Implementation of 2-D Discrete Wavelet Transform for Real-Time Video Signal Processing

RICARDO J. COLOM[1], RAFAEL GADEA[1], ANGEL SEBASTIA[1], MARCOS MARTINEZ[1], VICENTE HERRERO[1], VICENTE ARNAU[2]
[1]Departamento de Ingeniería Electrónica
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia
SPAIN
[2]Departamento de Informática
Universidad de Valencia
Av. Vicente Andrés Estellés s/n, 46100 Burjasot, Valencia
SPAIN
rcolom@eln.upv.es http://dsd.upv.es

Abstract: - This paper presents the architecture and implementation of a two-dimensional Discrete Wavelet Transform (2-D DWT) on a FPGA. This architecture works in a non-separable fashion using a parallel filter structure with distributed control to compute all the DWT resolution levels, so that the input sample can be processed at the rate of one sample per clock cycle. For the computation of an $N \times N$ still image with a filter length L, $N^2 + N$ clock cycles and $6N$ memory storage cells are required. Some of the most used image compression filters have been studied, emphasising the number of bits necessary to carry out a physical implementation of the Wavelet.

Key-Words: - FPGA, Wavelet, Biorthogonal, Digital Image Processing and VHDL.

1 Introduction

The Discrete Wavelet Transform has received considerable attentions in the field of image processing due to its flexibility in representing non-stationary image signals and its ability in adapting to human visual characteristics. It is closely related to multiresolution analysis and subband decomposition, which has been successfully used in image processing for a decade. This transform method describer was developed by Mallat [12], offers orthogonality and leads to multigrid representation. Many VLSI architectures for 1-D DWT have been proposed and implemented. Although 2-D DWT can be a direct extension of 1-D DWT, it is still difficult to design an efficient architecture with a low hardware cost and high throughput.

Basically, the 2-D DWT can be classified into separable and non-separable design approaches. In a separable design approach, the transform operations can be divided into row and column processes so that processes can be individually performed by the architecture of 1-D DWT. Therefore, Lewis and Knowles [13] proposed a simple architecture for the special case of Daubechies DWT, but it did not work efficiently for other wavelets. Vishwanath [4] constructed two systolic array filters, two parallel computation filters and a storage unit to complete the 2-D transform. Chakabarti [6] designed an architecture with regular structure by using linear systolic arrays. A parallel pipelined VLSI array architecture for the 2-D DWT was designed by Chuang and Chen [14]. Ming-Hwa [11] presented VLSI architecture with lower hardware costs and less memory for separable 2-D DWT. In non-separable design approaches, Chakabarti [6] used parallel filters and lots of shift registers to achieve a fast 2-D DWT. Chu Yu [5] used parallel-systolic filter structure and many registers to achieve the 2-D DWT.

In this paper, we design the architecture and implement a FPGA for the 2-D DWT decomposition, using a parallel-systolic filter structure with distributed control to compute all the resolution levels of DWT's.

This paper is organised as follows. Firstly, a brief review of the 1-D DWT and 2-D DWT is given in Section 2. A study of filters that can be used for the DWT are discussed in Section 3. Section 4 presents the architecture for 2-D DWT. Section 5 describes the FPGA implementation for the proposed DWT architecture. Performance and comparisons of various DWT architectures are discussed in Section 6. Finally, concluding remarks are given in Section 7.

2 Discrete Wavelet Transform

The Wavelet Transform is used for the compression of images and audio. In the process of analysing the wavelet, signals are represented using a group of basic functions produced by the displacement and scaled by a main function. The transformed wavelet is a decomposition of a signal in frequency.

The one-dimensional Discrete Wavelet Transform recursively decomposes the input signal, $S_0(n)$, into approximation and detail at the next lower resolution. Let $S_i(n)$ and $W_i(n)$ be the approximation and detail, respectively, of the signal at level i. The approximation of the signal at level $i + 1$ is computed using

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

and the detail of the signal at level $i + 1$ is computed using

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

The equations (1) and (2) describe the computation of DWT. This technique for computing the DWT is often referred to as the pyramid algorithm or Mallat's algorithm [12]. The three-level 1-D DWT computation is shown in Fig.1.

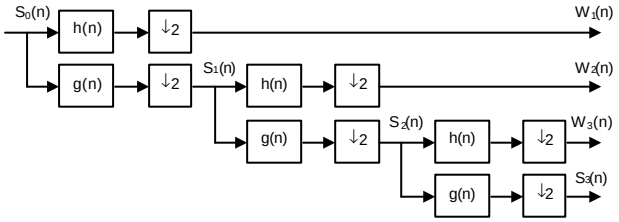

Fig.1. Block diagram of the DWT analysis filter banks.

The two-dimensional DWT has applications in multi-resolution analysis, computer vision, and image compression. The two-dimensional DWT operates on a 2-D signal, such as images. While 1-D filters are used to compute the 1-D DWT, the 2-D DWT uses 2-D filters in its computation. These 2-D filters may be separable or non-separable, where a 2-D filter $f(n_1,n_2)$ is separable if it can be written as $f(n_1,n_2)=f_1(n_1)f_2(n_2)$. The separable 2-D DWT decomposes an approximation image $S_i(n_1,n_2)$ into an approximation image and three detailed images according to

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

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

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

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

where $H(z)$ and $G(z)$ are 1-D wavelet filters. The signal $S_{i+1}(n_1,n_2)$ is an approximation of $S_i(n_1,n_2)$ at a lower resolution. This approximation is computed from $S_i(n_1,n_2)$ by lowpass filtering and decimating by 2 along its rows and columns. The signals $W_{i+1}^1(n_1,n_2)$, $W_{i+1}^2(n_1,n_2)$, and $W_{i+1}^3(n_1,n_2)$ contain the detail of $S_i(n_1,n_2)$. The one-level 2-D DWT computation is shown in Fig.2.

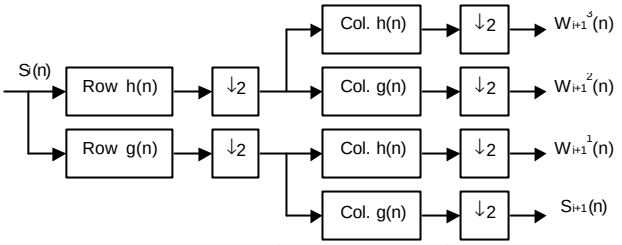

Fig.2. Block diagram of the analysis filter bank used to compute the 2-D DWT.

In contrast to the separable filters, the non-separable filters directly decompose an image into four resultant sub-images without column and row DWT's. Fig.3 illustrates the architecture of 2-D non-separable DWT with one-resolution levels. Note that each block in this architecture is a 2-D filter down-sampled by 2. Fig.4 illustrates the octave-band decomposition of an image using 2-D DWT.

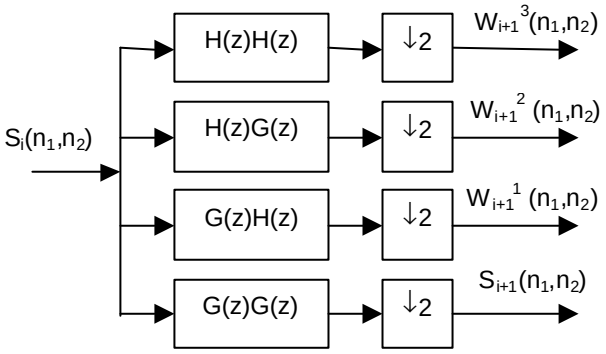

Fig.3. Block diagram of the analysis filter bank used to compute the 2-D non-separable DWT.

Fig.4. (a) Octave-band decomposition of an image. (b) Lena-image decomposition.

Chakrabarti and Vishwanath [7] proposed the 2-D non-separable DWT VLSI architecture with a parallel-filter structure to improve the disadvantages of the separable filters. The hardware components of this architecture include $2L^2$ programmable multipliers, $2(L^2-1)$ adders, $2NL$ memory storage, and a control unit, where L is the filter length and N is the row size of an image. The parallel-filter structure utilises the modified recursive pyramid algorithm in 2-D to construct the whole 2-D DWT architecture, such that a computation time can be achieved of around N^2. Since this architecture adopts non-separable filters, it needs no transpose memory between row and column DWT's. As a result, it reduces a large amount of memory space and latency time. However, this architecture holds a relatively large number of hardware components and complicated routing (proportion to the square of the filter length) so that its hardware cost is huge for the 2-D non-separable DWT realisation.

3 Study and Analysis of Filters for the DWT

The Wavelet Transform known as Daub-4 is performed using a four coefficients FIR filtering unit. The filter G(z) (low pass) used in the DWT analysing process, has the following coefficients:

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

As a well-known fact in sub-band filtering, symmetry of filters and exact reconstruction of the signal are incompatible concepts if the same filter is been used in reconstruction and decomposition. However, it is still possible to obtain symmetry in the filters using biorthogonals coefficients instead of the orthogonal ones. The biorthogonals coefficients under study are presented in the following table.

Filters	Coefficients
CDF 3/1	$G(z)=\frac{\sqrt{2}}{4}\left(-z^{-1}+3+3z-z^{2}\right)$ $G_S(z)=\frac{\sqrt{2}}{8}\left(z^{-1}+3+3z+z^{2}\right)$
CDF 2/2	$G(z)=\frac{\sqrt{2}}{8}\left(-z^{-2}+2z^{-1}+6+2z-z^{2}\right)$ $G_S(z)=\frac{\sqrt{2}}{4}\left(z^{-1}+2+z\right)$
CDF 9/7	$G(z)=\frac{\sqrt{2}}{64}\left(z^{-4}-8z^{-2}+16z^{-1}+46+16z-8z^{2}+z^{4}\right)$
	$G_S(z)=\frac{\sqrt{2}}{32}\left(-z^{-3}+9z^{-1}+16+9z-z^{3}\right)$

Once the algorithms in Matlab have been tested and verified, we carried out an initial implementation using a hardware description language like VHDL. This has shown to be useful for studying their behaviour with integer arithmetic.

In Table 1 we show the results of our study. The following information for each filter can also be found: signal-noise ratio; number of bits used in coefficient quantization; maximum number of necessary bits for the representation of the DWT after the 4 octaves; maximum number of necessary theoretical bits for the realisation of the internal arithmetic operations of the DWT; and maximum number of bits necessary in practice for the realisation of the internal arithmetic operations of the DWT without information loss.

4 Proposed Architecture for 2-D DWT

4.1 Direct Approach

A straightforward implementation of the 2-D non-separable DWT is the result of applying the pyramid algorithm. It essentially consists of four 2-D filters module (HH, HG, GH and GG) which are used repeatedly in the manner shown in Fig.5. The number of clock cycles needed to compute the $N \times N$ pixel DWT's with one level is $4N^2$.

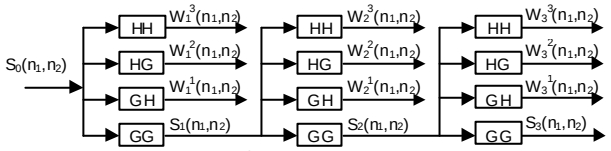

Fig.5. Architecture of a 2-D non-separable DWT with three levels

4.1 Parallel Even-Odd Architecture

To reduce the number of clock cycles we propose a 2-D non-separable filter architecture with a parallel even-odd structure. Fig.6 illustrated the block diagram of one 2-D non-separable filter with a parallel even-odd structure. It is composed mainly of two filter units, one multiplexor, and a simple controller. The even filter calculates the output of even pixel DWT's (output C) and the odd filter calculates the output of odd pixel DWT's (output D). The inputs A and B are, respectively, the even and

odd pixels of the input image. This structure computes two input pixels in parallel.

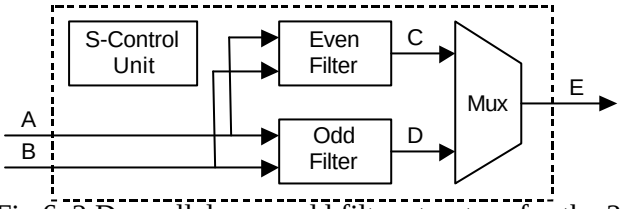

Fig.6. 2-D parallel even-odd filter structure for the 2-D non-separable DWT

This structure has a latency of eight clock cycles and a throughput of four clock cycles. Therefore, the number of clock cycles needed to compute the $N \times N$ pixel DWT's with one level is N^2+4. Thus, the number of clock cycles needed to compute the $N \times N$ pixel DWT's with three levels is:

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

But, we do not need to wait until the first level finishes to begin computing the second level. We can synchronise the computation of three levels and reduce the number of clock cycles to $N^2+4N+12$. We use storage units to connect the different levels. The storage units are dual port memories, and we store only six rows of the DWT.

4.3 Recursive Architecture

To conclude we have folded the architectures shown in Fig.5 and used recursive architecture. There is a filter unit to compute the first level and others to compute the second and third levels. Fig.7 shows the recursive architecture for a 2-D non-separable DWT with three levels. This architecture uses the filter unit described in Section 4.2 and shown in Fig.6. The number of clock cycles needed to compute the $N \times N$ pixel DWT's with three levels is $N^2+4N+12$.

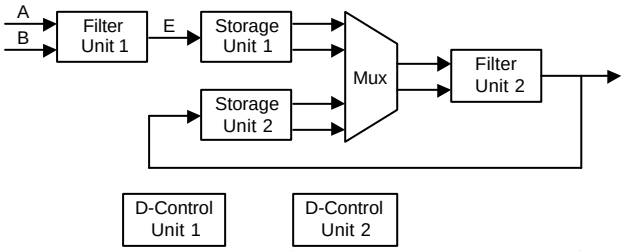

Fig.7. Block diagram of recursive architecture for 2-D non-separable DWT with three levels

Fig.8 shows the timing diagram for three resolution levels of computation in the 2-D non-separable DWT architecture. The first level (filter unit 1) this continuously computing. The second level (filter unit 2) begins when the first level has generated the first four rows. The second level computes when that the first level generates two new rows.

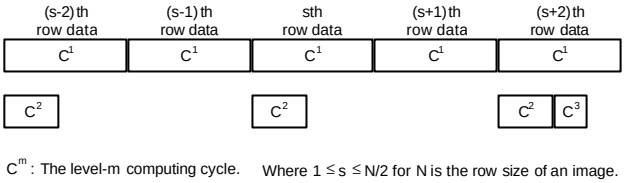

Fig.8. Timing diagram of the 2-D DWT recursive architecture.

4.4 Control Unit

To control the architecture shown in Fig.7, we propose a distributed control with two different control units. A simple control unit in the filter unit (S-control unit in Fig.6), and a complex control unit (D-control unit in Fig.7). The D-control unit controls the connection of data between one level and the following level. Thus, two D-control units they are needed to make a 2-D DWT with three levels. Fig.9 shows how the D-control unit is used in an unfolded architecture for 2-D DWT with three levels.

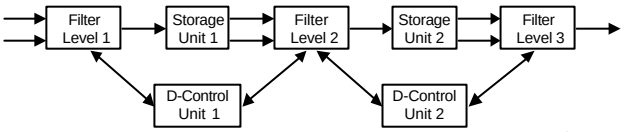

Fig.9. As using the distributed control in an unfolded architecture

The D-control unit communicates with the S-control unit of the filter unit. Therefore, the filter unit 2, shown in Fig.7, needs two S-control units one to communicate with the D-control unit 1, and another to communicate with the D-control unit 2.

Distributed control makes the architecture totally modulable and scalable. To increase the number of levels, it is only necessary to copy the D-control unit and the S-control unit.

5 FPGA Implementation

Following the proposed architecture as described in Section 4.3, we have implemented the 2-D non-separable DWT on an FPGA Xilinx XCV600E. The block diagram of the chip with 4-taps and 3-levels is shown in Fig.7. It is mainly composed of two storage units, two filter units, and two D-control units. The filter unit was shown in Section 4.2 and Fig.6.

The storage unit is a dual port block memory, which has two independent access ports permitting shared access to a central pool of memory. The data width and memory depth of each access port are independently configured. Access port A (write only) takes P-bit data value and a Q-bit address; and access port B (read only) takes 2P-bits data value and a (Q-1)-bit address. Since each port is accessing the same quantity of memory, data words formatted appropriately for port B will be equivalent to two data words formatted for port A. A single memory access on port B is equivalent to two accesses made to port A. The storage unit 1 has a memory depth of 8N/2 words, and the storage unit 2 has a memory depth of 8N/4 words. These storage units are Xilinx cores.

The implementation process is as follows. Firstly, we use Matlab simulations to validate the 2-D non-separable DWT. The simulation is also used to generate test patterns. Secondly, we use VHDL simulation for developing behavioural and structural VHDL code. A VHDL synthesiser produces a digital circuit that can be simulated for correctness. Another tool, called partitioning, placement and routing (PPR) tool, then generates FPGA bitstream that can be used for both static time analysis as well as downloading to an FPGA demoboard for real-time validation. Based on the resulting configuration of FPGA resources, it is possible to estimate the maximum clock cycle achievable. This clock cycle, in turn, determines the data throughput. The performance data of the FPGA, is summarised in Table 2.

Table 2. Performance Data

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

6 Performance and Comparisons of DWT Architectures

In this section, we examine the performance of the 2-D DWT architectures. The performance comparisons of our and another architecture, are summarised in Table 3. Our 2-D non-separable DWT architecture consists of regular 2-D filters, and so needs only 6N of memory storage, 8 multipliers, and 8 adders. When comparing our architecture to the other 2-D DWT, it is worth remembering that ours is the only one implemented in a FPGA. Our architecture processes video in real-time, using less storage units that [8].

Table 1. Results of the study of filters.

Filter	S/N (dB)	Bits Coefficients	Max. Bits in 4 Octaves	Max. Bits for the Theoretical Operations	Max. Bits for the Practical Operations
Daub-4	26.95	9	13	31	29
Daub-4 M.	26.91	9	9	27	25
CDF 3/1	52.96	4	14	22	19
CDF 3/1 M.	42.52	4	10	18	15
CDF 2/2	52.02	4	13	21	18
CDF 9/7	56	7	13	27	24

Table 3. Comparison between 2-D architectures

Architecture	Ours	[4]	[5]	[6]	[11]	[8]
Input order	Serial-Parallel	Parallel	Parallel	Parallel	Parallel	Parallel
Computing approach	Non-separable	Separable	Non-separable	Separable	Separable	Non-separable
Real-time Video Processing	Yes	Not	Not		Not	Yes
No. Of *	8	16			10	
No. Of +	8	16	3		10	2048

Architecture	Ours	[4]	[5]	[6]	[11]	[8]
MACs		8	8	24		
Storage size (byte)	3072	4096	3584	3968	896	7680
Frequency (MHz)	45		55		25	50
Computing Time ≈	N^2+N	N^2+N	N^2+N	N^2+N	N^2+N	

7 Conclusions

A fast and efficient architecture and an FPGA implementation for the 2-D DWT decomposition has been described in this paper. The 2-D non-separable DWT architecture is, in general, very complicated, and so it was not widely implemented in an FPGA. In this paper, we present a 2-D non-separable, parallel, and recursive DWT architecture, which uses distributed control, little storage memory, latency and throughput. Moreover, this 2-D architecture can be easily cascaded into a larger tap size of filters, scaled up with the resolution levels, and implemented in an FPGA. In addition, the designed FPGA can process digital video in real-time.

References:

[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-Frecuency Localization Operators: Ageometric 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 2-D 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 2-D 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.

[11] Ming-Hwa Sheu, Ming-Der Shieh and Sheng-Wel Liu. "A VLSI Architecture Design With Lower Hardware Cost and Less Memory for separable 2-D Discrete Wavelet Transform" IEEE,1998.

[12] S.G. Mallat. "Multifrequency Channal Decompositions of Images and Wavelet Models" IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 37, No 12, pp. 2091-2110, December 1989.

[13] A.S. Lewis y G. Knowles. "VLSI Architecture for 2-D Daubechies Wavelet Transform Without Multipliers" Electronics Letters, Vol. 27, No. 2, pp. 171-173, January 1991.

[14] Henry Y. H. Chuang and Ling Chen. "VLSI Architecture for Fast 2-D discrete Orthonormal Wavelet Transform" Journal of VLSI Signal Processing, Vol. 10, No. 3, pp. 225-236, August 1995.