

Transformada Discreta Wavelet 2-D para procesamiento de video en tiempo real.

Ricardo J. Colom, Rafael Gadea, Ángel Sebastiá, Marcos Martínez, Vicente Herrero, Vicente Arnau*

Resumen—Este trabajo presenta una arquitectura y una realización de la Transformada Wavelet Discreta 2D (2-D DWT) sobre una FPGA. Esta arquitectura trabaja en una forma no-separable utilizando una estructura de filtro paralela con control distribuido para computar todos los niveles de resolución de la DWT, de forma que la muestra de entrada puede ser procesada a razón de una muestra por ciclo de reloj. Para la computación de una imagen fija de $N \times N$ con un filtro de longitud L , se necesita $N^2 + N$ ciclos de reloj y $6N$ celdas de almacenamiento de memoria. Hemos estudiado algunos de los filtros para compresión de imágenes más usuales, analizando el número de bits necesarios para su almacenamiento y para su aplicación.

Palabras clave—Wavelets, Biortogonal, Procesamiento Digital de Imágenes y VHDL.

I. INTRODUCCIÓN

LA Transformada Wavelet Discreta está recibiendo mucha atención en el campo del procesamiento de imágenes debido a su flexibilidad en la representación de señal no estacionarias y a su adecuación por adaptarse a las características del sistema de visión humano. Su aplicación realiza fielmente un análisis multiresolución y una descomposición subbanda, por lo cual su utilización en los últimos 10 años está siendo fundamental en el procesamiento digital de imágenes y en todas sus disciplinas.

Este método de transformación descrito fue realizado por primera vez por Mallat [12], ofrece ortogonalidad y lidera las representaciones multicapa. Han sido propuestas y realizadas algunas arquitecturas VLSI para la DWT 1-D. Si bien la transformada DWT 2-D puede ser una extensión de la 1-D, es bastante difícil la realización de una arquitectura eficiente que permita su aplicación con un bajo coste en hardware y unas buenas prestaciones.

Básicamente, la DWT 2-D puede ser realizada utilizando tanto filtros separables como no separables. En la aproximación de diseño separable, las operaciones de la transformada pueden ser divididas en un procesamiento por filas y otro por columnas, que puede ser realizado mediante una arquitectura 1-D de la DWT. Sin embargo, Lewis y Knowles [13] proponen una arquitectura simple para el caso especial de la DWT de Daubechies, pero este trabajo no es eficiente para otras Wavelets. Vishwanath [4] construye dos filtros en array sistólico, dos filtros de computación paralelos y una unidad de almacenamiento para completar la transformada 2-D.

Chakabarti [6] diseña una arquitectura con estructura regular utilizando arrays sistólicos lineales. Una arquitectura de array VLSI con cauce segmentado para la DWT 2-D es presentada por Chang y Chen [14]. Ming-Hwa [11] presenta una arquitectura VLSI con un bajo coste en hardware y poca memoria para la DWT 2-D separable. Dentro de las aproximaciones no-separable, Chakabarti [6] utiliza filtros paralelos y bastantes registros de desplazamiento para almacenar una DWT 2-D rápida. Chu Yu [5] utiliza estructuras de filtros sistólico-paralelos y muchos registros de almacenamiento para realizar la DWT 2-D.

En este trabajo, diseñamos la arquitectura y realizamos una FPGA para la descomposición DWT 2-D, utilizando una estructura de filtro sistólica-paralela con control distribuido para computar todos los niveles de resolución de la DWT.

Este Trabajo lo hemos organizado de la siguiente forma: primero, en el apartado II, vamos a realizar una breve introducción a la DWT 1-D y la 2-D. En el apartado III vamos a realizar un estudio sobre que filtros utilizar para la DWT. En el apartado IV presentaremos la arquitectura de la DWT 2-D. En el apartado V describimos la realización sobre FPGA la arquitectura propuesta en el apartado anterior. En el apartado VI presentamos realizaciones y comparaciones de varias arquitecturas de la DWT que ya han sido propuestas por otros autores. Finalmente, en el apartado VII mostramos nuestras conclusiones.

II. TRANSFORMADA DISCRETA WAVELET.

La Transformada Wavelet se utiliza para la compresión tanto de imágenes como de audio. En el proceso de análisis de la Wavelet, las señales son representadas utilizando un grupo de funciones básicas producidas por el desplazamiento y el escalado de una función madre o función principal. La transformada Wavelet es una descomposición de una señal en frecuencias.

La Transformada Wavelet Discreta unidimensional descompone recursivamente la señal de entrada, $S_d(n)$, en una parte de detalle y otra de promedio en cada iteración. Sea $S_i(n)$ y $W_i(n)$ el promedio y el detalle respectivamente, y sea i el nivel de aplicación

Departamento de Ingeniería Electrónica, Universidad Politécnica de Valencia, Camino de Vera s/n, E-46022 Valencia. Correo electrónico: rcolom@eln.upv.es.

* Departamento de Informática. Universidad de Valencia, Av. Vicente Andrés Estellés s/n, 46100 Burjassot, Valencia: Vicente.Arnau@uv.es.

de la transformada. La aproximación de la señal en el nivel $i+1$ se calcula usando

$$S_{i+1}(n) = \sum_k g(k)S_i(2n-k) \tag{1}$$

Y el detalle de la señal en el nivel $i+1$ se calcula usando

$$W_{i+1}(n) = \sum_k h(k)S_i(2n-k) \tag{2}$$

Las ecuaciones (1) y (2) describen la computación de la DWT. Esta técnica de cálculo de la DWT es a menudo referenciada como el algoritmo piramidal o algoritmo de Mallat [2]. Los tres niveles de cálculo de la DWT 1-D se muestran en la siguiente figura

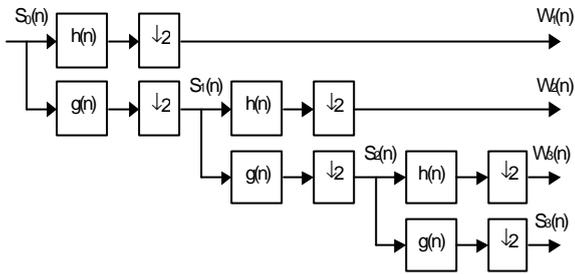


Fig. 1. Diagrama de bloques del banco de filtros de la DWT.

La DWT bidimensional se ha aplicado en análisis multi-resolución visión por computador, y compresión de imágenes. La DWT bidimensional trabaja sobre una señal 2-D como puede ser una imagen. Tal como los filtros 1-D se utilizan para computar la DWT 1-D, para computar la DWT 2-D utilizamos filtros 2-D calculados a partir de la convolución de los filtro 1-D sobre ellos mismos. Estos filtros 2-D pueden ser separables o no separables, donde un filtro 2-D $f(n_1, n_2)$ es separable si es expresable como $f(n_1, n_2) = f_1(n_1) f_2(n_2)$. La DWT 2-D separable descompone una imagen $S_i(n_1, n_2)$ en una imagen promedio y tres imágenes detalle, de acuerdo con las expresiones

$$S_{i+1}(n_1, n_2) = \sum_{k_1} \sum_{k_2} g(k_1)g(k_2)S_i(2n_1-k_1, 2n_2-k_2) \tag{3}$$

$$W_{i+1}^1(n_1, n_2) = \sum_{k_1} \sum_{k_2} g(k_1)h(k_2)S_i(2n_1-k_1, 2n_2-k_2) \tag{4}$$

$$W_{i+1}^2(n_1, n_2) = \sum_{k_1} \sum_{k_2} h(k_1)g(k_2)S_i(2n_1-k_1, 2n_2-k_2) \tag{5}$$

$$W_{i+1}^3(n_1, n_2) = \sum_{k_1} \sum_{k_2} h(k_1)h(k_2)S_i(2n_1-k_1, 2n_2-k_2) \tag{6}$$

donde $H(z)$ y $G(z)$ son los filtros Wavelet 1-D. La señal $S_{i+1}(n_1, n_2)$ es un suavizado de baja resolución de la imagen $S_i(n_1, n_2)$. Este suavizado se calcula desde $S_i(n_1, n_2)$ mediante un filtro pasa-baja y diezmando por 2 a lo largo de filas y columnas. Las señales $W_{i+1}^1(n_1, n_2)$, $W_{i+1}^2(n_1, n_2)$, y $W_{i+1}^3(n_1, n_2)$

contienen el detalle de $S_i(n_1, n_2)$. El nivel 1 de la DWT 2-D se calcula como muestra la figura 2.

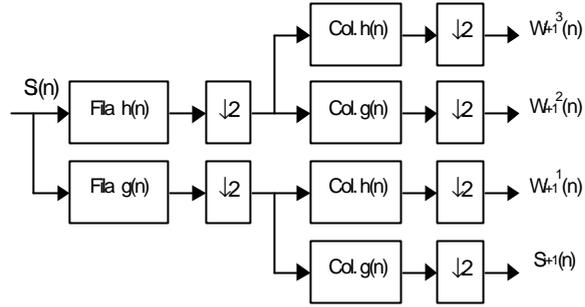


Fig. 2. Diagrama de bloques del banco de filtros de análisis utilizado para calcular la DWT 2-D.

En contraste con los filtros separables, los filtros no separables descomponen directamente una imagen en cuatro sub-ímagenes resultado de la aplicación de la DWT por filas y posteriormente por columnas. La figura 3 muestra la arquitectura de la DWT no-separable con dos niveles de resolución. Notar que cada bloque en esta arquitectura es un filtro 2-D submuestreado por 2. La figura 4 ilustra la descomposición en 8 bandas de una imagen utilizando la DWT 2-D.

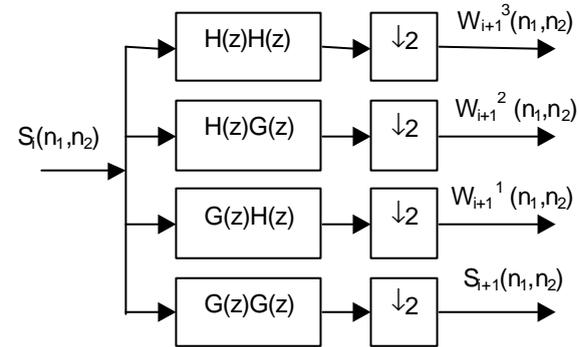


Fig. 3. Diagrama de bloques del banco de filtros de análisis utilizado para calcular la DWT 2-D no-separable.

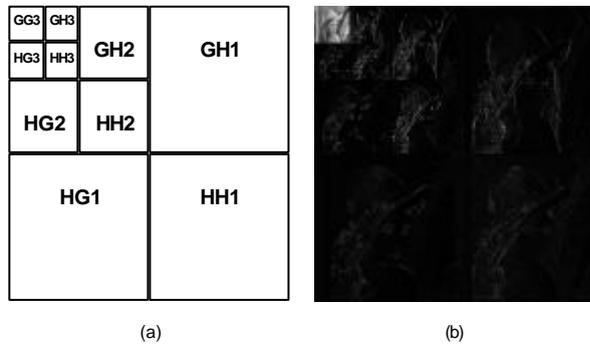


Fig. 4. (a) Descomposición en 8 bandas de una imagen (b) Descomposición de la imagen Lena.

Chakrabarti y Vishwanath [7] propusieron una arquitectura VLSI para la DWT no-separable 2-D con una estructura de filtro paralelo que mejora las

desventajas de los filtros separables. Los componentes hardware de esta arquitectura incluyen $2L^2$ multiplicadores programables, $2(L^2-1)$ sumadores, $2NL$ memorias de almacenamiento, y una unidad de control, donde L es la longitud del filtro y N es el tamaño de columna de una imagen. La estructura de filtro paralelo utiliza el algoritmo piramidal recursivo modificado en 2-D para construir la arquitectura de la DWT 2-D completa, tal que el tiempo de computación puede alcanzar alrededor de N^2 . Entonces esta arquitectura adopta filtros no separables, necesarios para no transponer memoria entre filas y columnas de la DWT. Como resultado, reduce considerablemente el espacio de memoria y el tiempo de latencia. Sin embargo, esta arquitectura contiene un número relativamente grande de componentes hardware y un complicado encaminamiento (proporcional al cuadrado de la longitud del filtro) ya que su coste hardware es grande para una realización de la DWT 2-D no separable.

III. ESTUDIO Y ANÁLISIS DE LOS FILTROS DE LA DWT.

La Transformada Wavelet conocida como Daub-4 es realizada utilizando una unidad de filtrados FIR de cuatro coeficientes. El filtro $G(z)$ (pasa-baja) usado en el proceso de análisis de la DWT, tiene los siguientes coeficientes:

$$g(0) = \frac{1+\sqrt{3}}{4\sqrt{2}}; \quad g(1) = \frac{3+\sqrt{3}}{4\sqrt{2}}; \quad g(2) = \frac{3-\sqrt{3}}{4\sqrt{2}}; \quad g(3) = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

Como es bien conocido en un filtro sub-banda, la simetría de los filtros y la reconstrucción exacta de las señales son conceptos incompatibles si el mismo filtro es usado en reconstrucción y descomposición. De cualquier forma, es posible obtener simetría en el filtro utilizando coeficientes biortogonales en lugar de unos ortogonales. Los coeficientes biortogonales estudiados son presentados en la siguiente tabla:

Filtros	Coefficientes
CDF 3/1	$G(z) = \frac{\sqrt{2}}{4}(-z^{-1} + 3 + 3z - z^2)$ $G_s(z) = \frac{\sqrt{2}}{8}(z^{-1} + 3 + 3z + z^2)$
CDF 2/2	$G(z) = \frac{\sqrt{2}}{8}(-z^{-2} + 2z^{-1} + 6 + 2z - z^2)$ $G_s(z) = \frac{\sqrt{2}}{4}(z^{-1} + 2 + z)$
CDF 9/7	$G(z) = \frac{\sqrt{2}}{64}(z^{-4} - 8z^{-2} + 16z^{-1} + 46 + 16z - 8z^2 + z^4)$ $G_s(z) = \frac{\sqrt{2}}{32}(-z^{-3} + 9z^{-1} + 16 + 9z - z^3)$

Una vez los algoritmos han sido probados y verificados en Matlab, continuamos con una realización inicial utilizando un lenguaje de descripción hardware como VHDL. Esto ha mostrado ser útil para el estudio del comportamiento de los filtros con aritmética entera.

En la Tabla I mostramos el resultado de nuestro estudio. La información que se muestra es la siguiente: relación señal-ruido, número de bits utilizados en la cuantización de los coeficientes; número máximo de bits necesarios para la representación de la DWT después de 4 octavas; máximo número de bits teóricamente necesarios para la realización de las operaciones aritméticas internas de la DWT; y el número máximo de bits necesarios en la práctica para la realización de las operaciones aritméticas internas de la DWT sin pérdida de información.

IV. ARQUITECTURA PROPUESTA PARA LA DWT 2-D.

A. Aproximación directa.

Una realización sencilla de la DWT 2-D no separable es el resultado de aplicar el algoritmo piramidal. En esencia consiste en 4 filtros 2-D modulados (HH, HG, GH y GG) los cuales son usados repetidamente de la forma que muestra la figura 5. El número de ciclos de reloj necesarios para calcular los $N \times N$ pixels de la DWT con un nivel es $4N^2$.

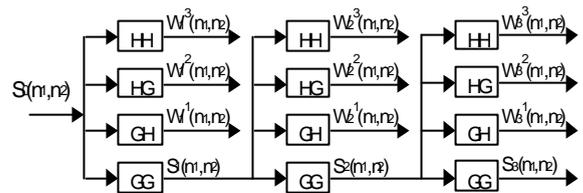


Fig. 5. Arquitectura de la DWT 2-D con 3 niveles.

B. Arquitectura par-impar paralela.

Para reducir el número de ciclos de reloj hemos propuesto una arquitectura de filtros 2-D no separables con una estructura par-impar paralela. La figura 6 muestra el diagrama de bloques de unos filtros 2-D no separables con una estructura paralela par-impar. Está realizada principalmente por dos unidades de filtros. El filtro impar calcula la salida de los pixeles impares de la DWT (salida C) y el filtro par calcula la salida de los pixeles pares de la DWT (salida D). Las entradas A y B son, respectivamente, los pixeles pares e impares de la señal de entrada. Esta estructura calcula en paralelo dos pixeles de la entrada.

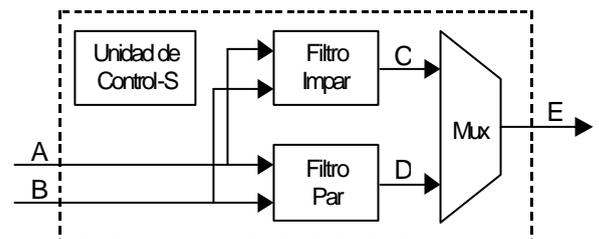


Fig. 6. Estructura del filtro par-impar 2-D paralelo para la DWT 2-D no separable.

Esta estructura tiene una latencia de 8 ciclos de reloj y una productividad de 4 ciclos de reloj. Por lo tanto, el número de ciclos de reloj necesarios para calcular la DWT sobre $N \times N$ píxeles con un nivel es $N^2 + 4$. De este modo, el número de ciclos de reloj necesarios para calcular la DWT sobre $N \times N$ píxeles con tres niveles es

$$[N^2 + 4] + \left[\left(\frac{N}{2} \right)^2 + 4 \right] + \left[\left(\frac{N}{4} \right)^2 + 4 \right] = \frac{11}{8}N^2 + 12 \quad (7)$$

Pero, nosotros no necesitamos espera hasta terminar el primer nivel para empezar a computar el segundo nivel. Podemos sincronizar la computación de los tres niveles y reducir el número de ciclos de reloj a $N^2 + 4N + 12$. Utilizamos unidades de almacenamiento para conectar los siguientes niveles. Las unidades de almacenamiento son memorias de doble puerto, y almacenamos solo seis filas de la DWT.

C. Arquitectura recurrente.

Para terminar nosotros hemos plegado las arquitecturas mostradas en la Figura 5, utilizando arquitectura recurrente. Hay una unidad de filtrado para calcular el primer nivel y otra para calcular el segundo y el tercer nivel.

La Figura 7 muestra la arquitectura recursiva para una DWT 2D no separable con tres niveles. Esta arquitectura utiliza las unidades de filtros descritos en la sección B, y mostradas en la figura 6. El número de ciclos de reloj necesarios para calcular la DWT sobre $N \times N$ píxeles con tres niveles es de $N^2 + 4N + 12$.

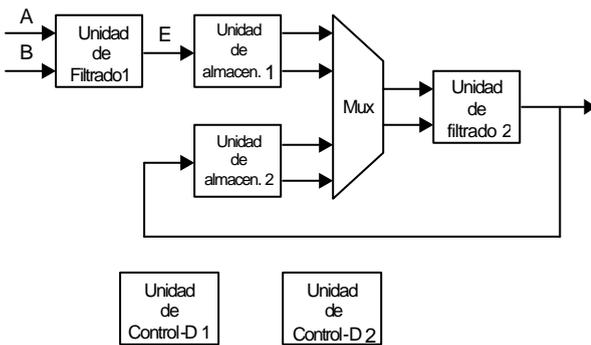
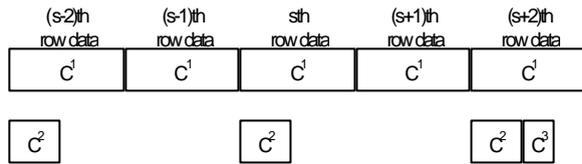


Fig. 7. Diagrama de bloques de la arquitectura recursiva para la DWT 2-D no-separable con tres niveles.

La figura 8 muestra el diagrama de tiempos para tres niveles de resolución del cálculo de la arquitectura DWT 2-D no separable. El primer nivel (unidad de filtrado 1) está continuamente calculando. El segundo nivel (unidad de filtrado 2) comienza cuando el primer nivel ha generado las primeras cuatro filas. El segundo nivel sigue calculando cada vez que el primero genera dos nuevas líneas.



C^m : Ciclo de computación del nivel m. Cuando $1 \leq s \leq N/2$ para N es el tamaño de la fila.

Fig. 8. Diagrama de tiempos de la arquitectura recursiva propuesta para la DWT 2-D

D. Unidad de control.

Para controlar la arquitectura mostrada en la figura 7, proponemos un control distribuido con dos unidades de control diferentes. Una unidad de control simple en la unidad de filtrado (S-control unit en la figura 6), y una unidad de control compleja (D-control unit en la figura 7). La unidad de control D controla la conexión de los datos entre un nivel y el siguiente nivel. De este modo, se necesitan dos unidades de control D para realizar la DWT 2-D con tres niveles. La figura 9 muestra como la unidad de control D es utilizada en una arquitectura desdoblada para la DWT 2-D con tres niveles.

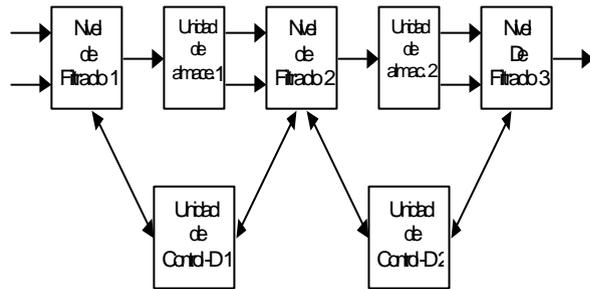


Fig. 9. Como usar el control distribuido en una arquitectura desdoblada.

La unidad de control D se comunica con la unidad de control S de la unidad de filtrado. Por consiguiente, la unidad de filtrado 2, mostrado en la figura 7, necesita 2 unidades de control S para comunicarse con la unidad de control-D 1, y otra para comunicarse con la unidad de control-D 2.

El control distribuido permite que la arquitectura sea totalmente modular y escalables. Para aumentar el número de niveles, solo es necesario copiar la unidad de control-D y la unidad de control-S.

V. REALIZACIÓN CON FPGA.

Siguiendo la arquitectura propuesta y descrita en la sección IV.C, hemos realizado la DWT 2-D no separables con una FPGA Xilinx XCV600E. El diagrama de bloques del integrado con filtro con 4 etapas y 3 octavas, se muestra en la figura 7. Está principalmente compuesto por 2 unidades de almacenamiento, 2 unidades de filtro, y 2 unidades de control. El filtro unidad fue mostrado en la sección IV.B y la figura 6.

La unidad de almacenamiento es un bloque de memoria de doble puerto, el cual tiene dos puertos de acceso independientes, permitiendo acceso compartido. El tamaño de los datos y la memoria dependen de cómo cada puerto esté configurado, independientemente. El puerto A (solo escritura) toma valores de datos de P-bits y direcciones de Q-bits. El puerto B (solo lectura) toma valores de datos de 2P-bits y direcciones de (Q-1)-bits. Entonces cada puerto está accediendo a la misma cantidad de memoria, el formato para la palabra de datos para el puerto B será equivalente al formato de las dos palabras de datos del puerto A. Un simple acceso a memoria en el puerto B es equivalente a dos accesos realizados en el puerto A. La unidad de almacenamiento 1 tiene una memoria de profundidad $8N/2$ palabras, y la unidad de almacenamiento 2 tiene una memoria de profundidad $8N/4$ palabras. Estas unidades de almacenamiento son el "core" de Xilinx.

El proceso de implementación es el siguiente. Primero, usamos simulaciones con el programa Matlab para validar la DWT 2-D no separable. Las simulaciones son también utilizadas para generar patrones de test. Después, usamos un modelo VHDL para desarrollar el comportamiento y la estructura del código VHDL. Un sintetizador VHDL produce un circuito digital que puede ser verificado para su corrección. Tras el particionado, emplazamiento y rótulo realizados por las herramientas de Xilinx, se generan los bits de configuración de la FPGA y la información del análisis temporal necesaria.

Basándonos en los resultados, es posible estimar la frecuencia máxima del reloj. Este ciclo de reloj, determina el rendimiento de procesamiento de los datos. Los resultados numéricos de la FPGA, se resumen en la Tabla II.

TABLA II
DATOS DE LA REALIZACIÓN CON UNA FPGA.

Processing Image Size	512 × 512
Input Data Precision	8 bits
Output Data Precision	14 bits
Filter Coefficients	Non-separable 4 × 4
Levels DWT	3
Device	XCV600E
No. Slices	2568 (37 %)
No. Block RAM's	14 (19 %)
Equivalent gate	285114
Clock Rate	45 MHz

VI. RENDIMIENTOS Y COMPARACIONES DE ARQUITECTURAS DE LA DWT.

En esta sección, examinaremos las prestaciones de las arquitecturas DWT 2-D. La comparación de resultados de nuestra y de otras arquitecturas, es resumida en la Tabla III. Nuestra arquitectura DWT consiste en filtros regulares 2-D, y solo necesita 6N unidades de almacenamiento, 8 multiplicadores, y 8

sumadores. Entonces, comparando nuestra arquitectura con las otras, es conveniente señalar que es la única que puede ser realizada en una FPGA. Nuestra arquitectura para procesamiento de vídeo en tiempo real, es la que menos unidades de almacenamiento utiliza [8].

VII. CONCLUSIONES.

Hemos descrito en este trabajo una rápida y eficiente arquitectura realizada sobre una FPGA de la DWT 2-D. La arquitectura de la DWT 2-D es no separable, en general, no es muy complicada, y es realizable sobre una FPGA. En este trabajo, presentamos una arquitectura de la DWT 2-D no separable, paralela y recursiva, la cual utiliza un control distribuido y pequeñas memorias de almacenamiento. Por otra parte, esta arquitectura 2-D puede ser fácilmente ampliable utilizando de filtros de distintos tamaños, permitiendo incrementar el número de octavas con facilidad, y es realizable sobre una FPGA. Además permite procesar vídeo en tiempo real.

VIII. REFERENCIAS.

- [1] M. Antonini, M. Barlaud and I. Daubechies. "Image Coding Using Wavelet Transform" IEEE, Transactions on Image Processing, Vol. 1, No. 2, April 1992.
- [2] I. Daubechies. "Time-Frequency Localization Operators: Geometric Phase Space Approach" IEEE Transactions on Information Theory, Vol. 34, No. 4, July 1988.
- [3] D. Wei, J. Tian, R. O. Wells and C.S. Burrus. "A New Class of Biorthogonal Wavelet Systems for Image Transform Coding" IEEE Transactions On Image Processing, Vol. 7, No. 7, July 1998.
- [4] Mohan Vishwanath, Robert Michael Owens, and Mary Jane Irwin. "VLSI Architectures for the Discrete Wavelet Transform" IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 42, No. 5, May 1995.
- [5] Chu Yu and Sao-Jie Chen. "VLSI Implementation of 2D Discrete Wavelet Transform for Real-Time Video Signal Processing." IEEE transactions on consumer electronics, 1997, vol. 43, no. 4, p. 1270 -1279.
- [6] C. Chakrabarti, and C. Mumford. "Efficient Realizations of Encoders and Decoders Based on the 2D Discrete Wavelet Transform" IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 7, No. 3, September 1999.
- [7] C. Chakrabarti, and M. Vishwanath. "Efficient Realizations of The Discrete and Continuous Wavelet Transform: From Single Chip Implementations to Mapping on SIMD Array Computers" IEEE Transactions Signal Processing, Vol. 43, No. 3, pp. 759-771, March. 1995.
- [8] Chien-Yu Chen, Zhong-Lan Yang, Tu-Chih Wang, and Liang-Gee Chen. "A Programmable VLSI Architecture for 2-D Discrete Wavelet Transform" IEEE, 2000.
- [9] Aleksander Grzeszczak, Mrinal K. Mandal and Sethuraman Panchanathan. "VLSI Implementation of Discrete Wavelet Transform" IEEE Transactions on VLSI Systems, Vol. 4, No 4, December 1996.
- [10] Keshb K. Parhi and Takao Nishitani. "VLSI Architectures for Discrete Wavelet Transforms" IEEE Transactions on VLSI Systems, Vol. 1, No. 2, June 1993.

