

Implementación en hardware reconfigurable de la Transformada Wavelet 2D

Estefanía Alcocer Espinosa, Otoniel López Granado, Roberto Gutiérrez Mazón

Dept. Física y Arquitectura de Computadores

Universidad Miguel Hernández

03202 Elche

estefania.alcocer@alu.umh.es, (otoniel,roberto)@umh.es

Resumen

Este artículo describe el diseño y realización de la Transformada Wavelet basada en las arquitecturas Line-Based y Lifting-Based y la implementación en hardware reconfigurable, en concreto, en la FPGA (*Field Programmable Gate Array*) de Xilinx Virtex-II Pro. La meta de este trabajo es mostrar dos arquitecturas VLSI (*Very Large Scale Integration*) de la Transformada Wavelet 2D para imágenes, evaluando su complejidad y sus ventajas.

1. Introducción

La rápida y exitosa evolución de los sistemas multimedia digitales es indispensable en el mundo digital de hoy en día. La omnipresencia multimedia se ha convertido en uno de los principales objetivos del desarrollo tecnológico actual. Ahora, muchos productos electrónicos y aplicaciones multimedia se desarrollan para lograr este objetivo, como la televisión broadcast móvil, servicios móviles multimedia, videollamadas, cámaras digitales y videocámaras [2]. Debido a la necesidad de progreso de estas aplicaciones, se ha producido una intensa actividad en el desarrollo de codificadores de imagen y video, plasmado en los diferentes estándares de compresión como pueden ser JPEG, JPEG2K, MPEG1, MPEG2, MPEG4, H.261, H.263, H.264, SPIHT y JPEG2000.

La *Transformada Wavelet Discreta* (DWT) presenta una importante técnica de aplicación multimedia, no sólo es un algoritmo clave para

el procesado de señal sino que también ha llevado a revolucionar los algoritmos de codificación de vídeo e imagen [2].

El interés en la Transformada Wavelet Discreta (DWT) se ha incrementado en los últimos años debido a sus aplicaciones en imagen y vídeo, tales como JPEG2000 y MPEG4. La representación multi-resolución derivada de la descomposición espacio-frecuencia en la DWT demuestra ventajas significativas en el análisis de señales y en compresión. A pesar de la simplicidad conceptual, los requerimientos críticos tales como consumo de memoria, ancho de banda, la sincronización y programabilidad pueden dar lugar a diseños costosos [5].

En este trabajo se describen dos esquemas de implementación de la transformada wavelet 2D con bajo consumo de memoria: Line-based y Lifting-based. La arquitectura Line-Based es un esquema de bajo consumo de memoria comparado con la convolución tradicional, que requiere tener toda la imagen en memoria. La arquitectura Lifting un método superior a la convencional convolución que proporciona una significativa reducción en memoria y en la complejidad computacional, el esquema lifting sobrescribe los coeficientes wavelet en los lugares de la memoria donde se contiene los valores de entrada, esto conlleva un coste menor en recursos hardware y un menor tiempo de computación.

Este artículo se organiza de la siguiente manera: En el apartado 2 se describen las bases teóricas de la Transformada Wavelet

Discreta, en el punto 3 se presentan las arquitecturas implementadas, en el apartado 4 se muestran los resultados de las distintas implementaciones y por último se comentan las conclusiones y posibles líneas futuras.

2. Bases teóricas de la Transformada Wavelet Discreta

Las Wavelets son funciones generadas por una única función (función base) llamada *prototipo* o *wavelet madre* mediante *dilataciones* (escalados) y *traslaciones* en el dominio del tiempo (frecuencia). Si la wavelet madre es denotada por $\psi(t)$, las otras wavelets $\psi_{a,b}(t)$ pueden ser representadas como

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \quad (1)$$

Las variables a y b representan los parámetros para las *dilataciones* y *traslaciones* respectivamente en el eje del tiempo.

Basándose en la definición de las wavelets, la *transformada wavelet* de una función (señal) $f(t)$ se representa matemáticamente mediante

$$W(a,b) = \int_{-\infty}^{\infty} \psi_{a,b}(t) f(t) dt \quad (2)$$

$W(a,b)$ es lo que se conoce como la *transformada wavelet continua* (CWT).

Cuando la señal de entrada es procesada por un dispositivo digital, como puede ser una imagen digital, es necesario describir la versión discreta de la transformada wavelet [3]. En general, los coeficientes wavelet para una función $f(t)$ son dados por

$$c_{m,n}(f) = a_0^{-\frac{m}{2}} \int f(t) \psi(a_0^{-m}t - nb_0) dt \quad (3)$$

$$c_{m,n}(f) = 2^{-\frac{m}{2}} \int f(t) \psi(2^{-m}t - n) dt. \quad (4)$$

Esta transformación es normalmente llamada como la *transformada wavelet discreta en tiempo* (DTWT). La *transformada wavelet discreta* (DWT) es una herramienta de procesado de señal muy versátil, proponiendo la representación multiresolución de señales basadas

en la descomposición wavelet. La función de la multiresolución es la de representar una señal con una colección de coeficientes, cada uno de ellos proporcionando información sobre la posición y la frecuencia de la señal. Como resultado, la DWT descompone una señal digital en diferentes subbandas de modo que las subbandas de baja frecuencia tienen una resolución en frecuencia más fina y una resolución en tiempo más gruesa comparadas con las subbandas de alta frecuencia. En análisis multiresolución, la descomposición de señales usando la transformada wavelet discreta puede ser expresada en términos de filtros FIR (*Finite Impulse Response*). Esto nos permite resumir la computación de la DWT en términos de un filtrado simple digital (FIR).

Una vez definidas las wavelet en una dimensión, es esencial describir la extensión a dos dimensiones de la DWT para transformaciones de señales bidimensionales. Una señal digital en dos dimensiones puede ser representada por un array $X[M, N]$ con M filas y N columnas, donde M y N son enteros no negativos. Un enfoque simple para la implementación en dos dimensiones de la DWT es el de realizar la DWT en una dimensión por filas para producir un resultado intermedio y entonces realizar la misma DWT por columnas en este resultado intermedio para producir el resultado final. Esto es posible porque las funciones de escalado en dos dimensiones puede expresarse como funciones separables que son el producto de dos funciones de escalado en una dimensión, tal como $\phi_2(x, y) = \phi_1(x)\phi_1(y)$. Aplicando la transformada unidimensional en cada fila, producimos dos subbandas en cada fila. Cuando las subbandas de baja frecuencia de todas las filas (L) se unen, aparece como una versión reducida (de tamaño $MxN/2$) de la señal de entrada. De manera similar si unimos las subbandas de alta frecuencia de todas las filas se produce la subbanda H de tamaño $MxN/2$, que contiene principalmente la información de alta frecuencia en torno a las discontinuidades (bordes en una imagen) de la señal de entrada. Entonces aplicando la DWT unidimensional por columnas en estas subbandas L y H (resultado intermedio), se producen cuatro subbandas LL ,

LH , HL y HH de tamaño $M/2 \times N/2$ respectivamente, LL es la versión aproximada de la señal original de entrada. LH , HL y HH son las subbandas de alta frecuencia que contienen la información de detalle. Se obtendría el mismo resultado si se aplicara la DWT unidimensional primero por columnas y luego por filas. Considerando que la señal de entrada es una imagen, la subbanda LL_1 puede ser considerado como una versión submuestreada 2:1 (tanto horizontalmente como verticalmente) de la imagen original. Las otras tres subbandas LH_1 , HL_1 y HH_1 contienen la alta frecuencia de la información de detalle. Estas subbandas espacialmente orientadas (horizontal, vertical o diagonal) contienen en su mayoría información de discontinuidades locales en la imagen y la mayoría de energía en cada una de estas tres subbandas esta concentrada en los alrededores de áreas correspondientes a los bordes en la imagen original. Donde LL_1 tiene características espacialmente y estadísticamente similares a la imagen original. Además, ésta puede ser a su vez descompuesta en cuatro subbandas más LL_2 , LH_2 , HL_2 y HH_2 basado en el principio del análisis de multiresolución. En consecuencia, la imagen se descompone en diez subbandas LL_3 , LH_3 , HL_3 , HH_3 , LH_2 , HL_2 , HH_2 , LH_1 , HL_1 y HH_1 después de tres niveles de descomposición [3] (véase Figura 1).

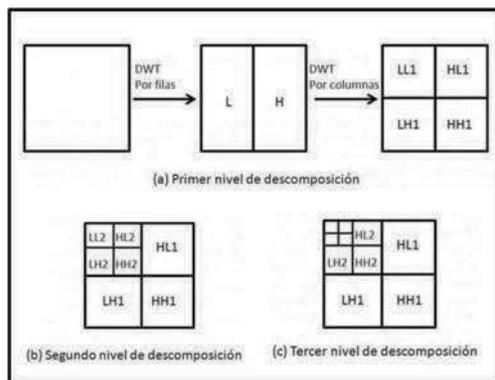


Figura 1: Cálculo Fila-Columna de DWT bidimensional

3. Implementación de las arquitecturas basadas en la Transformada Wavelet

Las arquitecturas propuestas se van a definir divididas en dos partes: En primer lugar se muestra el filtro FIR que se encarga de realizar la Transformada Wavelet para cada caso y en segundo lugar se describe la configuración de cada arquitectura para la implementación total de los diseños.

3.1. Arquitectura Line-Based

El filtro utilizado en este procedimiento es el que se muestra en la Figura 2. Este filtro se conoce como Daubechies 9/7F, es un filtro estándar de convolución y también denomina como implementación directa. Los valores de las constantes del filtro son los siguientes: $h_0 = 0,85269868$, $h_1 = 0,37740286$, $h_2 = -0,11062440$, $h_3 = -0,02384947$, $h_4 = 0,03782846$, $g_0 = 0,78848562$, $g_1 = -0,41809227$, $g_2 = -0,04068942$ y $g_3 = 0,06453888$.

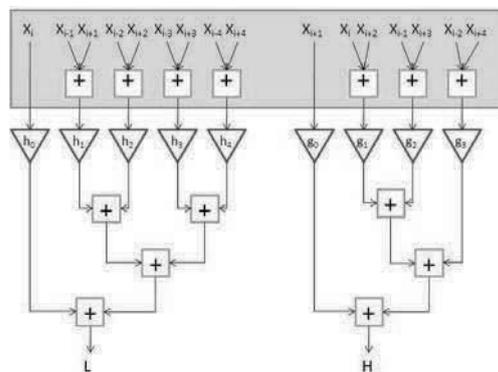


Figura 2: Filtro Daubechies 9/7F

El esquema Line-Based para la realización de la DWT bidimensional se muestra en la Figura 3. Esta configuración se compone de una memoria en línea cuyo tamaño horizontal es el mismo que el tamaño de los píxeles horizontales (ancho) de la imagen, un filtro Daubechies 9/7F *vertical* y otro filtro

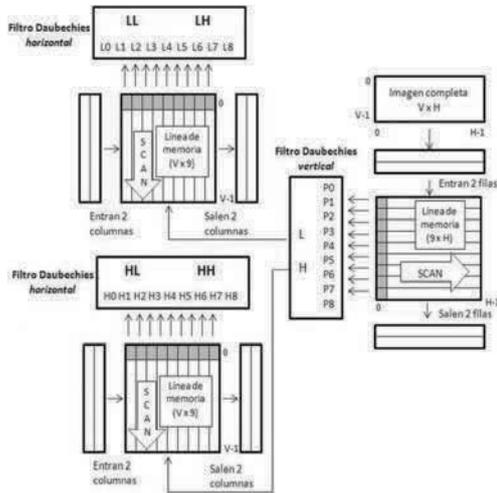


Figura 3: Configuración Line-Based para filtro Daubechies 9/7F

Daubechies 9/7F *horizontal*. La altura de la memoria en línea es de 9 líneas para el caso que nos atañe [4]. La denominación de filtro *vertical* u *horizontal* se debe a la forma de lectura de la memoria de los píxeles de entrada en cada caso, es decir, vertical por columnas y horizontal por filas.

La transformada DWT bidimensional Line-Based para un filtro Daubechies 9/7F se realiza de la siguiente manera [4]:

- Nueve píxeles consecutivos de la imagen en la dirección vertical (de P0 a P8) son la entrada al filtro *vertical* para calcular dos coeficientes wavelet (L, H).
- En este proceso nueve píxeles de cada columna son la entrada al filtro, siendo la salida de éste los coeficientes wavelets resultantes de forma iterativa, hasta que todos los nueve primeros píxeles de todas las columnas son procesados. Entonces se desplaza el buffer, descartando dos de las nueve filas e introduciendo dos nuevas filas de datos. Esta operación continúa hasta que todas las filas son procesadas.
- Los coeficientes L y H serán la entrada al filtro *horizontal*. Nueve píxeles consec-

utivos de los resultados intermedios en la dirección horizontal (de L0 a L8 y de H0 a H8) son la entrada al filtro *horizontal* para generar los coeficientes LL, HL, LH y HH.

- En este proceso nueve píxeles de cada fila son la entrada al filtro, siendo la salida de éste los coeficientes wavelets resultantes, hasta que todos los nueve primeros píxeles de todas las filas son procesados. Entonces se desplaza el buffer, descartando dos de las nueve columnas e introduciendo dos nuevas columnas de datos. Esta operación continúa hasta que todas las columnas son procesadas.

Debido a que la DWT line-based requiere suficiente memoria en línea para dar cabida a los datos asociados con nueve líneas, una gran capacidad de memoria es necesaria si el tamaño de la imagen es grande y, como resultado, los costes en hardware aumentan [4].

3.2. Arquitectura Lifting

El filtro utilizado en la arquitectura Lifting se observa en el diagrama de bloques de la Figura 4. Para los filtros D9/7F que pueden ser descompuestos en cuatro factores lifting, el cálculo se hace en cuatro etapas. El popular filtro D9/7F es un ejemplo de filtro DWT que requiere cuatro pasos lifting con $a = -1,586134342$, $b = -0,05298011854$, $c = 0,8829110762$, $d = 0,4435068522$ y $K = 1,149604398$.

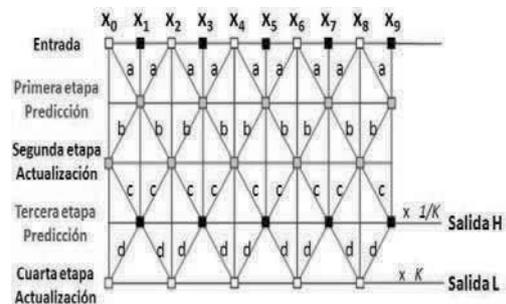


Figura 4: Diagrama con 4 pasos Lifting

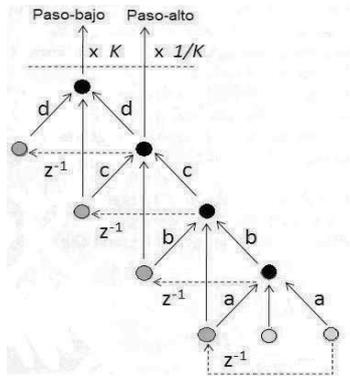


Figura 5: Diagrama de flujo de la arquitectura Lifting D9/7F

Mientras las arquitecturas convencionales lifting requieren menos operaciones aritméticas comparadas con la convolución para realizar la DWT, estas a veces, tienen grandes caminos críticos. Es decir, el camino crítico de la arquitectura lifting es mayor que el de la convolución. Una manera de mejorar esto es mediante la canalización (pipelining). Por tanto, este esquema es el que se adopta en la implementación de la arquitectura Lifting [3]. El correspondiente gráfico de flujo de señales puede verse en la Figura 5.

Para definir el comportamiento de la arquitectura Lifting completa se ilustra en la Figura 6 las funciones básicas con una máquina de estados finita. Como se observa, se utilizan dos memorias, en este caso memorias RAM (*Random-Access Memory*). En la memoria 1 se almacena la imagen original y se leen de ella los píxeles en dirección columnas que se introducen en el filtro de la Figura 5. En cada ciclo de reloj se leen dos muestras, una par y otra impar. Los coeficientes a la salida del filtro se escriben en la memoria 2, una vez la memoria 2 está llena, se leen de ella los píxeles en dirección filas y se introducen en el filtro. Los coeficientes obtenidos corresponden a la primera descomposición wavelet y se almacenan en la memoria 1. Esta operación se realiza de manera iterativa hasta las descomposiciones que se soliciten hayan

finalizado. La imagen comprimida siempre se guardará en la memoria 1 y la imagen de resultados intermedios se almacenará en la memoria 2.



Figura 6: Funcionamiento del diseño completo Lifting

4. Resultados

La implementación de las arquitecturas propuestas se ha realizado mediante la *toolbox* System Generator de Xilinx. Estos diseños han requerido un sistema de control que maneja las entradas y salidas a las memorias y al filtro que realiza la Transformada Wavelet Discreta. Este control se ha llevado a cabo mediante el bloque *black box* de System Generator y permite que se introduzca un archivo VHDL (*Very-High-Speed Integrated Circuit Hardware Description Language*) realizado mediante la herramienta Xilinx ISE 10.1, el cual modela todo el control que se muestra en las Figuras 3 y 6. Los resultados de área y periodo de funcionamiento también lo proporciona la herramienta Xilinx ISE 10.1. Los resultados en imágenes los proporciona Matlab, herramienta que mediante Simulink interactúa con System Generator para la implementación de hardware a nivel de bloques.

En la Figura 7 se muestra el resultado de la transformación de una imagen original, por filas y por columnas y el primer nivel de descomposición al que se llega. De igual manera si la transformada comienza por dirección vertical u horizontal se obtendrá el mismo resultado final.

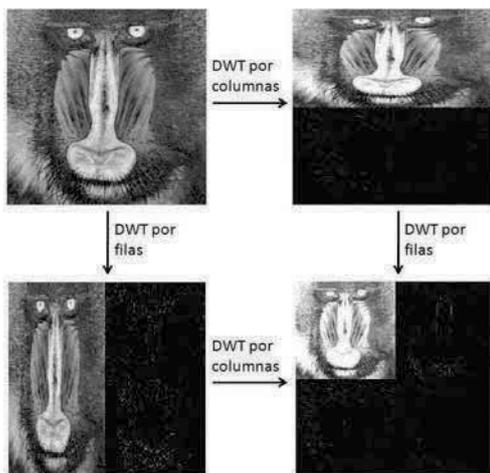


Figura 7: Cálculo DWT de la imagen original por fila-columnas

Una vez finalizados los diseños se ha realizado la comparación mediante una implementación en software y en hardware. En la Tabla 1 se muestra la diferencia en frecuencia, ciclos de reloj y tiempo en segundos necesarios para realizar 2 y 3 descomposiciones wavelet de la imagen original. Como se observa tanto la implementación en software como en hardware el tiempo de realización de la Transformada es mayor en la arquitectura Line-Based. La convolución siempre es más lenta. Por otro lado, se aprecia que a pesar de que los ciclos de reloj de la implementación software siempre son mayores tanto en las diferentes arquitecturas como en las descomposiciones, la frecuencia también es mayor, por lo que los tiempos son menores, esto se traduce en que se tarda menos tiempo en realizar la Transformada de la imagen. Esta ventaja temporal puede conseguirse e incluso superar en la implementación hardware en desarrollos futuros con la elaboración de un

nuevo código VHDL en el bloque de control, que tenga en cuenta la segmentación completa del diseño, puesto que en las arquitecturas implementadas solamente se consideró la segmentación al final del diseño, mediante bloques de retardo en la herramienta System Generator. También cabe destacar que una frecuencia menor se traduce en un menor consumo de potencia lo que conlleva a una ventaja de la tecnología hardware frente a la tecnología software.

DISEÑO LINE-BASED SOFTWARE		Frecuencia: 1805 MHz (0.554 ns)
Niveles de descomposición	Ciclos de reloj	Tiempo en segundos
2	3274830	1.814 ms
3	3544380	1.963 ms
DISEÑO LINE-BASED HARDWARE		Frecuencia: 135.135 MHz (7.4 ns)
Niveles de descomposición	Ciclos de reloj	Tiempo en segundos
2	1971906	14.592 ms
3	2067425	15.299 ms
DISEÑO LIFTING SOFTWARE		Frecuencia: 1805 MHz (0.554 ns)
Niveles de descomposición	Ciclos de reloj	Tiempo en segundos
2	2860524	1.584 ms
3	2994642	1.658 ms
DISEÑO LIFTING HARDWARE		Frecuencia: 51.546 MHz (19.4 ns)
Niveles de descomposición	Ciclos de reloj	Tiempo en segundos
2	155922	3.025 ms
3	161434	3.132 ms

Cuadro 1: Comparación en tiempo hardware-software

5. Conclusiones

En este artículo se han presentado dos arquitecturas hardware de la Transformada Wavelet Discreta sobre imágenes. Dichas arquitecturas se han implementado para imágenes en escala de grises y para un tamaño definido de 256x256 y se han obtenido unos resultados favorables en área y en frecuencia (bajo consumo de potencia). Los posibles desarrollos futuros irían orientados a: Completar el sistema con etapas previas y posteriores a la Transformada Wavelet, como puede ser cuantización y codificación. También se podría desarrollar las mismas arquitecturas para imágenes en color y para tamaños de imagen variable, incluso HD. Se podría trabajar con memoria externa y aprovechar dispositivos hardware más

pequeños en área. Y por último y más importante, como se ha tratado en el apartado anterior, realizar una mayor segmentación en el núcleo de control, el cual supone el cuello de botella en este tipo de diseños, y de esta manera conseguir un periodo de funcionamiento menor que con tecnología software.

Agradecimientos

Thanks to Spanish Ministry of education and Science under grants DPI2007-66796-C03-03 for funding.

Referencias

- [1] Rafael C. González, Richard E. Woods, *Digital Image Processing*, Pearson, 2008.
- [2] Liang-Gee Chen, Chao-Tsung Huang, Ching-Yeh Chen, Chih-Chi Cheng, *VLSI Design of Wavelet Transform*, Imperial College Press, 2007.
- [3] Tinku Acharya, Ping-Sing Tsai, *JPEG2000 Standard for Image Compression*, John Wiley and Sons, 2005.
- [4] Yamauchi et al.: *1440 x 1080 Pixel, 30 Frames per Second Motion-JPEG2000 Codec for HD-Movie transmission*, IEEE Journal of Solid-State Circuits, Vol.40, No.1, January 2005.
- [5] Chunhui Zhang, Yun Long, Fadi Kurdahi, *A hierarchical pipelining architecture and FPGA implementation for lifting-based 2D DWT*, Springer-Verlag, 2007.
- [6] Yang et al.: *A Block-Based Architecture for Lifting Scheme Discrete Wavelet Transform*, IEICE TRANS.FUNDAMENTALS, Vol.E90-A, No.5, May 2007.
- [7] Maurizio Martina, Guido Masera, *Low-Complexity, Efficient 9/7 Wavelet Filters VLSI implementation*, IEEE.
- [8] Keshab K. Parhi, Takao Nishitani *VLSI Architectures for Discrete Wavelet Transforms*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.1, No.2, June 1993.