FPGA REAL TIME ACCELERATION FOR DISCRETE WAVELET TRANSFORM OF THE 5/3 FILTER FOR JPEG2000 STANDARD

TAOUFIK SAIDANI[1], MOHAMED ATRI[2], YAHIA SAID[3] AND RACHED TOURKI[4]

[1]Department of Electronic and Microelectronic, FSM University, Monastir, Tunisia
Saidani_taoufik@yahoo.fr
Said.yahia1@gmail.com
Mohamed.atri@fsm.rnu.tn
Rached.Tourki@topnet.net

ABSTRACT

In recent years video and image compression have became very required . 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

JPEG2000 ,wavelet , FPGA, Matlab Xilinx System generator, VHDL.

1. INTRODUCTION

Implementation of DSP function in FPGA (Programmable Gate Arrays) has been a increasing trend in the last few years. Although for high performance applications ASICs and DSP chips have been the habitual solution ,now the technology implementation and the time to market are arresting new rules [1, 2]. So programmable DSP processors can be powerless to reach a desired performance due to their sequential-execution architecture, on the other hand, high development costs and time-to-market factors associated with ASICs can be prohibitive for certains applications . For this reasons, very attractive solution that balance high flexibility, time-to-market, cost and performance can be offered by FPGAs technology [1.2]. Digital video techniques have been used for a number of years, for example in the television broadcasting industry. Therefore, for stored or transmitted, the digital video information has to be compressed [6]. The growing demand for interactive multimedia technologies, in various application domains in this era of wireless and Internet communication, necessitated a number of desirable properties to be included in image and video compression algorithms. Accordingly, current and future generation image compression algorithms should not only demonstrate state-of-the-art performance, it should also provide desirable functionalities such as progressive transmission in

DOI : 10.5121/ijesa.2012.2101

terms of image fidelity as well as resolution, scalability, region-of-interest coding, random access, error resilience, handling large-size images of different types.

To address and improve on weaknesses in the JPEG standard, a new image compression scheme called JPEG 2000 has recently been introduced [6]. JPEG 2000 supports a rich set of features, including improved compression efficiency, optional lossless encoding, resolution and distortion (SNR) scalability, region of interest coding, and support for image editing on compressed images. Although they share the same name, JPEG 2000 is fundamentally different from the original JPEG standard. First, JPEG 2000 replaces the JPEG's discrete cosine transform (DCT) frequency decomposition with a discrete wavelet transformation (DWT). The DWT is a multiresolution decomposition, which allows for resolution scalability within the embedded bit stream. In addition, the DWT exhibits better energy compaction than the DCT, allowing for superior compression efficiency [6]. The DWT typically is applied on an image tile or on the entire image as a whole. This large scale application allows the DWT to minimize the blocking artifacts that plagued the 8x8 DCT in the original JPEG. The second major deviation of JPEG 2000 from its predecessor is the abandonment of the Huffman entropy encoding scheme for an adaptive binary arithmetic coder. The JPEG2000 algorithm has been developed based on the discrete wavelet transform (DWT) technique as opposed to the discrete cosine transform (DCT) based current JPEG. The nature of DWT helps to integrate both the lossless and lossy operations into the same algorithmic platform as well as it allows one to perform different kinds of progressive coding and decoding in the same algorithmic platform. Also the bit-plane coding of the transformed coefficients and the underlying structure of the bitstream syntax is very suitable to achieving different progressive operations during both encoding and decoding.

The paper is organized as follows. Section II discusses the design methodology for implementation on FPGA. Discrete wavelet transform and proposed architecture are presented simultaneously in section III and IV. The development of software implementation for our design is provided in Section V to demonstrate the effectiveness of our co-simulation approach. The real time implementation of DWT2D is presented in section V. Finally, we conclude in Section IV.

2. DESIGN METHODOLOGY FOR IMPLEMENTATION ON FPGA

For the designer it is easy to learn and easy to use for the graphical interface. 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 modelled using the System Generator blocks.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].

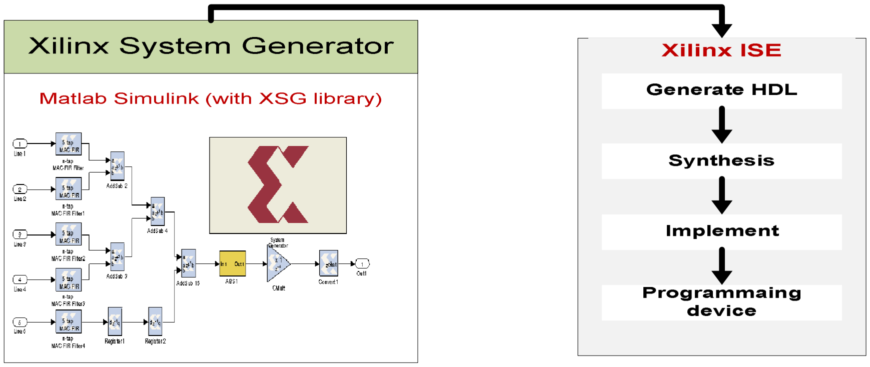

Figure.1 Flow design fo Matlab XSG

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. 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. 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. The generated HDL code is used as entry for the automatically performed implementation steps: logic synthesis and place-and-route [3,5].

3.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

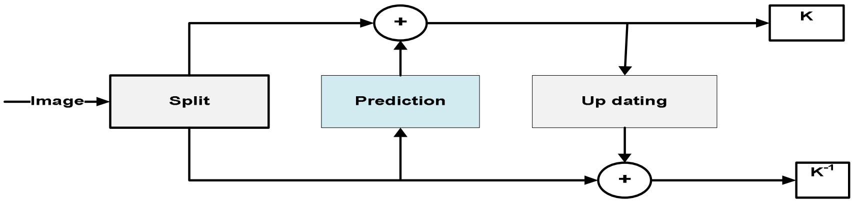

Figure.2 Lifting scheme forward transform

predictions and updating described by the Fig.2. The benefit of lifting scheme is the forward and inverse transform was obtained from the similar architecture. Going from right to the left, we can obtain the inverse transform by inversing the coefficients of normalized and changes the sign positive to negative. The constant k is the coefficient of normalisation and it is the coefficient for the steps of the predictions and the updating at decomposition in polyphase matrix. The polyphase representation of discrete filter h(n) is defined as:

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

where $\mathrm{h_e(z)}$ and $\mathrm{h_0(z)}$ 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 Laurent polynomials are represented by the filters $\mathrm{h_e(z)}$, $\mathrm{h_0(z)}$, $\mathrm{g_e(z)}$ and $\mathrm{g_0(z)}$. The polyphase p(z) is finally obtained as:

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

Where $\mathrm{S_i(z)}$ and $\mathrm{t_i(z)}$ present the primary lifting and dual lifting steps filters respectively. For lossless data compression t he 5/3 wavelet filter transform is adopted in JPEG2000. The architecture of 5/3 filter is decomposed on one prediction and one up-dating .The next steps are essential to get their wavelet coefficients as the subsequent tapes for 5/3 filter: the input signal (image) is divided into coefficients at odd and even positions. Execute a predict step, that is the operation specified below in [6,11]. Perform up-dating step which is the procedure given below in [6].

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

The implementation of lifting schemes is decomposed of two levels 2D-DWT, it may be computed using filter banks as shown in Fig.3. The input samples X(n) are approved through two stages of analysis filters.

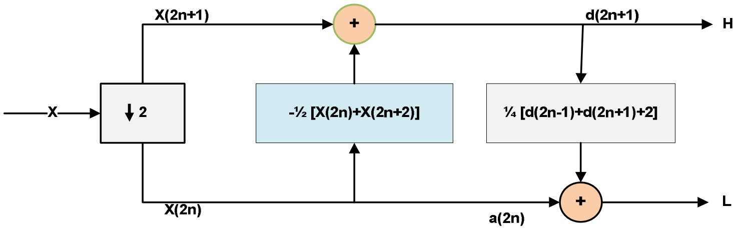

Figure.3 Lifting scheme decomposition of 5/3 filter

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 earlier structure, for a separable 2D-DWT with N levels of transformation,it 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.For image compression purposes, JPEG 2000 recommends an alternate row/column-based structure as the one presented in Figure 3. The sub-band decomposition of an image when the standard 2D-DWT with two transformation levels is presented in figure 4. “H” and “L” correspond to high and low-pass filter stages, respectively.

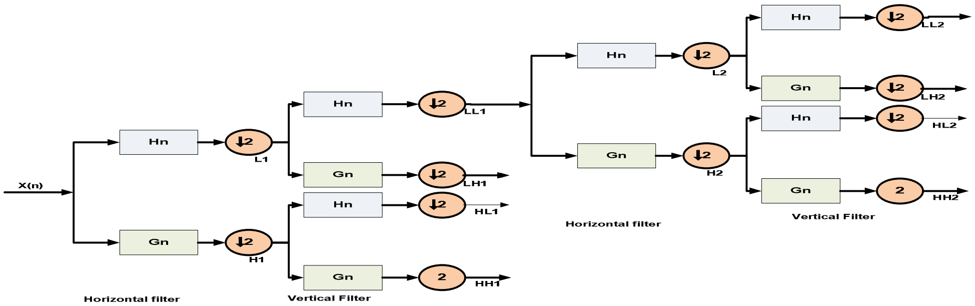

Figure.4 Subband decomposition for two-level 2D-DWT

4.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.

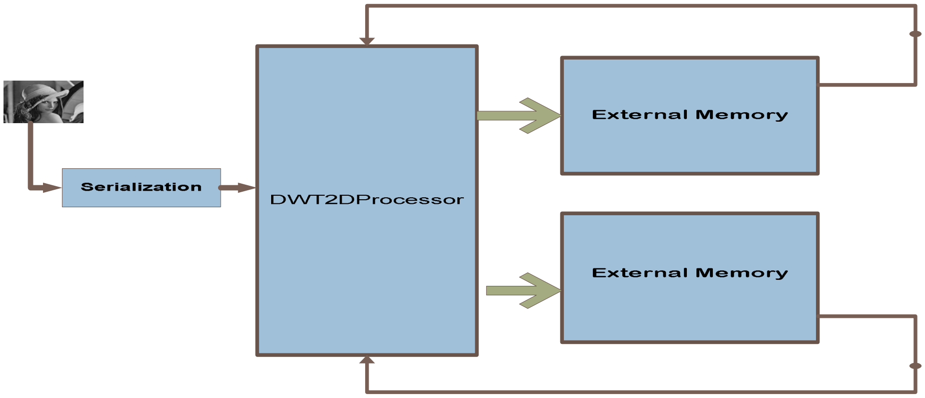

Figure.5 The proposed architecture for the Legall wavelet transform

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.

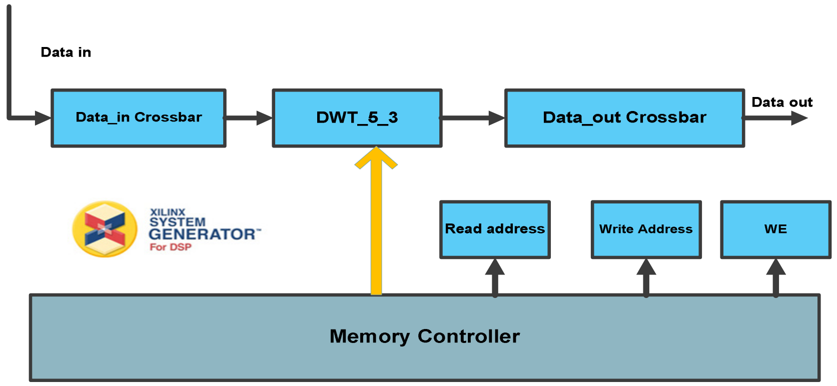

Figure .6 DWT2D Processor

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:

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

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

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. 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:

$$\mathbf{X(0) = X(1) - [\ X(2) - X(1)\ ]}$$

and

$$\mathbf{X(-1) = X(1) - 2\ [\ X(3) - X(1)\ ]}$$

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.

5. Software Implementation

Image and video processing algorithms are elaborated using MATLAB as it is considered industry standard for this field of research. Further, environment variables containing simulation and test data such as images or videos which were prepared during algorithm development are directly accessible from SIMULINK [7].

This fact eases the start into system level design where SIMULINK serves as modeling and simulation environment. Models are composed of blocks that have inputs and outputs, parameters and states. Inputs and outputs can be interconnected if data format and bus width match. At any time during model-composition the model can be validated by simulation. A validation step preceding the simulation checks the model for syntactical correctness and logical integrity. During simulation the model's semantic behavior is tested on the host.

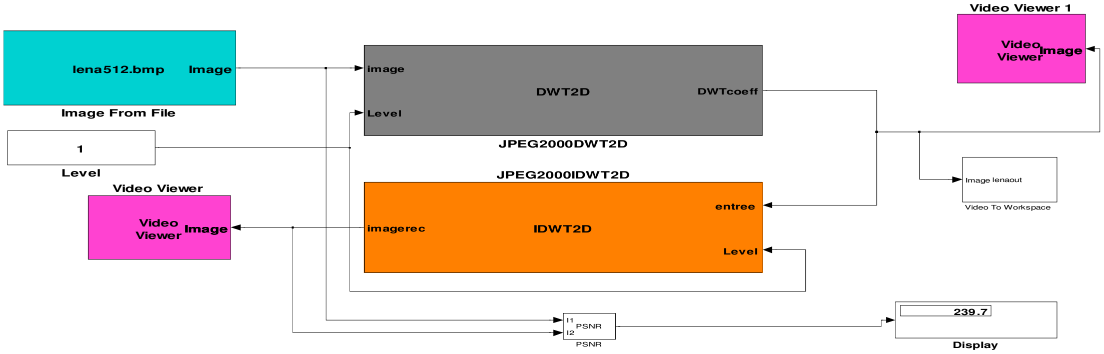

Figure. 7 Simulink Models for DWT2D-IDWT2D for jpeg2000

The DWT2D model in Simulink was therefore created using the embedded Matlab function block to gether with external constants feeding the function block with the DWT2D kernel.the DWT2D level is declared exeternally in the Matlab workspace as seen in fig.7. Demonstrate the results obtained using the software reference model.

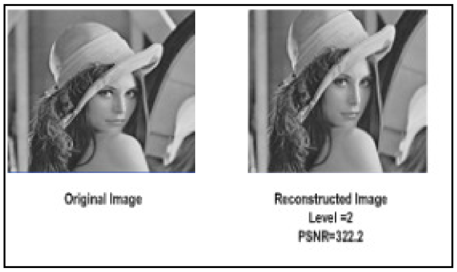

Figure. 8 Result for software implementation for two level

The PSNR, for the selected images, after forward and inverse Discrete Wavelet Transform with 8 bits coding coefficients, are given in Table I.

Table 1: PSNR for various Level For decomposition

Level	PSNR
1	325.6
2	322.2
3	320.6
4	319.4
5	318.5
6	317.2

6. 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.

6.1. Xilinx System Generator modelling

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.9. 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. 9 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.10.

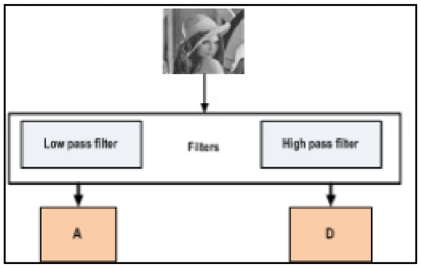

Figure. 10 Decomposition of the Image

Fig.11 present a hardware model of the “L” and“H” Filter by Xilinx system generator.

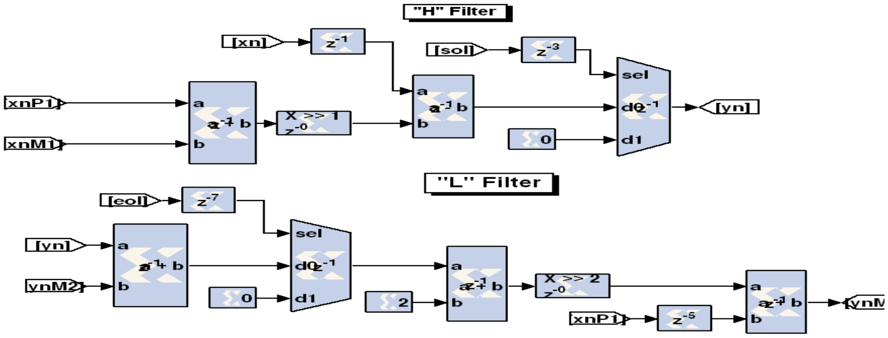

Figure.11 Xilinx System Generator for H and L filters

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.

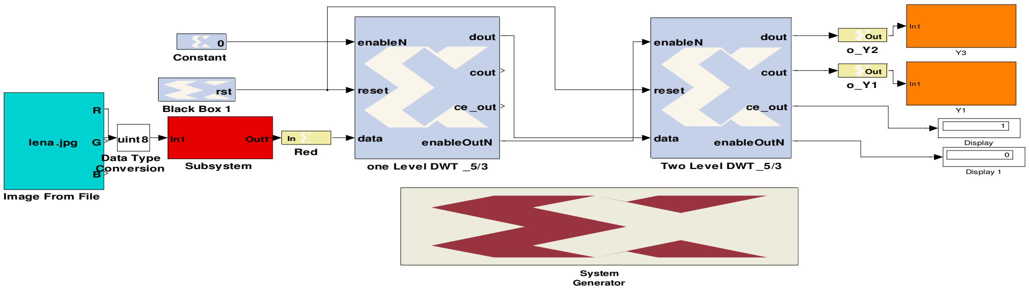

Figure.12 Model for two level of DWT2D transform

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.13

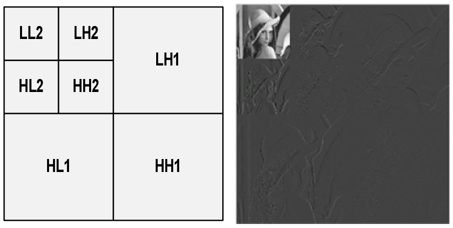

Figure.13 Second level of Decomposition

6.2. 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.

Table2: Logic Resources Consumed by DWT2D in XC3SD3400A-4FGG676C

	USED	Available	%
Number of slices	272	23872	1.13%
Number of Slice Flip Flops	497	47744	1.04%
Number of 4input LUTs	344	47744	0.7%
Number of bonded IOBs	91	469	19%
Number of GCLKs	1	24	4%

Table 3: Timing Summary for DWT2D in XC3SD3400A-4FGG676C

Minimum Periods	4.868 ns
Maximum Frequency	205.423Mhz

From the Simulink-to-FPGA design flow, architecture modeling is built up with Simulink basic blocks and Xilinx blocksets. Input and output data from hardware model are combined with Simulink workspace, which is suitable to convert number format and simulation.
The developed VHDL code has been simulated using ISE Xilinx. The waveform it presented in fig.14.

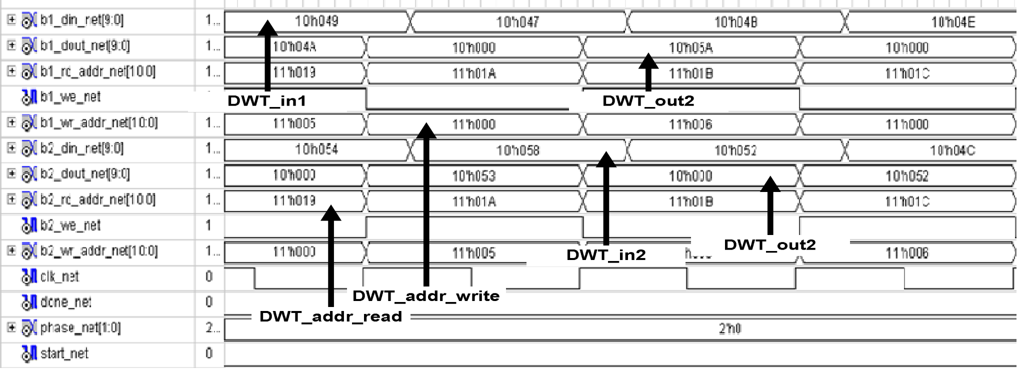

Figure.14 Waveform for DWT2D

6.3. 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 Xilinx system generator model can be used for hardware cosimulation. Once the design is simulated, for FPGA imlpemenation a hardware cosimulation block is generated and then will be used to be downloaded in the FPGA. Fig.15 shows the model with the hardware cosimulation block for the 2DWT architecture. JTAG cable for Xilinx is used to download the bit stream.

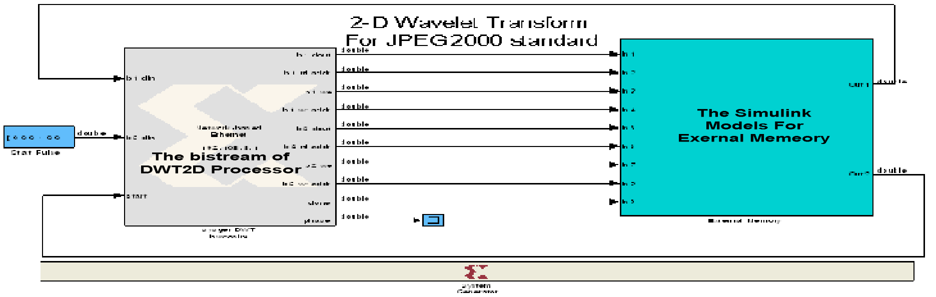

Figure.15 HW Co-Simulation for DWT2D Design

Fig.16 shows that almost there is no difference between result obtained from MATLAB/simulink and FPGA for the first decomposition of wavelet transform.

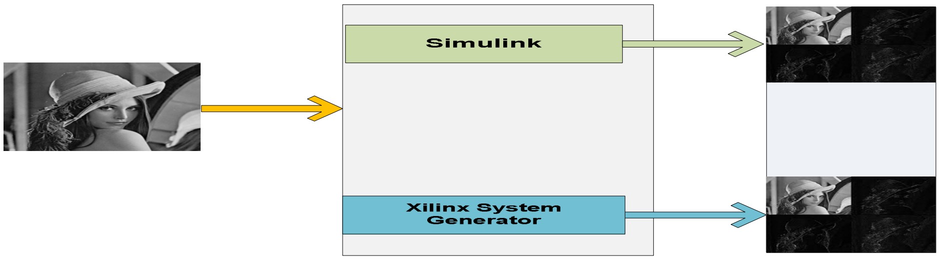

Figure . 16 Hardware and software results for the DWT2D transform

6.4. 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.

7. Conclusion

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.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).

Future works include the use of The design utilizes powerful design tool System Generator (SysGen) and Embedded Development Kit (EDK) for hardware-software codesign and integrates the DWT2D for jpeg2000 compression hardware as a peripheral to the Microblaze 32 bit soft RISC processor with an input from a CMOS camera and output to a DVI display and verified the results video in real time.

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 TimeHardware 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.

Authors

Taoufik Saidani received his M.S degree in Microelectronic from Faculty of science of Monastir !tunisia in 2007.His major research interest include VLSI and embedded system in video compression.

Yahia Saida received his M.S degree in Microelectronic from Faculty of science of Monastir !tunisia in 2010 .His major research interest include VLSI and embedded system in video tracking.

Atri Mohamed Non-menber born in 1971, received his Ph.D. Degree in Micro-electronics from the Science Faculty of Monastir in 2001. He is currently a member of the Laboratory of Electronics & Micro-electronics. His research includes Circuit and System Design, Image processing, Network Communication, IPs and SoCs.

Rached Tourki Non-menber was born in Tunis, on May 13, 1948. He received the B.S. degree in Physics (Electronics option) from Tunis University, in 1970; the M.S. and the Doctorat de 3eme cycle in Electronics from Institut d'Electronique d'Orsay, Paris-south University in 1971 and 1973 respectively. From 1973 to 1974 he served as microelectronics engineer in Thomson-CSF. He received the Doctorat d'etat in Physics from Nice University in 1979. Since this date he has been professor in Microelectronics and Microprocessors with the physics department, Faculte des Sciences de Monastir. His current research interests include digital signal processing and hardware-software codesign for rapid prototyping in telecommunications.