



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

⑪ Número de publicación: **2 283 236**

⑫ Número de solicitud: 200700983

⑬ Int. Cl.:
G06F 17/14 (2006.01)

⑭

PATENTE DE INVENCION CON EXAMEN PREVIO

B2

⑮ Fecha de presentación: **12.04.2007**

⑯ Fecha de publicación de la solicitud: **16.10.2007**

Fecha de la concesión: **26.02.2008**

⑰ Fecha de anuncio de la concesión: **16.03.2008**

⑱ Fecha de publicación del folleto de la patente:
16.03.2008

⑲ Titular/es: **Universidad Politécnica de Madrid
c/ Ramiro de Maeztu, 7
28040 Madrid, ES**

⑳ Inventor/es: **Garrido Gálvez, Mario y
Grajal de la Fuente, Jesús**

㉑ Agente: **No consta**

㉒ Título: **Procedimiento y arquitectura sin memoria para el cálculo de las rotaciones de la FFT.**

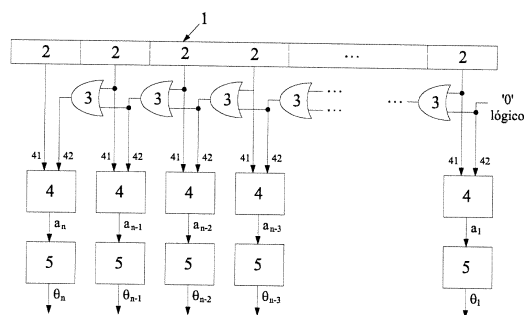
㉓ Resumen:

Procedimiento y arquitectura sin memoria para el cálculo de las rotaciones de la FFT.

Procedimiento y arquitectura que permiten calcular las rotaciones de cualquier FFT descompuesta según el algoritmo Cooley-Tukey, sin tener que recurrir a ningún dato previamente almacenado. Todos los ángulos de rotación que requieren las etapas de la FFT son generados a partir de un único contador para toda la FFT, y utilizando un sencillo circuito compuesto por sumadores y puertas lógicas. De esta forma se elimina la necesidad de almacenar datos relacionados con los ángulos de rotación, consiguiendo así una importante reducción en el área de la FFT, especialmente para FFTs de muchos puntos.

Además, las rotaciones se calculan empleando una modificación del algoritmo CORDIC que permite simplificar los bloques de cálculo de las microrrotaciones.

Finalmente, se presenta un sistema que utiliza únicamente dos restadores para compensar el escalado propio del algoritmo CORDIC.



Aviso: Se puede realizar consulta prevista por el art. 37.3.8 LP.

DESCRIPCIÓN

Procedimiento y arquitectura sin memoria para el cálculo de las rotaciones de la FFT.

5 Sector técnico

La invención que se describe se enmarca dentro del campo del procesamiento de señal, más en concreto en lo referente a la elaboración de algoritmos y el desarrollo de arquitecturas de circuito eficientes para llevar a cabo dichos algoritmos.

10

Estado de la técnica

15

La transformada de Fourier es una de las operaciones fundamentales en el campo del procesamiento de señal, especialmente en análisis espectral. Para su aplicación en sistemas digitales se hace uso de la DFT (*Discrete Fourier Transform*), que permite realizar la transformada de Fourier sobre datos almacenados en un ordenador o muestreados mediante un conversor analógico-digital, y cuya fórmula es:

20

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn}, \quad k = 0, 1, \dots, N-1$$

25

En la ecuación, el valor N indica el número de puntos sobre los que se calcula la DFT, $x[n]$ son las muestras en el dominio del tiempo, k es una variable discreta que representa la frecuencia, y $X[k]$ es la señal en el dominio de la frecuencia, que se define para los valores de k desde 0 a $N-1$.

30

Con el objetivo de calcular la DFT de forma eficiente, el nombre FFT (*Fast Fourier Transform*) engloba un conjunto de algoritmos que reducen el número de operaciones requerido por la DFT. De todos ellos, el algoritmo de Cooley-Tukey [CT65] es el más utilizado. Éste se basa en descomponer la DFT en $n = \log_r N$ etapas de cálculo en cascada, donde r es el *radix* empleado en la descomposición. Por otra parte, la descomposición se puede realizar siguiendo distintos métodos. Los más comunes son el diezmado en tiempo (DIT) y el diezmado en frecuencia (DIF). De esta forma se consigue una reducción en el número de operaciones, que pasa de un orden $O(N^2)$ en la DFT, a un orden $O(N \log_r N)$ en la FFT.

35

Cada una de las n etapas de la FFT se caracteriza por tener que calcular un conjunto de sumas, restas y rotaciones en el plano complejo. Las sumas y restas se realizan mediante elementos denominados mariposas, y las rotaciones complejas mediante rotadores.

40

Cada mariposa recibe r datos de entrada y ofrece r salidas que representan la FFT de r puntos de las entradas. Como la FFT de 2 puntos y la de 4 puntos se pueden realizar mediante rotaciones triviales (0° , 90° , 180° y 270°), lo más habitual es utilizar FFTs de *radix* 2 ó 4, ya que en otros casos es necesario incluir rotadores dentro de las mariposas para llevar a cabo las rotaciones no triviales que aparecen.

45

En cada etapa s de la FFT, $s \in \{1 \dots n\}$, se calculan N/r mariposas y N rotaciones. Sin embargo, en las arquitecturas de los circuitos de cálculo de la FFT, todas las mariposas se suelen calcular con un único circuito, aprovechando que los datos llegan de formas secuencial. Lo mismo ocurre con las rotaciones: un único rotador es suficiente para llevar a cabo todas las rotaciones de una etapa. De esta forma, se consigue un ahorro importante de componentes. Sin embargo, en el caso de las rotaciones es necesario conocer el ángulo de rotación (o algún dato relacionado con él) que hay que rotar cada una de las muestras que van llegando al rotador.

50

Existen dos métodos principales para la realización de las rotaciones de la FFT. El primero consiste en rotar el dato de entrada aplicando directamente la fórmula de una rotación, mediante el empleo de multiplicadores reales y sumadores. La otra opción, más utilizada por requerir en general menos recursos, es emplear el algoritmo CORDIC, propuesto por J. E. Volder [Vol59], y que se describe brevemente a continuación.

55

El algoritmo CORDIC se utiliza para calcular de forma eficiente operaciones matemáticas complicadas en sistemas digitales. Este algoritmo se basa en transformar estas operaciones en un conjunto de sumas y desplazamientos, que son fáciles de llevar a cabo en los circuitos digitales.

60

Si se utiliza para realizar rotaciones en el plano complejo, el algoritmo CORDIC descompone el ángulo de rotación, θ , en una suma de ángulos, α_i :

65

$$\theta = \sum_{i=m}^M \alpha_i + \epsilon$$

donde m y M son respectivamente los índices del primer y último ángulo considerado, ϵ es el error de la aproximación y

$$\alpha_i = \pm tg^{-1}(2^{-i})$$

Habitualmente $m = 0$ y, por lo tanto, el primer ángulo es $\alpha_0 = tg^{-1}(1) = 45^\circ$, lo que hace que el algoritmo sea válido para cualquier valor de θ en el intervalo $[-90^\circ, 90^\circ]$. Cuando θ se encuentra fuera de este intervalo se utilizan las rotaciones triviales de 180° y 90° para situarlo en él. De hecho, mediante estas rotaciones es posible dejar el ángulo restante de rotación, z , en el intervalo $[-45^\circ, 45^\circ]$. De acuerdo con esta idea, es posible prescindir de la rotación de 45° y considerar que la primera rotación del algoritmo CORDIC es para el caso $m = 1$, siendo $\alpha_1 = tg^{-1}(2^{-1}) \approx 26,5^\circ$.

La rotación del dato de entrada al rotador se lleva a cabo mediante una serie de etapas llamadas microrrotaciones, en cada una de las cuales se rota un ángulo α_i de acuerdo con las ecuaciones:

$$x_{i+1} = x_i - y_i \delta_i 2^{-i}$$

$$y_{i+1} = y_i + x_i \delta_i 2^{-i}$$

donde δ_i indica el sentido de la microrrotación y se calcula de acuerdo con:

$$\delta_i = \text{sign}(z_i)$$

$$z_{i+1} = z_i - \delta_i \cdot \alpha_i$$

siendo $\text{sign}(\eta) = 1$ si $\eta \geq 0$, y $\text{sign}(\eta) = -1$ si $\eta < 0$. Como $\delta_i \in \{-1, 1\}$, se realiza una rotación en cada una de las etapas, bien en sentido positivo o bien en sentido negativo. Ello da lugar a una ganancia constante en el rotador, que puede ser compensada, en caso de que sea necesario, simplemente multiplicando las muestras de salida por:

$$K = \prod_{i=m}^M \cos(\alpha_i) = \prod_{i=m}^M \cos(tg^{-1}(2^{-i}))$$

Para poder rotar adecuadamente cada uno de los datos de entrada al rotador se almacenan en una memoria los valores de los vectores de rotaciones; δ , correspondientes a cada una de las rotaciones que debe realizar el rotador. Estos datos son leídos de la memoria secuencialmente de tal forma que cada dato de entrada al rotador sea rotado el ángulo de rotación correspondiente.

Cuando el número de rotaciones que realiza el rotador de una cierta etapa de la FFT es elevado, la memoria de rotaciones deberá almacenar una gran cantidad de datos, lo que repercute negativamente en el área y en la velocidad de la FFT. Este problema es análogo en el caso de emplear multiplicadores para calcular las rotaciones, puesto que deben ser almacenados los valores del seno y el coseno de los ángulos de rotación.

Ciertas investigaciones han conseguido reducir la memoria de rotaciones a $\log_2 N$ [YCCO6], ó $0,5N$ [CP03]. Además, se han utilizado otros métodos de descomposición de la FFT (distintos al algoritmo de Cooley-Tukey) para reducir el número de rotaciones que hay que calcular, como ocurre en [ZH05], donde se emplea el método de descomposición en factores primos, el cual sólo es aplicable cuando el número de puntos de la FFT se puede representar como producto de factores primos entre sí.

Por el contrario, con el procedimiento y la arquitectura propuestos se consigue reemplazar las memorias de rotaciones de todas las etapas de la FFT por un sencillo circuito compuesto por un contador (único para toda la FFT), y unos pocos sumadores y puertas lógicas. Así, a partir del valor del contador se calculan las rotaciones de todas las etapas de la FFT aprovechando características comunes entre ellas. De esta forma, se consigue mejorar significativamente las prestaciones de la FFT, principalmente en cuanto a área. Además, la mejora es más significativa cuanto mayor sea el número de puntos de la FFT que se desea calcular, lo que hace que este procedimiento sea muy adecuado para el cálculo de FFTs de un elevado número de puntos.

Por otra parte, las rotaciones se calculan empleando una modificación del algoritmo CORDIC que permite simplificar los bloques de cálculo de las microrrotaciones respecto a otras opciones existentes [MGBS02, Hu92].

Descripción detallada de la invención

Esta invención presenta un procedimiento para el cálculo de las rotaciones de cualquier FFT descompuesta según el algoritmo Cooley-Tukey, y cuyo número de puntos, N , y *radix*, r , son ambos potencia de 2, con los siguientes pasos:

- obtener las secuencias de ángulos de rotación de todas las etapas de la FFT,
- calcular las rotaciones correspondientes a cada una de las etapas de la FFT a partir de los ángulos de rotación generados,

donde a partir de un único contador se obtienen las secuencias de ángulos de rotación de todas las etapas de la FFT. Este procedimiento tiene la ventaja de que no es necesario recurrir a ningún dato previamente almacenado.

Para implementar el procedimiento se utiliza una arquitectura de circuito que comprende:

- un módulo de generación de ángulos que incluye un único contador (1) para toda la FFT, a partir del cual se obtienen los ángulos de rotación de todas las etapas de la FFT sin necesidad de recurrir a ningún dato almacenado previamente,
- un módulo de cálculo de las rotaciones para cada una de las etapas de la FFT. Como se verá posteriormente, las rotaciones de alguna de las etapas de la FFT pueden resultar triviales, en cuyo caso se puede prescindir del módulo de cálculo de las rotaciones de dicha etapa.

Como se explicó anteriormente, una FFT de N puntos puede ser dividida en $n = \log_r N$ etapas, donde r es el *radix* de la FFT. Asumiendo que tanto N como r son potencia de 2, la secuencia de ángulos (de longitud N) de cualquier etapa s de la FFT, $s \in \{1 \dots n\}$, está formada por N/r^s secuencias iguales de longitud $L = r^s$.

La secuencia de ángulos depende del radix elegido en la descomposición de la FFT. Así, para *radix* 2, la secuencia de longitud L será:

$$a_s = \underbrace{0, 0, 0, 0, \dots}_{2^s-1}, \underbrace{0, 1, 2, 3, \dots}_{2^s-1}$$

y para *radix* 4,

$$a_s = \underbrace{0, 0, 0, 0, \dots}_{4^s-1}, \underbrace{0, 1, 2, 3, \dots}_{4^s-1}, \underbrace{0, 2, 4, 6, \dots}_{4^s-1}, \underbrace{0, 3, 6, 9, \dots}_{4^s-1}$$

Generalizando estas expresiones para cualquier valor de *radix*, la secuencia de ángulos se puede generar concatenando r subsecuencias:

$$0, p, 2p, \dots, (r^{s-1} - 1)p$$

donde $p = 0, 1, 2, \dots, r - 1$.

Para normalizar la secuencia de todas las etapas de la FFT al mínimo ángulo de rotación, $\theta_{\min}(\text{rad}) = 2\pi/N$, los valores de las secuencias deben ser multiplicados por $q = N/r^s$ o, lo que es lo mismo, desplazados $\log_2(q)$ bits:

$$\theta_s = q \cdot a_s = \underbrace{0, 0, 0, 0, \dots}_{2^s-1}, \underbrace{0, q, 2q, 3q, \dots}_{2^s-1}$$

La secuencia θ_s contiene los ángulos de rotación, θ , de la etapa s de la FFT, y cada ángulo de rotación $\theta \in [0, N - 1]$, siendo su equivalencia en radianes:

$$\theta(\text{rad}) = -\frac{2\pi}{N}\theta$$

Desde el punto de vista digital, esta representación de θ como un valor entre 0 y $N - 1$ resulta ventajosa por dos motivos. En primer lugar, el ángulo representado es exacto y el número de bits utilizado es mínimo; si el ángulo de rotación se representara en radianes siempre se cometería un error en la aproximación debido a que el número de bits

ES 2 283 236 B2

es finito. Por otra parte, el valor N equivale a una vuelta de circunferencia completa, por lo que, como N es potencia de 2, multiplicar por 2π radianes se transforma en un desplazamiento de bits.

5 Considerando ahora que se dispone de un contador, para una etapa cualquiera, s , de la FFT, la generación de la secuencia de ángulos de rotación de dicha etapa, θ_s , se realiza siguiendo los siguientes pasos:

- tomar los $s \cdot \log_2 r$ bits menos significativos proporcionados por el contador,
- 10 ■ de los bits tomados, multiplicar los $(s - 1) \cdot \log_2 r$ bits menos significativos por los $\log_2 r$ más significativos, obteniendo así los valores de la secuencia a_s . Los bits más significativos se corresponden con el valor de p mientras que los menos significativos contarán de 0 a $(r^{s-1} - 1)$.
- multiplicar el resultado por N/r_s , obteniendo así los valores de la secuencia θ_s .

15 El método explicado sirve para generar tanto los ángulos de la descomposición DIT (Diezmado en tiempo) como los de la descomposición DIF (Diezmado en frecuencia), considerando que $s = 1$ es la etapa de entrada de la FFT en el caso de la descomposición DIT, y la etapa de salida en el caso DIF.

20 Para implementar la generación de ángulos se emplea el módulo de generación de ángulos representado en la figura 1. El módulo de generación de ángulos comprende, además del contador (1), los siguientes elementos:

- lógica combinacional (3), que opera sobre el valor del contador,
- un bloque acumulador (4) por cada etapa de la FFT, para el cálculo de las secuencia a_s .

25 Por otra parte, como se puede observar en la figura 2, cada bloque acumulador (4) comprende los siguientes elementos:

- un sumador (43), que permite ir actualizando los valores de la secuencia a_s ,
- 30 ■ un registro (45) que proporciona los valores de la secuencia a_s ,
- una puerta lógica (44) para el control del bloque acumulador.

35 El contador (1) es un contador binario que cuenta de 0 a $N - 1$ de forma periódica. Éste se considera dividido en s partes (2) de $\log_2 r$ bits. Cada una de ellas contiene el valor p de una de las etapas, y se corresponde con la entrada (41) del bloque acumulador de dicha etapa. En concreto, el valor de p para una etapa s incluye los bits que van desde la posición $(s - 1) \cdot \log_2(r) + 1$ a la $s \cdot \log_2 r$ del contador, considerando que el bit menos significativo es el bit 1.

40 Según la estructura del bloque acumulador (4), el registro (45) se irá incrementando de acuerdo con el valor de p siempre que no se active la señal de control (42).

Por otra parte, los $(s - 1) \cdot \log_2 r$ bits menos significativos del contador (1) realizan una cuenta cíclica desde 0 hasta $(r^{s-1} - 1)$. Cuando la cuenta de estos bits vuelve a cero, se activa la señal de control (42) que hace que se ponga a cero el valor del registro (45), que va proporcionando los valores de la secuencia a_s .

50 Finalmente, los bloques de escalado (5) no incluyen ningún componente lógico. Ello se debe a que, al ser N y r potencias de 2, el factor de escalado $q = N/r^s$ representa únicamente un desplazamiento fijo de bits, para lo cual sólo hay que ajustar el interconexión de los bits con la etapa siguiente.

Así, la generación de las secuencias de ángulos de rotación θ_s de todas las etapas de la FFT se lleva a cabo mediante un sencillo circuito que no requiere ningún tipo de memoria ni la realización de multiplicaciones. Además, sólo es necesario el empleo de un único contador para toda la FFT, a partir de cual se generan todas las secuencias de ángulos de rotación.

55 El siguiente paso del procedimiento descrito es el cálculo de las rotaciones correspondientes a cada una de las etapas de la FFT a partir de los ángulos de rotación generados. La forma de proceder es la misma para cada una de las etapas de la FFT, y se compone de los siguientes pasos:

- determinar si se deben realizar las rotaciones de 180° y 90° ,
- calcular el vector de rotaciones, δ ,
- obtener el vector de rotaciones adaptado δ' ,
- 65 ■ rotar el dato de entrada al rotador.

ES 2 283 236 B2

En primer lugar, para cada ángulo de rotación θ perteneciente a la secuencia θ_s , se determina si se deben realizar las rotaciones de 180° y 90° con el fin de situar el ángulo restante de rotación, z , en el intervalo de definición del algoritmo CORDIC. Nótese que como $\theta(rad) = -2\pi\theta/N$, las rotaciones deberán realizarse en sentido negativo.

5 El hecho de representar θ como un valor entero, $\theta \in [0, N - 1]$, siendo N potencia de 2, presenta la ventaja de que en la arquitectura del circuito resulta muy sencillo determinar las rotaciones de 180° y 90° que permiten situar el ángulo restante de rotación z en el intervalo $[-45^\circ, 45^\circ]$. Para ello únicamente hay que sumar al conjunto de los dos bits más significativos de θ el siguiente bit más significativo. De esta forma, los dos bits resultantes de la operación indicarán respectivamente si se deben realizar las rotaciones de 180° y 90° . Además, z se obtiene directamente tomando todos los
10 bits de θ excepto los dos más significativos y considerando que el z está expresado en complemento a 2, y cambiado de signo respecto al ángulo en radianes.

A partir de z se determina el vector de rotaciones, δ , de acuerdo con:

$$15 \quad \delta_i = -sign(z_i)$$

$$z_{i+1} = z_i + \delta_i \cdot \alpha_i$$

20 siendo $sign(\eta) = 1$ si $\eta \geq 0$, y $sign(\eta) = -1$ si $\eta < 0$. Este cálculo se realiza a partir de $z_0 = z$ en el caso de emplear la microrrotación de $\alpha_0 = 45^\circ$ ($m = 0$); y a partir de $z_1 = z$ si se considera que la primera microrrotación es la de $\alpha_1 = tg^{-1}(2^{-1}) \approx 26,5^\circ$. El cambio de signo respecto al procedimiento habitual del CORDIC se debe únicamente a que, como se ha dicho anteriormente, la representación del ángulo en el intervalo $[0, N - 1]$ tiene signo contrario a la representación del ángulo en radianes.
25

Las rotaciones de 180° se calculan fácilmente cambiando de signo las componentes real e imaginaria de los datos de entrada, y la de 90° con un cambio de signo e intercambiando dichas componentes.

30 Habitualmente, a partir de los valores de δ , y una vez aplicadas las rotaciones de 180° y 90° sobre los datos de entrada al rotador, se procede a realizar las microrrotaciones de acuerdo con las ecuaciones:

$$x_{i+1} = x_i - y_i \delta_i 2^{-i}$$

$$35 \quad y_{i+1} = y_i + x_i \delta_i 2^{-i}$$

40 Sin embargo, es posible modificar este procedimiento para reducir la complejidad del módulo de cálculo de las rotaciones. Con esta idea, se propone un paso en el que se obtiene el vector de rotaciones adaptado, δ' , a partir del vector de rotaciones, δ , de la siguiente forma:

$$45 \quad \begin{aligned} \delta'_m &= \delta_m \\ \delta'_i &= -\delta_i \cdot \delta_{i-1}, \quad i = m + 1, \dots, M \\ \delta'_{M+1} &= \delta_M \end{aligned}$$

50 donde m es el índice correspondiente a la primera microrrotación y M el correspondiente a la última microrrotación.

Una vez obtenido el vector δ' , el cálculo de las microrrotaciones se realiza de la siguiente forma:

$$55 \quad a_{i+1} = \begin{cases} a_i + b_i 2^{-i} & \text{si } \delta'_i = -1 \\ b_i + a_i 2^{-i} & \text{si } \delta'_i = 1 \end{cases}$$

$$60 \quad b_{i+1} = \begin{cases} b_i - a_i 2^{-i} & \text{si } \delta'_i = -1 \\ a_i - b_i 2^{-i} & \text{si } \delta'_i = 1 \end{cases}$$

65

siendo a_i y b_i las entradas de la i -ésima etapa de microrrotación.

Las entradas a la primera etapa de microrrotación, a_m y b_m , valdrán:

$$a_m = x_{IN}$$

$$b_m = y_{IN}$$

siendo x_{IN} e y_{IN} respectivamente la parte real e imaginaria del dato sobre el que hay que realizar las microrrotaciones. Por otra parte, x_{OUT} e y_{OUT} , respectivamente la parte real e imaginaria del dato resultante de aplicar las microrrotaciones, valdrán:

$$x_{OUT} = \begin{cases} a_{M+1} & \text{si } \delta'_{M+1} = -1 \\ b_{M+1} & \text{si } \delta'_{M+1} = 1 \end{cases}$$

$$y_{OUT} = \begin{cases} a_{M+1} & \text{si } \delta'_{M+1} = -1 \\ b_{M+1} & \text{si } \delta'_{M+1} = 1 \end{cases}$$

El módulo de cálculo de las rotaciones de la FFT se representa en la figura 3 y se compone de los siguientes bloques:

- un generador del vector de rotaciones (6),
- un adaptador del vector de rotaciones (7),
- bloques de cálculo de las rotaciones de 180° y 90° (8),
- varios bloques de cálculo de las microrrotaciones (9).

El generador del vector de rotaciones (6) determina si hay que rotar 180° y/o 90° el dato de entrada al rotador y obtiene el vector de rotaciones, δ , todo ello a partir del ángulo de rotación, θ , que le llega del módulo de generación de ángulos.

A continuación, el adaptador del vector de rotaciones (7) calcula el vector de rotaciones adaptado, δ' , a partir del vector de rotaciones, δ , de acuerdo con el procedimiento descrito. Desde el punto de vista digital, los valores de δ_i y δ'_i , se representan con un bit, que puede tomar el valor “0” ó “1”, en lugar de los valores “-1” y “1” descritos en el procedimiento. Teniendo esto en cuenta, en la implementación de la arquitectura, el procedimiento se transforma en:

$$\delta'_m = \delta_m$$

$$\delta'_i = \delta_i \text{ XOR } \delta_{i-1}, \quad i = m + 1, \dots, M$$

$$\delta'_{M+1} = \delta_M$$

Los bloques de cálculo de las rotaciones de 180° y/o 90° (8) rotan el dato de entrada esos ángulos (en sentido negativo) en caso de que sea necesario, de acuerdo con el valor de θ . La rotación de 180° se puede realizar cambiando de signo las componentes real e imaginaria del dato y la de 90° con un cambio de signo e intercambiando dichas componentes.

En la figura 4 se presenta el esquema del bloque de cálculo de las microrrotaciones (9). Las entradas (91) y (92) se corresponden respectivamente con los valores de a_i y b_i descritos en el procedimiento, mientras que el conmutador (93) está controlado por la señal δ'_i (94). Después del conmutador, los datos son desplazados i posiciones (95). Como el valor es fijo para cada etapa de microrrotación, las señales están cableadas, lo que no constituye ningún elemento físico. Finalmente, el circuito se compone de un sumador (96) y de un restador (97).

Así, se consigue simplificar los bloques de cálculo de las microrrotaciones (9) siendo únicamente necesario un conmutador (93), un sumador (96), y un restador (97) en cada uno de ellos.

ES 2 283 236 B2

Como se dijo anteriormente, el dato rotado sale del circuito escalado por un factor constante y, por lo tanto, puede ser compensado. En el caso de $m = 1$ el factor de compensación será:

$$K = \prod_{i=1}^M \cos(\operatorname{tg}^{-1}(2^{-i})) \approx 0,8588$$

considerando que $\cos(\operatorname{tg}^{-1}(2^{-i})) \approx 1, \forall i > M$, que es una buena aproximación para $M \geq 7$. Así, teniendo en cuenta que:

$$K = 0.8588 \approx 0,8594 = 1 - 2^{-3} - 2^{-6}$$

la compensación del escalado se puede realizar exclusivamente empleando dos restadores.

Por lo tanto, en el caso de que el primer bloque de cálculo de las microrrotaciones sea el correspondiente a $\alpha_1 = \operatorname{tg}^{-1}(2^{-1})$, se puede añadir un módulo adicional de compensación del escalado que consiste en dos restadores.

Breve descripción de los dibujos

Figura 1: Esquema del módulo de generación de ángulos, que incluye un contador (1), lógica combinacional (3), bloques acumuladores (4) y bloques de escalado (5).

Figura 2: Esquema del bloque acumulador (4), que se compone de un sumador (43), un registro (45) y una puerta lógica (44).

Figura 3: Esquema del módulo de cálculo de las rotaciones, que incluye un generador del vector de rotaciones (6), un adaptador del vector de rotaciones (7), bloques de cálculo de las rotaciones de 180° y 90° (8) y bloques de cálculo de las microrrotaciones (9).

Figura 4: Esquema del bloque de cálculo de las microrrotaciones (9), que se compone de un conmutador (93), desplazadores (95), un sumador (96) y un restador (97).

Exposición de un modo de realización de la invención

A continuación se explica la invención para el caso en el que se calculen las rotaciones de una FFT de 8 puntos y *radix* 2. Como $N = 8$, el contador contará cíclicamente de 0 a 7, y se incrementará en cada ciclo de reloj. El número de etapas de la FFT considerada será $n = \log_2 N = 3$ y, por lo tanto, $s \in \{1, 2, 3\}$.

La siguiente tabla muestra la forma de obtener las secuencias de rotación θ_s a partir de los valores que va tomando el contador.

Contador		s =1			s =2			s =3		
Dec.	Bin.	p_1	a_1	θ_1	p_2	a_2	θ_2	p_3	a_3	θ_3
0	000	0	0	0	0	0	0	0	0	0
1	001	1	0	0	0	0	0	0	0	0
2	010	0	0	0	1	0	0	0	0	0
3	011	1	0	0	1	1	2	0	0	0
4	100	0	0	0	0	0	0	1	0	0
5	101	1	0	0	0	0	0	1	1	1
6	110	0	0	0	1	0	0	1	2	2
7	111	1	0	0	1	1	2	1	3	3

ES 2 283 236 B2

Las dos primeras columnas de la tabla indican el valor del contador en decimal y en binario respectivamente. Como estamos trabajando con *radix* 2, el valor de p_1 se corresponde con el del bit menos significativo del contador, el de p_2 con el siguiente, y el de p_3 con el más significativo.

Para obtener la secuencia a_3 se toma el valor de los dos bits menos significativos del contador y se multiplica por p_3 ; para obtener a_2 se multiplica el bit menos significativo del contador por p_2 ; y a_1 vale siempre cero, como debe ocurrir en todas las FFTs para $s=1$.

Por otra parte, $\theta_s = q_s \cdot a_s$, siendo $q_s = N/r^s$, por lo que $q_1 = 4$, $q_2 = 2$ y $q_3 = 1$.

A partir de la tabla también se puede entender el funcionamiento de la arquitectura de circuito que lleva a cabo el procedimiento. Así, por ejemplo, la secuencia a_3 empieza valiendo "0" y se incrementa con el valor de p_3 . Cuando el contador llega a 4, como los dos bits menos significativos valen cero, la secuencia se resetea y a continuación se vuelve a incrementar con el valor de p_3 hasta que el contador vuelve a 0, donde se resetea de nuevo la secuencia.

Para entender el sistema de rotaciones tomaremos el caso en el que $\theta = 3 = B' 011$, que se corresponde con un ángulo en radianes:

$$\theta(rad) = -\frac{2\pi}{N}\theta = -\frac{3\pi}{4}$$

Sumando a los dos bits más significativos de θ , "01", el siguiente bit, "1", obtenemos el valor "10", por lo que habrá que realizar una rotación de 180° , que situará el ángulo restante en $z = \pi/4$.

A partir de z se obtiene el vector de rotaciones que, en el caso de que $m = 1$ y $M = 8$ valdrá:

$$\delta = 1, 1, 1, -1, 1, -1, -1, 1$$

A continuación se calcula el vector de rotaciones adaptado de acuerdo con el procedimiento descrito:

$$\delta'_m = \delta_m$$

$$\delta'_i = -\delta_i \cdot \delta_{i-1}, \quad i = m + 1, \dots, M$$

$$\delta'_{M+1} = \delta_M$$

dando como resultado:

$$\delta' = 1, -1, -1, 1, 1, 1, -1, 1$$

Considerando ahora que el dato de entrada al rotador es el número complejo $10+3j$, se aplicará primero la rotación de 180° , con lo que las entradas a la primera etapa de microrrotación (para $m = 1$) serán:

$$a_1 = -10$$

$$b_1 = -3$$

Aplicando el procedimiento de cálculo de las microrrotaciones:

$$a_{i+1} = \begin{cases} a_i + b_i 2^{-i} & \text{si } \delta'_i = -1 \\ b_i + a_i 2^{-i} & \text{si } \delta'_i = 1 \end{cases}$$

$$b_{i+1} = \begin{cases} b_i - a_i 2^{-i} & \text{si } \delta'_i = -1 \\ a_i - b_i 2^{-i} & \text{si } \delta'_i = 1 \end{cases}$$

$$x_{OUT} = \begin{cases} a_{M+1} & \text{si } \delta'_{M+1} = -1 \\ b_{M+1} & \text{si } \delta'_{M+1} = 1 \end{cases}$$

$$y_{OUT} = \begin{cases} a_{M+1} & \text{si } \delta'_{M+1} = -1 \\ b_{M+1} & \text{si } \delta'_{M+1} = 1 \end{cases}$$

siendo δ' el vector de rotaciones adaptado que se ha calculado, y $M = 8$, se obtienen los valores de salida:

$$x_{OUT} = -5,80$$

$$y_{OUT} = -10,69$$

Se puede comprobar fácilmente que el ángulo que se ha rotado el dato de entrada es:

$$\theta(rad) = -2,3593 \approx -\frac{3\pi}{4}$$

Por otra parte, el módulo del dato de salida está escalado respecto al módulo del dato de entrada, y es posible compensarlo multiplicando la salida por $K = 1 - 2^{-3} - 2^{-6}$. También es posible realizar la compensación del escalado a la entrada de la FFT o en cualquier punto intermedio de la FFT.

Para realizar la compensación se tomará el dato que hay que multiplicar por K , el mismo dato desplazado 3 bits y finalmente el dato desplazado 6 bits. A continuación se restarán del dato sin desplazar los valores desplazados. Así, como los desplazamientos de bits son fijos, únicamente hacen falta dos restadores para llevar a cabo la compensación del escalado.

Aplicación industrial

Debido a que la FFT es un algoritmo ampliamente utilizado en las tecnologías de la información y las comunicaciones, la invención presentada puede tener aplicación en numerosos sistemas de comunicaciones y de procesamiento de señal. En especial, la arquitectura propuesta resulta útil cuando se pretende construir FFTs en plataformas *hardware*, como pueden ser las FPGAs (*Field Programmable Gate Array*). Por otra parte, la invención aporta una gran ventaja respecto a otros diseños en aplicaciones donde es necesario calcular FFTs de un elevado número de puntos.

Referencias

[CP03] Yun-Nan Chang and Keshab K. Parhi. An efficient pipelined FFT architecture. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 50:322-325, Jun 2003.

[CT65] J. W. Cooley and J. W. Tukey. An algorithm for machine computation of complex fourier series. *Math. Computation*, 19:297-301, Apr. 1965.

[Hu92] Y. H. Hu. CORDIC-based VLSI Architectures for Digital Signal Processing. *IEEE Signal Processing Magazine*, 9, Jul 1992.

ES 2 283 236 B2

[MGBS02] K. **Maharatna**, E. **Grass**, S. **Banerjee**, and A. **Sundar**. CORDIC UNIT. *Patente US 20060059215A1*, 20.12. 2002.

5 [Vo159] J. E. **Volder**. The CORDIC trigonometric computing technique. *IRE Trans. Electronic Computers*, EC-8:330-334, Sep. 1959.

[YCC06] **Cheng-Ying** Yu, **Sau-Gee** Chen, and **Jen-Chuan** Chih. Efficient CORDIC Designs for Multi-Mode OFDM FFT. *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 3:1036-1039, May 2006.

10 [ZH05] Li **Zou** and Xiao **Huang**. 3780-point Discrete Fourier Transformation processor. *Patente EP 1750206A1*, 04.08. 2005.

15

20

25

30

35

40

45

50

55

60

65

REIVINDICACIONES

1. Procedimiento para el cálculo de las rotaciones de cualquier FFT descompuesta según el algoritmo Cooley-Tukey, y cuyo número de puntos, N , y $radix$, r , son ambos potencia de 2, con los siguientes pasos:

- obtener las secuencias de ángulos de rotación de cada una de las etapas de la FFT,
- calcular las rotaciones de cada una de las etapas de la FFT a partir de los ángulos de rotación generados,

caracterizado porque a partir de un único contador se obtienen las secuencias de ángulos de rotación de todas las etapas de la FFT, y para una etapa cualquiera, s , de la FFT, la generación de la secuencia de ángulos de rotación de dicha etapa, θ_s , se realiza siguiendo los siguientes pasos:

- tomar los $s \cdot \log_2 r$ bits menos significativos proporcionados por el contador,
- de los bits tomados, multiplicar los $(s - 1) \cdot \log_2 r$ bits menos significativos por los $\log_2 r$ más significativos, obteniendo así los valores de la secuencia a_s ,
- multiplicar el resultado por N/r^s , obteniendo así los valores de la secuencia θ_s .

2. Procedimiento según la reivindicación 1, **caracterizado** porque para el cálculo de las rotaciones de cada una de las etapas de la FFT, se obtiene el vector de rotaciones adaptado, δ' , a partir del vector de rotaciones, δ , de la siguiente forma:

$$\begin{aligned}\delta'_m &= \delta_m \\ \delta'_i &= -\delta_i \cdot \delta_{i-1}, \quad i = m + 1, \dots, M \\ \delta'_{M+1} &= \delta_M\end{aligned}$$

donde m es el índice correspondiente a la primera microrrotación y M el correspondiente a la última microrrotación.

3. Procedimiento según las reivindicaciones 1 a 2, **caracterizado** porque para el cálculo de las rotaciones de cada una de las etapas de la FFT, el cálculo de las microrrotaciones se realiza de la siguiente forma:

$$\begin{aligned}a_{i+1} &= \begin{cases} a_i + b_i 2^{-i} & \text{si } \delta'_i = -1 \\ b_i + a_i 2^{-i} & \text{si } \delta'_i = 1 \end{cases} \\ b_{i+1} &= \begin{cases} b_i - a_i 2^{-i} & \text{si } \delta'_i = -1 \\ a_i - b_i 2^{-i} & \text{si } \delta'_i = 1 \end{cases}\end{aligned}$$

siendo a_i y b_i las entradas de la i -ésima etapa de microrrotación.

4. Arquitectura de circuito para implementar el procedimiento descrito en las reivindicaciones 1 a 3, **caracterizada** porque comprende:

- un módulo de generación de ángulos que incluye un único contador (1) para toda la FFT, a partir del cual se obtienen los ángulos de rotación de todas las etapas de la FFT sin necesidad de recurrir a ningún dato almacenado previamente;
- un módulo de cálculo de las rotaciones para cada una de las etapas de la FFT.

5. Arquitectura de circuito según la reivindicación 4, **caracterizada** porque el módulo de generación de ángulos comprende, además del contador (1), los siguientes elementos:

- lógica combinacional (3), que opera sobre el valor del contador,
- un bloque acumulador (4) por cada etapa de la FFT, para el cálculo de las secuencias a_s .

6. Arquitectura de circuito según las reivindicaciones 4 y 5, **caracterizada** porque, dentro del módulo de generación de ángulos, cada bloque acumulador (4) comprende los siguientes elementos:

- un sumador (43), que permite ir actualizando los valores de la secuencia a_s ,
- un registro (45) que proporciona los valores de la secuencia a_s ,
- una puerta lógica (44) para el control del bloque acumulador.

7. Arquitectura de circuito según las reivindicaciones 4 a 6, **caracterizada** porque el módulo de cálculo de las rotaciones de la FFT, comprende bloques de cálculo de las microrrotaciones (9) que consisten en un conmutador (93), un sumador (96), y un restador (97).

8. Arquitectura de circuito para el cálculo de las rotaciones de la FFT según las 5 reivindicaciones 4 a 7, **caracterizada** porque en el caso de que el primer bloque de cálculo de las microrrotaciones sea el correspondiente a $\alpha_1 = tg^{-1}(2^{-1})$, se puede añadir un módulo adicional de compensación del escalado que consiste en dos restadores.

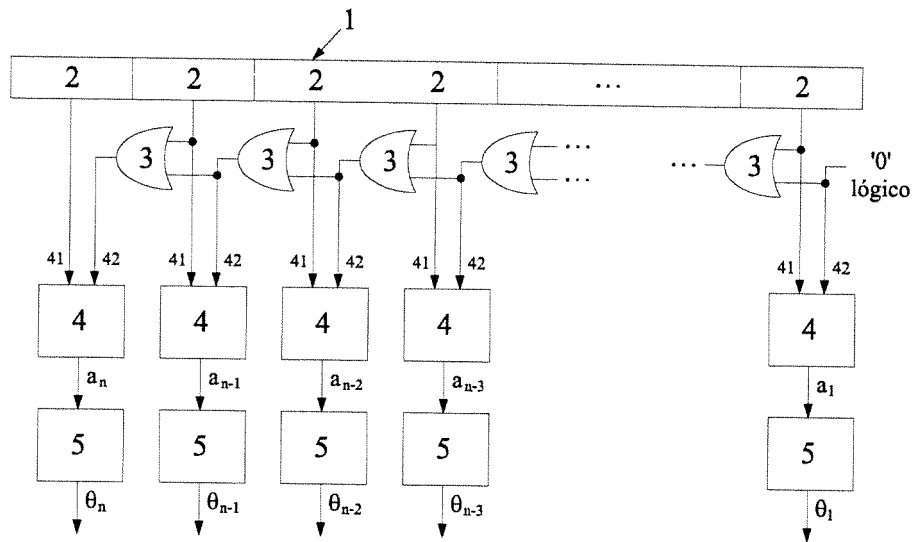


Figura 1

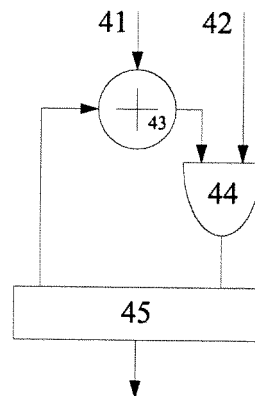


Figura 2

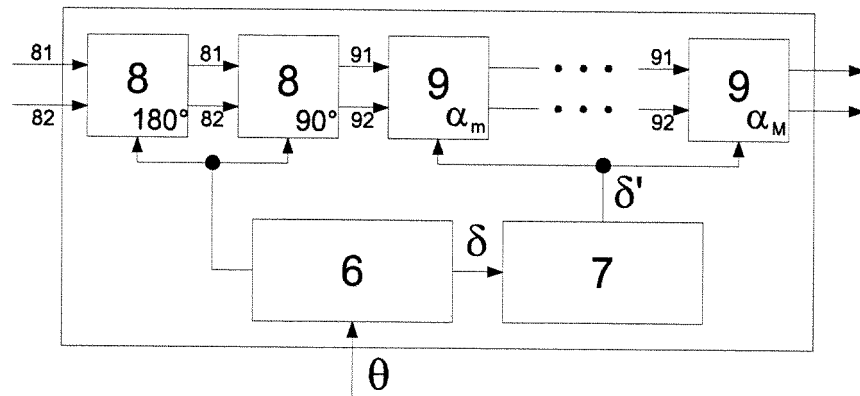


Figura 3

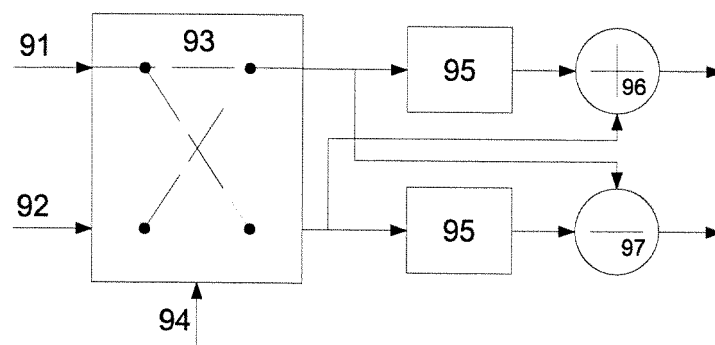


Figura 4



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

⑪ ES 2 283 236

⑫ Nº de solicitud: 200700983

⑬ Fecha de presentación de la solicitud: 12.04.2007

⑭ Fecha de prioridad:

INFORME SOBRE EL ESTADO DE LA TÉCNICA

⑮ Int. Cl.: G06F 17/14 (2006.01)

DOCUMENTOS RELEVANTES

Categoría	Documentos citados	Reivindicaciones afectadas
A	US 2005216540 A1 (WEN et al.) 29.09.2005	
A	CHENG-YING YU; SAN-GEE CHEN; JEN-CHUAN CHIH. "Efficient CORDIC Designs for Multi-Mode OFDM FFT", Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on Volumen 3, 14-19 Mayo 2006 Páginas: III-III [en línea] [recuperado el 04.09.2007]. Recuperado de internet: <URL:http://ieeexplore.ieee.org/iel5/11024/34759/01660834.pdf?tp=&arnumber=1660834&isnumber=34759>	
A	DICK, C. Computing the Discrete Fourier Transform on FPGA Based Systolic Arrays", Field-Programmable Gate Arrays, 1996. FPGA '96. Proceedings of the 1996 ACM Fourth International Symposium on 1996 Páginas: 129-135 [en línea] [recuperado el 04.09.2007]. Recuperado de internet: <URL:http://ieeexplore.ieee.org/iel5/9475/30061/01377297.pdf?tp=&arnumber=1377297&isnumber=30061>	
A	GROGINSKY, H.L.; WORKS, G.A. "A Pipeline Fast Fourier Transform", Computers, IEEE Transactions on Volumen C-19, Issue 11, Nov. 1970 Páginas: 1015-1019 [en línea] [recuperado el 04.09.2007]. Recuperado de internet: <URL:http://ieeexplore.ieee.org/iel5/12/35041/01671419.pdf?tp=&arnumber=1671419&isnumber=35041>	

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

24.09.2007

Examinador

J. Botella Maldonado

Página

1/2



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

⑪ ES 2 283 236

⑫ Nº de solicitud: 200700983

⑬ Fecha de presentación de la solicitud: 12.04.2007

⑭ Fecha de prioridad:

INFORME SOBRE EL ESTADO DE LA TÉCNICA

⑮ Int. Cl.: G06F 17/14 (2006.01)

DOCUMENTOS RELEVANTES

Categoría	Documentos citados	Reivindicaciones afectadas
A	SUNG, T.-Y. "Memory-efficient and high-speed split-radix FFT/IFFT processor based on pipelined CORDIC rotations", Vision, Image and Signal Processing, IEE Proceedings-Volumen 153, Issue 4, Agosto 2006 Páginas: 405-410 [en línea] [recuperado el 04.09.2007]. Recuperado de internet: <URL:http://ieeexplore.ieee.org/iel5/2200/35271/01677109.pdf?tp=&arnumber=1677109&isnumber=35271>	

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

24.09.2007

Examinador

J. Botella Maldonado

Página

2/2