(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 546 898**

(21) Número de solicitud: 201430455

(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 coma flotante y conversores**

(57) Resumen:

Dispositivos para convertir números a y desde un formato 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. Los dispositivos están configurados para ser conectados a unidades aritméticas. Las unidades aritméticas están configuradas para procesar al menos un primer número en coma flotante pre-procesado y generar al menos un segundo número en coma flotante pre-procesado. Los dispositivos están configurados para convertir un número de entrada a dicho al menos primer número en coma flotante pre-procesado o dicho al menos segundo número en coma flotante pre-procesado a un número de salida. En otras realizaciones, métodos son propuestos, ejecutados en sistemas de procesamiento de datos comprendiendo un procesador, para representar números reales en un formato pre-procesado. Los métodos comprenden computar, por el procesador, el mayor número de M dígitos, en base B, menor que el número real y fijar, por el procesador, los M MSDs del número en coma fija de M+1 dígitos igual a los M dígitos de dicho mayor número de M dígitos.

**ES 2 546 898 A1**

# Dispositivos coma flotante y conversores

## DESCRIPCIÓN

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

## ESTADO DE LA TÉCNICA

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

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

En coma flotante, el número se compone de la mantisa (Ma), la base (B) y el exponente (Ex). Por lo tanto, el valor (Va) representado sería Va = B * Ma ^ Ex. Entonces, solamente los números Ma y Ex necesitan almacenarse. El formato estándar IEEE-754 es el más extendido. El estándar define cinco formatos básicos que llevan el nombre de su base numérica y el número de bits usados en su codificación de intercambio. La precisión típica de los formatos binarios básicos es un bit más que la anchura de su mantisa (o mantisa). El bit de precisión extra proviene de un bit a uno implícito (oculto) en la parte más significativa. El número en coma flotante típico estará normalizado tal que el bit más significativo será un uno. Si conocemos que el bit más significativo es uno, entonces no se necesita codificarlo en el formato de intercambio.

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

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

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

Para realizar el complemento a la base de un número, primero se realiza el complemento a la base menos uno, una operación que se realiza sobre todos los dígitos en paralelo. Posteriormente se le suma al número una unidad-en-el-último lugar (ULP). En el caso binario, para que un circuito que lleva a cabo el complemento a dos de un número de N bits serían necesarios N inversores y un sumador de N bits. En el caso de una operación de resta (X-Y = X+(-Y)), que en realidad consiste en una suma con el complemento a dos del sustraendo, el bit de entrada de acarreo del sumador se suele utilizar para añadir el ULP. Sin embargo, esto no significa que cada vez que se requiere

llevar a cabo el complemento a dos el motivo es una resta. Tales casos son la operación de valor absoluto o la suma/resta de números en representación signo-magnitud, una representación típicamente usada en coma flotante.

Con respecto a los circuitos de redondeo, se utilizan varias formas de redondeo. Una que demuestra importantes propiedades y es la más utilizada es el "redondeo al par más cercano". En este modo, el valor que se utiliza como valor final es el valor que está más cerca del valor real y, en caso de empate, el valor par. Usando este tipo de redondeo, se obtiene un error inferior a +-0.5ULP y no presenta ningún sesgo en los errores.

Dado un número de D1 dígitos, para realizar una operación de redondeo a D2 dígitos, asumiendo D1 > D2, D1-D2 dígitos deben desecharse. Para que el redondeo sea al número más cercano, es importante examinar el valor del dígito más significativo de los que necesitan ser desechados (MD) y el dígito menos significativo de los que quedan (LD):

- Si MD < (B/2) entonces simplemente dichos dígitos son descartados.
- Si MD > (B/2) entonces dichos dígitos se descartan y se añade el valor uno al dígito menos significativo que permanece.
- Si MD = (B/2) entonces se debe verificar si alguno de los dígitos a descartarse no es cero (sticky bit). Si es así, entonces el redondeo se realiza según el segundo caso. Si todos son cero, entonces si el dígito LD es par entonces el redondeo se realiza según el primer caso y si es impar según el segundo caso.

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

Los circuitos de complemento a la base y redondeo son necesarios en las unidades funcionales tales como sumadores, multiplicadores, divisores, unidades FMAD, operadores de valor absoluto, conversores de formato o conversores de precisión etc. El coste adicional, por ejemplo en el área o retardo, que plantean dichos circuitos en las mencionadas unidades funcionales es generalmente substancial, sobre todo porque están típicamente en la vía crítica.

En el estado 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 conversores configurados para ser conectados 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 conversores configurados para ser conectados a 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 configurado para ser conectado a una unidad aritmética. Dicha unidad aritmética está configurada para procesar al menos un primer número en coma flotante pre-procesado y generar al menos un segundo número en coma flotante pre-procesado. Dichos números en coma flotante pre-procesados tienen una mantisa con un LSD igual a B/2, B siendo la base del sistema numérico. El dispositivo está configurado para convertir un número de entrada a dicho al menos primer número en coma flotante pre-procesado o dicho al menos segundo número en coma flotante pre-procesado a un número de salida.

Una ventaja del dispositivo es que permite operar, en una unidad aritmética para números en coma flotante pre-procesados, números representados en otro formato diferente. Además, realiza las operaciones mencionadas sin usar explícitamente el LSD de la mantisa de los números en coma flotante pre-procesados. El formato pre-procesado 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, en muchos casos, no hay ninguna necesidad de tener una lógica para calcular el sticky bit. La eliminación de la lógica para el cálculo del sticky bit es posible porque en estos casos, se conoce a priori que el sticky bit es uno, debido a que el último dígito oculto siempre es necesariamente B/2 (dígito KD). Por último, otra ventaja es que no puede ocurrir desbordamiento después del redondeo.

En las siguientes descripciones de las realizaciones se considera generalmente que losformatos coma flotante usan mantisas sin signo y normalizadas, con un bit de signo independiente, pero con el bit entero incluido en dicha mantisa, sin embargo, alguien experto en el estado de la técnica, podría aplicar las enseñanzas divulgadas aquí de una forma directa, también para otros formatos, tales como mantisas con signo y/o bit entero implícito. De manera similar, , los números en coma fija, tanto los no procesados, como los procesados, son representados en representación en complemento a dos, pero mínimas modificaciones de las realizaciones presentadas aquí son requeridas para soportar otros formatos,

En algunas realizaciones, cuando el número de entrada es un número en coma fija pre-procesado de N+2 bits y el primer número en coma flotante pre-procesado tiene una mantisa de M+2 bits, el conversor de entrada podría

comprender un calculador de cantidad de desplazamiento, un módulo para calcular el exponente, con una primera entrada para recibir la cantidad de desplazamiento del calculador de cantidad de desplazamiento, y una salida para generar el exponente del primer número coma flotante pre-procesado, y un calculador de mantisa. El calculador de 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 pre-procesado y una segunda entrada para recibir la 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, y completar las posiciones vacantes fijando el MSB de las posiciones vacantes a cero, y el resto, a unos, o fijando el MSB de las posiciones vacantes a uno, y el resto, a cero, para generar como mucho los M+1 MSBs de la mantisa. El signo del primer número coma flotante pre-procesado podría corresponder al MSB del número coma fija pre-procesado. Introduciendo un conversor de este tipo antes de la unidad aritmética permite que un número en formato de coma fija pre-procesado sea procesado por unidades aritméticas de acuerdo con 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. Estas configuraciones permiten al módulo de normalización eliminar el sesgo del redondeo.

En algunas realizaciones, el módulo de normalización del calculador de mantisa podría estar configurado además para generar selectivamente el complemento a uno del resultado de dicho desplazamiento. Esta configuración permite al módulo de normalización generar siempre una mantisa positiva.

En algunas realizaciones, cuando el número de entrada es un número en coma fija no procesado de R bits, y el primer número coma flotante pre-procesado tiene una mantisa de M+2 bits, dicho al menos un conversor de entrada podría comprender 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 primer número pre-procesado en coma flotante, y un calculador de exponentes, con una primera entrada para recibir una cantidad de desplazamiento proveniente del calculador de cantidad de desplazamiento, y una salida para generar el exponente del primer número en coma flotante pre-procesado. El signo del primer número en coma flotante pre-procesado podría corresponderse con el MSB del número no procesado en coma fija. Introduciendo un conversor de este tipo antes de la unidad aritmética permite que un número en formato de coma fija no procesado sea procesado por unidades aritméticas de acuerdo con las realizaciones descritas aquí.

En algunas realizaciones, el módulo de normalización podría comprender una primera entrada para recibir los R bits del número en coma fija no procesado y una segunda entrada para recibir la cantidad de desplazamiento. El módulo de normalización podría entonces estar configurado para generar un valor que corresponde a como mucho los M+1 MSBs de la mantisa del primer número coma flotante pre-procesado, 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. Esta configuración permite al módulo de normalización eliminar el sesgo del redondeo.

En algunas realizaciones, el módulo de normalización podría estar configurado además para generar selectivamente el complemento a uno de

dicho valor generado, si la entrada es negativa. Esta configuración permite al módulo de normalización generar siempre una mantisa positiva.

En algunas realizaciones, el módulo de normalización podría comprender un desplazador variable configurado para recibir un bit para completar las posiciones vacantes.

En algunas realizaciones, el desplazador variable 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)]. Cada multiplexor podría estar configurado para efectuar una operación de desplazamiento a la izquierda de 2^i posiciones, i$\epsilon$[0, número de multiplexores-1], y cada multiplexor configurado para completar las posiciones vacantes usando el valor de dicho bit recibido.

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

En algunas realizaciones, el módulo de normalización podría estar configurado además para generar selectivamente el complemento a uno del resultado de la mencionada operación de desplazamiento. Esta configuración permite al módulo de normalización generar siempre una mantisa positiva y eliminar el sesgo en algunos casos.

En algunas realizaciones, cuando el número de entrada es un número coma flotante pre-procesado con una mantisa de J+2 bits y el primer número en coma flotante pre-procesado tiene una mantisa con J+2-P bits, P<J+1, entonces el al menos primer conversor podría comprender una unidad de redondeo para eliminar los P+1 LSBs de los J+2 bits de la mantisa del número de entrada, para generar como mucho J+1-P MSBs de la mantisa del primer número en coma flotante pre-procesado. El LSB de la mantisa del primer número coma flotante pre-procesado es igual a 1 y podría estar implícito. El conversor podría comprender además un calculador de exponentes para generar el exponente del primer número en coma flotante pre-procesado.

En otras realizaciones, cuando el número de entrada es un número coma flotante pre-procesado con una mantisa de J+2 bits y el primer número en coma flotante pre-procesado tiene una mantisa con J+2+Q bits, entonces el primer conversor podría comprender un módulo de rellenado, configurado para recibir como mucho los J+1 MSBs de la mantisa del número de entrada y generar como mucho los J+Q+1 MSBs de la mantisa del primer 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. Los como mucho J+1 MSBs de la mantisa del primer número en coma flotante pre-procesado podrían ser los mismos que los como mucho J+1 MSBs de la mantisa del número de entrada. El conversor de entrada podría comprender además un calculador de exponentes para generar el exponente del primer número en coma flotante pre-procesado.

El módulo de rellenado podría entonces 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, del número coma flotante pre-procesado. En otras implementaciones, una nueva entrada podría configurarse. Estas

configuraciones permiten al módulo de rellenado eliminar el sesgo del redondeo.

Alguien experto en la técnica podría entender que el conversor descritos en estas realizaciones para convertir un número en coma flotante pre-procesado a otro, con al menos una mantisa de diferente tamaño, podría utilizarse para convertir el segundo número en coma flotante pre-procesado, en el número de salida.

En algunas realizaciones, cuando el número de entrada es un número coma flotante no procesado con una mantisa de E+2 bits y el primer número en coma flotante pre-procesado tiene una mantisa con E+2-D bits, D<E+1, entonces el conversor 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 los E+1-D MSBs de la mantisa del primer número coma flotante pre-procesado. El LSB de la mantisa del primer número en coma flotante pre-procesado es igual a uno y podría estar implícito. El conversor de entrada podría comprender además un calculador de exponentes para generar el exponente del primer número en coma flotante pre-procesado.

En algunas realizaciones, la unidad de redondeo podría estar configurada además para, selectivamente, poner a cero el segundo LSB de la mantisa del primer 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. Esta configuración permite a la unidad de redondeo eliminar el sesgo del redondeo.

En algunas realizaciones, cuando el número de entrada es un número coma flotante no procesado con una mantisa de E+2 bits y el primer número en coma flotante pre-procesado tiene una mantisa con E+2+G bits, entonces el primer conversor podría comprender un módulo de rellenado, configurado para recibir la mantisa del número en coma flotante no procesado y generar los E+G+1

MSBs de la mantisa del primer número en coma flotante pre-procesado, fijando los E+2 MSBs del primer número en coma flotante pre-procesado al mismo valor que 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 primer número en coma flotante pre-procesado es igual a uno y podría estar implícito. El conversor de entrada podría comprender además un calculador de exponentes configurado para generar el exponente del primer número en coma flotante pre-procesado.

En algunas realizaciones, módulo de rellenado podría estar configurado además para generar selectivamente el valor correspondiente a restar uno del segundo LSB de la mencionada mantisa generada cuando un bit seleccionado, o una combinación de bit seleccionados, de la mantisa no procesada de entrada es igual a uno. Esta configuración permite al módulo de rellenado eliminar el sesgo del redondeo.

En algunas realizaciones, cuando el segundo número en coma flotante pre-procesado tiene una mantisa con F+2 bits y el número de salida es un número coma fija pre-procesado de L bits, donde L<F+4, entonces el segundo conversor podría comprender un calculador de la cantidad de desplazamiento que recibe el exponente del segundo número en coma flotante pre-procesado, en una entrada, y que genera una cantidad de desplazamiento, en una salida. El conversor de salida podría comprender además un módulo de desplazamiento con una primera entrada, para recibir los L-1 MSBs de la mantisa del segundo 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 segundo 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 podría comprender un desplazador aritmético a la derecha acoplado a un inversor de bit condicional.

En algunas realizaciones, cuando el segundo número en coma flotante pre-procesado tiene una mantisa de F+2 bits y el número de salida es un número coma fija pre-procesado de F+C+3 bits, C>0, el conversor podría comprender un calculador de cantidad de desplazamiento que recibe el exponente del segundo número pre-procesado en una entrada, y que genera una cantidad de desplazamiento en una salida. El conversor podría comprender además 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 segundo número en coma flotante pre-procesado, y el MSB de los C LSBs puesto a cero y el resto a uno, o el MSB de los C LSBs puesto a uno y el resto a cero.

En estas realizaciones de dicho conversor , el módulo de desplazamiento aritmético a la derecha podría estar 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. En algunas implementaciones dicho bit (o bits) podrían seleccionarse, del segundo número coma flotante pre-procesado. En otras implementaciones, una nueva entrada podría configurarse. Estas configuraciones permiten al módulo de desplazamiento aritmético a la derecha eliminar el sesgo del redondeo.

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. Esta configuración permite al

módulo de desplazamiento aritmético a la derecha generar números en formato complemento a dos, si la mantisa es sin signo.

Introduciendo un conversor de este tipo para convertir números coma flotante pre-procesados a números coma fija pre-procesados después de la unidad aritmética permite que el resultado de las operaciones sea usado por circuitos que funcionan con formato coma fija pre-procesado.

En algunas realizaciones, cuando el segundo número coma flotante pre-procesado tiene una mantisa de Z bits y el número de salida es un número coma fija no procesado de H+1 bits, el dispositivo podría comprender un conversor para convertir un número coma flotante pre-procesado a un número coma fija pre-procesado, de acuerdo con algunas realizaciones presentadas aquí, teniendo una salida de H+2 bits, conectada a un módulo de redondeo. Introduciendo un conversor de este tipo después de la unidad aritmética permite que el resultado de las operaciones sea usado por circuitos que funcionen con formato coma fija no procesado.

El módulo de redondeo 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 dicha entrada si el LSB de dicha salida es igual a 1. Esta configuración permite al módulo de redondeo realizar redondeo al más cercano. Alguien experto en el estado de la técnica podría apreciar que diferentes modos de redondeo podrían implementarse de una forma directa.

En algunas realizaciones, cuando el segundo número en coma flotante pre-procesado tiene una mantisa de U+2 bits y el número de salida es un número coma flotante no procesado con una mantisa de U-V+2 bits, V>0, el conversor

de salida podría comprender un módulo de redondeo, configurado para recibir como mucho los U+3-V MSBs de la mantisa del segundo número en coma flotante pre-procesado, y generar como mucho U+2-V bits de la mantisa del número en coma flotante no procesado. El conversor podría comprender además un calculador de exponentes configurado para generar el exponente del número en coma flotante no procesado.

En algunas realizaciones, el módulo de redondeo podría comprender un sumador. Dicho sumador podría estar configurado para recibir, en una entrada, como mucho los U+2-V MSBs de la mantisa del segundo 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 estas realizaciones, el calculador de exponentes podría estar configurado, además, para incrementar el exponente de salida cuando se genera la mencionada instrucción del módulo de redondeo.

En algunas realizaciones, cuando el segundo número en coma flotante pre-procesado tiene una mantisa de U+2 bits y el número de salida es un número coma flotante no procesado con una mantisa de U+2+W bits, entonces el conversor de salida podría comprender un módulo de rellenado, configurado para recibir como mucho los U+1 MSBs de la mantisa del segundo número en coma flotante pre-procesado y generar 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. El conversor de salida podría comprender además un calculador de exponentes configurado para generar el exponente del número en coma flotante pre-procesado.

Introduciendo un conversor de este tipo para convertir números coma flotante pre-procesados a números coma flotante no procesados después de la unidad aritmética permite que el resultado de las operaciones sea usado por circuitos en coma flotante comunes.

En un otro aspecto un procedimiento, en un sistema de procesado de datos comprendiendo un procesador, para representar un número real en un formato pre-procesado, se propone. Dicho formato pre-procesado es la representación del número real mediante el número más cercano en coma fija de M+1 dígitos teniendo el LSD igual a B/2, B siendo la base del sistema numérico.

El procedimiento comprende las etapas de calcular, por el procesador, el mayor número en base B de M dígitos menor que el número real, y fijar, por el procesador, los M MSDs del número en coma fija de M+1 dígito igual a los M dígitos de dicho mayor número de M dígitos.

En algunas realizaciones, el procedimiento podría comprender además la etapa de fijar, por el procesador, el LSD del número en coma fija de M+1 dígito igual a B/2.

En algunas realizaciones, cuando el número real es un número coma fija ya representado en un formato en base B y comprende N+1 dígitos, N>M, el procedimiento podría comprender además las etapas de fijar, por el procesador, los N-M-1 MSDs de los N-M LSDs del número en coma fija de N+1 dígitos a cero y fijar, por el procesador, el (M+1)-ésimo MSD a B/2.

En algunas realizaciones, el procedimiento podría comprender además las etapas de fijar, por el procesador, el LSD del número en coma fija de N+1 dígitos a cero; detectar, por el procesador, si dichos N-M-1 MSDs y dicho LSB eran igual a cero antes de dicha operación de fijar; y fijar, por el procesador, el M-ésimo MSD a cero en respuesta a dicha identificación.

En algunas realizaciones del procedimiento el número real podría ser la mantisa de un número en coma flotante.

## 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 muestra un ejemplo de una unidad aritmética conectada a un conversor de entrada y un conversor de salida;

Fig. 2 ilustra un ejemplo de implementación de un conversor de números coma fija pre-procesados a números coma flotante pre-procesados;

Fig. 2a ilustra un ejemplo de implementación de un desplazador a la izquierda pre-procesado;

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

Fig. 3 ilustra un ejemplo de implementación de un conversor de números coma fija no procesados a números coma flotante pre-procesados;

Fig. 3a and 3b 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. 4a, 4b and 4c ilustran ejemplos de implementación de un conversor de números coma flotante pre-procesados a números coma flotante pre-procesados;

Fig. 5, 6a and 6b ilustran ejemplos de implementación de un conversor de números coma flotante pre-procesados a números coma fija pre-procesados;

Fig. 7, 8a, 8b 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. 9 ilustra un ejemplo de implementación de un conversor de números coma flotante pre-procesados a números coma flotante no procesados;

Fig. 9a 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. 10 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

El dispositivo descrito aquí requiere números FP que hayan sido pre-procesados de acuerdo con la invención como se describió aquí. Estos números pre-procesados podrían ser generados por circuitos, tales como las mencionadas unidades aritméticas, que están diseñados para funcionar con números pre-procesados o podrían ser generados por conversores, diseñados para convertir número no procesados o números pre-procesados no FP en números pre-procesados. Además, los números pre-procesados generados por las unidades aritméticas descritas arriba podrían, en concordancia, requerir conversores tales que los números generados podrían ser usados por circuitos que no estén diseñados para operar números pre-procesados. Fig.1 muestra un ejemplo de dispositivo de acuerdo con las realizaciones descretas aquí. El dispositivo 100 comprende una unidad aritmética 100C configurado para procesar números en coma flotante pre-procesados y generar números en coma flotante pre-procesados. Un conversor de entrada 110C está conectado a la entrada de dicho dispositivo. El conversor de entrada 110C está configurado para convertir un número de entrada a un primer número en coma flotante pre-procesado. En concordancia, el dispositivo comprende un conversor de salida 120C, conectado a la salida de la unidad aritmética 100C, y configurado para recibir un segundo número en coma flotante pre-procesado y generar un número de salida. Dichos números de entrada y salida podrían ser números pre-procesados o no procesados, o en coma fija o en coma flotante. Además el conversor 110C y/o 120C podrían ser internos a la unidad aritmética 100C. En otras implementaciones solo uno de los conversores podría estar presente a la entrada o la salida de la unidad aritmética 100C. En otras implementaciones, el dispositivo podría comprender una pluralidad de conversores en la entrada y/o

la salidade dicha unidad aritmética 100C para convertir, por ejemplo, en paralelo, una pluralidad de números de entrada respectivamente.

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

a) en FP
   - representación implícita del MSB de la mantisa, o
   - representación fusionada del signo y la mantisa mediante representación en complemento a dos o cualquier otra representación
b) en coma fija: representación signo-magnitud, o representación natural

Una categoría de tales conversores es la de conversores para convertir números en coma fija pre-procesados a números FP pre-procesados. La Fig. 2 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 entero pre-procesado de m+2 bits. El MSB de 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 605. La salida de m bits del inversor de bits condicional 605 es la entrada al desplazador a la izquierda pre-procesado 610. En implementaciones alternativas el desplazador a la izquierda pre-procesado 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. 2a. 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. 2b. El desplazador a la izquierda especial 245, mostrado en Fig. 2b, 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 especial, 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 la Fig. 2b, la máxima cantidad de desplazamiento es m-1.

Volviendo al ejemplo de la Fig. 2, 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. 2a 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 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. 3 ilustra un conversor de este tipo. El conversor 700 comprende un módulo de normalización 705 configurado para recibir los m LSBs de un número en coma

fija m+1 bits. El MSB del número en coma fija es el signo y se utiliza para controlar el módulo de normalización 705, y para poner el signo del número FP pre-procesado. El módulo de normalización 705 podría comprender un desplazador a la izquierda convencional, para desplazar el valor de entrada hasta eliminar los bits no significativos de la izquierda, seguido de un inversor de bit condicional para calcular el complemento a uno de dicho valor desplazado si el número de entrada es negativo. Esta configuración evita el sesgo redondeando hacia arriba los números de entrada positivos y hacia abajo los negativos. Además, el módulo de normalización podría ser implementado de acuerdo con los ejemplos descritos en la Fig. 3a y en la Fig. 3b. En la Fig. 3a, 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. 2a. En este caso el desplazador a la izquierda especial 706a recibe los m-1 MSBs de los m LSBs del número en coma fija no procesado, extendidos a la derecha con un bit con valor cero y el LSB del número en coma fija se utiliza como la tercera entrada del desplazador a la izquierda especial 706a. La salida del desplazador a la izquierda especial 706a corresponde a los n MSBs 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 MSBs 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. 3b 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 700de la Fig. 3, un camino paralelo comprende módulo LOD 710 que tiene una entrada que recibe el número en coma fija no procesado y una salida para la generación de la cantidad de desplazamiento para el módulo de normalización 705 que también se utiliza como entrada al módulo de computación del exponente 715 para generar el exponente Ez del número FP pre-procesado. En otras implementaciones que podrían utilizar el módulo de normalización 705b, la entrada del módulo LOD 710 podría recibir la salida del inversor de bits condicional 706b en su lugar.

Otra categoría de conversores son los conversores para convertir números FP pre-procesados a números FP pre-procesados de diferente tamaño de mantisa. La Fig. 4a 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. 4b 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. 4c 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, tal como se muestra en la Fig. 4c.

Otra categoría de conversores son los conversores para convertir números FP pre-procesados a números en coma fija pre-procesados. La Fig. 5 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 MSBs 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. 6a 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 de bit condicional 1005a. La salida del inversor de bits condicional 1005a junto con el signo (sign_x) son introducidos al desplazador a la derecha 1010a. La salida del inversor de bits condicional se expandió mediante la adición por la derecha 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. 6b 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 MSBs 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. 4 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. 7. Para una mantisa de entradade 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. 8a. El LSB será igual a 1 y está implícito.

b) en el caso de redondeo sin sesgo, los n-1 MSBs son los mismos. El enésimo bit se fuerza a cero. Los m +1 bits a la derecha se hacen igual al LSB de la mantisa no procesada. Esto se ilustra en la Fig. 8b. El LSB de la mantisa pre-procesada será 1 y está implícito, 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. 9 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. 9a. Un sumador 1310a se usa para incrementar en uno los n MSBs de la mantisa pre-procesada si el (n+1)-ésimo MSB es uno. En implementaciones alternativas diferentes unidades de redondeo que realizan diferentes modos de redondeo podrían ser usadas. Cuando la mantisa del número FP pre-procesado tiene menos bits (m+1) que la mantisa no procesada (m+n), entonces se podría utilizar el circuito ilustrado en la Fig. 4b. 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 números en coma fija no procesados. La Fig. 10 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. 5. 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. 9a 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 del número en coma fija de salida, un conversor de este tipo podría ser idéntico al conversor 1000a descrito en la Fig. 6a

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 configurado para ser conectado a una unidad aritmética, dicha unidad aritmética configurada para procesar al menos un primer número en coma flotante pre-procesado para generar al menos un segundo número en coma flotante pre-procesado, dichos números en coma flotante pre-procesados teniendo una mantisa con un LSD igual a B/2, B siendo la base del sistema numérico, dicho dispositivo siendo configurado para convertir un número de entrada a dicho al menos primer número en coma flotante pre-procesado o dicho al menos segundo número en coma flotante pre-procesado a un número de salida.

2. Dispositivo según reivindicación 1 en el que, cuando el número de entrada es un número coma fija pre-procesado de N+2 bits y el primer número coma flotante pre-procesado tiene una mantisa de M+2 bits, el dispositivo comprende:

un calculador de cantidad de desplazamiento,

un módulo para calcular el exponente, con una primera entrada para recibir la 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 mantisa, comprendiendo:

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 cantidad de desplazamiento; dicho módulo de normalización configurado para desplazar a la izquierda dichos N MSBs de acuerdo con dicha cantidad de desplazamiento, y completar las posiciones vacantes fijando el MSB de las posiciones vacantes

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

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

3. Dispositivo según reivindicación 2 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.

4. Dispositivo según reivindicación 2 ó 3, en el que dicho módulo de normalización está configurado además para generar selectivamente el complemento a uno del resultado de dicho desplazamiento.

5. Dispositivo según las reivindicación 1, en el que, cuando el número de entrada es un número coma fija no procesado de R bits y el primer número coma flotante pre-procesado tiene una mantisa de M+2 bits, dicho dispositivo 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 la mantisa del primer número en coma flotante pre-procesado,

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

donde el signo del primer número en coma flotante pre-procesado se corresponde con el MSB del número no procesado en coma fijo.

6. Dispositivo según la reivindicación 5, 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 cantidad de desplazamiento, donde el módulo de normalización está configurado para generar un valor que corresponde a como mucho los M+1 MSB de la mantisa del primer número coma flotante pre-procesado 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.

7. Dispositivo según la reivindicación 6, 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.

8. Dispositivo según cualquiera de las reivindicaciones 2 a 7, en el que el módulo de normalización contiene un desplazador variable configurado para recibir un bit para completar las posiciones vacantes.

9. Dispositivo según la reivindicación 8, 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.

10. Dispositivo según la reivindicación 5, 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 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 del primer número coma flotante pre-procesado mediante el desplazamiento a la izquierda de los R-1 LSBs de la primera entrada.

11. Dispositivo según la reivindicación 10, 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.

12. Dispositivo según la reivindicación 1, en el que, cuando el número de entrada es un número coma flotante pre-procesado con una mantisa de J+2 bits y el primer número en coma flotante pre-procesado tiene una mantisa con J+2-P bits, P<J+1, entonces el dispositivo comprende:

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

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

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

13. Dispositivo según reivindicación 1 , en el que, cuando el número de entrada es un número coma flotante pre-procesado con una mantisa de J+2 bits y el primer número en coma flotante pre-procesado tiene una mantisa con J+2+Q bits, entonces el dispositivo comprende:

un módulo de rellenado, configurado para recibir como mucho los J+1 MSBs de la mantisa del número de entrada y generar como mucho los J+Q+1 MSBs de la mantisa del primer 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 primer número en coma flotante pre-procesado son los mismos que los como mucho J+1 MSBs de la mantisa del número de entrada, y

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

14. Dispositivo según la reivindicación 13, 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.

15. Dispositivo según la reivindicación 1, en el que, cuando el número de entrada es un número coma flotante no procesado con una mantisa de E+2 bits y el primer número en coma flotante pre-procesado tiene una mantisa con E+2-D bits, D<E+1, entonces el dispositivo comprende:

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

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

16. Dispositivo según la reivindicación 15, en el que la unidad de redondeo está configurada además para, selectivamente, poner a cero el segundo LSB de la mantisa del primer 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.

17. Dispositivo según la reivindicación 1, en el que, cuando el número de entrada es un número coma flotante no procesado con una mantisa de E+2 bits y el primer número en coma flotante pre-procesado tiene una mantisa con E+2+G bits, entonces el dispositivo 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 primer número en coma flotante pre-procesado, fijando los como mucho E+2 MSBs del primer 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 primer número en coma flotante pre-procesado es igual a uno, y

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

18. Dispositivo según la reivindicación 17, 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.

19. Dispositivo según cualquiera de las reivindicaciones 1 a 18, en el que, cuando el segundo número en coma flotante pre-procesado tiene una mantisa con F+2 bits y el número de salida es un número coma fija pre-procesado de L bits, donde L<F+4 entonces el dispositivo comprende:

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

un módulo de desplazamiento con una primera entrada para recibir como mucho los L-1 MSBs de la mantisa del segundo 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 segundo número en coma flotante, para generar como mucho los L-1 MSBs del número en coma fija pre-procesado en una salida.

20. Dispositivo según la reivindicación 19, en el que el módulo de desplazamiento comprende un desplazador aritmético a la derecha acoplado a un inversor de bit condicional.

21. Dispositivo según cualquiera de las reivindicaciones 1 a 18, en el que, cuando el segundo número en coma flotante pre-procesado tiene una mantisa de F+2 bits y el número de salida es un número coma fija pre-procesado de F+C+3 bits, C>0, el dispositivo comprende:

un calculador de cantidad de desplazamiento que recibe el exponente del segundo número 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 segundo 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.

22. Dispositivo según la reivindicación 21, 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.

23. Dispositivo según las reivindicaciones 21 o 22, 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.

24. Dispositivo según cualquiera de las reivindicaciones 1 a 18, en el que cuando el segundo número coma flotante pre-procesado tiene una mantisa de Z bits y el número coma fija no procesado tiene H+1 bits, el dispositivo comprende un dispositivo para convertir números coma flotante pre-procesados a números coma fija pre-procesados, de acuerdo con cualquiera de las reivindicaciones 19 a 23, teniendo una salida de H+2 bits conectada a un módulo de redondeo.

25. Dispositivo según reivindicación 24, 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 dispositivo para convertir 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.

26. Dispositivo según cualquiera de las reivindicaciones 1 a 18, en el que, cuando el segundo número en coma flotante pre-procesado tiene una mantisa de U+2 bits y el número de salida es un número coma flotante no procesado con una mantisa de U-V+2 bits, V>0, el dispositivo comprende:

un módulo de redondeo, configurado para recibir como mucho los U+3-V MSBs de la mantisa del segundo 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.

27. Dispositivo según la reivindicación 26, 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 segundo 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.

28. Dispositivo según las reivindicaciones 26 ó 27, 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.

29. Dispositivo según cualquiera de las reivindicaciones 1 a 18, en el que, cuando el segundo número en coma flotante pre-procesado tiene una mantisa de U+2 bits y el número de salida es un número coma flotante no procesado con una mantisa de U+2+W bits, entonces el dispositivo comprende:

un módulo de rellenado, configurado para recibir como mucho los U+1 MSBs de la mantisa del segundo 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.

30. Procedimiento, en un sistema de procesado de datos comprendiendo un procesador, para representar un número real en un formato pre-procesado, dicho formato pre-procesado siendo la representación del número real mediante el número más cercano en coma fija de M+1 dígitos teniendo el LSD igual a B/2, B siendo la base del sistema numérico, el método comprende:

calcular, por el procesador, el mayor número en base B de M dígitos menor que el número real;

fijar, por el procesador, los M MSDs del número en coma fija de M+1 dígito igual a los M dígitos de dicho mayor número de M dígitos.

31. Procedimiento, según la reivindicación 30, que comprende además fijar, por el procesador, el LSD del número en coma fija de M+1 dígito igual a B/2.

32. Procedimiento,, según la reivindicación 30, en el cual, cuando el número real es un número coma fija ya representado en un formato en base B y comprende N+1 dígitos, N>M, el método comprende:

fijar, por el procesador, los N-M-1 MSDs de los N-M LSDs del número en coma fija de N+1 dígitos a cero;

fijar, por el procesador, el (M+1)-ésimo MSD a B/2.

33. Procedimiento, según la reivindicación 32 que comprende además:

fijar, por el procesador, el LSD del número en coma fija de N+1 dígitos a cero;

detectar, por el procesador, si dichos N-M-1 MSDs y dicho LSB eran igual a cero antes de dicha operación de fijar;

fijar, por el procesador, el M-ésimo MSD a cero en respuesta a dicha identificación.

34. Procedimiento, según cualquiera de las reivindicaciones 30 a 33, en el que el número real es la mantisa de un  número en coma flotante.

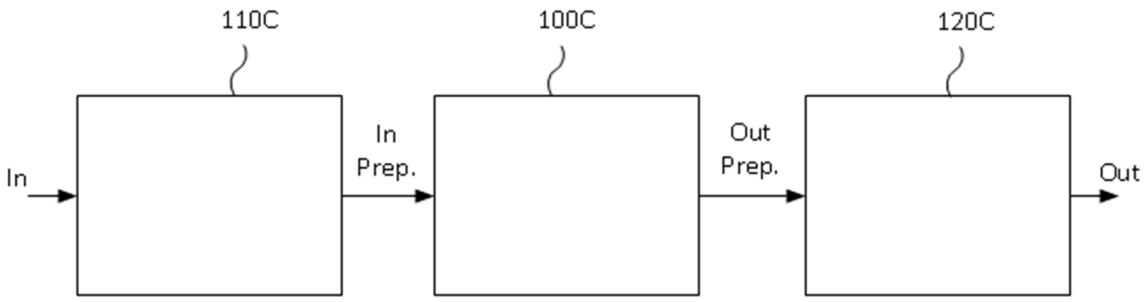



**Fig. 1**

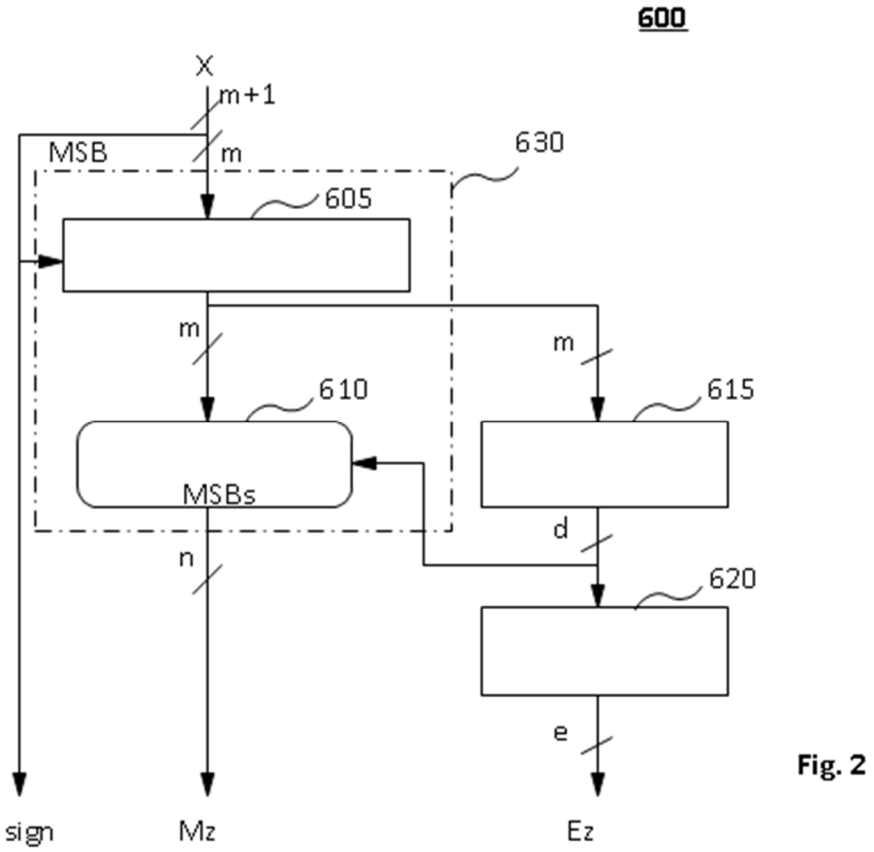



Fig. 2

Fig. 2a

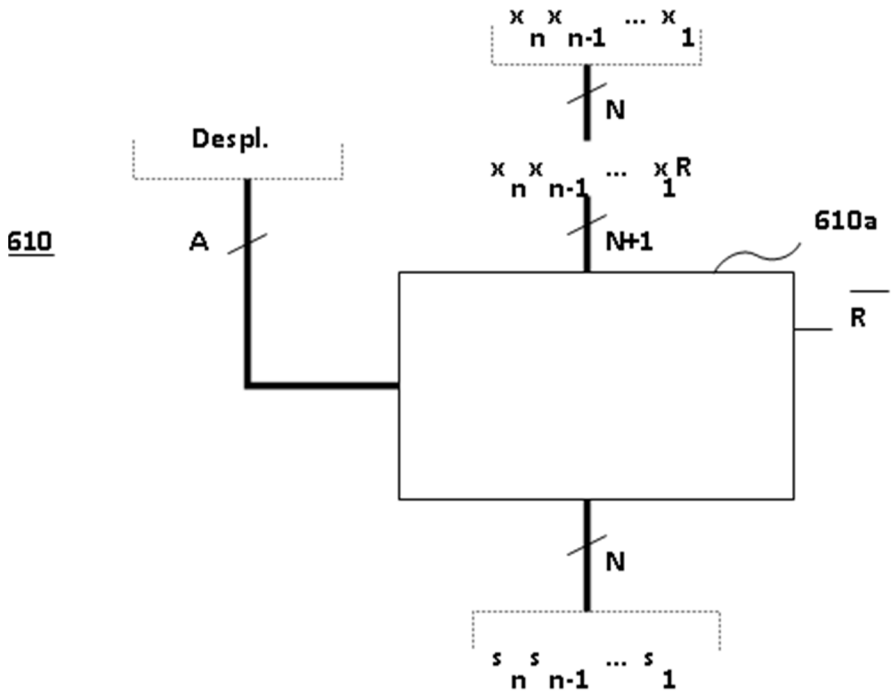


Fig. 2b

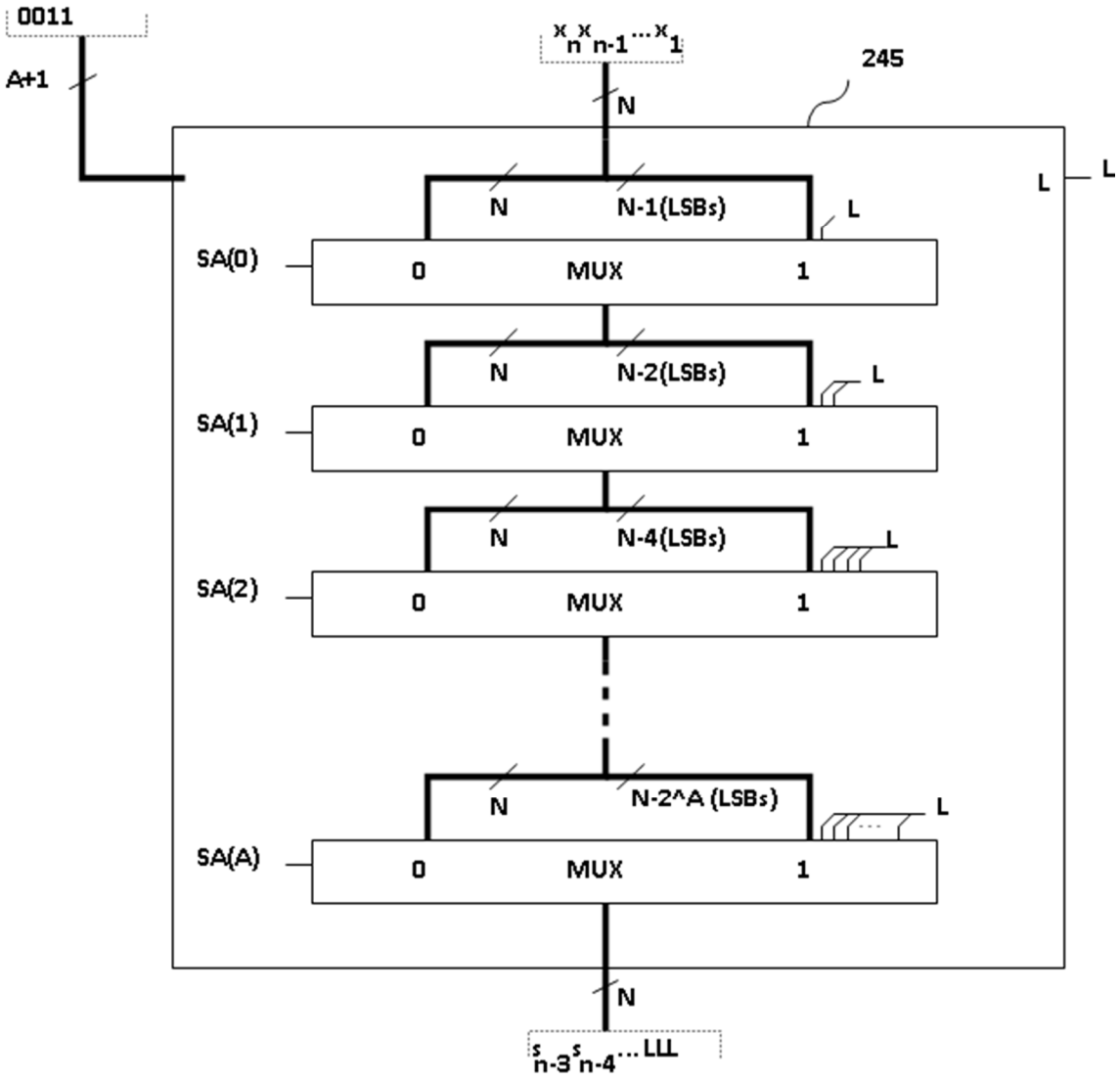

0011
A+1
$x_n x_{n-1} \ldots x_1$
N
245
L
L
N
N-1(LSBs)
L
SA(0)
0
MUX
1
N
N-2(LSBs)
L
SA(1)
0
MUX
1
N
N-4(LSBs)
L
SA(2)
0
MUX
1
N
N-2^A (LSBs)
L
SA(A)
0
MUX
1
N
$s_{n-3} s_{n-4} \ldots LLL$

Fig. 3

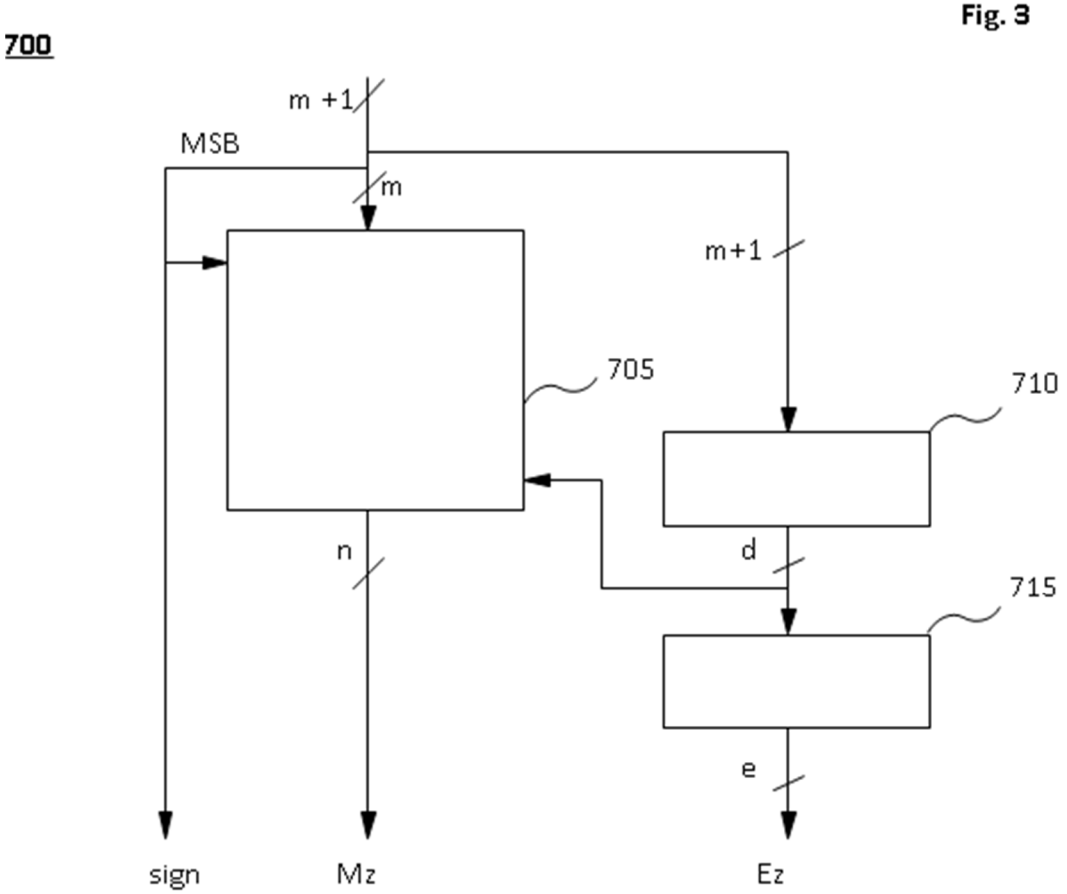


Fig. 3a

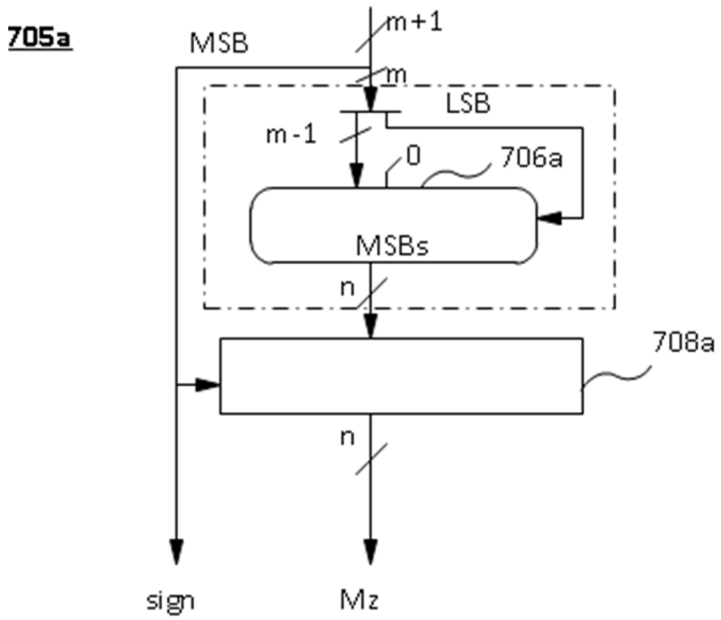
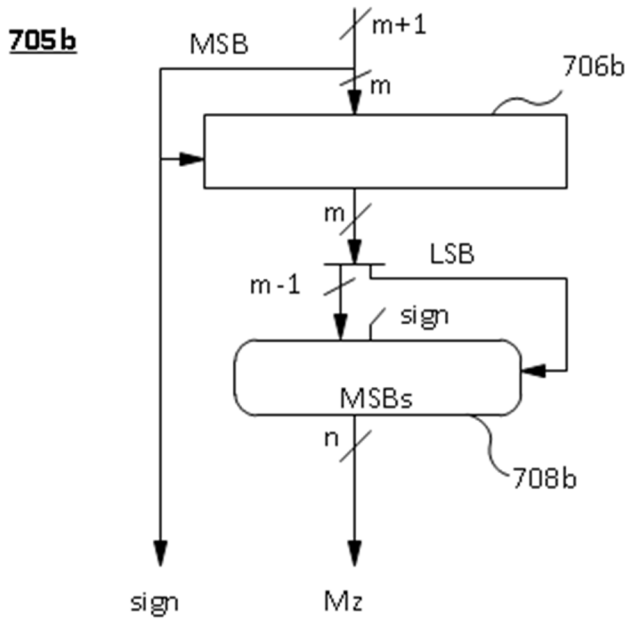

705a
MSB
m+1
m
LSB
m-1
0
706a
MSBs
n
708a
n
sign
Mz

Fig. 3b


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

800a

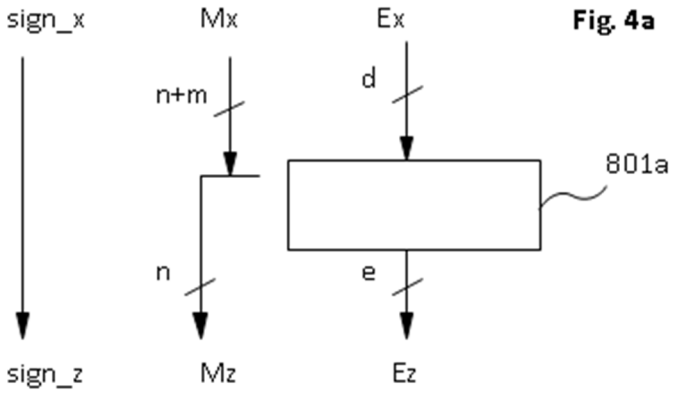



Fig. 4a

800b

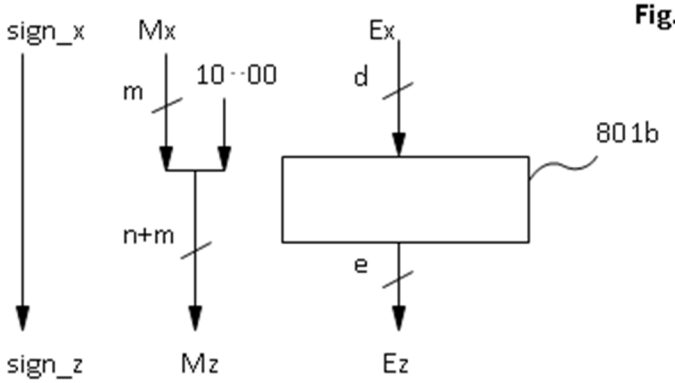



Fig. 4b

800c

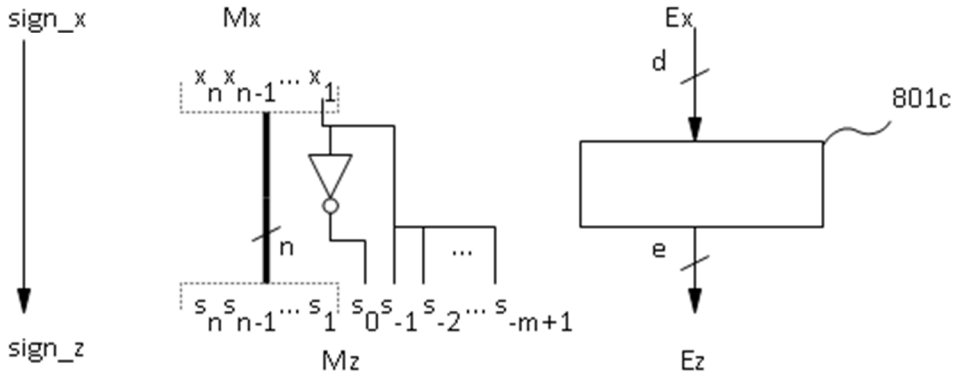



Fig. 4c

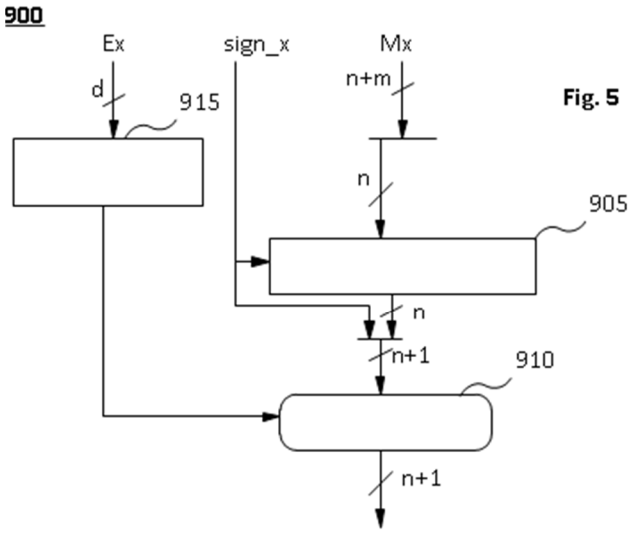



Fig. 5

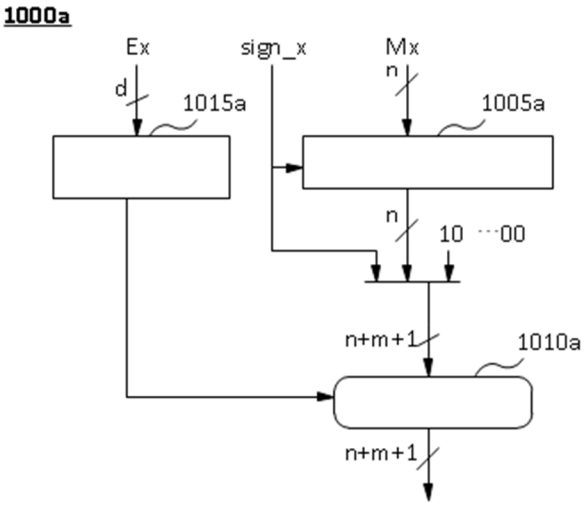



Fig. 6a

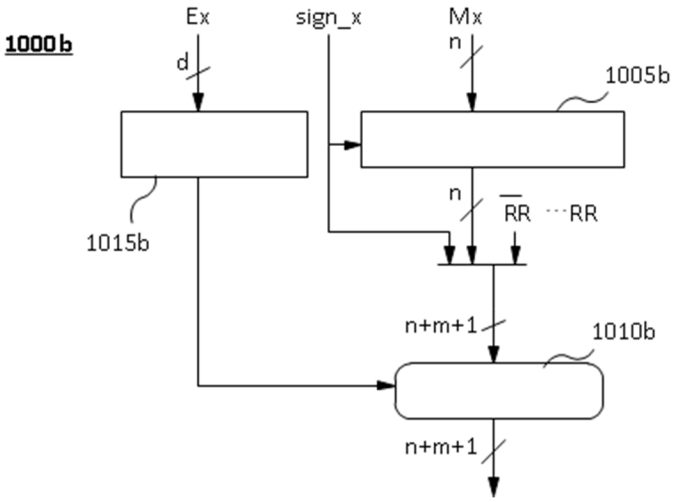



Fig. 6b

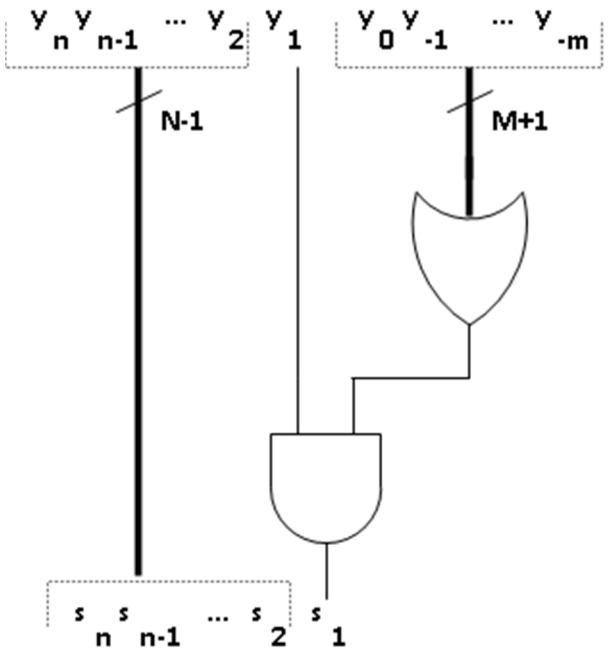



Fig. 7

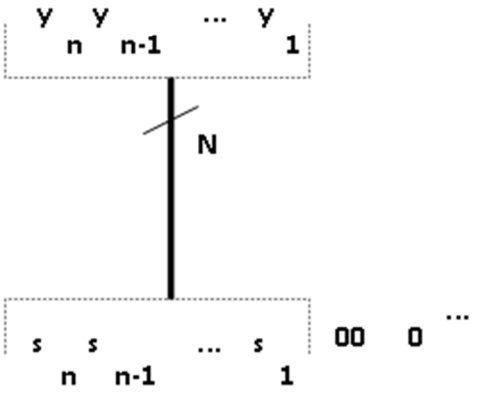



Fig. 8a

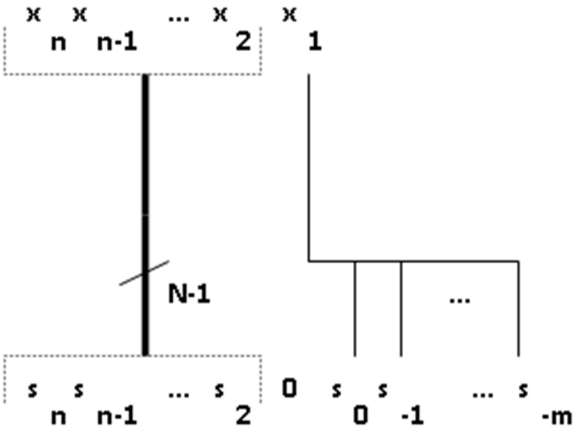



Fig. 8b

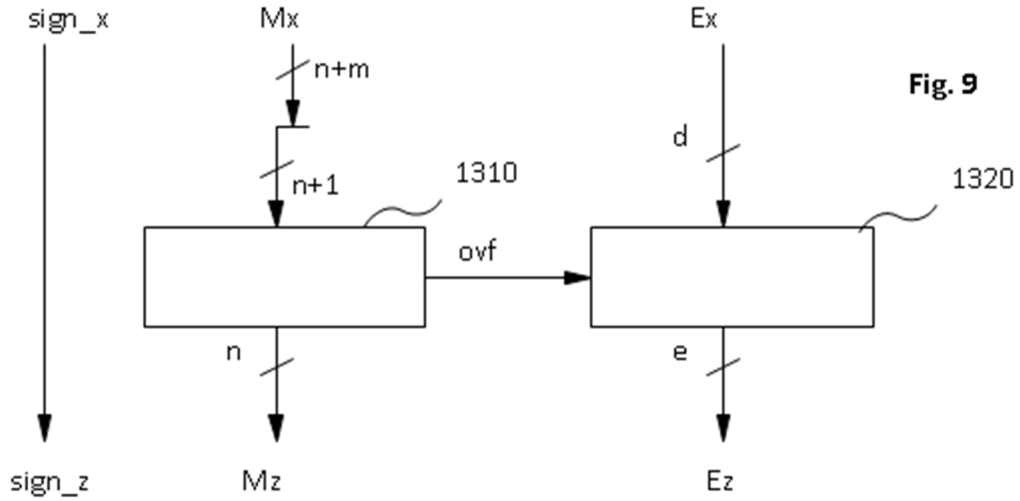

sign_x
Mx
n+m
n+1
1310
Ex
d
1320
ovf
n
e
sign_z
Mz
Ez

Fig. 9

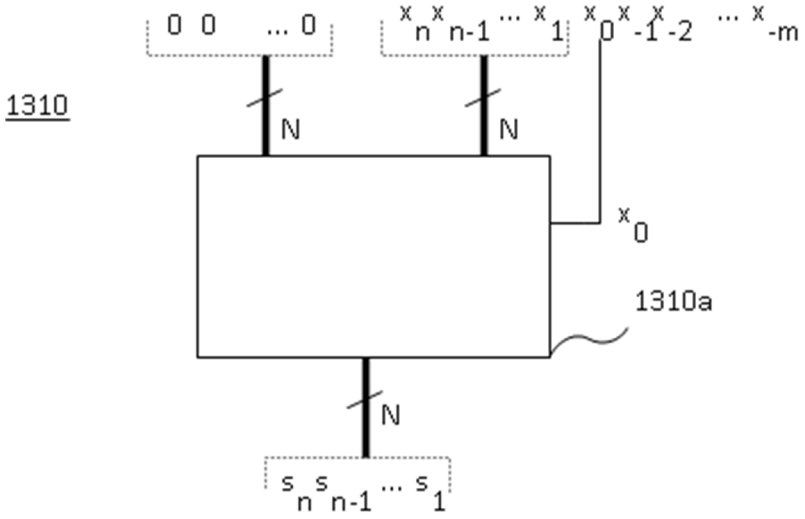
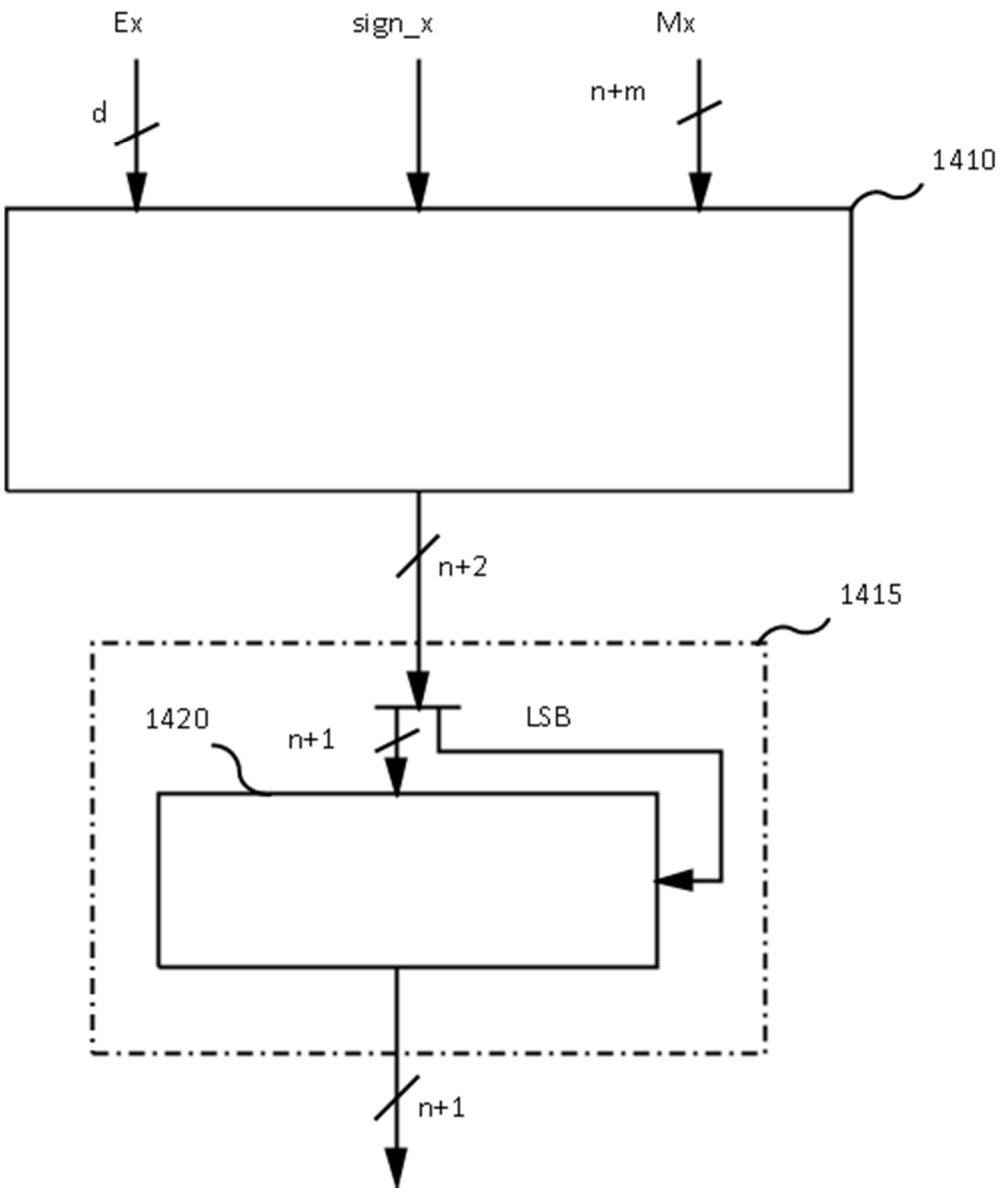

1310
0 0 ... 0
x n x n-1 ... x 1
x 0 x -1 x -2 ... x -m
N
N
x 0
1310a
N
s n s n-1 ... s 1

Fig. 9a

Fig. 10



OFICINA ESPAÑOLA
DE PATENTES Y MARCAS

ESPAÑA

(21) N.º solicitud: 201430455

(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 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. | 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 |
|---|---|---|
| 05.02.2015 | 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, XPI3E, XPIEE, NPL

Fecha de Realización de la Opinión Escrita: 05.02.2015

**Declaración**

| | | | |
|---|---|---|---|
| **Novedad (Art. 6.1 LP 11/1986)** | Reivindicaciones | 1-34 | **SI** |
| | Reivindicaciones | | **NO** |
| **Actividad inventiva (Art. 8.1 LP11/1986)** | Reivindicaciones | 1-34 | **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 EEE 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: 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.

Reivindicación 30: el procedimiento reivindicado se corresponde directamente con la estructura y funciones del dispositivo de la reivindicación 1 y, por tanto, resulta también válido el análisis hecho de dicha reivindicación. Así, la reivindicación 30 posee actividad inventiva según el art. 8.1 de la Ley de Patentes.

## Reivindicaciones dependientes

Reivindicaciones 2-29 y 31-34: estas reivindicaciones poseen actividad inventiva según el art. 8.1 de la Ley de Patentes porque dependen de las reivindicaciones 1 y 30 respectivamente que, como se ha mencionado, también la tienen.