Comparative Analysis of Implementing Discrete Wavelet Transform on Reconfigurable Platform.

Shweta S. Swami[1], Altaf O. Mulani[2]
[1, 2]*Electronics & Telecommunication Department,*
SKNSCOE Korti Pandharpur, Solapur University Solapur, MS, India
[1]swami.shweta27@gmail.com, [2]aksaltaaf@gmail.com

Abstract—Image compression is the science of reducing the amount of data required to represent an image. It is one of the most useful and commercially successful technologies in the field of digital image processing. Discrete Wavelet Transform (DWT) is one of the most used techniques for image compression and is applied in a large category of applications for multi resolution analysis of signals. Using VLSI technology, system requires low power, less area and high speed constraints while designing. In this paper, various kinds of implementing Discrete wavelet Transform along with their performances are discussed.

Keywords— Discrete Wavelet Transform (DWT), Distributed Arithmetic (DA), Lifting schema, VLSI, FPGA.

I. INTRODUCTION

In today's world, Electronic equipment comes with userfriendly interfaces such as keypads and graphical displays. As we know, Images convey more information than words, these equipment's today have image displays and interfaces. Image storage on these smaller, handled devices is a challenge as they occupy huge storage space; also image transmission requires higher bandwidth. Hence most of the signal processing technologies today has dedicated hardware's that act as co-processors to compress and decompress images.

The Discrete Wavelet Transform (DWT) has become one of the most used techniques for image compression and is applied in a large category of applications. The application of DWT comes with digital photography, medical imaging, internet, satellite imaging, FBI fingerprint image compression. DWT can provide significant compression ratios without great loss of visual quality than the previous techniques such as the Discrete Cosine Transform (DCT) and the Discrete Fourier Transform (DFT). The DWT present the main part of theJPEG2000 standard, which permits both lossy and lossless compression of digital images. With the increasing use of multimedia technologies, image compression requires higher performance. To address needs and requirements of multimedia and internet applications, many efficient image compression techniques, with considerably different features, have been developed. As the technology of semiconductor industries has enormous growth it has led to unpredictable demand for low power, high speed complex and reliable integrated circuits. These circuits are used for medical, defence and consumer applications.

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, pattern recognition, data compression and so on. Implementing DWT on FPGA can extend FPGA's application on digital signal processing.

II. OVERVIEW OF DISCRETE WAVELET TRASFORM

A .One-dimensional discrete wavelet transformAlgorithm

In the 1990s, the French scholar Mallat raised a fastdiscrete wavelet transform algorithm named the Mallat algorithm using of the concept of multiresolution analysis. The algorithm can be described as follows:

$$y_h(n) = \sum_k x(2n-k)h(k)$$
$$y_g(n) = \sum_k x(2n-k)g(k)$$

where x(n) are the input signals, y_g(n) and y_h(n) are the output sequence. It is a one dimensional FIR filtering process in essence. The h (k) and g (k) are high-pass and low-pass filters. The input sequence x (n) bypass the filter banks h (k) and g (k) and extracted by 2, and finally generates the low-frequency signal y_g(n) and high frequency details signal y_h(n).[6]

B. Two-dimensional discrete wavelet transform Algorithm

In the field of image processing, we need to deal with the two-dimensional data matrix. Therefore, the Mallat algorithm should be extended to the two-dimensional space. This is essentially to make twice one-dimensional wavelet transform to the image matrix. Firstly, the filter groups h (n) and g (n) are used for each row of the image to filter and 2 extract, and then each column of the results use the same to filter and 2 extract. In this way, the original image is decomposed into four sub-band images, denoted as LL, LH, HL and HH. Where the LL stands for the horizontal and vertical low signal; the LH for the horizontal direction's low pass and vertical direction's high-pass signal; the HL for the horizontal direction's high pass and vertical direction's signal, and the HH is the

horizontal and vertical direction high-pass signal.[6] The two-dimensional Mallat algorithm structure is shown in Fig.1.

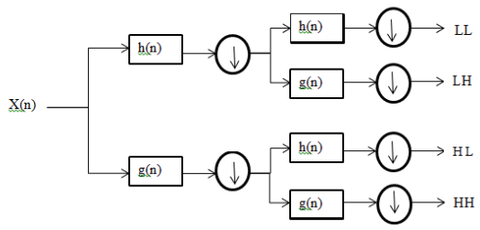

Fig.1 Two dimensional Mallat algorithm structure[6]

III. APPROCHES FOR IMPLEMENTING DWT

A. Distributive Arithmetic based 2D DWT architecture

In this section, we first outline how to perform multiplication by using memory based architecture. Following this, we briefly explain architecture for DWT filter bank. Using this we show complete design for block based DWT. The memory based approach provides an efficient way to replace multipliers by small ROM tables such that the DWT filter can attain high computing speeds with a small silicon area as shown in Figure 2. Traditionally, multiplication is performed using logic elements such as adders, registers etc. However, multiplication of two n-bit input variables can be performed by a ROM table of size of 22n entries. Each entry stores the pre-computed result of a multiplication. The speed of the ROM lookup table is faster than that of hardware multiplication if the look-up table is stored in the on-chip memory. In DWT, one of the input variables in the multiplier can be fixed. Therefore, a multiplier can be realized by 2n entries of ROM. Distributed arithmetic implementation of the Daubechies 8-tap wavelet FIR filter consists of an LUT, a cascade of shift registers and a scaling accumulator

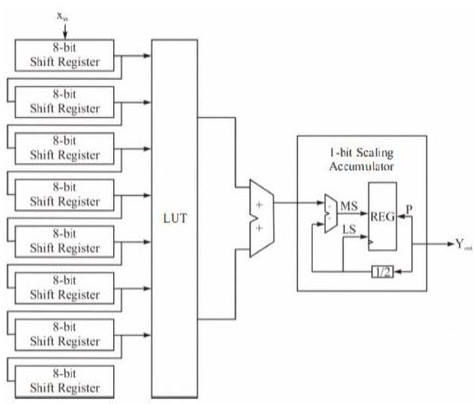

Fig. 2 Distributive Arithmetic[1]

To speed up the process parallel implementation of the Distributive Arithmetic (DA) architecture shown in Figure 3 is realized. In parallel implementation, the input data is divided into even samples and the odd samples based on their position. This scheme reduces the memory size to half due to the symmetric property of the filter coefficients. This increases the through put as the input samples are simultaneously used to read the data from two LUTs and hence speed is increased.

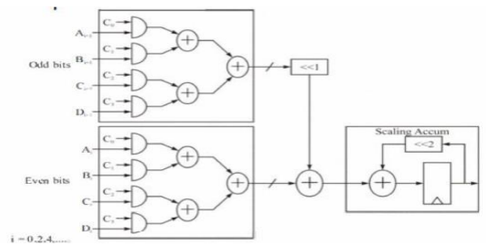

Fig. 3 Parallel Implementation of DA architecture[1]

B. Lifting Schema based DWT Architecture

The lifting scheme is an algorithm used for implementation hardware and software of DWT; it is constituted of steps of

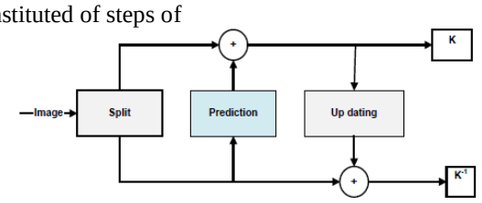

Fig. 4:-Lifting Scheme Forword Transform[4]

predictions and updating described by the Fig.4 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:

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

where $h_e(z)$ and h_o (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:

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

The filters $h_e(z)$, $h_o(z)$, $g_e(z)$ and $g_o(z)$ 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 polyphasep(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

IV. LIETRATURE REVIEW

[1]In this paper, a modified Distributive Arithmetic based DWT architecture is proposedand isimplemented on FPGA.

V. MODIFIED DA-DWT ARCHITECTURE

The modified DA-DWT architecture shown in Figure 5 consists of four LUTs, each of the LUTs are accessed by the even and odd samples of input matrix simultaneously. Odd and even input samples are divided into 4 bits of LSB and 4 bits of MSB, each 4-bit data read the content of four different LUTs that consist of partial products of filter values computed and stored as per the DA logic. Input samples are split into even and odd in the first stage, the data is further loaded sequentially into the serial inserial out shift registers, top four shift register store MSB bits and bottom four shift register stores the LSB bits. It requires 40 clocks cycles to load the shift register contents. At the end of 40th clock cycle, the control logic configures the shift register as serial in parallel out, thus forming the address for the LUT. The partial products stored in the LUT are read simultaneously from all the four LUTS and are accumulated with previous values available across the shift register in the output stage. The output stage consisting of adders, accumulators and right shift registers are used to accumulate the LUT contents and thus compute the DWT output. This architecture has a latency of 44 clock cycles in computing the first high pass and low pass filter coefficients, and has a through put of 4 clock cycles. This architecture is faster architectures as the latency is reduced by half clock cycles and through put is increased by a factor of 2.

[2]In this paper, Field programmable gate array (FPGA) Spartan3EXC3S1200e-4fg320 of Xilinx Inc. over Nexys2 development board of Digilent Inc. was selected as primary target technology. Design tools used were MATLAB R2010b (Wavelet Toolbox 4.6, Filter Design Toolbox 4.7.1 and Filter Design HDL Coder 2.7) from The MathWorks Inc. for wavelet filter banks design, DA HDL code generation and user interface for running and debugging purposes. Other tools used were ISE WebPack 12.4 from Xilinx Inc. for implementation ofdesigned filters, ADEPT 2.6.1 System and Adept 2.0.1 SDK from DigilentInc. for programming and data transmission from the register bus implemented on the FPGA system.First, both Daubechies wavelets coefficients filters bank db1 to db8 are loaded into the MATLAB environment by the Wavelet Toolbox and passed as parameters to the Filter Design Toolbox to generate a FIR structures.Then, LUT partition is calculated by considering the lengthof filters and LUTs of 4 inputs.The sum of the partitions must be equal to the length of the filter.

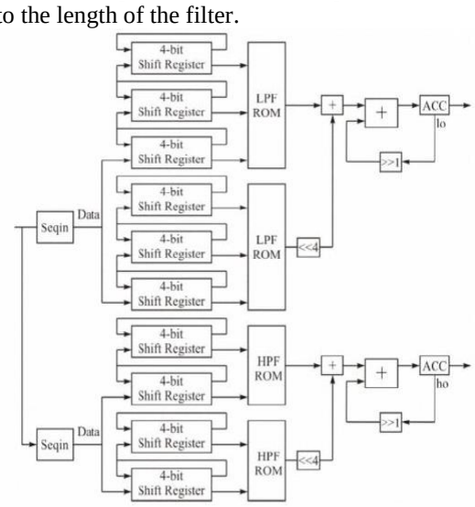

Fig 5:- Practical parallel implementation of DA-DWT technique[1]

Finally, filters are coded in VHDL language by the Filter Design HDL Coder for DA and parallel architectures. Test bench files are also generated with impulse, chirp and white noise signals for both architectures types. Once the filters designed, system is implemented with the ISE WebPack tool. Epp bus with 8 bits registers was implemented with USB-EPP and EPP-Wavelet interfaces along with a set of eight wavelet analysis filters banks, and decimation,data path control and synchronization modules. Each filter bank is similar to filter prototype Wdb1DA shown in fig. 6.

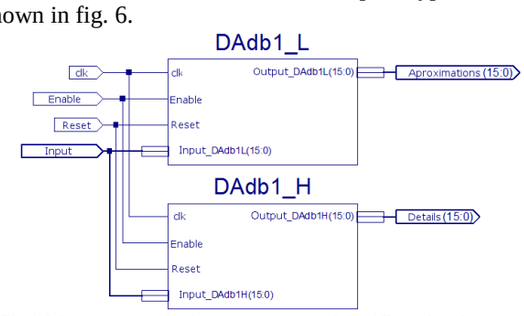

Fig 6 Wavelet analysis filter bank module Wdb1DA with signals and routings[2].

The implementation include a set of eight wavelet analysis filter banks with the necessary control for multiplexing data path as shown in fig. 7. Total filters implemented was sixteen, i.e. low pass filter and high pass filter for each bank. Each filter ranges from first to eighth order.

Wavelet Module implemented with synchronization anddecimation modules are shown in fig.8. EnablePulse sub modulegenerate a pulse of 16 clock cycles to process

every data when enable signal goes high in one clock cycle. Signal generated from this sub-module serves as clock to the decimation sub-modules decimationby2. Module also outputs a decimation state signal.

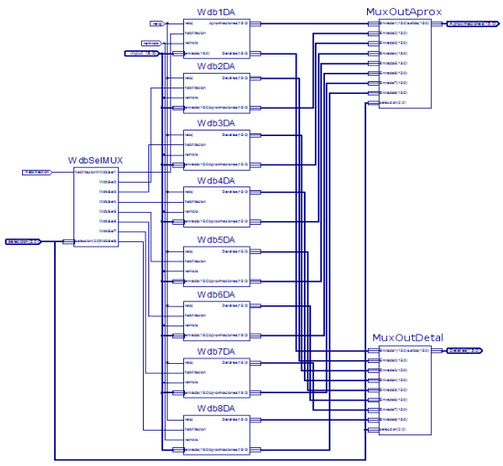

Fig. 7 Module WdbNda01; Set of eight wavelet analysisfilters banks with multiplexing data path modules for wavelet selection.[2]

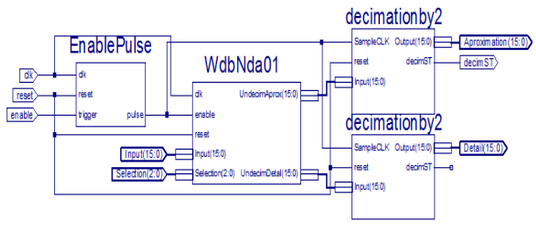

Fig 8 Implemented wavelet module

[3] In this paper, they proposed architecture of high-speed DWT is shown in Figure 9, it includes the input interface module, the high/low filter module and the output interface module. Boundary extension module eliminates data redundancy as a filter pre-treatment. High-pass and low-pass filters are realized by distributed algorithm. After a 2-extraction module, the results of high-pass and low-pass filters are selected to output by a multiplexer module.In this architecture, high-pass and low-pass filters are parallel for optimizing processing speed.

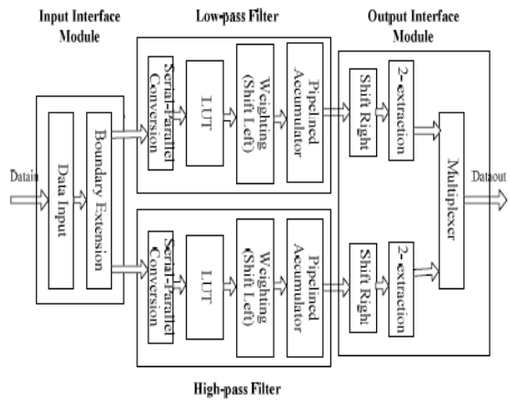

Fig 9 Architecture of high speed DWT[3]

It is worth to mention that in order to obtain high precisionoutput, internal filter coefficients in Figure 9 are enlarged by 213 and then decreased the same scale when data is output. By this way, although it is still a fixed-point calculation,the accuracy of this design will greatly increased. Since above calculation need long chain of adders, the system clock frequency is limited. In order to meet the requirement of real-time application, we propose the pipelined adder chain structure based on the improved DA Algorithm.

The direct sum of long data list is break into some two-input adders, in which middle registers are involved to storage intermediate results. These two input adders are parallel, so the rate of algorithm calculation is greatly improved and the system clock frequency is improved. After above optimizations, the maximum clock frequency of DWT module is increased from 93.26MHz to 217.72MHz.

[4]In this paper, lifting schema technique is used.

Fig.10 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.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.

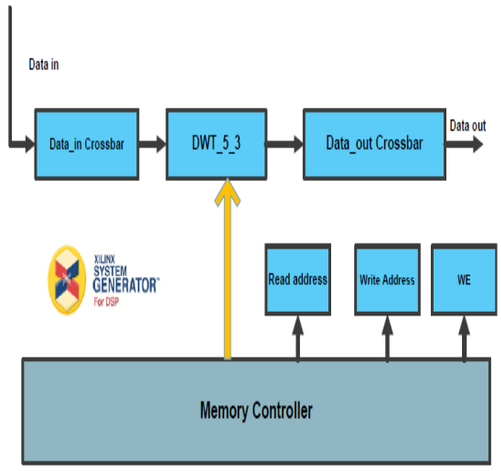

Fig. 10 DWT2D Processor[4]

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.

VI. SYNTHESIS RESULTS AND DISCUSSIONS

In this paper, we have seen various approaches for implementing Discrete Wavelet Transform on FPGA. So in this section we compare the results of these techniques. The comparison is given below.

Paper	Device	No. of slices	No.ofFlipflops	No. of LUT's	No.of GCLK's
1	XC2Vp30ff8987	832	634	1196	1
2	XC3S1200e-4fg320	220	---	583	1
3	StratxIIEP2S15F484C3	FPGA Area –10719 LE's			
4	XC3SD3400A-4FGG676C	272	497	344	1

VII. CONCLUSION

From above paper, we can conclude that,discrete wavelet transform can be implemented using various methods. Mostly Distributed Arithmetic approach is used because of its high speed.

REFERENCES

[1] Mahesh Goparaju, S Mohan" Design and Implementation ofDWT-IDWT Algorithm for Image Compression by using FPGA" ISSN 2250-3153, 2013.

[2] E.A.Raviera-Juarico,J.M.Ramirez-Cort'es,V.Alarcon-Aquino and J.Escamilla-Ambrosio"Design and Implementation of the Discrete Wavelet Transform on an FPGA Platform to Process Data Sets of up to Three Dimensions"978-1-61284-1325-5/12 ©2012 IEEE

[3] Qijun Huang, YajuanWang and ShengChang, "High-performance FPGA Implementation of Discrete Wavelet Transform for Image Processing"978-1-4244-6554-5/11.2011 IEEE.

[4] Taoufik SAIDANI Mohamed ATRI,Yahia SAID and Rached TOURKI, " Real Time FPGA acceleration for Discrete Wavelet Transform of the 5/3 Filter for JPEG 2000 "

[5] Sani John, Manu Jose, "Optimized Implementation of Discrete Wavelet Transform with Area Efficiency" 978-1-4673-5090-7/13©2013 IEEE

[6] Juan Li1 Binghua Su1, Yongming Yan2 , Caigao Jiang, "Discrete Wavelet TransformImplementation Based on FPGA"ICSP2012 Proceedings

[7] Ms.Yamini S. Bute,Prof. R. W. Jasutkar "Implementation of Discrete wavelet Transform Processor for Image Compression "ISSN 2277-5420,2012.