Convolution algorithm for implementing 2D Discrete Wavelet Transform on the FPGA

I.SLIMANI[1], A.ZAARANE[1], A.HAMDOUN[1]
[1]Laboratoire du Traitement de l'Information
Faculté des Sciences Ben M'sik, Université Hassan II, Casablanca
ibtissamslimani7@gmail.com, z.abdelmoghit@gmail.com

Abstract. Nowadays, the two dimensional discrete wavelet transform 2D-DWT has become a very important tool in the field of image processing. Several architectures for implementing DWT are proposed. In this paper we present an architecture for implementing the DWT based on the convolution algorithm. This implementation is fully described in VHDL and implemented in an FPGA component.

Keywords: image processing, Field Programmable Gate Arrays (FPGAs), discrete wavelet transform (DWT), VHSIC Hardware Description Language (VHDL).

I. INTRODUCTION

NOWADAYS, implementing digital signal processing algorithms on field programmable gate arrays (FPGAs) becomes a growing trend, for the reason that FPGAs have merit on merging digital signal processing algorithms with other control logic. Discrete wavelet transform (DWT) is a classical signal transform algorithm, which is increasingly applied on many areas, such as signal instantaneous analysis, image edge detection, image denoising, and pattern recognition. It has also become a basic tool of many new compression standard image, such as JPEG2000. Implementations of the wavelet transform is based on the convolution algorithms [1]. Several architecture are proposed to calculate the DWT [3, 4, 5, 6, 7]. In this paper, FPGA implementation of two dimensional discrete wavelet transform using Mallat algorithm is introduced. We have proposed an architecture for implementing the two dimensional convolution algorithm. We tested the proposed architecture on an image of 128x128 pixels.

II. DISCRETE WAVELET TRANSFORM

A. *Mallat Algorithm*

The Mallat algorithm is proposed in 1988 [2] as a fast algorithm for the discrete wavelet transform. As the main theory behind the hardware implementation, it is widely used in DWT, as the fast Fourier transform what used conventional Fourier analysis.
The principle of the algorithm is to divide the image into four images at each iteration three blocks on the details of the image and the fourth is the approximation corresponding to the most important information for the eye (low frequencies) which is the basis for the next iteration.

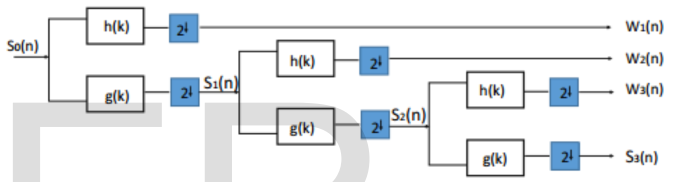


Fig.1: Block diagram of the DWT analysis filter banks

The one dimensional discrete wavelet transform decomposes the input signal $S_0(n)$ into two signals at multiple levels: the approximation $S_i(n)$ and the detail W_i (n) relating to the level i [3]. The approximation of the signal at level i + 1 is calculated using the convolution of the input signal $S_i(n)$ and the filter g(k):

$$S_{i+1}(n) = \sum_{k=0}^{L-1} g(k) S_i(2n-k)$$

The detail signal at level i + 1 is calculated using the convolution of the input signal Si (n) and the filter h (k):

$$W_{i+1}(n) = \sum_{k-0}^{L-1} h(k) S_i(2n-k)$$

g (k) and h (k) are respectively the coefficients of the low-pass and high-pass and L is the filters size. The low-pass filter provides the approximation (low frequency) and high-pass filter provides the details (high frequencies).

B. *The* two Dimensional *Discrete Wavelet Transform*

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 no-separable. A 2-D filter $f(n_1, n_2)$ is separable if it can be written as $f(n_1, n_2) = f_1(n_1)f_2(n_2)$, where $f_1(n_1)$ and $f_2(n_2)$ are 1-D filters. The separable 2-D DWT decomposes an approximation image $S_i(n_1,n_2)$ into an approximation image and three details images according to:

$$S_{i+1}(n_1, n_2) = \sum_{k_1=0}^{L-1}\sum_{k_2=0}^{L-1} g(k_1)g(k_2)S_i(2n_1 - k_1, 2n_2 - k_2)$$

$$W^3_{i+1}(n_1, n_2) = \sum_{k_1=0}^{L-1}\sum_{k_2=0}^{L-1} h(k_1)h(k_2)S_i(2n_1 - k_1, 2n_2 - k_2)$$

$$W^2_{i+1}(n_1, n_2) = \sum_{k_1=0}^{L-1}\sum_{k_2=0}^{L-1} h(k_1)g(k_2)S_i(2n_1 - k_1, 2n_2 - k_2)$$

$$W^1_{i+1}(n_1, n_2) = \sum_{k_1=0}^{L-1}\sum_{k_2=0}^{L-1} g(k_1)h(k_2)S_i(2n_1 - k_1, 2n_2 - k_2)$$

The 2-D DWT can be separable or no-separable design [4]. The separable design can be calculated by applying the 1D DWT along the rows and columns of the input image of each level in steps of horizontal and vertical filtering as shown in Figure 2. For the no-separable design can be calculated by applying 2D filters as shown in fig.3.

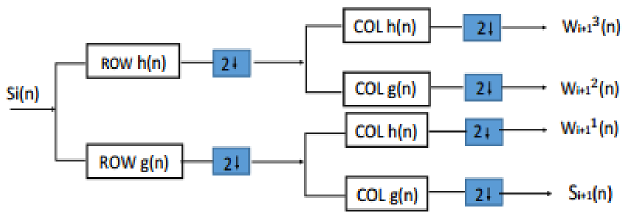


Fig.2: Block diagram of the separable 2-D DWT.

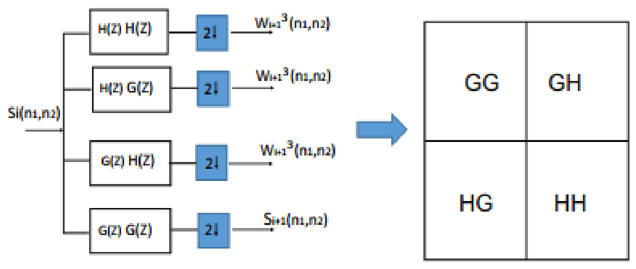


Fig.3: Block diagram of no-separable 2D DWT

III. Proposed Architecture of the 2D Convolution Algorithm

The convolution is a mathematical operator that is used to multiply matrices between them. It is widely used for performing the image filtering operation. For us, we want to apply a two-dimensional filter to our image to achieve no-separable 2D-DWT. In our case, we are giving two very different matrices: the image array and the filter array.
We can extend the two dimensions convolutional by performing two successive sums and adding an index for the second dimension. The pixel results S_{i+1} of the image S_i multiplied by the filter g is thus expressed:

$$S_{i+1}(n_1, n_2) = \sum_{k_1=0}^{L-1}\sum_{k_2=0}^{L-1} g(k_1)g(k_2)S_i(2n_1 - k_1, 2n_2 - k_2)$$

$$= \sum_{k_1=0}^{L-1}\sum_{k_2=0}^{L-1} g(k_1, k_2)S_i(2n_1 - k_1, 2n_2 - k_2)$$

To implement the algorithm above in an FPGA component, we used three memories, the first to record the original image, the second to save the filter array and the last to save the resulting image. Concerning the addressing of each of these memories, we used two counters, one counter for rows and the other for columns. We also used a multiplier and an adder for performing the convolution operation. To control and synchronize all of these modules we have designed a state machine that manages the entire system (see Fig.4).

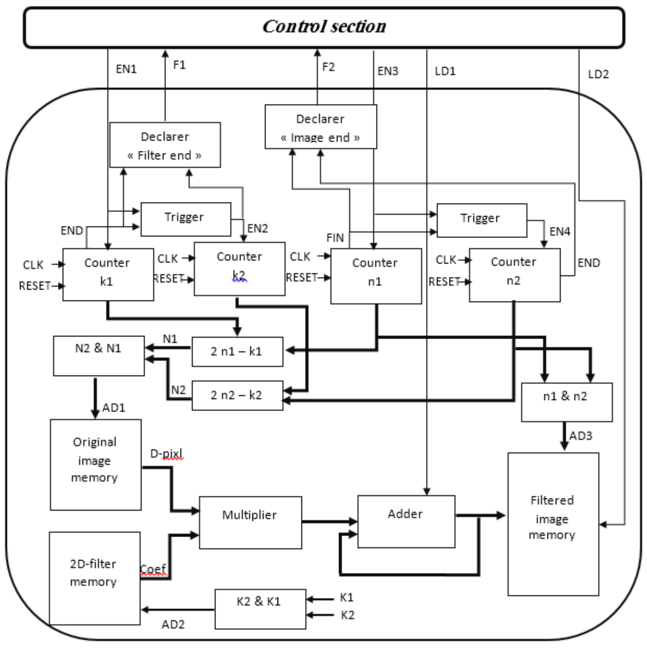
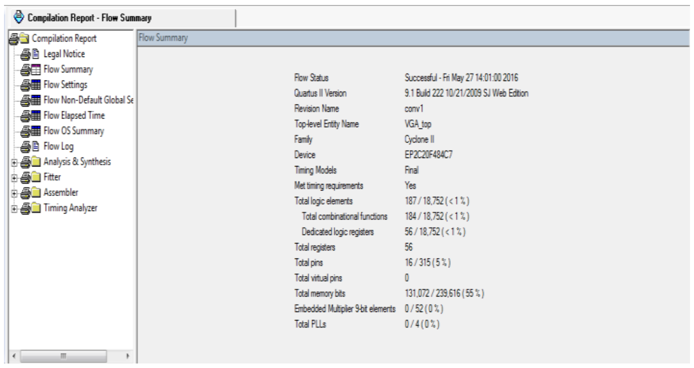


Fig.4: Our architecture of 2D convolution algorithm

IV. SYNTHESIS RESULT

We proceeded to the realization and simulation of this architecture with the tools of the Quartus II development system and using the VHDL description language. We tested it on an image of 128x128 pixels, with a 3x3 two-dimensional filter and the resulting image was displayed on a VGA screen. According Fig.5 this architecture requires 187 logic elements: 184 combinational functions and 56 register. This is an encouraging result and we intend to reduce the complexity of the system. This is the subject of our current research.



```
Flow Status                          Successful - Fri May 27 14:01:00 2016
Quartus II Version                   9.1 Build 222 10/21/2009 SJ Web Edition
Revision Name                        conv1
Top-level Entity Name                VGA_top
Family                               Cyclone II
Device                               EP2C20F484C7
Timing Models                        Final
Met timing requirements              Yes
Total logic elements                 187 / 18,752 ( < 1 % )
  Total combinational functions      184 / 18,752 ( < 1 % )
  Dedicated logic registers          56 / 18,752 ( < 1 % )
Total registers                      56
Total pins                           16 / 315 ( 5 % )
Total virtual pins                   0
Total memory bits                    131,072 / 239,616 ( 55 % )
Embedded Multiplier 9-bit elements   0 / 52 ( 0 % )
Total PLLs                           0 / 4 ( 0 % )
```

Fig.5: Synthesis Result

V. CONCLUSIONS

In this paper we presented two methods of 2D discrete wavelet transform, the 2D-DWT separable and no-separable. After we presented our design of the 2D-DWT based on 2D convolution algorithm. The corresponding VHDL program was tested on an image of 128x128 pixels with a 2D filter (3x3) using the Quartus II system. This work was implemented in an FPGA device Cyclone IV from Altera, it required 187 cells as logic resources.

References

[1] Q. Huang, Y. Wang and S. Chang, High-performance FPGA Implementation of Discrete Wavelet Transform for Image Processing, 2011.

[2] Mallat S.G., A theory for multiresolution signal decomposition: The Wavelet Representation [J], IEEE Trans on PAMI,1989,11(7):674-693.

[3] R. J. C. PALERO, R.G.GIRONES, A. S. CORTES. A Novel FPGA Architecture of a 2-D Wavelet Transform, 2006.

[4] C. Cheng, and Keshab K. Parhi, High-Speed VLSI Implementation of 2-D Discrete Wavelet Transform.

[5] 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.

[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.

[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.