

ARQUITECTURA FLEXIBLE Y CONFIGURABLE DE LA TRANSFORMADA DISCRETA BIDIMENSIONAL DE WAVELET IMPLEMENTADA EN FPGAS

Daniel Iparraguirre Cárdenas, Chris Tomás Horna, Manuel Paredes Castro
danielux@ieee.org, a19987174@pucp.edu.pe, a20000421@pucp.edu.pe

Grupo de Microelectrónica – Pontificia Universidad Católica del Perú
Av. Universitaria Cdra. 18 S/N – San Miguel – Lima 32, Perú
Telf. (+51-1) 460-2870 ext 304,318 Fax: (+51-1) 2618861

ABSTRACT

The present paper shows an FPGA implementation of the Bidimensional Discrete Wavelet Transform (DWT). The designed architecture is prepared for the computing of the forward transform (FWT) so as the inverse transform (IWT), applying the image standard or non-standard decomposition. The architecture description is made with VHDL, and the result is a design whose compilation parameters are the data width, maximum image size, filterbank order and parallelism degree (multiplier array size). The architecture has been implemented on an Altera's FLEX 10K device, inside which the design architecture takes advantage of the FLEX EABs in order to achieve the implementation of high-order filterbanks without compromising the logic cell consumption. The high flexibility of the architecture allows its application inside processing systems with an ample variety of requirements.

RESUMEN

El presente trabajo muestra la implementación FPGA de una arquitectura para el cálculo de la Transformada Discreta Bidimensional de Wavelet. La arquitectura diseñada está preparada para hacer tanto la transformación directa (FWT) como la transformación inversa (IWT), usando descomposición estándar o no estándar de la imagen. La descripción está hecha en VHDL, resultando un diseño el cual tiene como parámetros para compilación el tamaño de palabra, tamaño máximo de la imagen, orden del banco de filtros y grado de paralelismo (tamaño del arreglo de multiplicadores). La arquitectura ha sido implementada sobre un dispositivo FLEX 10K de Altera, en el cual la arquitectura aprovecha los EABs del FLEX con objeto de hacer posible la implementación de bancos de filtros de alto orden sin comprometer el consumo de celdas lógicas. La alta flexibilidad de la arquitectura permite su aplicación en sistemas de procesamiento de los requerimientos más diversos.

ARQUITECTURA FLEXIBLE Y CONFIGURABLE DE LA TRANSFORMADA DISCRETA BIDIMENSIONAL DE WAVELET IMPLEMENTADA EN FPGAS

Daniel Iparraguirre Cárdenas, Chris Tomás Horna, Manuel Paredes Castro
danielux@ieee.org, a19987174@pucp.edu.pe, a20000421@pucp.edu.pe

Grupo de Microelectrónica – Pontificia Universidad Católica del Perú
Av. Universitaria Cdra. 18 S/N – San Miguel – Lima 32, Perú
Telf. (+51-1) 460-2870 ext 304,318 Fax: (+51-1) 2618861

RESUMEN

El presente trabajo muestra la implementación FPGA de una arquitectura para el cálculo de la Transformada Discreta Bidimensional de Wavelet. La arquitectura diseñada está preparada para hacer tanto la transformación directa (FWT) como la transformación inversa (IWT), usando descomposición estándar o no estándar de la imagen. La descripción está hecha en VHDL, resultando un diseño el cual tiene como parámetros para compilación el ancho del bus de datos, tamaño máximo de la imagen, orden del banco de filtros y grado de paralelismo o número de flujos (tamaño del arreglo de multiplicadores). La arquitectura ha sido implementada sobre un dispositivo FLEX 10K de Altera, en el cual la arquitectura aprovecha los EABs (celdas de memoria) del FLEX con objeto de hacer posible la implementación de bancos de filtros de alto orden sin comprometer el consumo de celdas lógicas. La alta flexibilidad de la arquitectura permite su aplicación en sistemas de procesamiento de los requerimientos más diversos.

1. INTRODUCCIÓN

El desarrollo de una arquitectura para la Transformada de Wavelet está en función del requerimiento del sistema que vaya a usar dicha arquitectura. De acuerdo a ello, existen implementaciones orientadas a lograr un alto grado de velocidad de procesamiento, mientras que otras implementaciones están orientadas a lograr un bajo consumo de recursos. El logro de dichos objetivos es particularmente difícil en un FPGA, ya que la satisfacción de un requerimiento significa sacrificar de manera sensible el otro. Un dispositivo FLEX contiene un número limitado de celdas lógicas, cada una de las cuales puede dar lugar a la implementación de un único registro. Es por ello que la implementación de bancos de filtros de alto orden

conlleva necesariamente a un elevado consumo de celdas lógicas.

La arquitectura diseñada contiene las siguientes características:

- Alta flexibilidad, en la cual el usuario no sólo puede modificar el ancho de palabra y el orden del banco de filtros, sino también el grado de paralelismo del banco, expresado en el número de flujos de datos y multiplicadores a usar en el mismo, con el fin de lograr un cálculo más veloz.
- Relativamente bajo consumo de celdas lógicas, debido a la ausencia de registros de desplazamiento en el banco de filtros. Esto hace que posible en un FLEX de mediana capacidad la implementación de arquitecturas con bancos de filtros de alto orden.
- Fácil configuración, en la cual la misma arquitectura puede efectuar la FWT o la IWT de la imagen, usando descomposición estándar o no estándar.

La arquitectura presentada llega a tener mejores resultados y mayor flexibilidad que las implementaciones presentadas en [2] y [4].

El presente trabajo está presentado como sigue. La sección 2 presenta las justificaciones de la arquitectura global presentada. La sección 3 presenta la arquitectura general así como la descripción de sus bloques conformantes, la sección 4 muestra el procesamiento de datos dentro de la arquitectura presentada, la sección 5 presenta los resultados de compilación sobre el dispositivo FLEX y en la sección 6 aparecen las conclusiones del presente trabajo.

2. FUNDAMENTOS

En todos los casos posibles de transformación (directa o inversa, y descomposición estándar o no-estándar), el proceso de transformación involucra la transformación

unidimensional de cada una de las filas y columnas de la imagen original o transformada, según sea el caso.

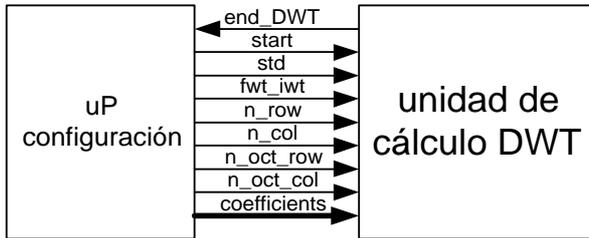


Fig. 1. Componentes y señales de comunicación en la arquitectura de la DWT.

En el procesamiento de la FWT e IWT unidimensional, el bloque principal es el procesamiento de la octava. En base a ello, y tomando en cuenta el análisis realizado en [1] para la implementación del banco de filtros, resulta que la arquitectura de la DWT está basada en un bloque de cálculo principal que efectúa el procesamiento de una única octava unidimensional para una fila o columna.

Para el caso de la FWT, el procesamiento de la octava involucra el filtraje y decimación sucesivas de la secuencia de entrada, generando dos secuencias, una resultado del filtraje paso-alto y otra resultado del filtraje paso-bajo. En base al análisis realizado en [1], el procesamiento descrito es efectuado descomponiendo la secuencia de entrada en dos flujos de datos pares e impares respectivamente, para luego efectuar la convolución de ambas secuencias con los coeficientes de orden par e impar respectivamente, primero los coeficientes paso-alto y luego los coeficientes paso-bajo.

Para el caso de la IWT, el procesamiento de la octava involucra la interpolación de las secuencias de entrada paso-bajo y paso-alto, para su posterior filtraje y adición con los coeficientes correspondientes. En base al análisis realizado en [1], el procesamiento es efectuado mediante la convolución de dos porciones de secuencia paso-alto y paso-bajo respectivamente, con los coeficientes de orden par paso-alto y paso-bajo respectivamente, siendo el resultado la suma de ambas convoluciones. El resultado inmediato posterior resulta trabajando con los coeficientes de orden impar.

La arquitectura de banco de filtros presentada en [1] procesa los datos siguiendo la técnica de multiplexión de productos, en la cual dos multiplicadores-acumuladores llevan a cabo todos los productos y sumas correspondientes al proceso de convolución de la secuencia de entrada con los coeficientes almacenados. En la presente arquitectura, existe un parámetro de compilación (grado de paralelismo) que determina el número de flujos de datos y multiplicadores conformantes del banco de filtros. Esto significa que la secuencia a procesar es distribuida en un número de bloques de

memoria, los cuales alimentan la información a los multiplicadores del banco. De la misma manera, los coeficientes están distribuidos en un número similar de bloques de memoria.

El banco de filtros de la arquitectura trabaja con conjuntos de coeficientes Wavelet ortogonales, de tal manera que los coeficientes del filtro paso-alto se derivan de los coeficientes paso-bajo, de la forma descrita en [5]. De esta manera, la arquitectura guarda únicamente la información de los coeficientes del filtro paso-bajo.

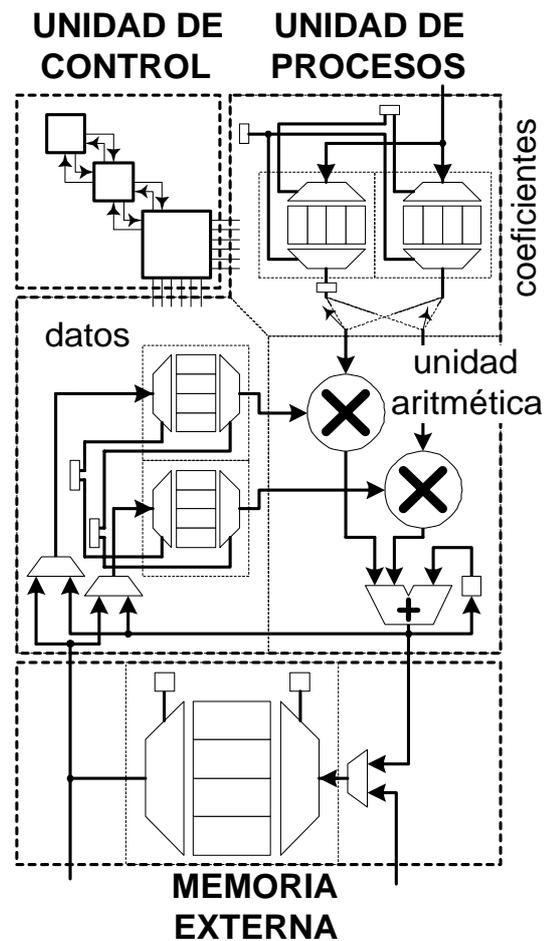


Fig. 2. Estructura de la unidad de cálculo de la DWT.

3. ARQUITECTURA

La arquitectura realizada está mostrada en la fig. 1 e incluye un microprocesador de configuración y la unidad de cálculo de la DWT, los cuales interactúan por medio de señales de control.

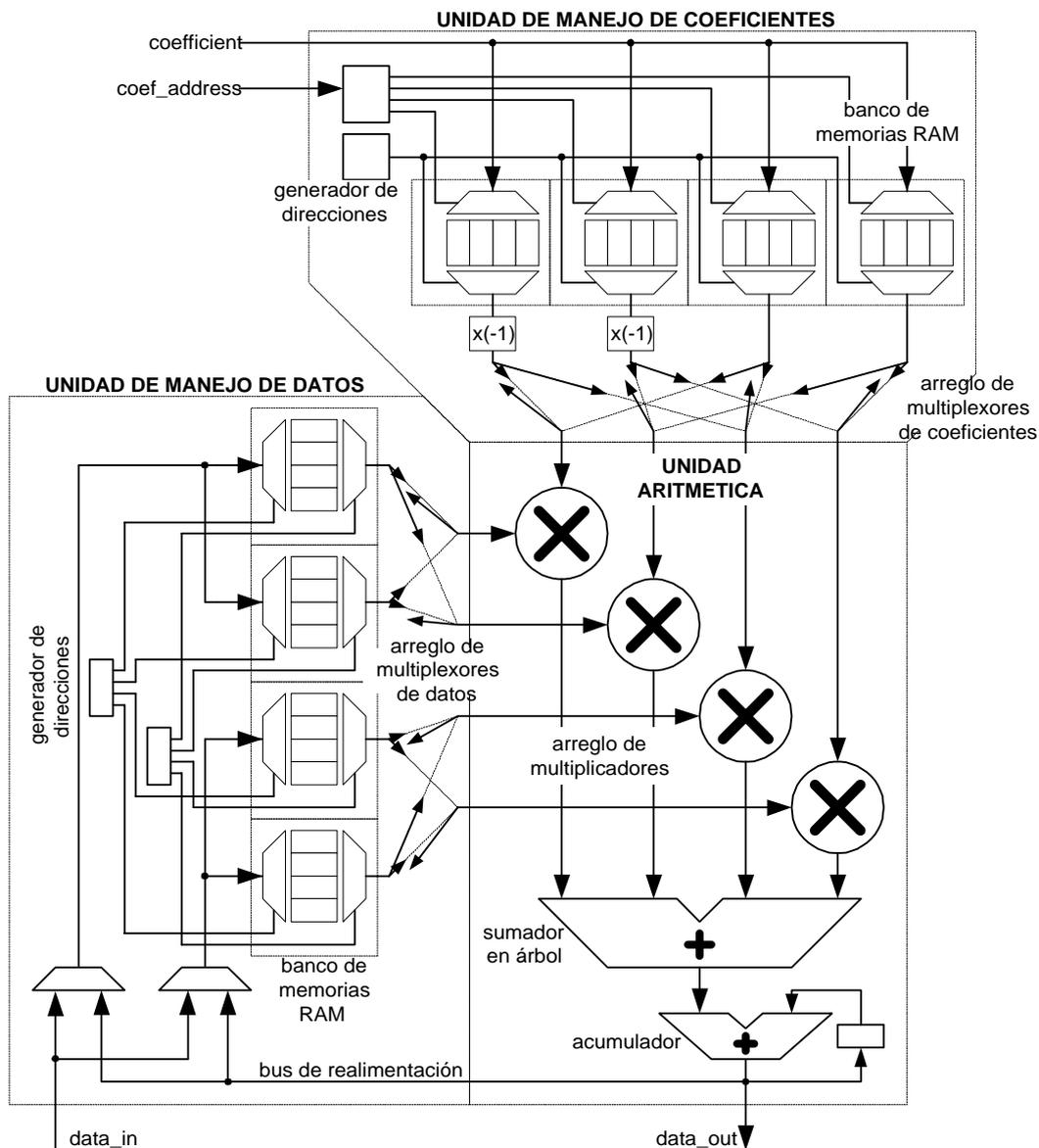


Fig. 3. Estructura de la unidad de procesos, para una configuración de 4 flujos

El microprocesador de configuración está hecho para las labores de carga de los coeficientes a la unidad de cálculo (señal *coefficients*), así como las señales necesarias para el cálculo directo o inverso (señal *fwf_iwt*), por descomposición estándar o no estándar de la imagen (señal *std*), tamaño de imagen, (*n_row* y *n_col*), número de octavas a procesar (*n_oct_row* y *n_oct_col*), e inicio y fin de la transformación (*start* y *end_DWT*). Asimismo, dicho microprocesador interactúa de manera independiente con la memoria externa, para las labores de carga y lectura de imagen a procesar o procesada. La arquitectura del microprocesador es RISC, con objeto de mantener un alto desempeño.

La unidad de cálculo de la DWT está mostrada en la fig. 2 y comprende la unidad de procesos, la unidad de control y la memoria externa para el almacenamiento de la imagen a procesar.

La unidad de procesos está encargada de la transformación octava de una hilera de la imagen (porción de fila o columna), esto es, una sola etapa filtraje-decimación o interpolación-filtraje, según sea FWT o IWT. Dicha unidad está mostrada en la fig. 3 y está dividida en la unidad de manejo de datos, la unidad de manejo de coeficientes y la unidad aritmética. Para el caso de la FWT, la unidad de procesos recibe la hilera a descomponer y la almacena en la unidad de manejo de

datos, la cual alberga a los datos de orden par e impar. Una vez almacenada, la hilera es alimentada a la unidad aritmética para efectuar la convolución de la secuencia con los coeficientes almacenados en la unidad de manejo de coeficientes, siendo el resultado almacenado en la RAM externa y en la unidad de manejo de datos (esto sólo si se trata del componente paso-bajo). Para el caso de la IWT, los componentes paso-bajo y paso-alto a sintetizar son alimentados a la unidad de manejo de datos, para luego ser filtrados por los coeficientes respectivos y sumados, generando la secuencia sintetizada la cual es realimentada a la unidad correspondiente.

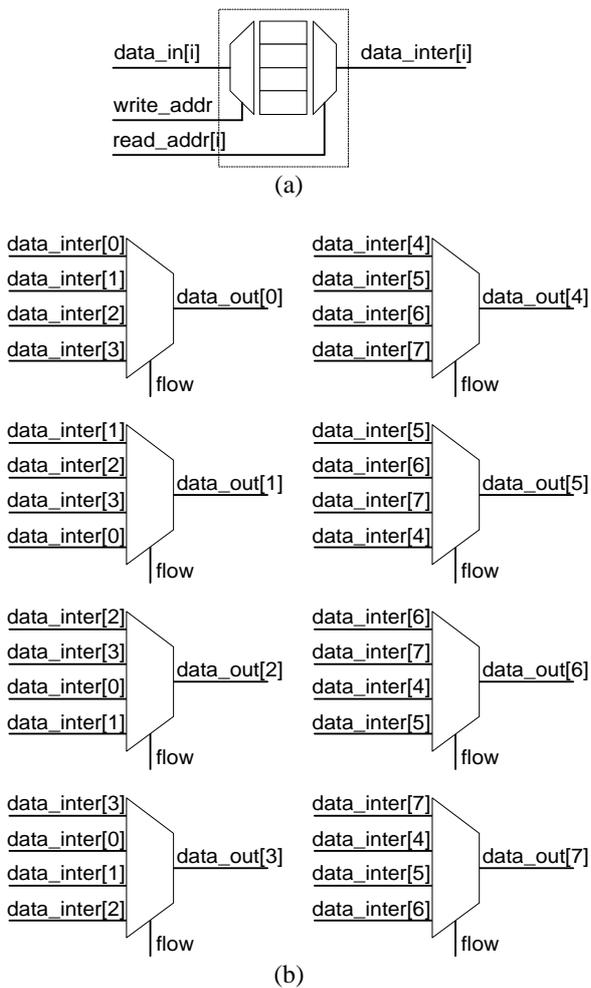


Fig. 4. (a) Banco de memoria y (b) disposición de los arreglos de multiplexores dentro de la unidad de manejo de datos, para una configuración de 8 flujos.

La unidad de manejo de datos contiene dos arreglos de bancos de memoria. Para el cálculo de la FWT, los arreglos albergan los datos de orden par e impar respectivamente, de la secuencia a descomponer. Para el

cálculo de la IWT, los arreglos albergan los componentes paso-bajo y paso-alto de una octava a sintetizar.

A la salida de datos del arreglo de bancos, existe un arreglo de multiplexores, mostrado en la fig. 4(b) (para una configuración de 8 flujos) los cuales hacen posible la lectura de los datos a fin de efectuar la convolución con los coeficientes respectivos.

La unidad de manejo de coeficientes contiene asimismo dos arreglos de bancos de memoria, el primer arreglo destinado al almacenamiento de los coeficientes de orden par y el segundo para los coeficientes de orden impar. Adicionalmente, a la salida de datos de los arreglos de bancos, existe un arreglo de multiplexores y un arreglo de inversores a la salida del segundo arreglo, para la generación de los coeficientes paso-alto a partir de los paso-bajo.

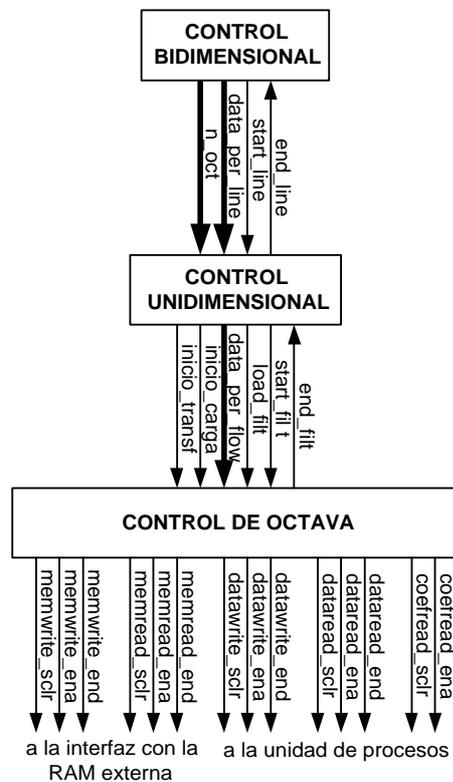


Fig. 5. Bloques y señales de interconexión dentro de la unidad de control

La unidad de control está mostrada en la fig. 5 y se encarga de generar las señales necesarias para el cálculo de la DWT por parte de la unidad de procesos. Está formada por tres bloques que llevan a cabo los tres niveles de control existentes: el control bidimensional, el control unidimensional y el control de octava.

El control bidimensional controla la transformación de la imagen dependiendo si se trata de la FWT o la IWT, en

descomposición estándar o no estándar. Lleva el control del número de filas y columnas que tiene la imagen, y de acuerdo a ello, envía al control unidimensional las señales correspondientes al inicio de transformación de cada fila o columna.

El control unidimensional controla la transformación de una fila o columna de la imagen a procesar. A partir de la señal de inicio dada por el control bidimensional, genera las señales de primera carga de línea, e inicio de transformación de octava para el control de octava, llevando cuenta del número de octavas a procesar.

El control de octava está relacionado con la carga, transformación y almacenamiento de octava, generando las señales correspondientes a la lectura y escritura en la unidad de manejo de datos (unidad de procesos), así como la selección en la unidad de manejo de coeficientes.

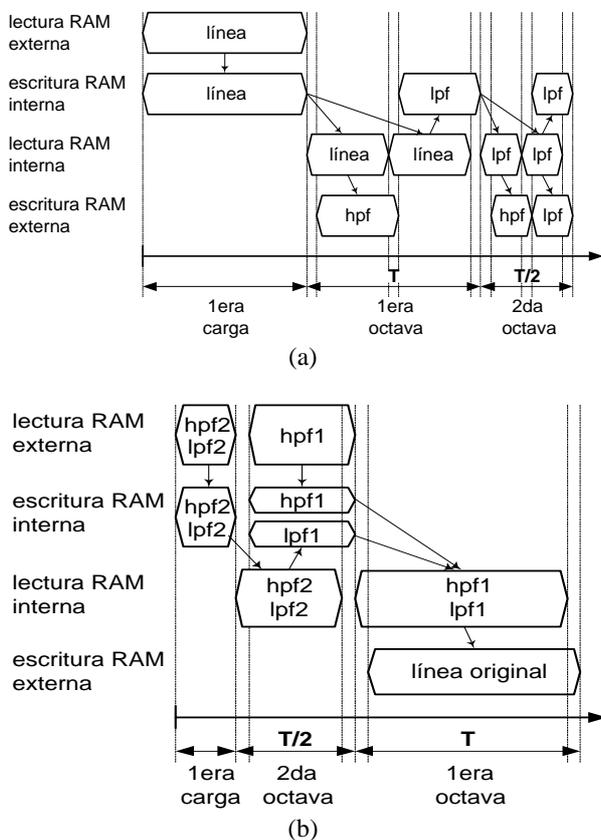


Fig. 6. Diagrama de tiempos para la transformada unidimensional (a) directa, y (b) inversa

5. PROCESAMIENTO DE DATOS

La unidad de procesos está encargada del procesamiento de una octava. En el caso de la FWT, la octava a procesar está almacenada en la unidad de manejo de datos y dividida en datos de orden par e impar, los cuales van en

los dos arreglos de bancos de memoria respectivos. Los datos almacenados son leídos y alimentados a la unidad aritmética, para ser filtrados con los coeficientes provenientes de la unidad de manejo de coeficientes, la cual divide a los mismos en coeficientes de orden par e impar. La lectura de la octava almacenada se lleva a cabo dos veces, para generar primero el resultado paso-alto que es almacenado en la RAM externa, y después el resultado paso-bajo que será realimentado a la unidad de manejo de datos y descompuesto en elementos de orden par y elementos de orden impar. Por tanto, la unidad de procesos es configurada para procesar los datos siguiendo el flujo descrito en la fig. 7(a).

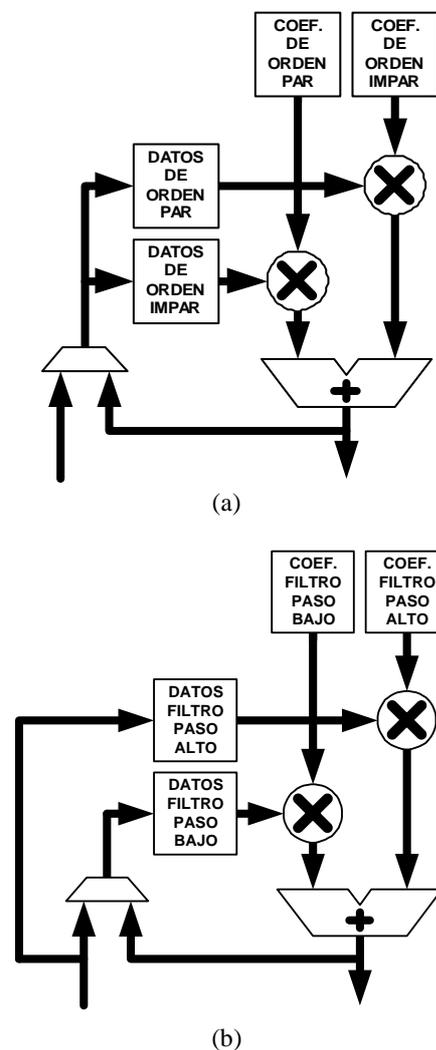


Fig. 7. Flujo de datos dentro de la unidad de procesos, para la (a) FWT, y para la (b) IWT.

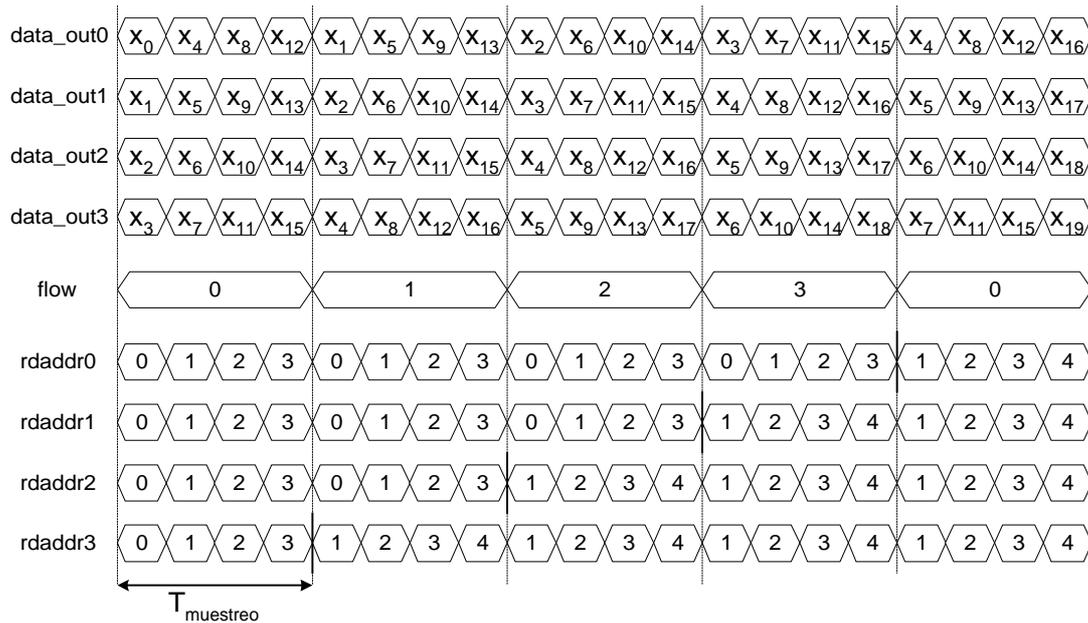


Fig. 8. Generación de direcciones y lectura de datos en la unidad de manejo de datos

En el caso de la IWT, los componentes paso-alto y paso-bajo a sintetizar están almacenados en los arreglos de bancos de memoria respectivos. Dichos componentes son leídos y procesados en la unidad aritmética, la cual recibe los coeficientes provenientes de la unidad de manejo respectiva, divididos en coeficientes paso-alto y paso-bajo. La unidad de procesos lleva el flujo de datos de la forma mostrada en la fig. 7(b).

La lectura de un banco de memoria de datos es realizada en base a un contador principal, que determina el punto de partida del vector a filtrar para la generación de un elemento del resultado, y un contador desplazamiento, que recorre todos los elementos del vector. El recorrido efectuado dura el número de ciclos correspondientes al período de muestreo, dado por el cociente de la división del orden del banco de filtros por el grado de paralelismo. Los contadores empleados son de módulo variable, de tal manera que el retorno de cuenta (del final al inicio o del inicio al final) depende de la longitud de octava a procesar. Esto es para la implementación de la convolución circular de los datos.

Una vez calculado un elemento del resultado, el vector de datos a procesar debe sufrir un desplazamiento. Para ello, el arreglo de multiplexores efectúa una “rotación” del arreglo de bancos, el cual está reflejado en un incremento de la señal de selección *flow* del arreglo de multiplexores mostrado en la fig. 4(b). Esto va junto con el incremento del contador base del banco que resulta trasladado de un extremo a otro del arreglo.

La generación de las direcciones de lectura de los bancos y la señal *flow*, así como los flujos de datos resultantes para cada flujo, están mostrados en la fig. 8, para una

configuración de 8 flujos y orden 32 (16 coeficientes de orden par o impar).

6. RESULTADOS EXPERIMENTALES

La arquitectura ha sido implementada para sus distintas configuraciones sobre un dispositivo EPF10K200S de Altera. La variación del orden del banco de filtros no influye significativamente en el consumo de celdas lógicas, aunque sí en el consumo de celdas de memoria.

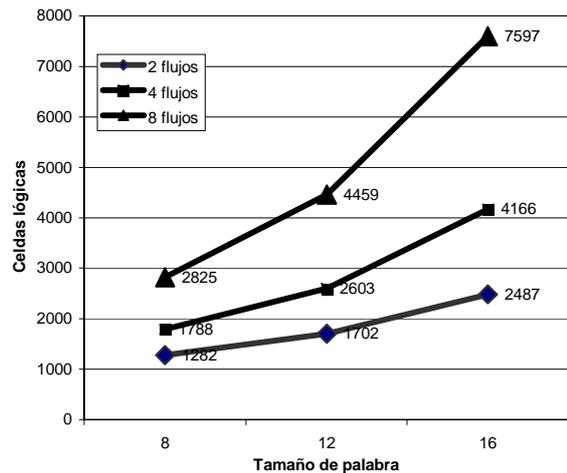


Fig. 9. Consumo de celdas lógicas vs. Tamaño de palabra para una configuración de imagen 1024x1024 y orden 128. La fig. 9 muestra los resultados de consumo de celdas lógicas para una configuración de tamaño de imagen

1024x1024, orden del banco de filtros 128, grados de paralelismo de 2, 4 y 8 flujos y bus de datos de 8, 12 y 16 bits. Para 2 flujos y 8 bits, la arquitectura entra sin inconveniente alguno en un EPF10K30, mientras que para 4 flujos y 16 bits la arquitectura entra en un EPF10K100.

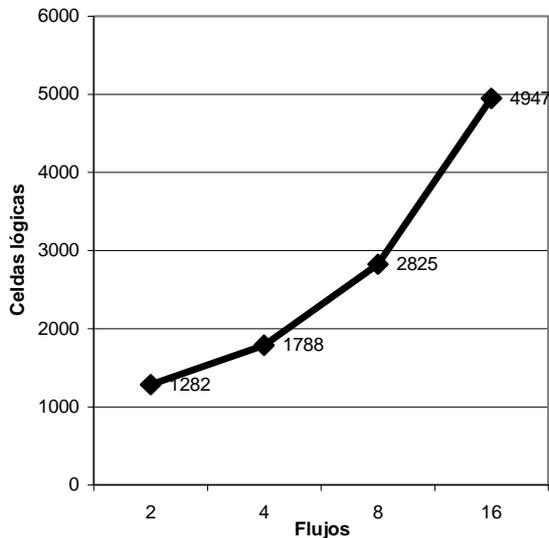


Fig. 10. Consumo de celdas vs. número de flujos para una configuración de 8 bits, imagen 1024x1024 y orden 128.

La fig. 10 muestra los resultados en el consumo de celdas respecto a la variación en el grado de paralelismo, para una configuración de 8 bits. En tal sentido, es posible la implementación de la arquitectura con un alto paralelismo sobre un dispositivo menor EPF10K100, dependiendo del tamaño de imagen y orden del banco de filtros.

La fig. 11 presenta los resultados de desempeño de reloj. Ya que la arquitectura está optimizada para mantener el desempeño independiente del número de flujos, es posible lograr una configuración de alta velocidad sobre un dispositivo de mediana capacidad. Para una configuración de imagen de 256x256, 16 bits, orden 16 y 4 flujos, la arquitectura alcanza un rendimiento de 58 cuadros por segundo, mientras que el consumo es de 4166 celdas lógicas (EPF10K100).

7. CONCLUSIONES

Los resultados presentados en la sección 6 muestran que la arquitectura posee la suficiente flexibilidad para diversas aplicaciones que requieran una configuración de banco de filtros de alto orden, mínimo consumo de celdas o alta velocidad de procesamiento. El desempeño de reloj de la arquitectura depende únicamente del ancho del bus de datos, lo cual permite una rápida determinación de la

configuración a usar dependiendo de los requerimientos iniciales.

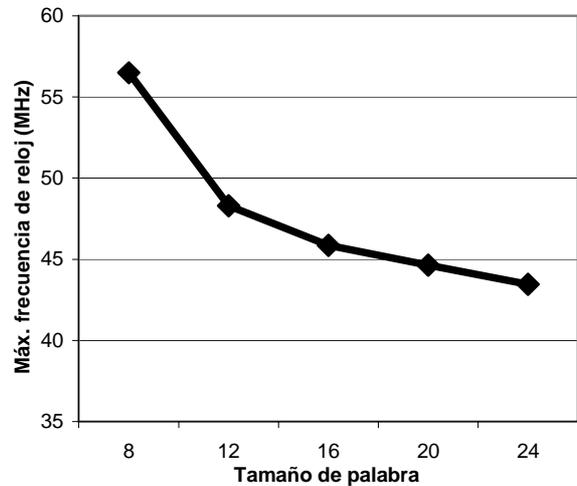


Fig. 11. Desempeño de reloj vs. tamaño de palabra, sobre un dispositivo EPF10K200S

La posibilidad de obtener una configuración de alto orden y bajo consumo hace a la arquitectura idónea para aplicaciones de sonido, en las cuales los requerimientos de velocidad de procesamiento son menores. En tal sentido, es posible la reducción en el consumo de celdas a costa de la velocidad de procesamiento mediante la combinación de técnicas dígito-seriales.

8. REFERENCIAS

- [1] Ing. Daniel Iparraguirre Cárdenas, "Arquitectura de Banco de Filtros para la Transformada Discreta de Wavelet empleando la Técnica de Multiplexión de Productos", *Anales del VIII Workshop Iberchip*, Guadalajara - México, marzo 2002.
- [2] Ing. Daniel Iparraguirre Cárdenas e Ing. Gerard Franz Santillán Quiñonez, "Implementación de la Transformada Discreta de Wavelet de una y Dos Dimensiones usando la Técnica Dígito-Serial", *Tesis de Licenciatura, Pontificia Universidad Católica del Perú*, Abril 2001.
- [3] Daniel Iparraguirre Cárdenas y Gerard F. Santillán Quiñonez, "Diseño de un banco de filtros dígito-serial para la implementación de la transformada discreta de wavelet", *Anales del VI Workshop Iberchip*, págs. Del 286 al 291, Sao Paulo, Marzo 2000.
- [4] Gustavo Bellora y Jean-Pierre Deschamps, "Diseño de una Arquitectura Hardware para el Cálculo de una Transformada Wavelet", *Anales del VIII Workshop Iberchip*, Guadalajara - México, marzo 2002.
- [5] Gilbert Strang, "Wavelets and Filter Banks", 1996.
- [6] Altera Data Book, 1999.