# Real Time FPGA acceleration for Discrete Wavelet Transform of the 5/3 Filter for JPEG 2000

Taoufik SAIDANI Mohamed ATRI
Faculty of sciences of Monastir Tunisia
Saidani_taoufik@yahoo.fr
Mohamed.atri@fsm.rnu.tn

Yahia SAID and Rached TOURKI
Faculty of sciences of Monastir Tunisia
Said.yahia1@gmail.com
Rached.Tourki@topnet.net

*Abstract*—**In the programmable device technologies The continuous advancements enable the widespread use of FPGAs in an increasing number of industrial applications. The availability of powerful software design tools is a fundamental requirement to take advantage of the many advanced and specialized resources included in the latest devices. Video acceleration and processing technologies have become critical for the development of many consumer electronics products. In this paper, we investigate Real Time FPGA implementation of 2-D lifting-based Daubechies 5/3 transforms using a Matlab/Simulink/Xilinx System Generator tool that generates synthesizable VHSIC Hardware Description. This system offers significant advantages: portability, rapid time to market and real time, continuing parametric change in the DWT transform. The proposed model has been designed and simulated using Simulink and System Generator blocks, synthesized with Xilinx Synthesis tool (XST) and implemented on Spartan 3A DSP based XCSD 3400A-4fg476 target device.**

***Keywords-component; formatting; style; styling; insert (key words)***

## I. INTRODUCTION

In the last few years, there has been a growing trend to implement DSP functions in Field Programmable Gate Arrays (FPGAs), which offer a balanced solution in comparison with traditional devices. Although ASICs and DSP chips have been the traditional solution for high performance applications, now the technology and the market are imposing new rules [1, 2]. On one hand, high development costs and time-to-market factors associated with ASICs can be prohibitive for certain applications and, on the other hand, programmable DSP processors can be unable to reach a desired performance due to their sequential-execution architecture. In this context, FPGAs offer a very attractive solution that balance high flexibility, time-to-market, cost and performance[1.2]. In the digital video and image processing area, there are many different methods to compress images. Each compression method has different aspects and different characteristics. Among those methods, the compression standard developed by Joint Photographic Expert Group (JPEG) [6] is currently the most widely used in many applications such as web applications. It is because that JPEG compression method can compress large amount of image data into relatively small amount of data size.

About ten years later when JPEG image standard was developed, the new compression standard for image compression was developed in year 2000 and it was named JPEG2000[6]. JPEG2000 is new image compression standard from JPEG and it adapts different algorithm from previous JPEG standard. With this new algorithm, JPEG2000 can perform much improved image compression with higher quality. However, JPEG2000 is not yet widely used in both commercial and non-commercial applications and most of general image application users do not know about this new image compression standard. Although wavelets have been around for a while in the area of mathematics, it has been until very recently that they were formally formulated and began to be extensively used. Just in 1986, a joint effort between Mallat and Meyer gave birth to multiresolution analysis with wavelets, and in 1988, based on this study, Daubechies discovered the most widely used and known family of wavelets, called after her the Daubechies Wavelets. Nowadays, wavelet transform is intensively used in speech, image and video processing, and in signal processing in general because of its attractive characteristics to represent non-stationary signals in both frequency and time domains [8]. Researchers have switched from STFT (Short-Time Fourier Transform) to DWT because the former uses a fixed resolution at all times while the latter provides a variable resolution following the observed pattern on most applications: high frequency components of signals have a short duration while low frequency components have a long duration [4]. Furthermore, DWT has been adopted by recent still image and video coding standards, JPEG2000 [6] and MPEG-4 [6], given its high performance for image and video compression showing superior results when compared to the traditionally used Discrete Cosine Transform (DCT).

The paper is organized as follows. Section II discusses related work. Section III describes our approach for building a MATLAB/Simulink/Xilinx System Generator based hardware/software cosimulation environment for application development using FPGA configured soft processors. Due to its wide availability, we focus on integrating System Generator. Discrete wavelet transform and proposed architecture are presented simultaneously in section 3 and 4. The real time implementation of DWT2D is presented in section 5. Finally, we conclude in Section 6.

## II. DESIGN METHODOLOGY FOR IMPLEMENTATION ON FPGA

Unlike the standard HDL languages Xilinx System Generator provides a model-based design interface using an extended library of building blocks to create hardware. Rather than at textual code level, Xilinx System Generator takes the abstraction level one step higher, and uses the Mathworks Simulink environment to provide a graphical approach. Connected graphical building blocks and their parameterization form the description of the model.

Xilinx System Generator is integrated as a blockset in the model-based design environment Simulink from the Mathworks [4, 5].

The graphical user interface is powerful, yet easy-to-learn and easy-to-use. System Generator provides the Simulink environment with a list of specific Xilinx building blocks which can be used to create designs optimized for Xilinx FPGA's. The Matlab workspace can be used during simulation, while the hardware itself is modeled using the System Generator blocks.

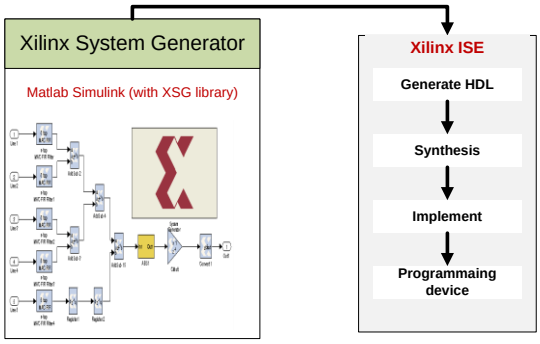


Figure 1. Lifting scheme forward transform

Designs that are to be implemented on the FPGA are generally captured in Hardware Description Languages (HDLs) such as Verilog or VHDL. HDLs insulate designers from the details of hardware implementation. Some of the EDA tools allow design description in high-level software languages such as C and C++ which are eventually converted to HDL designs [3]. Designs can also be captured using model-based tools such as the Xilinx System Generator (XSG) which convert schematic models to HDL descriptions. XSG allows users with little or no HDL background to work with FPGAs. XSG uses MATLAB's Simulink tool [7] to model designs by connecting hardware blocks together. Hardware blocks are IP cores or pieces of tested logic supplied by Xilinx. XSG relieves the designer from low-level algorithmic complexity and helps to implement designs. The XSG library has a set of DSP hardware blocks that can perform complex functions such as FFT, FIR fillter design, or Viterbi decoding. XSG uses the Xilinx ISE design suite to automate HDL code generation which can then be integrated with other designs or used as a stand-alone design. Fig.1 gives design using XSG.

The system model can be simulated in the Simulink environment [2, 3]. This higher abstraction level reduces the analysis and debugging time. For real hardware testing, Xilinx System Generator supports the possibility to perform hardware in-the-loop co-simulation. Computationally intensive parts of the design can be downloaded into the FPGA and run in hardware, while another part runs on the workstation (PC) itself. This further accelerates the simulation. Interfacing between the Simulink environment (on PC/workstation) and FPGA platform is done using a slower standard JTAG connection or a gigabit Ethernet connection which allows a fast data transfer.

When the design is modeled, automatic VHDL or Verilog code generation can take place. Each library building block corresponds with a generic HDL file. Xilinx System Generator instantiates these HDL files with the correct parameters and applies the needed connections. The generated HDL code is used as entry for the automatically performed implementation steps: logic synthesis and place-and-route [3,5].

## III. THE DISCRETE WAVELET TRANSFORM

The lifting scheme is an algorithm used for implementation hardware and software of DWT [11, 12]; it is constituted of steps of

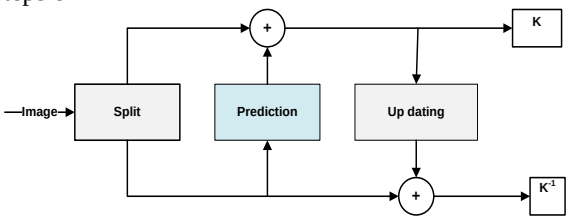


Figure 2. Lifting scheme forward transform

predictions and updating described by the Fig.2 The advantage of lifting scheme is the forward and inverse transform was obtained from the same architecture. The inverse goes from right to the left, by inversing the coefficients of normalized and changes the sign positive to negative. Where k is the constant of normalisation and the steps of the predictions and the updating at decomposition in polyphase matrix. The polyphase representation of discrete filter h(n) is defined as:

$$\mathbf{h(z) = h_e(z^2) + z^{-1} h_0}$$

where $h_e(z)$ and $h_0(z)$ are respectively obtained from the even and odd zeta transform respectively. If we represent h(z) and g(z) the low pass and high pass coefficients of the synthesis filter respectively, the polyphase matrix written as:

$$\mathbf{p(z)} = \begin{bmatrix} \mathbf{h_e(z)} & \mathbf{g_e(z)} \\ \mathbf{h_0(z)} & \mathbf{g_0(z)} \end{bmatrix}$$

The filters $h_e(z)$ , $h_0(z)$ , $g_e(z)$ and $g_0(z)$ are Laurent polynomials, as the set of all polynomials exhibits a commutative ring structure, within which polynomial division with remainder is possible, long division between two Laurent polynomials is not a unique operation. In Euclidean algorithm decomposition can be used, the polyphase p(z) is finally obtained as:

$$p(z) = \prod_{i=1}^{m} \begin{bmatrix} 1 & S_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} k & 0 \\ 0 & \frac{1}{k} \end{bmatrix}$$

Where $S_i(z)$ and $t_i(z)$ primary lifting and dual lifting steps filters respectively, k is a constant of normalisation at low and high coefficients filters. The 5/3 wavelet filter transform is more suitable lossless data compression adopted in JPEG2000 and 9/7 filter is used in JPEG2000 for lossy compression data. The 5/3 filter has one prediction and one up-dating compared to two predictions and two up-dating for 9/7 filter. The following steps are necessary to get their wavelet coefficients as the following tapes for 5/3 filter: Split the input signal (image) into coefficients at odd and even positions. Perform a predict step, that is the operation given below in [6,11]. Perform up-dating step which is the operation given below in [6].

$$d(2n+1) = x(2n+1) - \left[\frac{x(2n) + x(2n+2)}{2}\right]$$

$$a(2n) = x(2n) + \left[\frac{d(2n-1) + d(2n+1) + 2}{4}\right]$$

direct lifting scheme of bi-orthogonal 5/3 filter, where the constant k is equal unit.

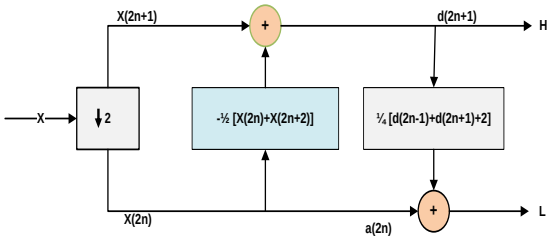


Figure 3. Lifting scheme decomposition of 5/3 filter

The lifting based implementation of two levels 2D-DWT may be computed using filter banks as shown in Fig.3. The input samples X(n) are passed through two stages of analysis filters. They are first processed by low-pass (h(n)) and high-pass (g(n)) horizontal filters and are sub sampled by two. Subsequently, the outputs (L1, H1) are processed by low-pass and high-pass vertical filter. Note that: L1, H1 are the outputs of 1D-DWT; LL1, LH1, HL1 and HH1 one-level decomposition of 2D-DWT.

From the previous structure, a separable 2D-DWT with N levels of transformation can be easily achieved by concatenation of 1D-DWT units, with the first stage processing N transformation levels on rows and the second one with N transformation levels on columns. However, that scheme is still not the most efficient one. For image compression purposes, JPEG 2000 recommends an alternate row/column-based structure as the one presented in Figure 2.

The standard filterbank shown in figure 2 processes alternatively rows and columns in every stage, with iteration only in the low-pass filter sub-band. Fig. 4 shows the sub-band decomposition of an image when the standard 2D-DWT with three transformation levels is applied. In figure 3, "H" and "L" correspond to high and low-pass filter stages, respectively.

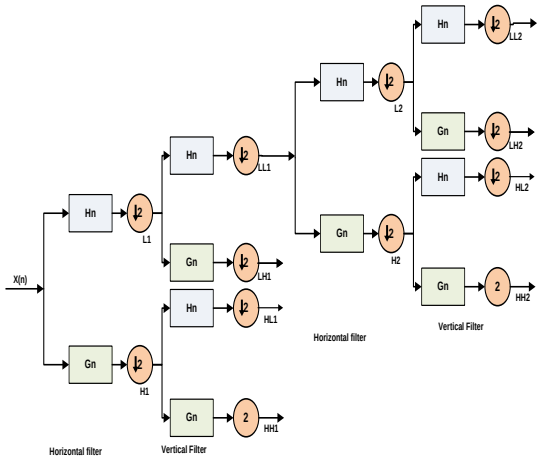


Figure 4. Subband decomposition for two-level 2D-DWT

## IV. PROPOSED ARCHITECTURE

The block diagram of the proposed design is shown in fig.5. It consists of a DWT processor and a pair of external dual-port memories. The two memories

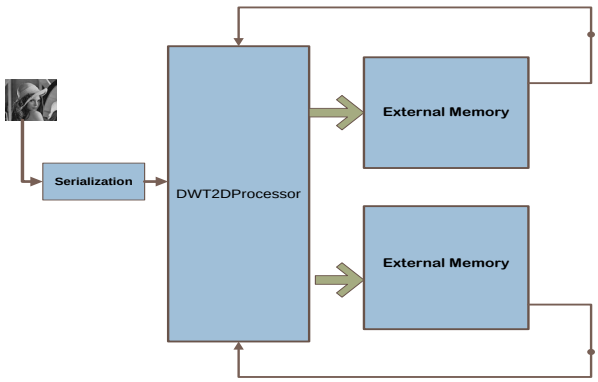


Figure 5. The proposed architecture for the Legall wavelet transform

Fig.6 represent the DWT processor, It includes DWT filter, memory controller and crossbars for the read and write address. The crossbars are used for interleaving the image pixels. The output of the high pass and low pass filter will be distributed alternatively to the two memory banks.

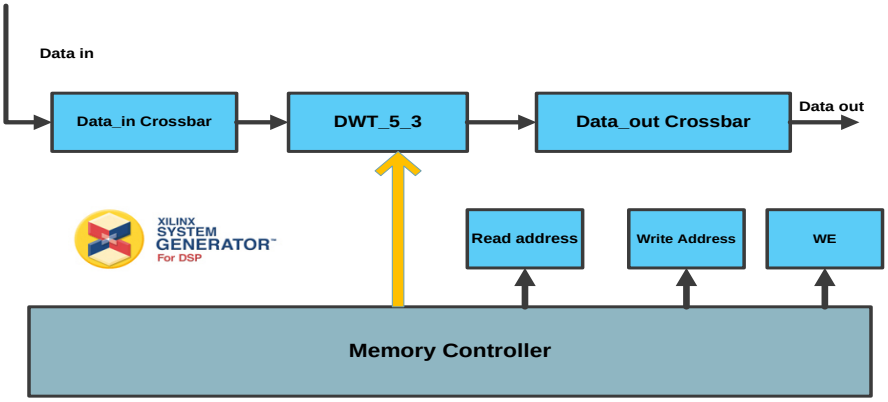


Figure 6. DWT2D Processor

The Discrete Wavelet Transform can be implemented using high pass and low pass filters. The high pass and low pass filters are designed using following transformations:

$$H(2n+1) = X(2n+1) - \text{floor}( [ X(2n) + X(2n+2) ] / 2 )$$

$$L(2n) = X(2n) + \text{floor}( [ H(2n-1) + H(2n+1) + 2 ] / 4 )$$

The high pass and low pass filters decompose the image into detail and approximate information respectively. The detail information is basically low scale, high frequency components of the image and it imparts nuance. Whereas the approximate information is high scale, low frequency components of the image and it impart the important part of the image. In the high pass and low pass filter, the new inputs are accepted at one end before previously accepted inputs appear as outputs at the other end. This process is known as pipelining which helps to enhance the speed of the processor. The output of the H and L filters will be alternately distributed to the two memory banks. The data on the 'H' outputs are delayed by 32 cycles relative to the 'L' outputs. Without this delay, the data being written from the 'H' and the 'L' filters would always be trying to write to the same memory bank. With the delay added, they end up always writing to opposite banks.

on a per-line basis, where lines are defined by the sol (start-of-line) and eol (end-of-line) signals, and where the odd samples ( X(2n+1)) and even

samples (X(2n)) are supplied on separate input channels. Extrapolation from the endpoints of the line is required to compute the above transformation. We have chosen a Neuman extrapolation, wherein, effectively, the second Derivative of the data becomes zero at the endpoint and the data is inflected. Thus, with x(1) being an endpoint, we construct:

$$X(0) = X(1) - [ X(2) - X(1) ]$$

and

$$X(-1) = X(1) - 2 [ X(3) - X(1) ]$$

The memory controller works as though the writes are happening simultaneously to the reads. it does not account for the latency of getting data from memory, or the latency of the filter engine. The necessary pipeline balancing adjustments are handled outside of the controller. This makes for a much simpler control structure.

By pipelining the DWT, the highpass and lowpass coefficients can be computed during the time the memory is being accessed, rather than having to wait until the reads are complete, computing the coefficients, and then writing them. Due to the nature of the lifting scheme DWT, the low pass coefficients depend not only on the high pass coefficient in the stage immediately before them, but also the pixel values of higher stages. As a result, it is necessary to ensure that the pipeline data is valid at all times.

## V. HARDWARE IMPLEMENTATION RESULTS

This section describes the modeling desing for DWT2Din hardware, figures of merit, obtained from our architecture, as well as the resources occupied. The adopted hardware platform is the XtremeDSP Video Starter Kit equipped with a Xilinx Spartan-3A DSP 3400A [13], an FPGA chip optimized for DSP applications. The FPGA resource occupation and the timing performances are estimated using the Xilinx development tools. The FPGA model was tested on both synthetic and real data sets, and the results are compared with those obtained using a behavioral software model of the algorithm and with the expected ground truth, respectively. Tests are first performed in simulation, and then in co-simulation with the hardware in the loop, providing the FPGA with input data coming from a software simulation environment.

### *A. Xilinx System Generator modeling*

The proposed model has designed and simulated using Simulink and Xilinx System Generator block sets.

2D-DWT is applied on grayscale image which is shown in fig.7. It transforms an image into sub-bands such that the wavelet coefficients in the lower level sub-bands typically contain more energy than those in higher level sub-bands.

Figure 7. Original Image (512*512*8bits)

The simulated has been accomplished by using DWT filter in the proposed model. The DWT filter uses high pass and low pass filter to decompose the image into its detail and approximate information respectively. The decomposition of the image is shown in the fig.8.

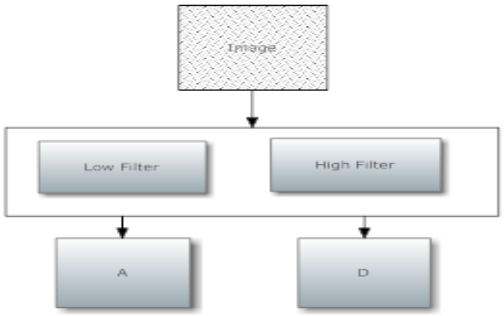


Figure 8. Decomposition of the Image

Fig.9 present a hardware model of the "L" and"H" Filter by Xilinx system generator.

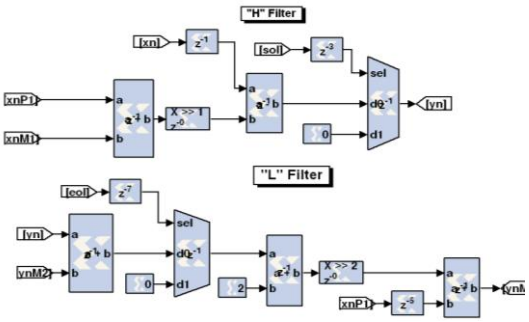


Figure 9. Xilinx System Generator for H and L filters

This extrapolation scheme becomes very simple in the hardware implementation, adding only the two muxes appearing below.

The decomposition process can be iterated with successive approximations being decomposed into many lower resolution components. This is also called as the wavelet decomposition tree.

It can be accomplished by applying one-dimensional DWT filter in a separable manner. The first stage of the DWT divides an image into four sub-bands by applying low-pass and high pass filters. The first level of decomposition is consists of two steps. In the first step, each row of an image is again transformed using same filter bank horizontally. Thus first level of decomposition produces four filtered and sub-sampled images.

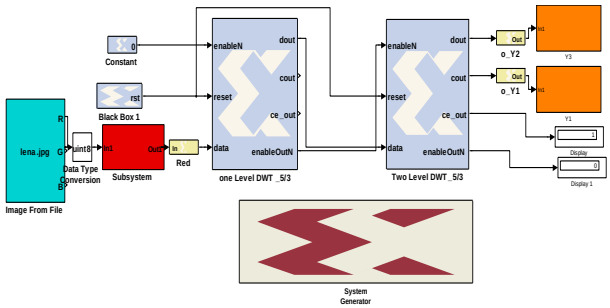


Figure 10. Model for two level of DWT2D transform

For the second level of decomposition (fig.12), DWT further divides the lowest sub-band using the same filtering method as above. The lowest sub-band has been decomposed into further four sub-bands. Each row and column of the lowest sub-band has been replaced by 1D-DWT. The result of the second level of decomposition has been shown in fig.11.

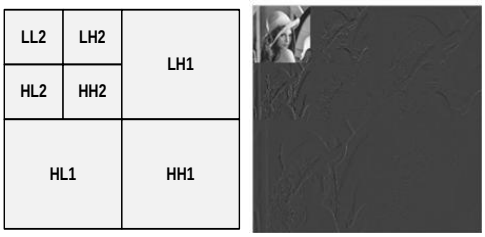


Figure 11. Second level of Decomposition

## B. *Implementation*

When simulation results are correct the implementation steps are done automatically by the tool. In this experiment VHDL code was generated for a Xilinx Spartan-3A DSP 3400A FPGA. On an Intel Pentium 4 running at 3GHz code is generated in less than 2 minutes. Code generation is realized using the instantiation of generic library blocks which are parameterized based on the settings of the designer. The synthesized design takes an area of 272 slices in the FPGA, and has a maximum clock frequency of 205.423 MHz.

The logic resource consumed by the DWT2D module along with its other important timing constraints are shown in Table 2 and Table 3.

TABLE I. LOGIC RESOURCES CONSUMED BY DWT2D IN XC3SD3400A-4FGG676C

| | Used | Available | % |
|---|---|---|---|
| N. of slices | 272 | 23872 | 1.13 |
| N. of F.F | 497 | 47744 | 1.01 |
| N. of LUT | 344 | 47744 | 0.7 |
| N.of IOBs | 91 | 469 | 19 |
| N. of GCLK | 1 | 24 | 4 |

TABLE II. TIMING SUMMARY FOR DWT2D IN XC3SD3400A-4FGG676C

| | |
|---|---|
| Minimum Periods | 4.868 ns |
| Maximum Frequency | 205.423Mhz |

During the Simulink-to-FPGA design flow, circuit modeling is built up with Simulink basic blocks and Xilinx specified blocks. Input and output data are combined with MATLAB workspace, which is convenient to convert number format and debug.

The developed VHDL code has been simulated using ISE Xilinx. The waveform it presented in fig.12.

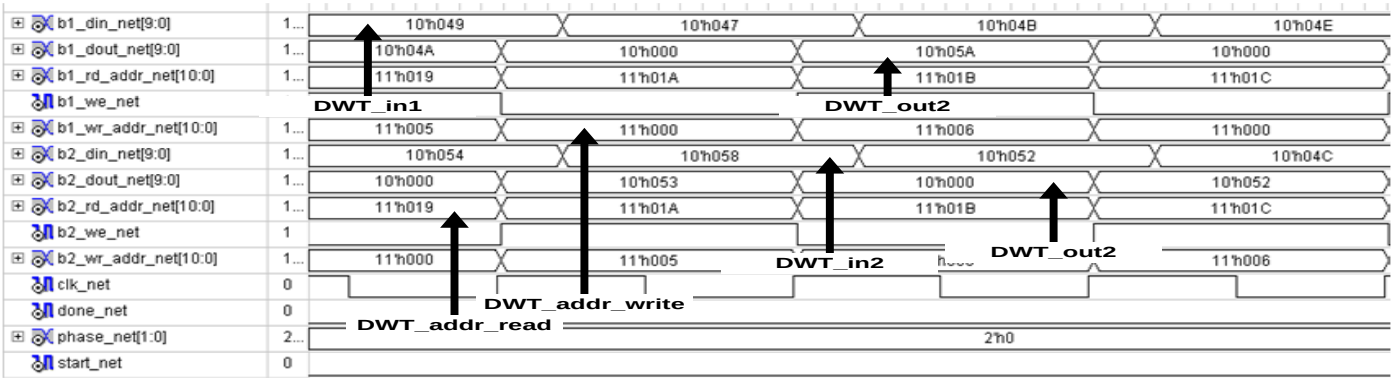


Figure 12. Waveform for DWT2D

## C. *Hardware-in-the-loop*

Xilinx System Generator extends Simulink through a direct interface to hardware platforms, hardware-in-the-loop (HIL) co-simulation. Xilinx System Generator automatically generates an FPGA bit stream for a selected part of the model, mostly a computational intensive function. Communication between the hardware platform and the Matlab/Simulink environment is done through a JTAG or a gigabit Ethernet connection. This hardware co-simulation allows the user to exploit the processing power of the FPGA hardware to significantly accelerate simulations. In the meantime the model is validated on the hardware platform.

This model can be used for co-simulation. Once the design is verified, a hardware co-simulation block can be generated and then will be used to program the FPGA for the non uniform illumination correction model implementation. Fig.13 shows the model with the hardware co-simulation block. The bit stream download step is performed using a JTAG cable. Here Xilinx System Generator Token is used which is necessary for the design of such models.

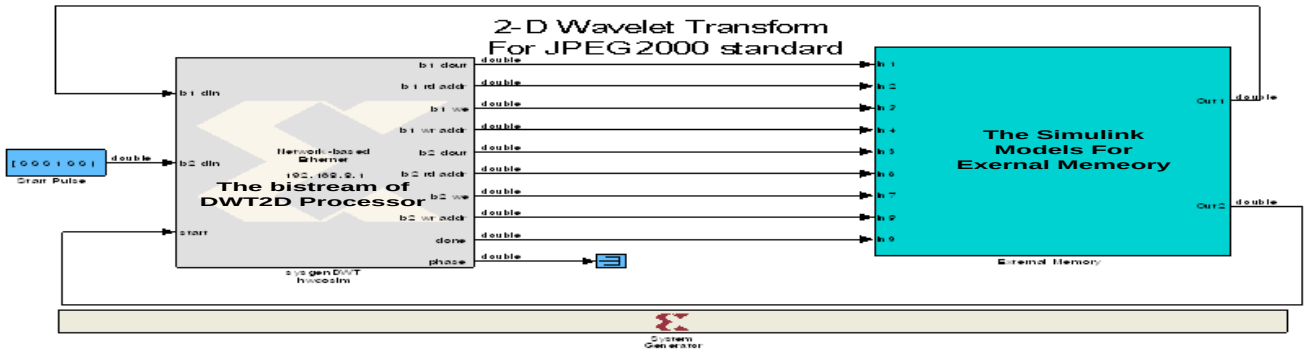


Figure 13. HW Co-Simulation for DWT2D Design

Fig.14 shows that almost there is no difference between result obtained from MATLAB/simulink and FPGA for the first decomposition of wavelet transform.

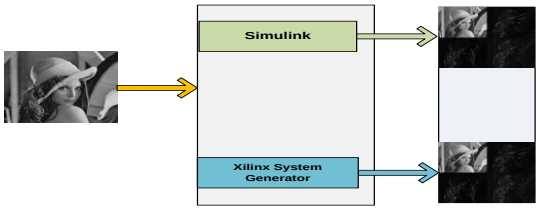


Figure 14. Hardware and software results for the DWT2D transform

## D. *Discussion*

From the development of FPGA technology, the methodology challenges the update of various EDA tools [2]. Based on the standard development flow, initial efforts have been transferred to high-level design and synthesis. There are many conversion tools such as C-to-FPGA, Stateflow diagram to VHDL Matlab-to-FPGA. The features of Simulink/Xilinx System Generator-to-FPGA [3, 4] flow can be discussed as follows.

- Fast time-to-market for DSP development. With the assistance of specified DSP blocks for FPGA, the Simulink/Xilinx System generator-to-FPGA flow can greatly shorten the development cycle from algorithm to hardware.
- Friendly graphics interface. Although the schematic entry is also a GUI interface, the Simulink is easier to organize input data and much convenient to observe output in many ways [8].
- Flexible modeling and simulation. The design can be well organized into hierarchical modules and easy to be combined with other entry method for design decision and convenient to debug and simulation.

## VI. CONCLUSION

We conclude that Xilinx system generator is a very useful tool for developing computer vision algorithms. It could be described as a timely, advantageous option for developing in a much more comfortable way than that permitted by VHDL or Verilog hardware description languages (HDLs). The purpose of this paper was to demonstrate the use of System Generator to design a system wavelet processor for video processing. This design is implemented in the device Spartan 3A DSP 3400 (XC3SD3400A-4FGG676C). The implemented DWT2D architecture using low cost available Spartan-3A DSP 3400 development system with Xilinx chip XC3SD3400A-4FGG676C has 205.423 MHz maximum frequency and uses 272 CLB slices.

## REFERENCES


[1] Zemcik, P.: Hardware acceleration of graphics and imaging algorithms using FPGAs. SCCG'02: Proceedings of the 18th Spring Conference On Computer Graphics, pp. 25–32. ACM, New York (2002).

[2] Fons, M. Fons, F. Cantó, E. "Fingerprint Image Processing Acceleration Through Run-Time Reconfigurable Hardware "Circuits and Systems II: Express Briefs, IEEE Transactions on 2010.

[3] Xilinx System Generator User's Guide, www. Xilinx.com.

[4] Moctezuma, J.C., Sanchez, S., Alvarez, R., S´anchez, A.: "Architecture for filtering images using Xilinx system generator" World scientific advanced series in Electrical and Computer Engineering. Proceedings of the 2nd WSEAS International Conference on Computer Engineering and Applications, pp. 284–289 (2008).

[5] T. Saidani , M. Atri ,D. Dia, , and R. Tourki, "Using Xilinx System Generator for Real Time Hardware Co-simulation of Video Processing System", Electronic Engineering and Computing Technology,.Lecture Notes in Electrical Engineering Springer 2010.

[6] ITU-T Recommend. T.800-ISO FCD15444-1: JPEG2000 Image Coding System. International Organization for Standardization, ISO/IEC JTC1 SC29/WG1 (2000).

[7] The MathWorks Inc. Embedded Matlab Language User Guide (2007).

[8] Marusic, B., Skocir, P.: Video post-processing with adaptive 3-D filters for wavelet ringing artifact removal. IEICE Transactions on Information and Systems 88(5), 1031–1040 (2005).

[9] G. Seetharaman B. Venkataramani G. Lakshminarayanan, "Automation techniques for implementation of hybrid wave-pipelined 2D DWT" Reconfigurable Architecture for Real-Time Image Processing sringer 2008.

[10] Agustin Ramirez-Agundis Rafael Gadea-Girones Ricardo Colom-Palero Javier Diaz-Carmona, A wavelet-VQ system for real-time video compression" J Real-Time Image Proc,"Special Issue on: Field Programmable Technology" Springer 2007.

[11] Tinku Acharya and Ping-Sing Tsai. JPEG2000 Standard for Image Compression: Concepts, algorithms and VLSI architectures. Wiley-Interscience, New York, 2004.

[12] Dhaha Dia, Medien Zeghid, Taoufik Saidani, Mohamed Atri, Belgacem Bouallegue, Mohsen Machhout and Rached Tourki." Multi level Discrete Wavelet Transform Architecture Design" Proceedings of the World Congress on Engineering 2009 Vol I, London, UK.

[13] Xilinx, Spartan 3A DSP FPGA Family Datasheet.