como, en este caso, el LSB del resultado de la suma(o resta) es siempre uno, el módulo de normalización podría ser fácilmente configurado, como se describió anteriormente, para producir aleatoriamente un redondeo hacia abajo que eliminaría dicho sesgo. Si dicho LSB es recibido explícitamente, esto se realiza restando aleatoriamente 1 del LSB del valor desplazado. Ahora bien, si no se recibe el LSB explícitamente esto podría lograrse poniendo aleatoriamente o bien el MSB de las posiciones vacantes a cero y el resto a uno o bien poniendo el MSB de las posiciones vacantes a uno y el resto a cero. Las mismas soluciones pueden utilizarse cuando la operación es una suma única y el exponente del tercer número de entrada es mayor que el exponente del otro sumando. Llamamos suma única al caso cuando, o bien el primero, o bien el segundo número de entrada, es igual a uno, y como resultado la operación FMAD es, a efectos prácticos, tan sólo una suma entre el tercer número de entrada y el número de entrada que no es uno. Del mismo modo, otro caso en el que se podría producir sesgo es si después de una suma única, cuando el exponente del tercer número de entrada es uno menos que el exponente del otro sumando, la normalización requiere un desplazamiento a la izquierda de más de 2 * m + 2 bits. En este caso, el sesgo podría evitarse poniendo aleatoriamente o bien el MSB de los posiciones vacantes a cero y el resto a uno o bien poniendo el MSB de los posiciones vacantes a uno y el resto a cero, ya que el LSB del resultado de la suma es implícito e igual a uno. Finalmente, otro caso es después de una suma única, cuando el exponente del tercer número de entrada y el exponente del otro sumando son iguales. Como en este caso el resultado de la suma podría ser positivo o negativo y su LSB es cero, el sesgo podría evitarse de dos maneras. Una forma es simplemente completar los posiciones vacantes con ceros. Otra forma es completando con ceros y, además, restando uno del LSB del valor desplazado si un bit seleccionado, o combinación de éstos, del resultado de la suma única es uno.

En algunas realizaciones, el módulo de normalización podría estar configurado además para forzar cero el segundo LSB del valor que corresponde a la mantisa del cuarto número pre-procesado cuando la operación es una suma única, el tercer número de entrada y el otro sumando tienen el mismo exponente y signo, y los valores del segundo LSB de las mantisas pre-procesadas de dichos operandos son diferentes. Esto permite eliminar el sesgo en el redondeo para la suma única alineada.

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 operación de resta posterior. Esto permite una salida positiva cuando el módulo de suma proporciona un número pre-procesado negativo. Como es un número pre-procesado, esta negación podría ser implementada simplemente invirtiendo todos los bits excepto el LSB y no se requiere ninguna suma. El sumador podría proveer un número no procesado negativo solamente cuando realiza una suma única de dos números con el mismo exponente y signos diferentes. En este caso, la inversión de bits cambiaría el signo y también eliminaría el sesgo del redondeo. En implementaciones alternativas, la mantisa del formato coma flotante podría ser con signo y la negación no sería necesaria.

En algunas implementaciones, el camino de datos del exponente podría ser configurado para distinguir entre una operación multiplicación-suma fusionada, o una multiplicación única, o una suma única. La multiplicación única podría ser reconocida si el tercer número de entrada es el valor especial cero, y el dispositivo podría ser instruido para producir el resultado de una multiplicación única. En algunas implementaciones, la suma única podría ser reconocida si, o bien el primero, o bien el segundo, número de entrada es un valor especial uno, mientras que en otras, podría ser reconocido por una instrucción externa. En algunas implementaciones el camino de multiplicación podría ser instruido para generar una salida correspondiente a la mantisa del primer o segundonúmero, si se reconoce una suma única. En algunas implementaciones, el módulo de normalización podría instruirse, si se reconociera una suma única, para generar una salida en consecuencia.

En algunas implementaciones, 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 de salida.

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 multiplicación-suma permite que un número en formato de coma fija pre-procesado sea procesado por dispositivos de

multiplicación-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 de un número no procesado en coma fija y generar como mucho los M+1 MSBs de la mantisa del número pre-procesado en coma flotante, y un calculador de exponentes con una primera entrada para recibir la cuarta cantidad de desplazamiento proveniente de dicho calculador de cantidad de desplazamiento y una salida para generar el exponente del número pre-procesado en coma flotante. El signo del número pre-procesado en coma flotante podría corresponder con el MSB del número en coma fija no procesado. Introduciendo un conversor de este tipo antes del módulo de multiplicación-suma permite que un número en formato de coma fija no-procesado sea procesable por dispositivos de multiplicación-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 MSBs 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, iє[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 cuarto número en coma flotante pre-procesado a un cuarto 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 a salida si el LSB de la dicha salida es igual a 1. Introduciendo un conversor de este tipo después de los dispositivos de acuerdo a las realizaciones descritas aquí permite que el resultado de las operaciones sea usado por circuitos que funcionan con formato coma fija no procesado.

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 FMAD 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 FMAD, 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 convertidor 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 K 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 de un 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 podría es 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 además configurado 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 un circuito de multiplicación-suma fusionadas (FMAD) en coma flotante de acuerdo a un ejemplo

Fig. 2 y 2b ilustran ejemplos de implementación de un multiplicador en coma fija pre-procesado;

Fig. 3 ilustra un circuito FMAD en coma flotante de acuerdo a otro ejemplo, el cual elimina el sesgo y está optimizado en velocidad

Fig. 3a and 3b ilustran ejemplos de implementación del módulo de desplazamiento a la izquierda de un circuito FMAD;

Fig. 3c ilustra un ejemplo de implementación de un desplazador a la izquierda especial;

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

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

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

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

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

Fig. 7, 8a and 8b ilustran ejemplos de implementación de un conversor de números coma flotante pre-procesados a números coma fija pre-procesados;

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

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

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

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

DESCRIPCION DETALLADA DE LAS REALIZACIONES

Fig. 1 ilustra un circuito de multiplicación-suma fusionadas (FMAD) en coma flotante (FP) de acuerdo a un ejemplo. FMAD 100F recibe tres números en coma flotante pre-procesados X, Y, y Z, y genera un resultado S que es la suma del tercer número coma flotante con el producto de los otros dos (S=Z+X*Y). El LSB de las mantisas es igual a uno. FMAD 100 comprende un camino de datos del exponente 105F y un camino de datos de la mantisa 110F. El camino de datos del exponente 105F comprende una lógica de exponente 107F para recibir los exponentes Ex, Ey, Ez de los tres números FP y genera un valor intermedio de exponente en una salida, de acuerdo al máximo valor entre Ez y Ex+Ey. La salida de la lógica de exponente 107F está conectada a la primera entrada del módulo de actualización de exponente 109F. Una segunda entrada del módulo de actualización de exponente 109F está conectada al camino de datos de la mantisa 110F para recibir el número de ceros por la izquierda del resultado de la operación de suma o el número de unos por la izquierda si dicho resultado es negativo. Una tercera entrada del módulo de actualización de exponente 109F está conectada al camino de datos de la mantisa 110F para recibir un bit de desbordamiento (ovf). En una implementación alternativa, las dos últimas entradas, es decir, el número de bits no significativos por la izquierda y el bit de desbordamiento podrían combinarse en un único valor. El módulo de actualización de exponente 109F está configurado para generar el exponente Es del número coma flotante S, incrementando o decrementando el valor intermedio de exponente de acuerdo al número de bits no significativos por la izquierda y la señal de desbordamiento. Además un circuito lógico de signo calcula la señal de operación efectiva (op) para la suma final y el signo del resultado, de una forma estándar, basándose en el signo de las entradas y en el signo del resultado de la suma final.

El camino de datos de la mantisa 110F comprende un módulo de multiplicación 115F para recibir los m MSBs de las mantisas de los números FP pre-procesados X e Y. Las mantisas se representan por los símbolos Mx y My en Fig. 1. Las mantisas Mx y My (así como Mz) ambas

tienen m+1 bit. Sin embargo como ambas mantisas pertenecen a números pre-procesados, el LSB de ambas mantisas es igual a uno y no necesita ser introducido en el FMAD a la entrada. Además, en el ejemplo de Fig. 1 los tres número coma flotante están normalizados. Sin embargo, para simplificar la descripción, el MSB del número normalizado, se incluye en los m bits que se introducen en FMAD 100F. En una implementación alternativa, este bit podría omitirse en las entradas e introducirse, o bien antes del módulo de multiplicación 115F, o bien internamente a dicho módulo de multiplicación 115F, para Mx y My, y, o bien antes del primer módulo de desplazamiento 120F, o bien internamente a dicho módulo, para Mz. En el ejemplo de Fig. 1 el LSB de las mantisas de entradas son introducidas como una entrada separada del módulo de multiplicación 115F. Alternativamente, este podría estar implícito e introducirse dentro del módulo de multiplicación 115F. Este es meramente ilustrado en el ejemplo de Fig. 1 y otros ejemplos posteriores, para indicar la necesidad de la introducción funcional del LSB implícito. El módulo de multiplicación 115F recibe los m MSBs de las mantisas Mx y My y genera los 2*m+1 MSBs del producto de las mantisas de X e Y (incluyendo su bit implícito) en un valor de salida. El LSB de dicho producto es siempre uno y no se requiere explícitamente. Dicho de otra forma, si los m MSBs de Mx se representan con A, y los m MSBs de My se representan con B, entonces el valor de 2*m+1 bits en la salida es igual a A*B+1/2A+1/2B.

Una implementación de módulo de multiplicación 115F se ilustra en Fig. 2. El módulo de multiplicación representado en Fig. 2 recibe los m MSBs de dos números coma fija pre-procesados (mantisas), ya que el LSB es constante e igual a uno, y genera los 2*m+1 MSBs del resultado de la multiplicación de ambos números de entrada, siendo el LSB de dicho resultado también implícito e igual a uno. El multiplicador coma fija 300M comprende un multiplicador redundante 305 y un módulo de conversión 335.

El multiplicador redundante 305 comprende un módulo generador de productos parciales 325 y un árbol de compresores 330. El módulo generador de productos parciales 325 recibe dichos m MSBs de los dos números coma fija pre-procesados, en una primera y una segunda entrada, respectivamente, y genera todos los productos parciales correspondientes al producto de la primera entrada por cada bit de la segunda entrada. En una implementación alternativa, la segunda entrada podría estar dividida en varios grupos de bits y los productos parciales generados podrían corresponder a los productos de la primera entrada por cada uno de dichos grupos de bits. El árbol de compresores 330 recibe la salida del módulo generador de productos parciales 325 y una copia de las dos entradas del módulo generador de productos parciales 325 y genera una salida de 2*m+1 dígitos redundantes correspondiente a la suma de todas sus entradas. En este ejemplo particular, como se usa representación de acarreo almacenado, se producen dos números de 2*m+1 bits correspondientes a las palabras de suma y acarreo. Si se desea una salida no redundante, el módulo de conversión 335 es usado para transformar la salida del árbol de compresores 330, a un número no redundante de 2*m+1 bit correspondiente a los 2*m+1 MSBs del producto de los números de entrada iniciales.
El módulo de multiplicación representado en Fig. 2b es similar al anterior, pero la segunda entrada se recodifica (por ejemplo, mediante recodificación de Booth) antes de entrar en el generador de productos parciales 325b para producir menos productos parciales, mediante el uso del módulo de recodificación 320b. El valor uno se inserta también en la entrada del módulo recodificador 320b, tal que los m bits de la segunda entrada son aumentados por la derecha con un bit correspondiente al LSB implícito. Sin embargo, en otras implementaciones, la introducción del uno adicional podría realizarse internamente al módulo de recodificación 320b sin necesidad de una entrada especial. Esto es meramente ilustrado en el ejemplo para indicar la necesidad de la introducción funcional del LSB implícito. De forma similar, el LSB de la otra entrada está también ilustrado en la primera entrada del generador de productos parciales 325.

En un camino paralelo , los m MSBs de la mantisa Mz del tercer número FP pre-procesado es

entrada a primer módulo de desplazamiento 120F que está configurado para alinear Mz tal que pueda ser sumado con el resultado de la multiplicación. El primer módulo de desplazamiento 120F comprende un inversor de bit condicional 122F, que es controlado por el bit op, y un desplazador aritmético a la derecha 124F. Este bit op indica la operación efectiva, la cual depende del signo de los números coma flotante de entrada (XOR de los tres signos). La salida de m bits de inversor de bit condicional 122F aumentada por la izquierda con el bit op, como su bit de signo, y por la derecha con el LSB de Mz, es entrada al desplazador aritmético a la derecha 124F. El desplazador aritmético a la derecha 124F es controlado por una salida de la lógica de exponente 107F que indica la diferencia (d) entre el exponente de Z y la suma de los otros dos exponentes. La salida de primer módulo de desplazamiento 120F es un número de 3*m+3 bits. En principio dicho número debería tener 3*m+4 bits para cubrir todos los casos de desplazamientos con el mínimo error. Sin embargo, el bit de signo (MSB del valor desplazado) se omite y el segundo MSB se usa en su lugar, ya que ambos bits son iguales excepto si no se realiza ningún desplazamiento. En este último caso, no se realiza realmente ninguna suma ya que ningún desplazamiento significa que los dos números están demasiado separados (Ez>>Ex+Ey, y más concretamente Ez>Ex+Ey+m+1). Por lo tanto, el signo del resultado de la suma no es su MSB, sino el bit que indica la operación efectiva (op). En una implementación alternativa, la inversión en ambos inversores de bit condicional 122F y 144F podría evitarse cuando esta situación (Ez>Ex+Ey+m+1) se produce, y consecuentemente, el signo del resultado sería siempre positivo en esta situación. En otras implementaciones alternativas, el signo del resultado de la suma podría ser siempre su MSB y la señal de desbordamiento podría evitarse, si 3*m+4 bits son usados para representar la mantisa alineada y el resultado de la suma.

La salida del primer módulo de desplazamiento 120F y la salida del módulo de multiplicación 115F son entradas al módulo de suma 130F. La salida de 2*m+1 bits del módulo de multiplicación 115F es aumentada con un LSB igual a 1 y m+1 MSBs iguales a cero, tal que las dos entradas al módulo de suma 130F tienen igual cantidad de bits antes de la suma. El módulo de suma 130F comprende un sumador en complemento a dos 134F. En el ejemplo de Fig. 1, dicho LSB se introduce como una entrada separada en el sumador 134F. Alternativamente, éste podría estar implícito e introducirse internamente al sumador 134F. La misma regla es aplicable a dichos m+1 MSBs. El módulo de suma 130F genera un número de 3*m+3 bits en una primera salida, que es el resultado sin normalizar de la suma alineada de la mantisa Mz con el producto de las mantisas Mx y My. En una segunda salida, el módulo de suma 130F genera un bit de desbordamiento el cual es la tercera entrada del módulo de actualización del exponente 109F. Dicho número de 3*m+3 bits es entrada al módulo de normalización 140F y al detector del uno de cabecera (LOD) 135F. LOD 135F también recibe una instrucción (no incluida en la figura) sobre la operación efectiva cuando no se realiza ningún desplazamiento en el primer módulo de desplazamiento 120F. Aunque se utilice el nombre LOD en el ejemplo Fig. 1, alguien experto en el estado de la técnica podría apreciar que, dependiendo del signo del resultado de la suma, o el primer uno por la izquierda o el primer cero por la izquierda, debería detectarse, respectivamente. El propósito del módulo 135F es detectar el primer bit significativo por la izquierda del número de 3*m+3 bits con objeto de instruir al módulo de normalización como corresponda. En el mismo sentido, en una implementación alternativa podría usarse en su lugar un circuito anticipador, tal como un anticipador de ceros de cabecera (LZA) para predecir este valor basándose en algunas señales antes del módulo de suma. El LOD 135F provee la segunda entrada del módulo de actualización de exponente 109F.

El módulo de normalización 140F comprende un desplazador a la izquierda 142F y un inversor de bit condicional 144F. El desplazador a la izquierda 142F está conectado, en una primera entrada, a la salida del módulo de suma 130F y, en una segunda entrada, a la salida del LOD 135F. El desplazador a la izquierda 142F desplaza a la izquierda la salida del módulo de suma

130F de acuerdo con la salida del LOD 135F para normalizarla, y sólo mantiene los m MSBs. Por lo tanto, el redondeo al más cercano se realiza mediante truncado. Sin embargo, este redondeo podría producir sesgo, si la normalización requiere un desplazamiento grande. Dichos m MSBs son entonces entrada al inversor de bit condicional 144F para invertirlos si su MSB es cero, lo cual indica un resultado negativo de la suma, ya que el MSB es el bit entero y debería ser uno (número normalizado). Alguien experto en el estado de la técnica podría apreciar que diferentes opciones para detectar un resultado negativo en la suma podrían usarse. Por otro lado, en una implementación alternativa, el inversor de bit condicional 144F podría estar antes del desplazador a la izquierda. La salida de m bits del inversor de bit condicional 144F corresponde a los m MSBs de la mantisa del resultado final S. El LSB de dicha mantisa pre-procesada está implícito y es igual a uno. Se debe indicar que en esta implementación los m MSBs de la mantisa incluyen el bit entero, el cual siempre vale uno. Por lo tanto, en una implementación alternativa, el bit entero podría ser descartado después de la normalización.

Fig. 3 ilustra un circuito de multiplicación-suma fusionadas (FMAD) en coma flotante (FP), de acuerdo a otro ejemplo, configurado para eliminar el sesgo del redondeo y mejorar la velocidad del camino de datos de la mantisa. FMAD 200F recibe tres números en coma flotante pre-procesados X, Y, y Z, y genera un resultado S que es la suma del tercer número coma flotante con el producto de los otros dos (S=Z+X*Y). El LSB de las mantisas es igual a uno. FMAD 200 comprende un camino de datos del exponente 205F y un camino de datos de la mantisa 210F. El camino de datos del exponente 205F comprende una lógica de exponente 207F para recibir los exponentes Ex, Ey, Ez de los tres números FP de entrada y genera un valor intermedio de exponente en una salida, de acuerdo al máximo valor entre Ez y Ex+Ey. La salida de la lógica de exponente 207F está conectada a la primera entrada del módulo de actualización de exponente 209F. Una segunda entrada del módulo de actualización de exponente 209F está conectada al camino de datos de la mantisa 210F para recibir el número de ceros por la izquierda del resultado de la operación de suma (o el número de unos por la izquierda si dicho resultado es negativo). Una tercera entrada del módulo de actualización de exponente 209F está conectada al camino de datos de la mantisa 210F para recibir un bit de desbordamiento (ovf). De forma similar al anterior ejemplo, en una implementación alternativa, las dos últimas entradas, es decir, el número de bits no significativos por la izquierda y el bit de desbordamiento podrían combinarse en un único valor. El módulo de actualización de exponente 209F está configurado para generar el exponente Es del número en coma flotante S, incrementando o decrementando el valor intermedio de exponente de acuerdo al número de bits no significativos por la izquierda y la señal de desbordamiento. Además un circuito lógico de signo (no mostrado) calcula la señal de operación efectiva (op) para la suma final y el signo del resultado, de una forma estándar, basándose en el signo de las entradas y en el signo del resultado de la suma final.

El camino de datos de la mantisa 210F comprende un módulo de multiplicación 215F para recibir los m MSBs de las mantisas de los números FP pre-procesados X e Y. De nuevo, las mantisas se representan por los símbolos Mx y My en Fig. 3. Las mantisas Mx y My (así como Mz) ambas tienen m+1 bit. 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 FMAD a la entrada. Además, como en el ejemplo de Fig. 1, los tres número coma flotante están normalizados. Sin embargo, para simplificar la descripción, el MSB del número normalizado, se incluye en los m bits que se introducen en FMAD 200F. En una implementación alternativa, este bit podría omitirse en las entradas e introducirse, o bien antes del módulo de multiplicación 215F, o bien internamente a dicho módulo de multiplicación 215F, para Mx y My, y, o bien antes del primer módulo de desplazamiento 220F, o bien internamente a dicho módulo, para Mz. En el ejemplo de Fig. 3 el LSB de las mantisas de entradas son introducidas como una entrada separada del módulo de multiplicación 215F. Alternativamente, este podría estar

implícito e introducirse internamente al módulo de multiplicación 215F. El módulo de multiplicación 215F recibe los m MSBs de las mantisas Mx y My y genera, en un formato de representación redundante los 2*m+2 del producto de las mantisas de X e Y (incluyendo su bit implícito). El LSD de dicho producto es siempre uno pero, aunque no se requiere explícitamente, y podría ser omitido como en el ejemplo de la Fig. 1, se incluye en la señal de salida de este ejemplo para mostrar diferentes alternativa. El módulo de multiplicación 215F mostrado en Fig. 3 genera el resultado en formato de acarreo almacenado y entonces dicho resultado se entrega en una primera y una segunda salida de 2*m+2 bits, correspondientes a la palabra de suma y la palabra de acarreo, respectivamente. Sin embargo alguien experto en el estado de la técnica podría apreciar que otros formatos de representación redundante podrían usarse con modificaciones menores sobre el circuito mostrado, tal como representación de dígitos con signo. Las salidas del módulo de multiplicación 215F están conectadas al módulo de suma 230F. El módulo de multiplicación 215F podría ser similar a las implementaciones que se han descrito para los multiplicadores redundantes 305 y 305b con referencia a Fig. 2 y 2b, respectivamente.

En un camino paralelo , los m MSBs de la mantisa Mz del tercer número FP pre-procesado son entrada al primer módulo de desplazamiento 220F que está configurado para alinear Mz tal que pueda ser sumado con el resultado de la multiplicación. El primer módulo de desplazamiento 120F comprende un inversor de bit condicional 222F, que es controlado por el bit op, y un desplazador aritmético a la derecha 224F. Este bit op indica la operación efectiva, la cual depende del signo de los números coma flotante de entrada (XOR de los tres signos). La salida de m bits de inversor de bit condicional 222F, aumentada por la izquierda con el bit op, como su bit de signo, y por la derecha con el LSB de Mz, es entrada al desplazador aritmético a la derecha 224F. De nuevo, el desplazador aritmético a la derecha 224F es controlado por una salida de la lógica de exponente 207F que indica la diferencia (d) entre el exponente de Z y la suma de los otros dos exponentes de entrada. La salida de primer módulo de desplazamiento 220F es un número de 3*m+3 bits y está conectada al módulo de suma 230F. En principio dicho número debería tener 3*m+4 bits para cubrir todos los casos de desplazamientos con el mínimo error. Sin embargo, el bit de signo (MSB del valor desplazado) se omite y el segundo MSB se usa en su lugar, ya que ambos bits son iguales excepto si no se realiza ningún desplazamiento. En este último caso, no se realiza realmente ninguna suma, ya que ningún desplazamiento significa que los dos números están demasiado separados (Ez>>Ex+Ey y más concretamente Ez>Ex+Ey+m+1). Por lo tanto, el signo del resultado de la suma no es su MSB, sino el bit que indica la operación efectiva (op). En una implementación alternativa, la inversión en ambos inversores de bit condicional 222F y 244F podría evitarse cuando esta situación (Ez>Ex+Ey+m+1) se produce, y consecuentemente, el signo del resultado sería siempre positivo en esta situación. En otras implementaciones alternativas, el signo del resultado de la suma podría ser siempre su MSB y la señal de desbordamiento podría evitarse, si 3*m+4 bits son usados para representar la mantisa alineada y el resultado de la suma.

El módulo de suma 230F genera, en una representación no redundante, la suma entre la salida redundante del módulo de multiplicación 215F y la salida alineada del primer módulo de desplazamiento 220F. En este ejemplo particular, como se usa acarreo almacenado como representación redundante, el módulo de suma 230F comprende un compresor 3:2 232F, para sumar las dos salidas del módulo de multiplicación 215F y los 2*m+2 LSBs de la salida del primer módulo de desplazamiento 220F. El compresor 3:2 323F genera dos palabras de 2*m+2 bits como salida en representación de acarreo almacenado. El módulo de suma 230F comprende además un sumador en complemento a dos 234F, conectado a la salida del compresor 3:2 232F, y un módulo de incremento 235F, con una primera entrada para recibir los m+1 MSBs de la salida del primer módulo de desplazamiento 220F, y una segunda entrada para recibir un bit de acarreo final desde el sumador en complemento a dos 234F, para

producir una mantisa en una representación no redundante. En una implementación alternativa, ambos módulos podrían ser sustituidos por un sumador en complemento a dos de 3*m+3 bits, teniendo los m+1 MSBs de una de sus entradas conectados a cero, o un circuito diferente, si la representación redundante seleccionada es otra. La salida de m+1 bits del módulo de incremento 235F y la salida de 2*m+2 bits del sumador en complemento a dos 234F conforman un número de 3*m+3 bits que corresponde a la mantisa del resultado de la operación de multiplicación-suma fusionadas antes de normalizarla. Dicho número de 3*m+3 bits es entrada a un módulo de normalización 240F. El módulo de incremento 235F produce además un bit de desbordamiento en una segunda salida. En otras implementaciones, la información de desbordamiento podría obtenerse de la salida del anticipador de ceros de cabecera (LZA) y esta salida explícita no sería necesaria.

El camino de datos de la mantisa 210F comprende además un Anticipador de Ceros de Cabecera (LZA) 237F, teniendo una primera entrada conectada a la salida del compresor 3:2 232F y una segunda entrada para recibir los m+1 MSBs de la salida del primer módulo de desplazamiento 220F. LZA 237 también recibe una instrucción (no mostrada en la figura), sobre la operación efectiva cuando no se realiza desplazamiento en el primer módulo de desplazamiento 220F. LZA 237F calcula el desplazamiento a la izquierda requerido para normalizar el resultado. En una implementación alternativa, el LZA podría tomar sus entradas directamente de la salida del módulo de multiplicación 215F y del primer módulo de desplazamiento 220F, o en una etapa posterior, desde la salida del módulo de suma 230F.

Alguien experto en el estado de la técnica podría apreciar que el módulo de suma 230F y el LZA 237 podrían ser implementados (en conjunto o separadamente) de muchas formas diferentes, sin desviarse del alcance (objeto) de esta invención.

El módulo de normalización 240F comprende un módulo de desplazamiento a la izquierda 242F y un inversor de bit condicional 244F. El módulo de desplazamiento a la izquierda 242F recibe el número de 3*m+3 bits desde el módulo de suma 230F, en una primera entrada, y genera un número pre-procesado de m+1 bits normalizado y redondeado, teniendo el LSB implícito e igual a uno. Esta operación la realiza en base a una segunda cantidad de desplazamiento recibida desde el LZA 237F, en una segunda entrada. Los m MSBs de dicho número pre-procesado son entonces introducidos en inversor de bit condicional 244F para negarlo si su MSB es cero. Esto último indica un resultado negativo de la suma, ya que dicho MSB es el bit entero y debería valer uno (número normalizado). Alguien experto en el estado de la técnica podría apreciar que diferentes opciones para detectar un resultado negativo en la suma podrían usarse. Por otro lado, en una implementación alternativa, el inversor de bit condicional podría estar antes del módulo de desplazamiento a la izquierda. La salida de m bits del inversor de bit condicional 244F se corresponde con los m MSBs de la mantisa pre-procesada del resultado final de la operación FMAD. El LSB de dicha mantisa pre-procesada está implícito y es igual a uno. Se debe indicar que en esta implementación los m MSBs de la mantisa incluyen el bit entero que siempre vale uno. Por tanto, en una implementación alternativa, el bit entero podría descartarse después de la normalización.

Fig. 3a y 3b ilustran diferentes implementaciones alternativas del módulo de desplazamiento a la izquierda 242F de acuerdo a otros ejemplos. El módulo de desplazamiento a la izquierda 242F permite evitar el sesgo producido por el redondeo, en ciertos casos, cuando un desplazador a la izquierda estándar es usado, como en el ejemplo de la Fig. 1. El módulo de desplazamiento a la izquierda 242F representado en Fig. 3a comprende un desplazador a la izquierda especial 370F teniendo una primera entrada conectada a la primera entrada del módulo de desplazamiento a la izquierda 242F. Sin embargo, el LSB está conectado a un bit con valor aleatorio. Una segunda entrada del desplazador a la izquierda especial 370F está conectada a la cantidad de desplazamiento desde la segunda entrada del módulo de

desplazamiento a la izquierda 242F. Este 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 que, en este caso, está conectado al inverso de dicho bit aleatorio. El bit aleatorio podría ser cualquier bit seleccionado, o el resultado de la combinación de varios bits seleccionados, de la primera entrada, o cualquier otro bit con las adecuadas características estadísticas. La salida de desplazador a la izquierda especial 370F comprende los m MSBs del valor desplazado, el cual es la salida del módulo de desplazamiento a la izquierda 242F. Este ejemplo de implementación de módulo de desplazamiento a la izquierda 242F evita el sesgo producido en una operación FMAD, como en el ejemplo de la Fig. 1, cuando la cantidad de desplazamiento (el número de bits no significativos por la izquierda) es mayor que 2*m+3 (cuando una operación efectiva de resta produce una cancelación). En una implementación alternativa, como el LSB de la primera entrada es descartado, este bit podría no generarse a la salida del módulo de suma 230F.

Una implementación del desplazador a la izquierda especial 370F basada en la implementación del desplazador variable clásico es ilustrada en la Fig. 3c. El desplazador a la izquierda especial 370F 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 a 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 de implementación de un desplazador a la izquierda especial, la máxima cantidad de desplazamiento es m-1

El circuito de FMAD en coma flotante podría ser usado también para tan solo multiplicar dos números coma flotante, fijando la entrada Z al número cero, o también para tan solo sumar dos números coma flotante, fijando, ya sea la entrada X, o la Y, al número FP uno. Debemos indicar que el número coma flotante cero es normalmente un valor especial en representaciones convencionales y, en concordancia, en representaciones pre-procesadas. Por tanto, podría reconocerse la instrucción de una multiplicación única mediante la detección del valor especial cero en la entrada Z, como en un circuito convencional. En cambio, el número coma flotante uno es representado exactamente en formatos coma flotante convencionales, pero no en un formato pre-procesado normalizado. Por lo tanto, para realizar una operación de suma única, se podría requerir o bien tener un caso especial para representar el número uno, o una instrucción para indicar la operación deseada (FMAD o suma). En los ejemplos de la Fig. 1 y 3 un valor especial es usado para representar el valor uno. Sin embargo, alguien experto en el estado de la técnica podría apreciar que un patrón diferente para el caso especial, o una instrucción independiente para indicar una suma única podrían usarse en su lugar, sin afectar la idea principal de la invención propuesta.

Los ejemplos de Fig. 1 y 3 permiten la operación de multiplicación única, fijando la salida del primer módulo de desplazamiento a cero, cuando la entrada Z es un cero. Esto se realiza fijando la primera cantidad de desplazamiento a un valor mayor que 3*m+3 (ésta es la misma solución que podría usarse cuando Ez<<Ex+Ey, y más concretamente Ez<Ex+Ey-2*m-2). En una implementación alternativa, el LSB de Mz que es entrada al desplazador aritmético a la derecha 124 o 224 de Fig.1 o 3, respectivamente, podría ser selectivamente fijado a cero.

Alguien experimentado en el estado de la técnica podría apreciar que otras modificaciones menores al circuito propuesto podrían tener el mismo efecto.

Los ejemplos de Fig. 1 y 3 permite la operación de suma única fijando todos los bits explícitos de una de las entradas, ya sea X, o Y, a cero y ajustando el exponente en consecuencia. En una implementación alternativa, donde el MSB, el bit entero, es también implícito e igual a uno, dicho bit podría fijarse selectivamente a cero. En otra implementación alternativa un multiplexor a la salida del módulo de multiplicación podría ser usado para seleccionar, o bien el resultado de la operación de multiplicación (para FMAD), o directamente una de las mantisas de entrada (para suma única). Alguien experimentado en el estado de la técnica podría apreciar que otras modificaciones menores al circuito propuesto podrían tener el mismo efecto.

Cuando los ejemplos de Fig. 1 y 3 se usan para una suma única, el redondeo y normalización final podría producir cierto sesgo cuando la diferencia entre los exponentes de entrada es cero o uno. El módulo de desplazamiento a la izquierda representado en Fig. 3b permite evitar este sesgo y también el producido por la operación FMAD. El módulo de desplazamiento a la izquierda representado en Fig. 3b comprende un desplazador a izquierda especial, similar al desplazador a izquierda especial 370F discutido con referencia a Fig. 3a, teniendo los 3*m+2 MSBs de la primera entrada conectados a los 3*m+2 MSBs de la primera entrada del módulo de desplazamiento a la izquierda 242F, mientras que el segundo y primer LSB se conectan, al multiplexor 371, y a la puerta AND 372, respectivamente. El multiplexor 371 comprende una primera entrada conectada al LSB de la primera entrada del módulo de desplazamiento a la izquierda 242F, y una segunda entrada conectada a un primer bit aleatorio (R1) y una tercera entrada conectada a un primer bit de control (c1), el cual indica, o una operación FMAD, o una suma única. La puerta AND 372 comprende una primera entrada conectada a un multiplexor 373 y una segunda entrada conectada al inverso de un segundo bit de control (c2), elcual es uno cuando una suma única de números de entrada con el mismo exponente, pero diferente signo, es realizada. El multiplexor 373 comprende una primera entrada conectada al inverso de R1, una segunda entrada conectada a un segundo bit aleatorio (R2), y una tercera entrada conectada al primer bit de control (c1). Una segunda entrada del desplazador a la izquierda especial 370F está conectada a la cantidad de desplazamiento desde la segunda entrada del módulo de desplazamiento a la izquierda 242F. Una tercera entrada del desplazador a la izquierda especial 370F, la que indica con qué valor se rellenaran los bits vacantes, está conectada a una puerta AND 374, teniendo una primera entrada conectada al multiplexor 375 y una segunda al inverso de c2. El multiplexor 375 comprende una primera, y una segunda, entrada conectadas al inverso de R1 y R2, respectivamente, y una tercera entrada conectada a c1. El módulo de desplazamiento a la izquierda representado en Fig. 3b fija el MSB de las posiciones vacantes a R2, y el resto, a su inverso, cuando una suma única se realiza, el exponente de la tercera entrada es uno menos que el exponente del otro sumando, y dichos números FP tienen diferente signo. En otro caso, cuando una suma única se realiza, los exponentes de entrada son iguales, pero los números FP tienen diferente signo, entonces dicho módulo fija las posiciones vacantes a cero. Para el resto de los casos, dicho módulo fija el LSB de la salida del módulo de suma a R1 y las posiciones vacantes a su inversa. Esto evita el sesgo después del redondeo en ciertos casos. Los bits R1 y R2 podrían ser cualquier bit seleccionado, o el resultado de la combinación de varios bits seleccionados, de la primera entrada, o cualquier otro con las características estadísticas adecuadas. En una implementación alternativa, R1 y R2 podrían tener el mismo valor, y el circuito podría ser más simple. En otra implementación alternativa el módulo de desplazamiento a la izquierda 242F podría estar configurado además para restar uno del LSB, si un bit seleccionado, o el resultado de la combinación de varios bits seleccionados, del resultado de la suma es uno, cuando una suma única se realiza teniendo los mismos exponentes de entrada pero diferente signo.

El módulo de desplazamiento a la izquierda 242F comprende además un módulo de redondeo lejano 385F teniendo una entrada conectada a la salida del desplazador a la izquierda especial 370F, el cual, en este caso, saca los m+1 MSBs del valor desplazado. La salida del módulo de redondeo lejano 385F podría estar directamente conectado a los correspondientes bits de salida del módulo de desplazamiento a la izquierda 242F. El LSB de dicha salida está conectado a la salida de una puerta AND 387, teniendo una primera entrada conectada al segundo LSB de la salida del desplazador a la izquierda especial 370F, y una segunda entrada conectada a una puerta NAND 386. La puerta NAND 386 tiene una primera entrada conectada al inverso del LSB de la entrada del módulo de redondeo lejano 385F y una segunda entrada conectada a un tercer bit de control (c3), el cual indica una suma única alineada (Ez=Ex+Ey). El módulo de redondeo lejano 385F previene el redondeo con sesgo en este último caso. Su salida es igual a los m MSBs de su entrada, excepto si la operación efectiva es una suma única, los exponentes y signos de los sumandos de entrada son iguales y el LSB de la entrada es cero. En este caso el LSB de la salida es puesto a cero. En una implementación alternativa el módulo de redondeo lejano podría estar justo en la primera entrada del módulo de desplazamiento a la izquierda 242F, y podría fijar acero el tercer LSB de la entrada en caso de que sea una suma única alineada y el segundo LSB de la entrada sea cero.

En una implementación alternativa, alguien experimentado en el estado de la técnica podría elegir controlar el sesgo en menos casos, y por tanto producir un circuito más simple para el módulo de desplazamiento a la izquierda 242F.

Los dispositivos FMAD descritos arriba requieren números FP que hayan sido pre-procesados de acuerdo a la invención como se describió arriba. Estos números pre-procesados podrían ser generados por circuitos, tales como los mencionados dispositivos FMAD, que están diseñados para funcionar con números pre-procesados o podrían ser generados por convertidores, diseñados para convertir número no procesados, o números pre-procesados no FP, en números FP pre-procesados. Además, los números pre-procesados generados por los dispositivos FMAD descritos arriba podrían, en concordancia, requerir convertidores tales que los números generados podrían ser usados por circuitos que no estén diseñados para operar números FP 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 punto fijo: 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. 4 ilustra un ejemplo de tal conversor para números en coma fija pre-procesados de m+2 bits y un número FP pre-procesado con una mantisa de n+1 bits. El conversor 600 comprende un módulo de normalización 630 que tiene un inversor de bits condicional 605 en serie con un desplazador a la izquierda pre-

procesado 610. El inversor de bits condicional 605 tiene una primera entrada para recibir los m LSBs de los m+1 MSBs del número en coma fija pre-procesado de m+2 bits. El MSB dicho número de m+2 bits es el signo y será el signo del número FP pre-procesado así como será usado para controlar el inversor de bits condicional. La salida de m bits del inversor de bits condicional 605 es la entrada al desplazador a la izquierda pre-procesado 610. En implementaciones alternativas el desplazador a la izquierda pre-procesado precede al inversor de bits condicional 605. La función del desplazador a la izquierda pre-procesado 610 es descrito con más detalle en la Fig. 4a. El desplazador a la izquierda pre-procesado 610 requiere un desplazador a la izquierda especial 610a con una nueva tercera entrada de un bit, el cual permite seleccionar el valor usado para rellenar las posiciones vacantes después del desplazamiento. Una implementación del desplazador a la izquierda especial 610a podría ser similar al del desplazador a la izquierda especial 245 ilustrado en la Fig. 3b. En este caso, la máxima cantidad de desplazamiento es m o m+1. Si el número en coma fija es igual a cero y el bit R en la Fig. 4a es también igual a cero, requiere una máxima cantidad de desplazamiento que tiene un bit adicional (m+1) de manera que la mantisa está normalizada. Alternativamente, si cuando el número en coma fija es igual a cero, puede ser tratado como un caso especial y convertido a cero en FP. Entonces la máxima cantidad de desplazamiento podría ser igual a m. Usando este desplazador a la izquierda especial 610a, el valor de entrada del desplazador a la izquierda pre-procesado 610 es aumentado con un LSB adicional fijado a cualquier bit aleatorio (por ejemplo, el LSB del valor de entrada inicial) y la tercera entrada del desplazador a la izquierda especial se pone al inverso de dicho valor aleatorio, para rellenar ambas, las posiciones vacantes requeridas para completar el tamaño n si n>m+1 y los posiciones vacantes producidas después del desplazamiento. La salida del desplazador a la izquierda pre-procesado 610 comprende los n MSBs de la mantisa Mz del número FP pre-procesado. Dicha salida se corresponde sólo con los n MSBs del valor desplazado si n<m. El LSB de la mantisa 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 números en coma fija a FP, cuando M>N, no hay redondeo hacia arriba después de la operación de desplazamiento y por lo tanto hay una reducción en los componentes y en el procesamiento. Cuando M<N, entonces no hay sesgo producido por el redondeo con la utilización del conversor propuesto.

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

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

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

Volviendo al conversor 700 de la Fig. 5, un camino paralelo comprende un módulo LOD 710 que tiene una entrada que recibe el número en coma fija no procesado y una salida para la generación de la cantidad de desplazamiento para el módulo de normalización 705 que también se utiliza como entrada al módulo de computación del exponente 715 para generar el exponente 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. 6a es un ejemplo de un conversor de este tipo. El conversor 800a ilustra un conversor adaptado para convertir un número FP pre-procesado que tiene n+m+1 bits de mantisa a una mantisa de n+1 bits. El LSB de ambas mantisas es igual a 1 y por lo tanto no se representa. El signo (sign_x) del número FP pre-procesado original va a seguir siendo el mismo en el número FP pre-procesados objetivo (representado como sign_z). Los n MSBs de la mantisa original serán los n MSBs de la mantisa pre-procesada objetivo. Es decir, tiene lugar una simple función de truncamiento. Por lo tanto, no se genera un bit de desbordamiento, y un calculador de exponentes 801a podría generar el exponente objetivo Ez basándose simplemente en el exponente original Ex.

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

La Fig. 6c es otro ejemplo de un conversor de pre-procesados FP a pre-procesados FP. El conversor 800c ilustra un conversor adaptado para 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 que se usa en la Fig. 6c.

Otra categoría de conversores son los conversores para convertir números FP pre-procesados a números en coma fija pre-procesados. La Fig. 7 ilustra 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. 8a ilustra un conversor con sesgo para la conversión de un número FP pre-procesado que tiene n+1 bits de mantisa y un exponente de d bits a un número en coma fija pre-procesado de n+m+2 bits. Los n MSBs de la mantisa se introducen en el inversor de bits condicional 1005a. El LSB de la mantisa es igual a 1 y no se introduce. El signo del número FP pre-procesado se utiliza para controlar el inversor convencional 1005a. La salida del inversor de bits condicional 1005a junto con el signo (sign_x) son introducidos al desplazador a la derecha 1010a. La salida del inversor de bits condicional es expandida mediante la adición por la derecha de un bit a uno y tantos bits a cero como sean necesarios para completar el nuevo tamaño. En una implementación alternativa esta expansión se podría realizar con un bit a cero y tantos bits a uno como fuesen necesarios. Este número expandido entra al desplazador a la derecha 1010a. El desplazador a la derecha 1010a tiene otra entrada para recibir la cantidad de desplazamiento del calculador de cantidad de desplazamiento 1015a. El calculador de cantidad de desplazamiento 1015a recibe el exponente del número FP pre-procesado y genera la cantidad de desplazamiento. La salida del desplazador a la derecha 1010a son los n+m+1 MSBs del número en coma fija pre-procesado. El LSB es, similarmente, igual a 1 y no es ni generado ni representado.

La Fig. 8b ilustra un conversor sin sesgo para la conversión de un número FP pre-procesado, que tiene n+1 bits de mantisa y un exponente de d bits, a un número en coma fija pre-procesado de n+m+2 bits. Los n bits más significativos de la mantisa se introducen en el inversor de bits condicional 1005b. El LSB de la mantisa es igual a 1 y no se introduce. El signo del número FP pre-procesado se utiliza para controlar el inversor de bits condicional 1005b. La salida del inversor de bits condicional 1005b junto con el signo (sign_x) son introducidos al desplazador a la derecha 1010b. La salida del inversor de bits condicional es expandida mediante la adición por la derecha un bit seleccionado al azar y tantos bits con el valor inverso de dicho bit aleatorio como sean necesarios para completar el nuevo tamaño. El bit aleatorio podría ser cualquiera de la mantisa inicial. Este número expandido entra al desplazador a la derecha 1010b. El desplazador a la derecha 1010b tiene otra entrada para recibir la cantidad de desplazamiento del calculador de cantidad de desplazamiento 1015b. El calculador de

cantidad de desplazamiento 1015b recibe el exponente del número FP pre-procesado y genera la cantidad de desplazamiento. La salida del desplazador a la derecha 1010b son los n+m+1 MSBs del número en coma fija pre-procesado. El LSB es, similarmente, igual a 1 y no es ni generada ni representado.

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

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

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. 10a. El LSB será igual a 1 y está implícito.

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

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

Aún, otra categoría de conversores son los conversores para convertir números FP pre-procesados a en coma fija no procesados. La Fig. 12 ilustra un conversor de este tipo en el que el número de bits de la mantisa de entrada es mayor que el número de bits del número en coma fija de salida. Se compone de un sub-conversor 1410, que corresponde a un conversor de pre-procesado FP a número en coma fija pre-procesado 900 como se discutió con referencia a la Fig. 7. El sub-conversor 1410 recibe el exponente Ex, el bit del signo del número FP (sign_x) y la mantisa Mx que comprende n+m bits. Genera un número en coma fija pre-procesado de n+2 bits a la salida. Conectada a la salida de dicho sub-conversor 1410 hay una unidad de redondeo 1415 que incluye un incrementador 1420 similar al sumador 1310a descrito con referencia a la Fig. 11a para incrementar los n+1 MSBs si el LSB es uno. La salida del sumador 1420 y por lo tanto de la unidad de redondeo 1415 es un número en coma fija no procesado de n +1 bits. En una implementación alternativa el redondeador podría realizar otro

tipo de redondeo.

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

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

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

A pesar también de que las realizaciones descritas comprenden dispositivos informáticos, la invención también se extiende a programas informáticos, más particularmente a programas informáticos en unos medios portadores, adaptados para llevar a cabo la invención. El programa informático puede estar en forma de código fuente, código objeto o un código intermedio entre 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. Un dispositivo para realizar una operación de multiplicación-suma fusionada en coma flotante entre tres números coma flotante pre-procesados y generar un cuarto número coma flotante pre-procesado, cada número teniendo una mantisa pre-procesada de m+2 dígitos, el dispositivo comprende:

un camino de datos del exponente configurado para recibir los exponentes de los tres números pre-procesados de entrada y generar el exponente del resultado de la operación de multiplicación-suma en coma flotante; y

un camino de datos de la mantisa, comprendiendo

un camino de multiplicación comprendiendo

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

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

el camino de multiplicación configurado para multiplicar dichas mantisas pre-procesadas del primer y segundo número y generar un resultado de la multiplicación en una salida,

un camino de suma configurado para recibir como mucho los m+1 MSDs de la mantisa pre-procesada del tercer número en una primera entrada y el resultado de la multiplicación en una segunda entrada y generar como mucho los m+1 MSDs de la mantisa del cuarto número pre-procesado, mientras 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.

2. El dispositivo según reivindicación 1, donde el camino de datos del exponente está configurado para

definir la operación efectiva entre la tercera mantisa y el resultado de la multiplicación según los signos de las entradas;

calcular el exponente de la salida;

calcular el signo de la salida; y

detectar y resolver excepciones y valores especiales de las entradas o de dicha operación.

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

5. Dispositivo según cualquiera de las reivindicaciones 1 a 3, en el que comprende además una cuarta 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 suma comprende:

un primer módulo de desplazamiento, configurado para recibir como mucho los m+1 Bits Mas Signigicativos (MSBs) de la tercera mantisa pre-procesada en una primera entrada y alinear la tercera mantisa pre-procesada con la salida del camino de multiplicación, y

un módulo de suma configurado para sumar la salida alineada del primer módulo de desplazamiento con la salida del camino de multiplicación.

8. Dispositivo según reivindicaciones 6 o 7, en el que el camino de multiplicación comprende:

un módulo de multiplicación, configurado para recibir, en una primera y una segunda entrada, como mucho los m+1 MSBs de la mantisa del primer y segundo número pre-procesado pre-proc, respectivamente, y generar los 2*m+3 MSBs del resultado de la multiplicación entre dichas mantisas pre-procesadas, en una salida.

9. Dispositivo según reivindicaciones 6 o 7, en el que el camino de multiplicación comprende:

un multiplicador redundante, configurado para recibir, en una primera y una segunda entrada, como mucho los m+1 MSBs de la mantisa del primer y segundo número pre-procesado pre-proc, respectivamente y generar, en un formato de representación redundante, como mucho los 2*m+3 MSDs del valor correspondiente a la operación de multiplicación entre dichas mantisas pre-procesadas.

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

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

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

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

12. Dispositivo según cualquiera de las reivindicaciones 7 a 11, en el que el primer módulo de desplazamiento está configurado para recibir como mucho los m+1 MSBs de la mantisa del tercer número pre-procesado, 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.

13. Dispositivo según reivindicación 12, en el que el primer módulo de desplazamiento está configurado para negar selectivamente el valor de salida.

14. Dispositivo según reivindicaciones 12 o 13, en el que el primer módulo de desplazamiento comprende además una tercera entrada con el valor uno.

15. Dispositivo según cualquiera de las reivindicaciones 12 a 14, en el que el primer módulo de desplazamiento comprende un desplazador a la derecha conectado a un inversor de bit condicional.

16. Dispositivo según cualquiera de las reivindicaciones 12 a 15, en el que el módulo de suma comprende:

un sumador configurado para recibir la salida del camino de multiplicación, en una primera entrada, y la salida del primer módulo de desplazamiento, en una segunda entrada, y generar un valor correspondiente a la suma con signo del resultado de la multiplicación entre las mantisas del primer y segundo número pre-procesado, y la mantisa alineada del tercer número pre-procesado, en una salida.

17. Dispositivo según reivindicación 16, en el que el sumador está configurado

para recibir los 2*m+3 MSBs de la multiplicación de la mantisa del primer y segundo número pre-procesado, en una primera entrada.

18. Dispositivo según reivindicación 16, en el que el sumador está configurado para recibir los 2*m+3 MSDs de la multiplicación de la mantisa del primer y segundo número pre-procesado, en un formato de representación redundante, en una primera entrada.

19. Dispositivo según reivindicaciones 17 o 18, en el que la suma con signo comprende n bits, n>m, y el sumador está configurado para generar como mucho los n-1 MSBs de dicha suma con signo en una primera salida.

20. Dispositivo según reivindicación 19, en el que el sumador está configurado además para generar el LSB de dicha suma con signo en una segunda salida.

21. Dispositivo según cualquiera de las reivindicaciones 7 a 20 en el que el camino de datos de la mantisa comprende además un módulo de normalización, teniendo una primera entrada conectada al módulo de suma y una segunda entrada para recibir una segunda cantidad de desplazamiento, en el que el módulo de normalización está configurado para generar como mucho los m+1 MSBs de la cuarta mantisa pre-procesada mediante el desplazamiento a la izquierda de la salida del módulo de suma.

22. Dispositivo según reivindicación 21, 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, es igual a uno.

23. Dispositivo según cualquiera de las reivindicaciones 21 a 22, en el que el módulo de normalización está 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.

24. Dispositivo según reivindicación 23, 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.

25. Dispositivo según cualquiera de las reivindicaciones 21 a 24, en el que el módulo de normalización está configurado además para forzar a cero el segundo LSB del valor que corresponde a la mantisa del cuarto número pre-procesado, cuando la operación es una suma única, el tercer número de entrada y el otro sumando tienen el mismo exponente y signo, y los valores del segundo LSB de las mantisas pre-procesadas de dichos operandos son diferentes.

26. Dispositivo según cualquiera de las reivindicaciones 21 a 25, en el que el módulo de normalización está configurado además para generar selectivamente el complemento a uno del resultado de dicha operación.

27. Dispositivo según cualquiera de las reivindicaciones 6 a 26, 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.

28. Dispositivo según reivindicación 27 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.

29. Dispositivo según reivindicación 28 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.

30. Dispositivo según reivindicación 28 ó 29, en el que dicho módulo de normalización está configurado además para generar selectivamente el complemento a uno del resultado de dicho desplazamiento.

31. Dispositivo según cualquiera de las reivindicaciones 6 a 30, 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.

32. Dispositivo según la reivindicación 31, 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 MSBs 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.

33. Dispositivo según la reivindicación 32, en el que el módulo de normalización está configurado además para generar selectivamente el complemento a uno de dicho valor generado si la entrada es negativa.

34. Dispositivo según cualquiera de las reivindicaciones 24, 29, 30, 32 or 33, en el que el módulo de normalización comprende un desplazador variable configurado para recibir un bit para completar las posiciones vacantes.

35. Dispositivo según la reivindicación 34, 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, iϵ[0, número de multiplexores-1], y cada multiplexor configurado para completar las posiciones vacantes usando el valor de dicho bit recibido.

36. Dispositivo según la reivindicación 31, 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.

37. Dispositivo según la reivindicación 36, 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.

38. Dispositivo según a cualquiera de las reivindicaciones 25 a 37, 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.

39. Dispositivo según la reivindicación 38, 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.

40. Dispositivo según cualquiera de las reivindicaciones 6 a 39, en el que comprende además un conversor de números coma flotante pre-procesados a números coma fija no procesados para convertir el cuarto número en coma flotante pre-procesado a un cuarto número en coma fija no procesado.

41. Dispositivo según la reivindicación 40, 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.

42. Dispositivo según reivindicación 41, 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.

43. Dispositivo según cualquiera de las reivindicaciones 6 a 39, 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.

44. Dispositivo según la reivindicación 43, 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.

45. Dispositivo según la reivindicación 43, 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.

46. Dispositivo según la reivindicación 45, 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. 47. Dispositivo según cualquiera de las reivindicaciones 6 a 46, 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.

48. Dispositivo según la reivindicación 47, 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.

49. Dispositivo según la reivindicación 48, en el que el segundo módulo de desplazamiento comprende un desplazador aritmético a la derecha acoplado a un inversor de bit condicional.

50. Dispositivo según reivindicación 47, 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.

51. Dispositivo según la reivindicación 50, 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..

52. Dispositivo según las reivindicaciones 50 o 51, 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.

53. Dispositivo según cualquiera de las reivindicaciones 6 a 52, 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.

54. Dispositivo según la reivindicación 53, 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.

55. Dispositivo según la reivindicación 54, 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.

56. Dispositivo según la reivindicación 53, 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 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, 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.

57. Dispositivo según la reivindicación 56, 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.

58. Dispositivo según cualquiera de las reivindicaciones 6 a 57, 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.

59. Dispositivo según la reivindicación 58, en el que cuando el número en coma flotante no procesado tiene una mantisa de U+2-V bits, V<U, entonces el conversor comprende:

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

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

60. Dispositivo según la reivindicación 59, 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.

61. Dispositivo según las reivindicaciones 59 ó 60, 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.

62. Dispositivo según la reivindicación 58, 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.

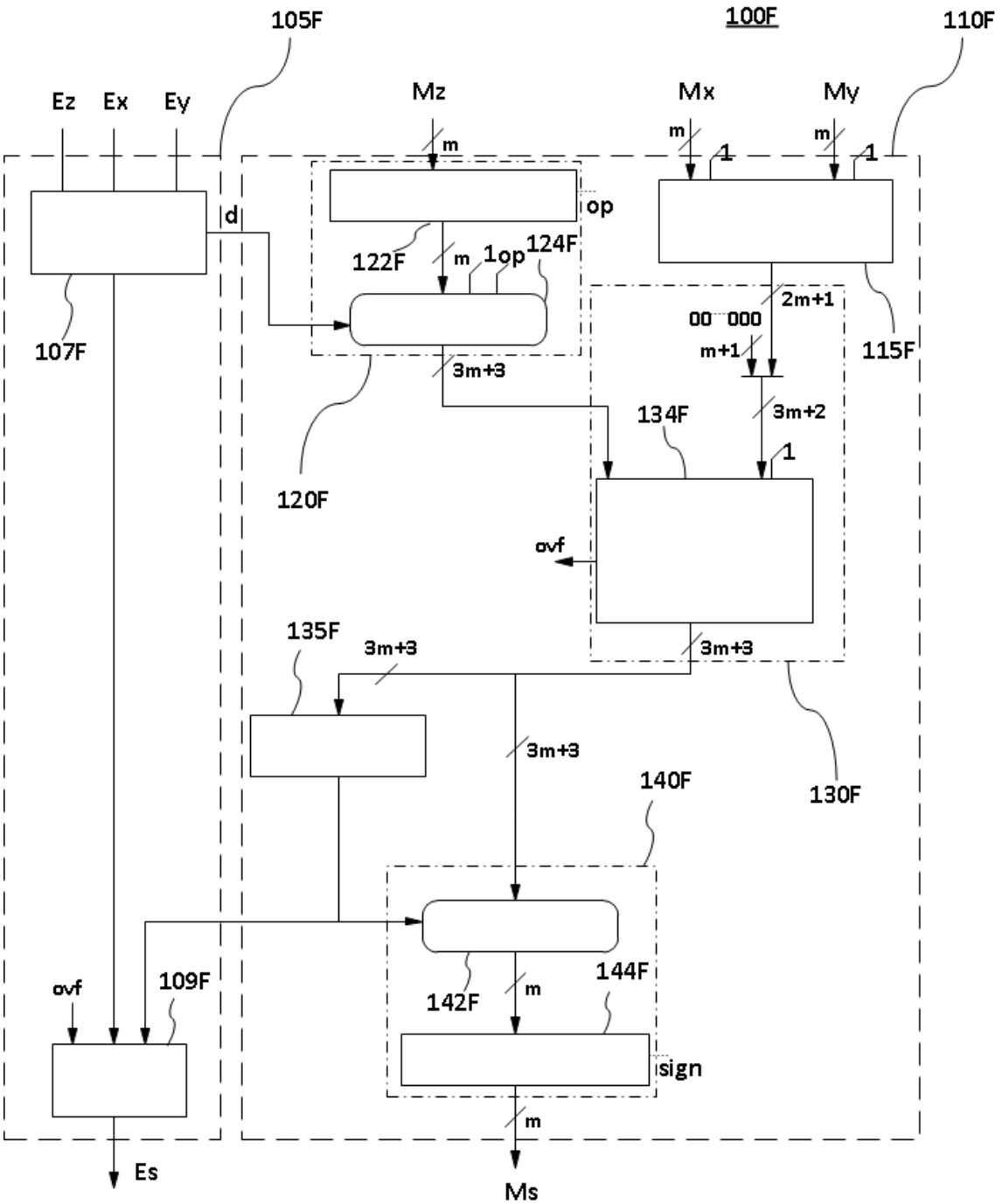

105F
100F
110F
Ez
Ex
Ey
Mz
Mx
My
m
1
d
op
122F
124F
1op
107F
115F
2m+1
00 000
m+1
3m+3
134F
3m+2
120F
ovf
135F
130F
140F
109F
142F
144F
sign
Es
Ms

FIG. 1

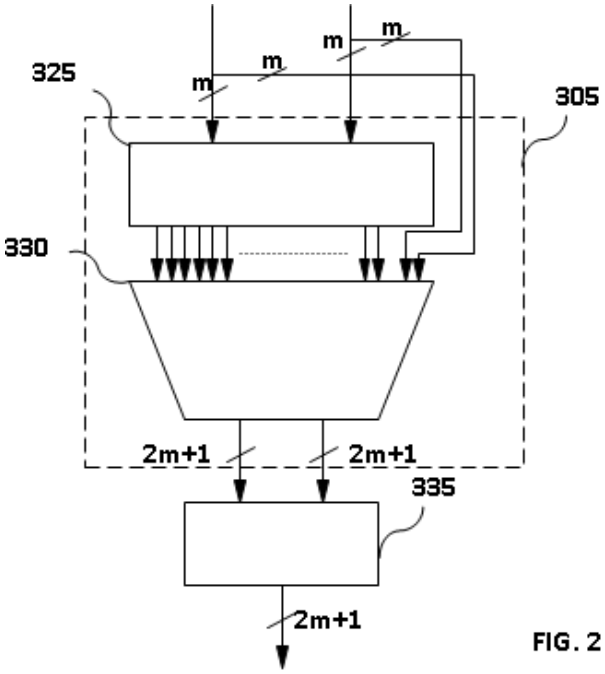

m
m
m
m
325
305
330
2m+1
2m+1
335
2m+1

FIG. 2

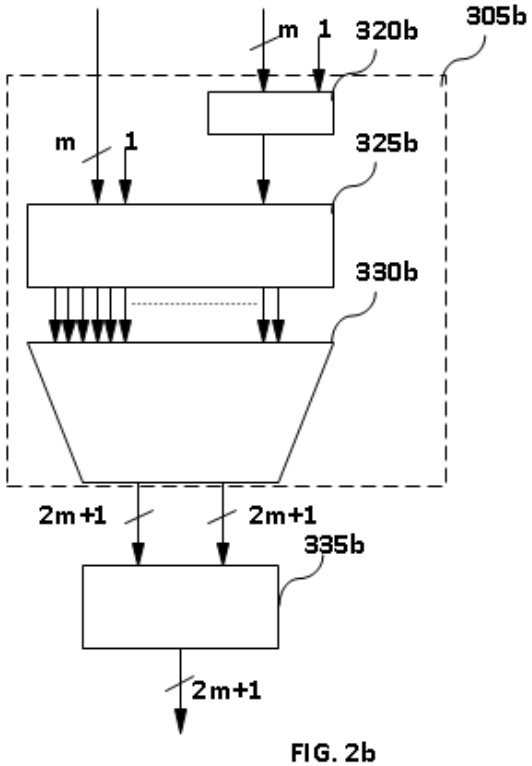

m
1
320b
305b
m
1
325b
330b
2m+1
2m+1
335b
2m+1

FIG. 2b

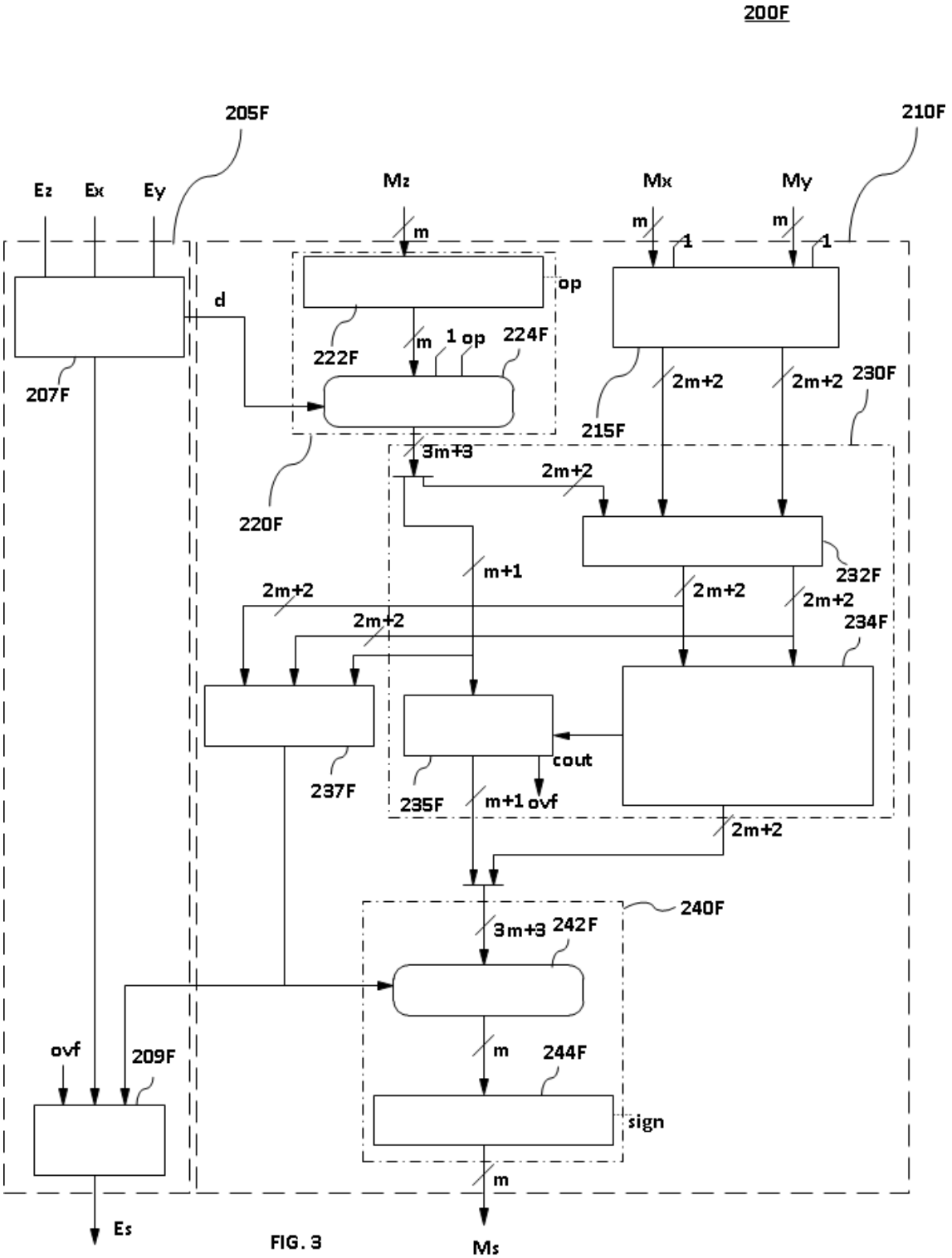

200F
205F
210F
Ez
Ex
Ey
Mz
Mx
My
m
1
op
d
222F
224F
207F
215F
230F
3m+3
2m+2
220F
232F
m+1
234F
237F
235F
cout
ovf
240F
242F
244F
sign
209F
Es
Ms

FIG. 3

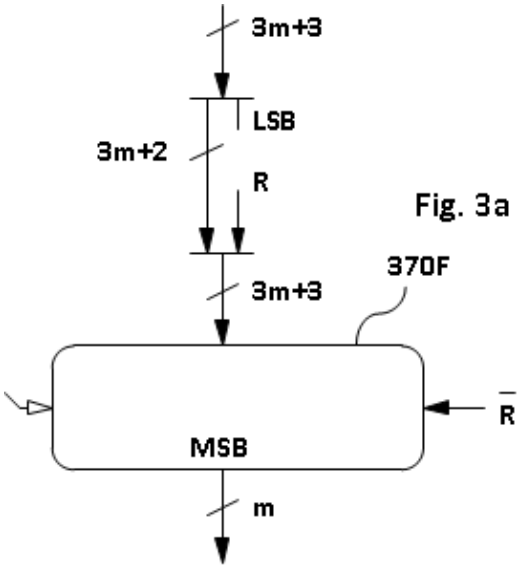
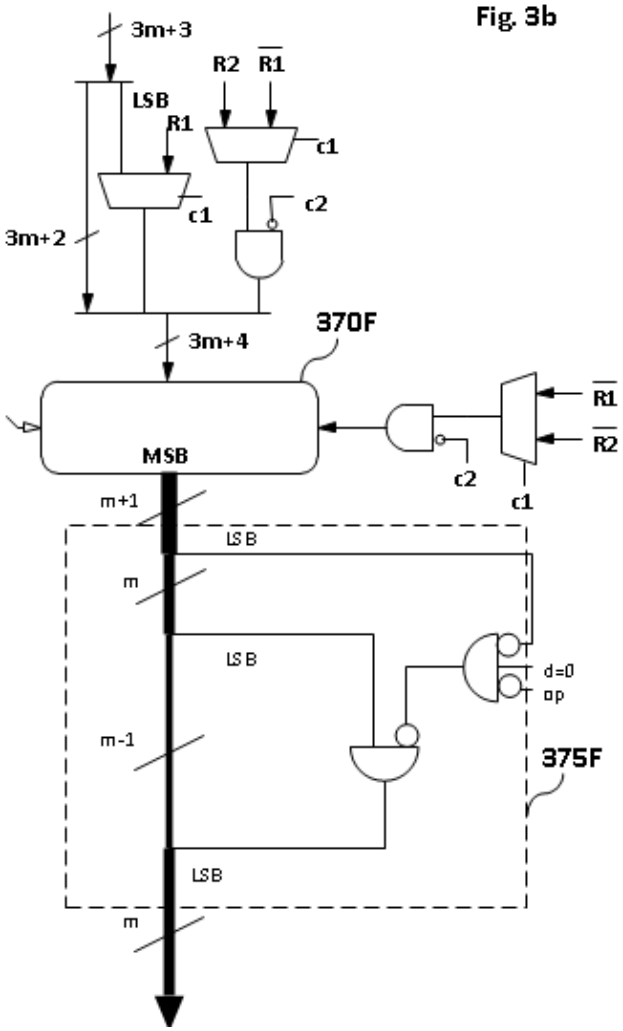

3m+3
LSB
3m+2
R
3m+3
370F
MSB
$\overline{R}$
m

Fig. 3a

Fig. 3b


3m+3
R2
$\overline{R1}$
LSB
R1
c1
c1
c2
3m+2
3m+4
370F
$\overline{R1}$
$\overline{R2}$
MSB
c2
c1
m+1
LSB
m
LSB
d=0
op
m-1
375F
LSB
m

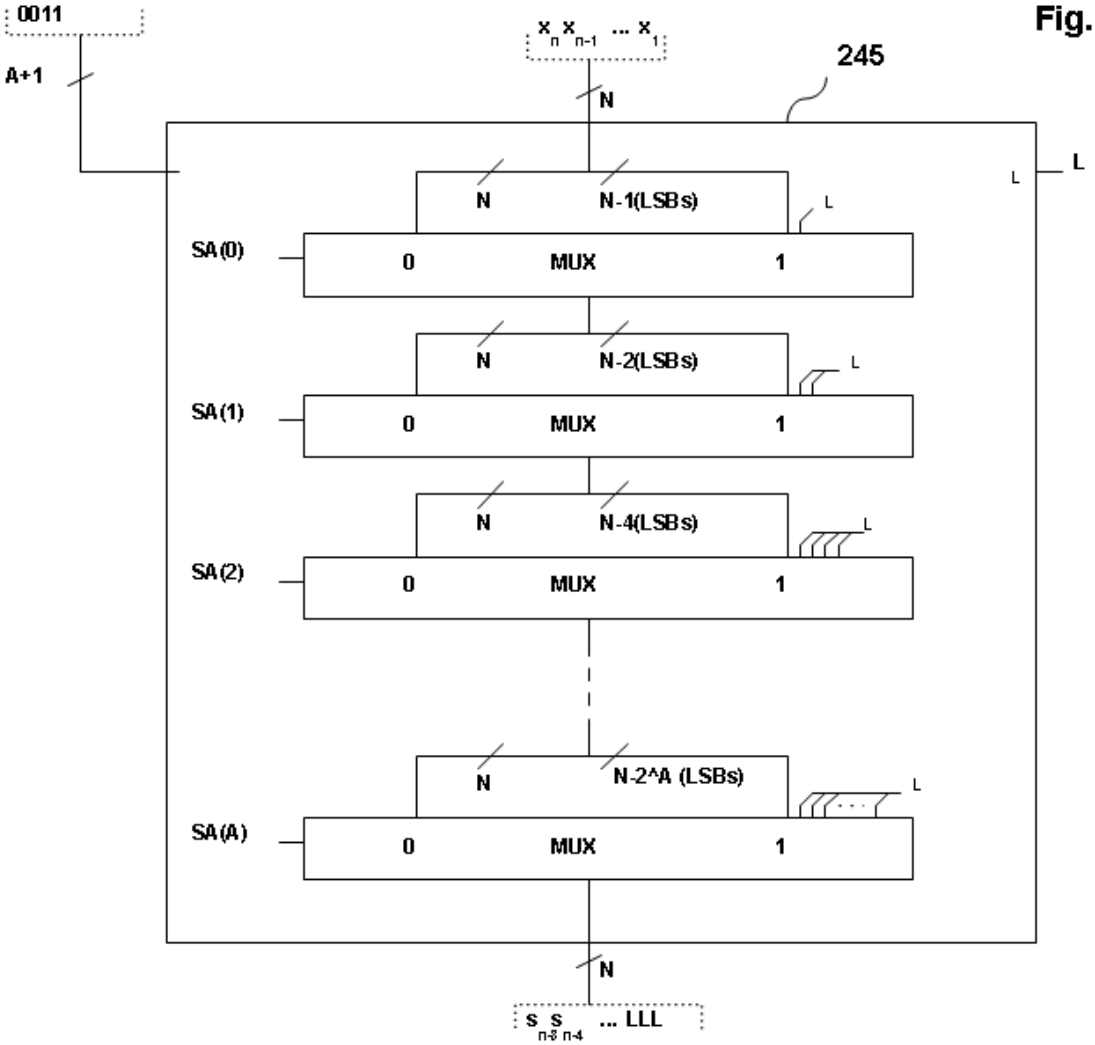

0011
A+1
X n X n-1 ... X 1
N
245
L
L
N
N-1(LSBs)
L
SA(0)
0
MUX
1
N
N-2(LSBs)
L
SA(1)
0
MUX
1
N
N-4(LSBs)
L
SA(2)
0
MUX
1
N
N-2^A (LSBs)
L
SA(A)
0
MUX
1
N
S S n-3 n-4 ... LLL

Fig. 3c

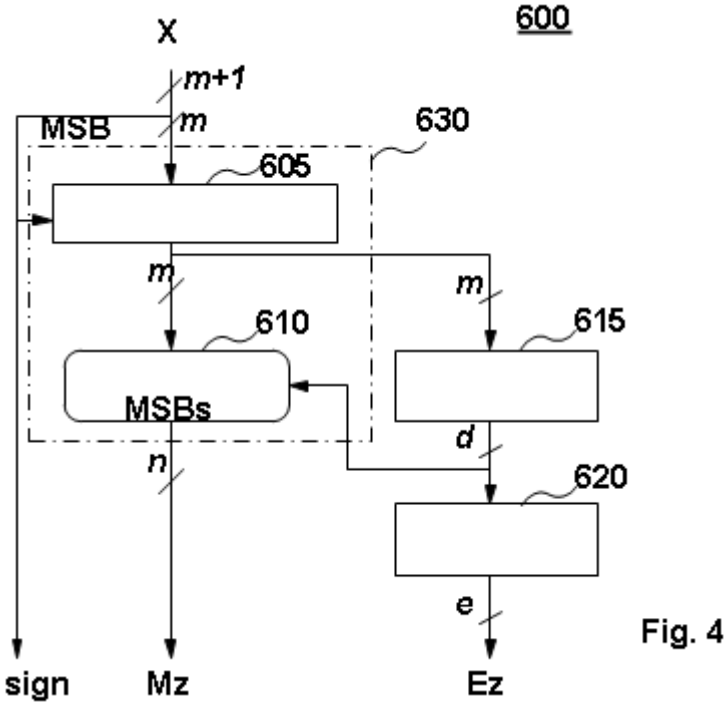


Fig. 4

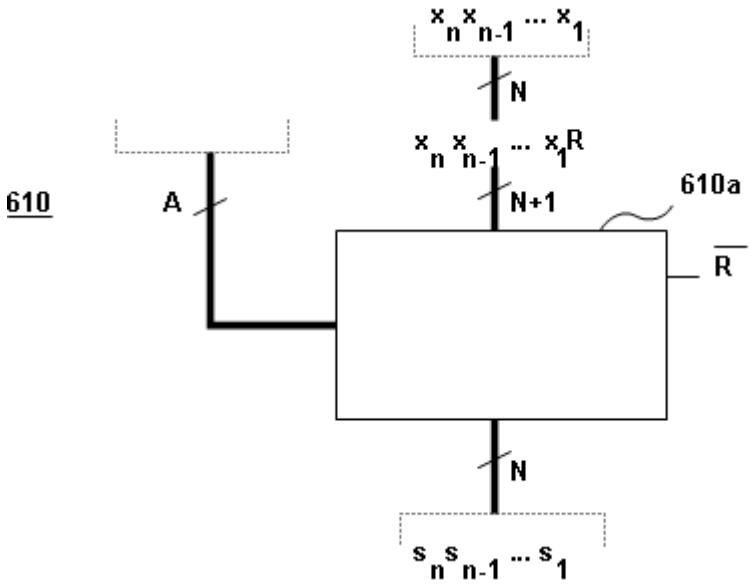

$x_n x_{n-1} \dots x_1$
N
$x_n x_{n-1} \dots x_1 R$
610
A
N+1
610a
$\overline{R}$
N
$s_n s_{n-1} \dots s_1$

Fig. 4a

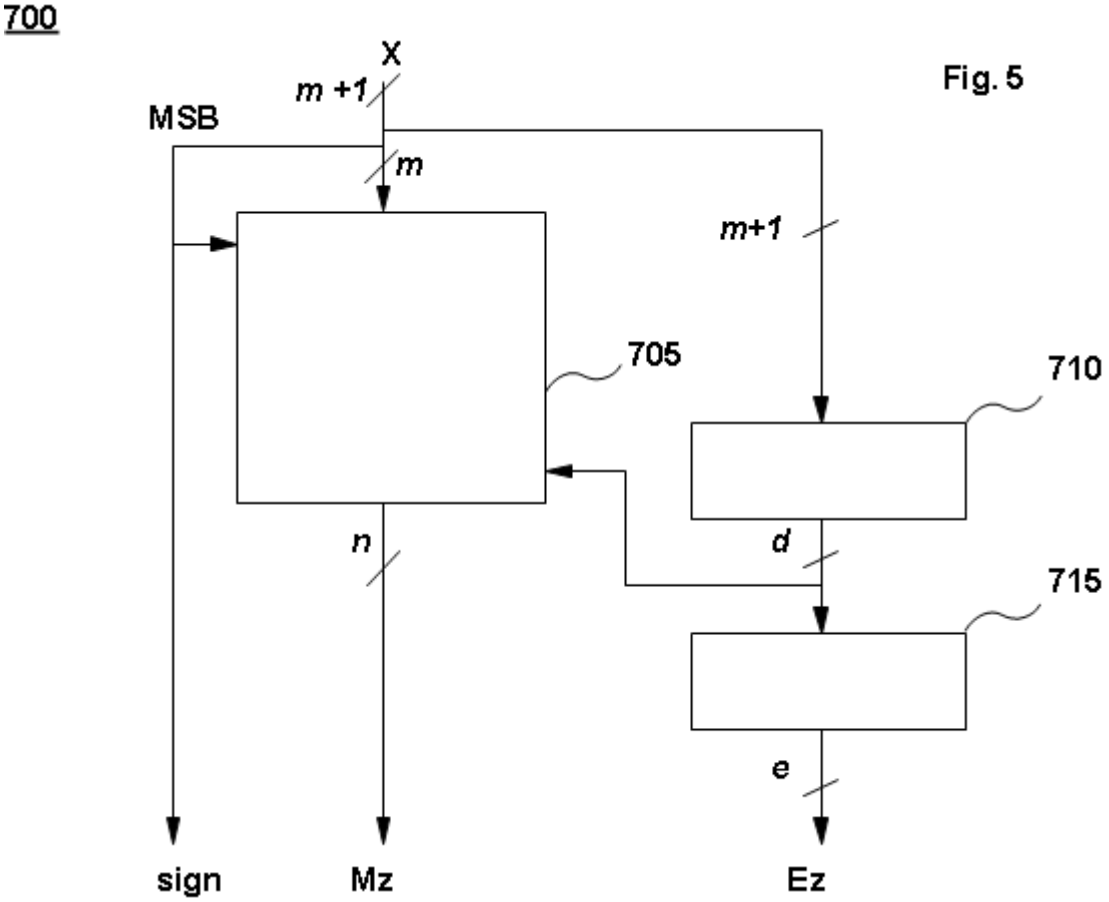


Fig. 5

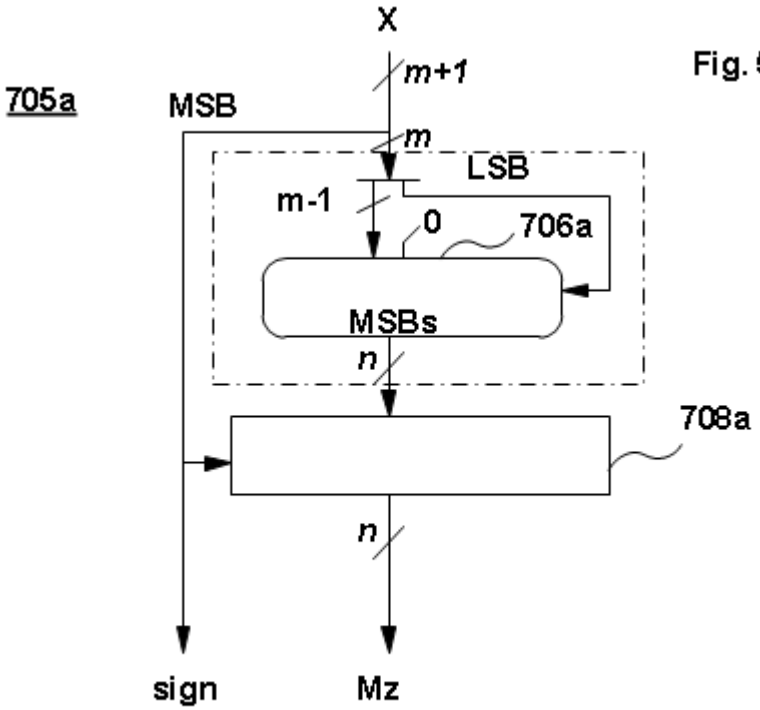

X
m+1
705a
MSB
m
LSB
m-1
0
706a
MSBs
n
708a
n
sign
Mz

Fig. 5a

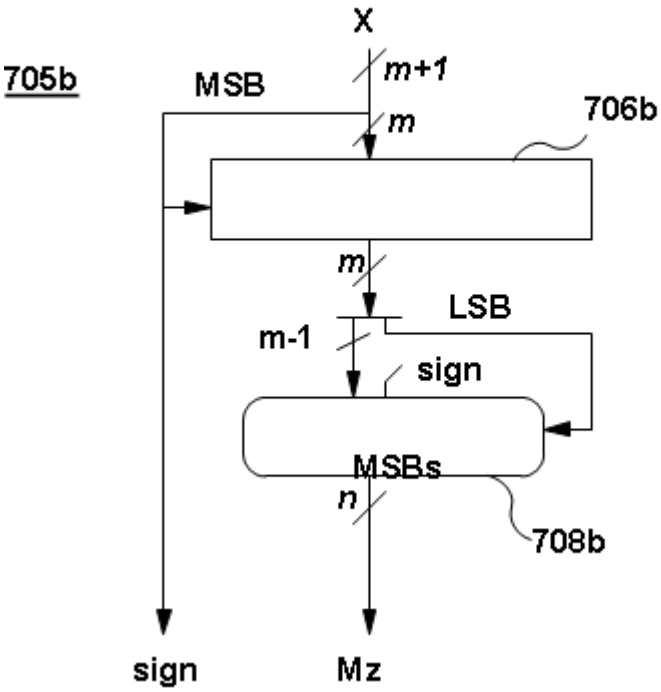

X
705b
MSB
m+1
m
706b
m
LSB
m-1
sign
MSBs
n
708b
sign
Mz

Fig. 5b

800a

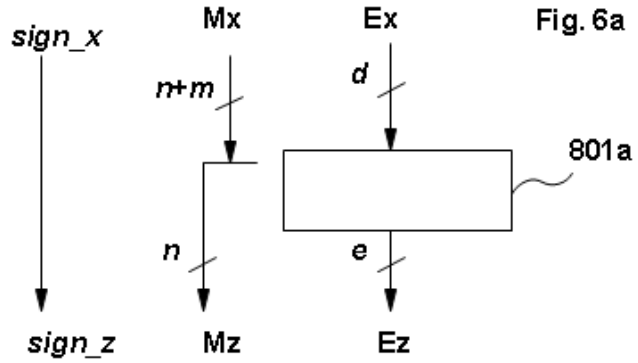


Fig. 6a

800b

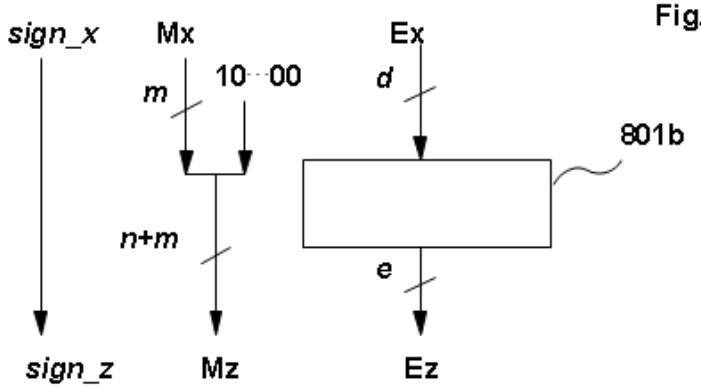


Fig. 6b

800c

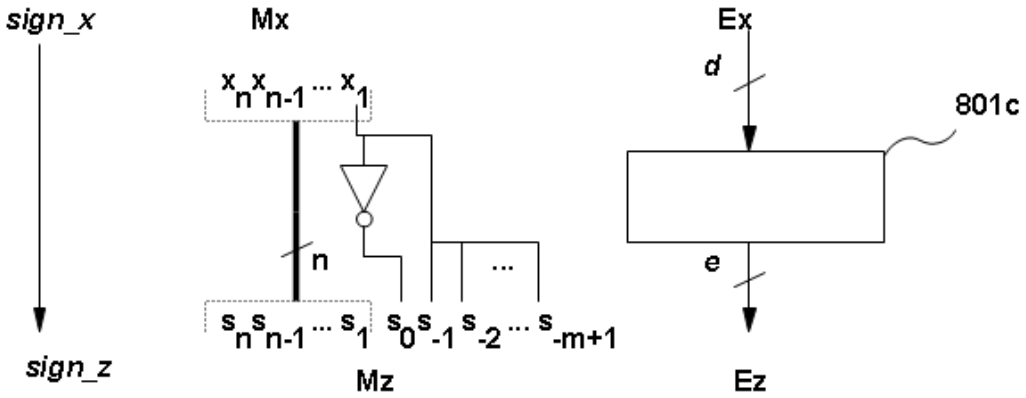


Fig. 6c

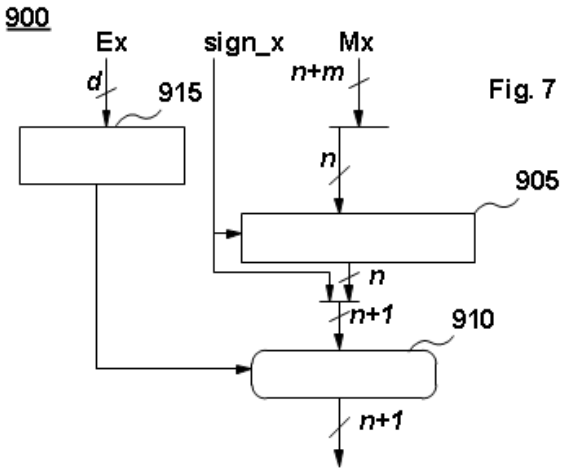


Fig. 7

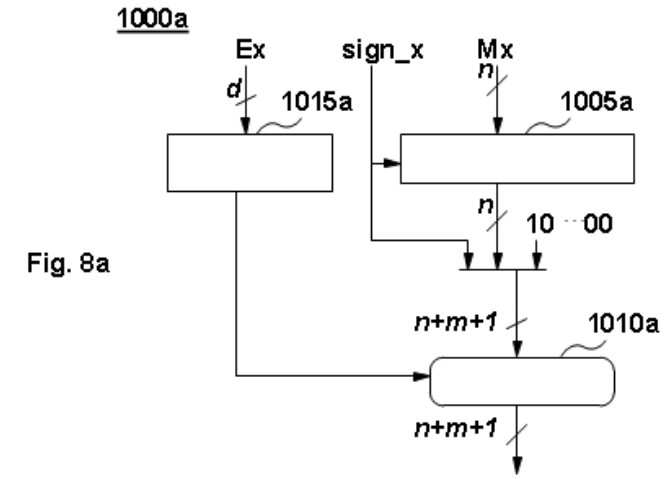


Fig. 8a

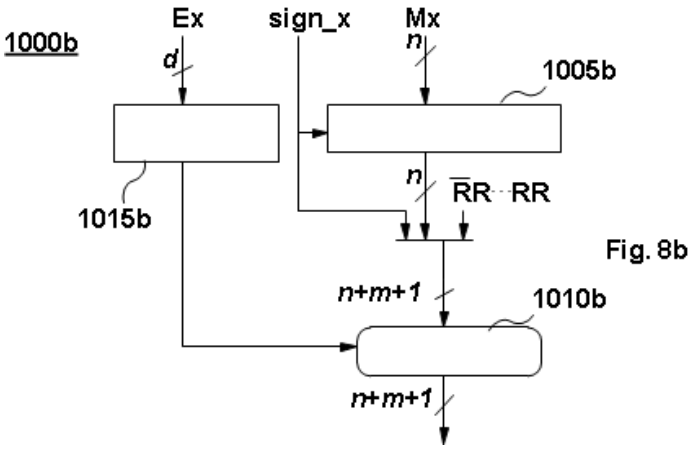


Fig. 8b

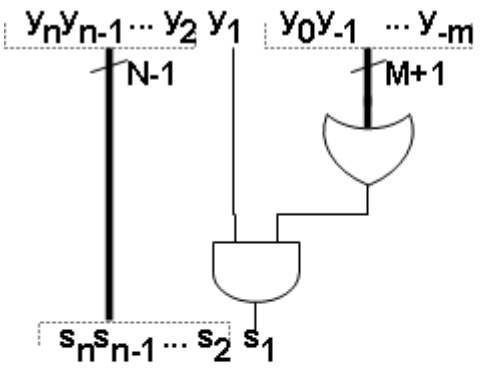


Fig. 9

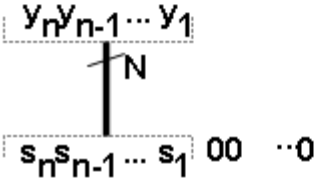


Fig. 10a

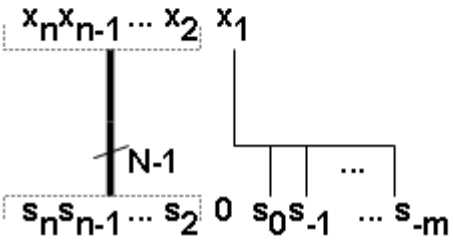


Fig. 10b

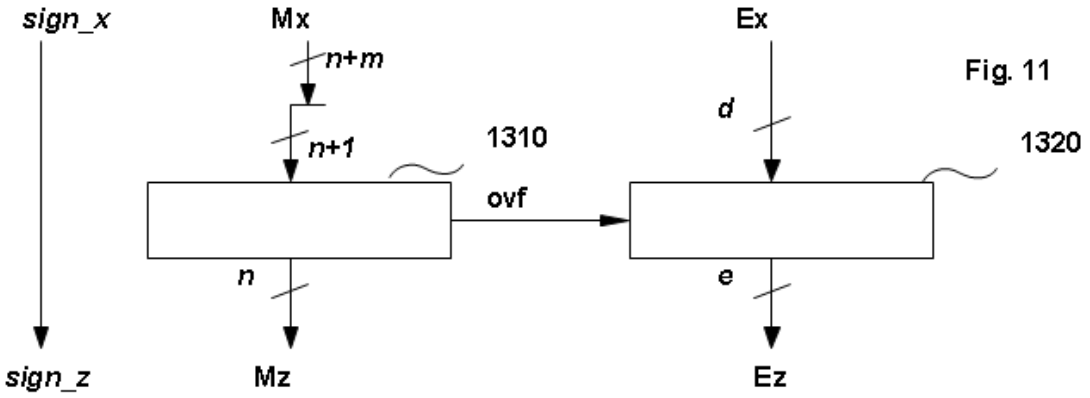


Fig. 11

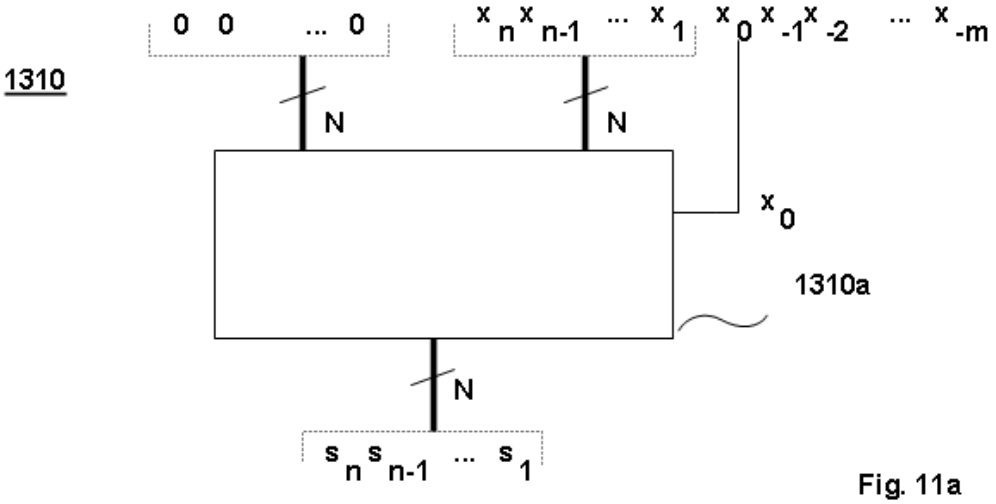
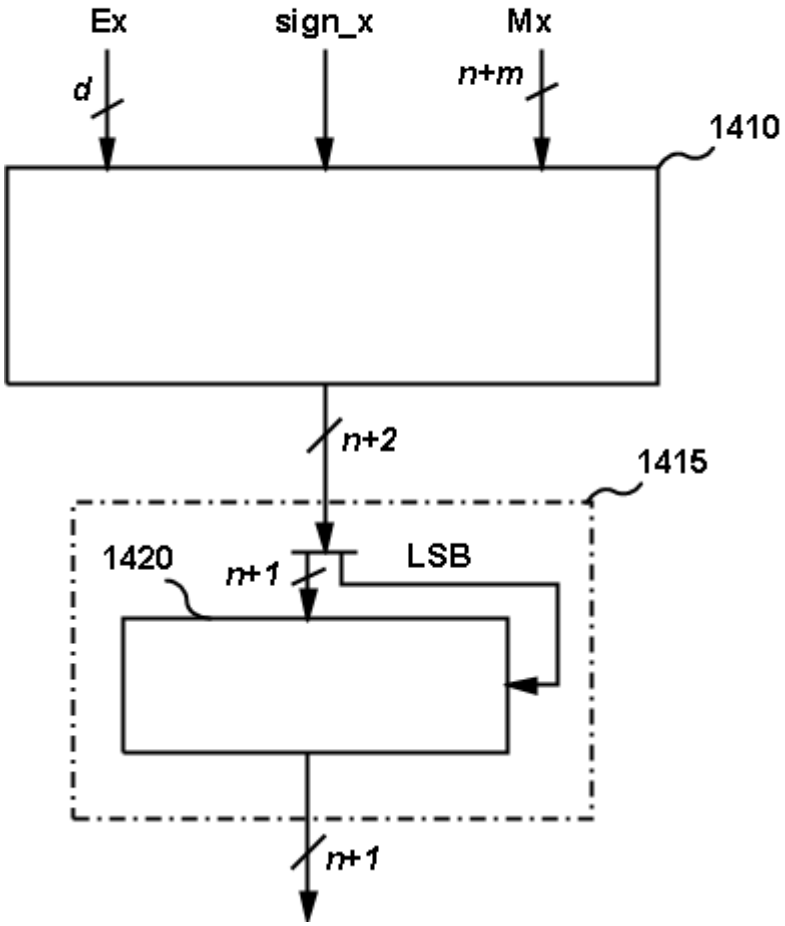


Fig. 11a

Fig. 12



OFICINA ESPAÑOLA
DE PATENTES Y MARCAS

ESPAÑA

(21) N.º solicitud: 201430454

(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,	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.	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: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: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**
02.02.2015	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: 02.02.2015

Declaración

Novedad (Art. 6.1 LP 11/1986)	Reivindicaciones	1-62	**SI**
	Reivindicaciones		**NO**
Actividad inventiva (Art. 8.1 LP11/1986)	Reivindicaciones	1-62	**SI**
	Reivindicaciones		**NO**

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

Base de la Opinión.-

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

1. Documentos considerados.-

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

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

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

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

Reivindicaciones independientes

Reivindicación 1: El documento D01, divulga una unidad de cómputo aritmético para realizar operaciones de suma o multiplicación de coma flotante con caminos de datos para la mantisa y el exponente respectivo. Los operandos tienen un bit de valor implícito 1 (el más significativo). Los resultados se redondean y normalizan.
La diferencia entre el documento D01 y la reivindicación 1 es que el bit implícito es el menos significativo y su efecto técnico es la simplificación de los cálculos de redondeo y truncamiento. El problema técnico objetivo consistiría así en cómo simplificar los cálculos habituales que se realizan en las operaciones de suma y multiplicación.
El documento D02 por su parte divulga una implementación de un sumador en 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 2-62: 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.