International Journal of Electrical and Electronic Engineering & Telecommunications

ISSN 2319 – 2518 www.ijeetc.com
Vol. 3, No. 4, October 2014

Research Paper

A HIGH SPEED OPTIMIZATION FOR FAST COMPUTATION OF 2-D DWT FOR COMPENSATING MEMORY ERRORS IN JPEG2000

Kartheek Itha[1]* and H Raghunadh Rao[2]

Corresponding Author: Kartheek Itha ✉ kartheek.itha1@gmail.com

The fundamental idea behind wavelets is to analyze according to scale. Indeed, some researchers in the wavelet field feel that, by using wavelets, one is adopting a perspective in processing data. Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. In this paper, separable pipeline architecture for fast computation of the 2D DWT with a less memory and low latency is proposed. The low latency and less memory is achieved by proper designing of two 1-D DWT filtering processes and also efficiently transferring the data between the two 1-D DWT filters. This 2D DWT is responsible for checking is there any errors are there or not in Memories.

Keywords: DWT, Separable and Non-Separable, Xilinx ISE, Modelsim, Verilog

INTRODUCTION

The 2-D discrete wavelet transforms (DWT) have been widely used in many applications like image compression, signal processing, speech compression because of their multi-resolution of signals with localization both in time and frequency. In the past, much architecture have been proposed aimed at providing high – speed 2-D DWT computation with the requirement of utilizing a reasonable amount of hard ware resources. These architectures can be broadly classified into separable (Vishwanath M *et al.*, 1995; Chakrabarti C and Vishwanath M, 1995; Liao H Y *et al.*, 2004; Guevorkian D *et al.*, 2005; Alam M *et al.*, 2005) and non separable architectures (Chakrabarti C and Vishwanath M, 1995; Liao H Y *et al.*, 2004; Guevorkian D *et al.*, 2005; Alam M *et al.*, 2005; Yu C and Chen S J, 1997). The separable method is the most straight forward implementation method. In separable method, a 2-D filtering operations, one for processing the data row-wise and the other column-wise. In this method the intermediate coefficients stores in a frame memory first. Then it performs 1-D DWT in

[1] M.Tech. Student, Department of ECE, Chirala Engineering College, Chirala 523155, Prakasam Dt., AP.
[2] Associate Professor, Department of ECE, Chirala Engineering College, Chirala 523155, Prakasam Dt., AP.

other direction with these intermediate coefficients to complete one-level 2-D DWT .Because the size of this frame memory is usually assumed to off chip.

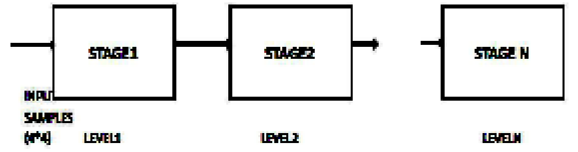
However, the separable method performs 1-D DWT in both directions simultaneously. Thus in separable architectures, in which a 1-D filtering structure is used to perform the 2-D DWT, have an additional requirement of transposing the intermediate data between the two 1-D filtering processes. Hence the separable method does not require a frame memory to store the intermediate data. Instead, some internal line buffers are used to store the intermediate data, and the required size is proportional to the image width. Vishwanath *et al*. (1995) have proposed low-storage short-latency separable architectures in which the row wise operations are performed by systolic filters and the column operations by parallel filters. In non separable architectures the 2-D transforms are computed directly by using 2-D filters. chakrabarti *et al*. (1995) have proposed two non separable architectures, one using parallel 2-D filters and the other an SIMD 2-D array ,both based on a modified RPA. In non separable method internal line buffers are use to store the boundary data among neighbor blocks such as to keep the required external frame memory bandwidth as low as the separable method. However, the external memory access would consume the most power (Vishwanath M *et al*., 1995) and become very sensitive in the case of system on chip. In addition, the required external memory bandwidth of the non-separable is more than the double of the separable method. The most existing separable architectures aim at providing fast computation of the DWT by using pipeline and a large number of parallel filters and systolic filters. However these existing architectures have large latency and memory size increases because of by providing additional requirement of transposing the intermediate data between the two 1-D filtering processes.

PIPELINE FOR THE 2-D DWT COMPUTATION

In a pipeline structure for the DWT computation, multiple stages are used to carry out the computations of the various decomposition levels of the transform [4]. The computation corresponding to each decomposition level needs to be mapped to a stage or stages of the pipeline. In order to design a pipeline structure capable of performing a fast computation of the DWT with low expense on hardware resources and low design complexity, an optimal mapping of the overall task of the DWT computation to the various stages of the pipeline needs to be determined. Any distribution of the overall task of the DWT computation to stages must consider the inherent nature of the sequential computations of the decomposition levels that limit the computational parallelism of the pipeline stages, and consequently the latency of the pipeline. Further, in order to minimize the expense on the hardware resources of the pipeline, the number of filter units used by each stage ought to be minimum and proportional to the amount of the task assigned to the stage.

A straightforward of mapping of the overall task of the DWT computation to a pipeline is one-level to one-stage mapping, in which the

Figure 1: Pipeline Structure with N Stages



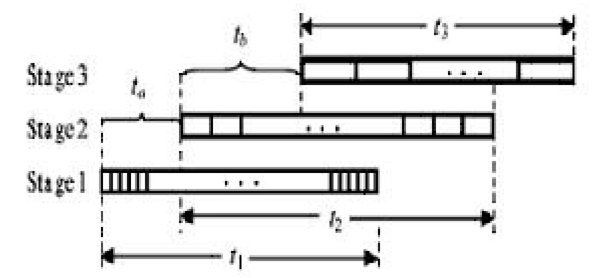
tasks of J decomposition levels are distributed to J stages of the pipeline. However, dividing a stage of the one-level to one stage pipeline into multiple stages would require a division of the task associated with the corresponding decomposition level into sub-tasks, which in turn, would call for a solution of even a more complex problem of synchronization of the sub-tasks associated with divided stages. On the other hand, merging multiple small-size stages of the pipeline into one stage would not create any additional synchronization problem. As a matter of fact, such a merger could be used to reduce the overall number of filter units of the pipeline.

SYNCHRONIZATION OF STAGES

The distribution of the computational load among the three stages, and the hardware resources made available to them are in the ratio 8:2:1. The stages of pipeline need to be synchronized in such a way that each stage starts the operation at an earliest possible time when the required data become available for its operation. Once the operation of a stage is started, it must continue until the task assigned to it is fully completed. Consider the timing diagram given in Figure 2 for the operation of the three stages, where t_1,t_2 and t_3 are the times taken individually by stages 1,2 and 3, respectively, to complete their assigned tasks, and t_a and t_b are the times elapsed between the starting points of the tasks, by stages 1 and 2, and that stages 2 and 3 respectively.

Note that the lengths of the times t_1,t_2 and t_3 to complete the tasks by individual stages are approximately the same, since the ratios of the tasks assigned and the resources made available to the three stages are the same. The average times to compute one output sample by stages 1,2 and 3 are in the ratio 1:4:8. In Figure 2 the relative widths of the slots in the three stages are shown to reflect this ratio. Our objective is to minimise the total computation time t_a+t_b+t_3 by minimizing t,t and t individually.

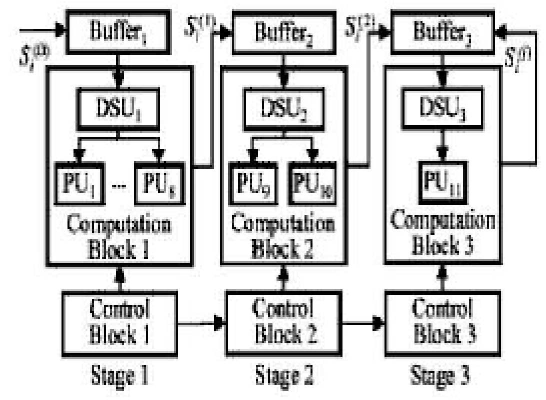
Figure 2: Timing Diagram for the Operations Three Stages



A. Design of stages

In the proposed three-stage architecture, stages 1 and 2 perform the computations of levels 1 and 2 respectively, and stage 3 that of all the remaining levels. Figure 3 shows the block diagram of the three-stage architecture.

Figure 3: Block Diagram of Diagram of the Three-stage Architecture



DIFFERENT TYPES OF TRANSFORMS

1. FT (Fourier Transform).
2. DCT (Discrete Cosine Transform).

3. DWT (Discrete Wavelet Transform).

A. Discrete Fourier Transform

The DFT representation for a finite duration sequence is

$$X(j\omega) = \sum_{n=-\infty}^{\infty} e^{-j\omega n} \quad ...(1)$$

$$X(n) = \frac{1}{2\pi}\int X(j\omega)e^{-j\omega n}d\omega \quad ...(2)$$

where x(n) is a finite duration sequence, X(jω) is periodic with period 2π.It is convenient sample X(jω) with a sampling frequency equal an integer multiple of its period =m that is taking N uniformly spaced samples between 0 and 2π.

Let

$$\omega = \frac{2\pi k}{N} \quad 0 \le k \le N-1 \quad ...(3)$$

Therefore

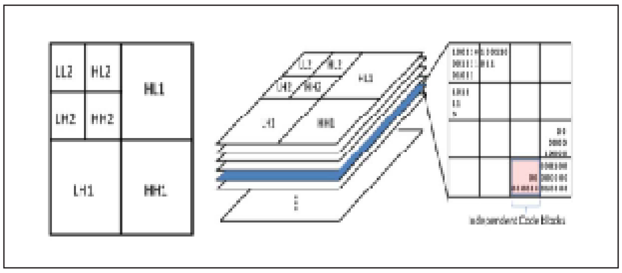
$$X(j\omega) = \sum_{n=-\infty}^{\infty} xne^{\frac{-j2\pi kn}{N}} \quad ...(4)$$

Since X(jω) is sampled for one period and there are N samples X(jω) can be expressed as

$$X(k) = X(j\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{\frac{-j2\pi kn}{N}} \quad ...(5)$$

B. The Discrete Cosine Transform (DCT)

The discrete cosine transform (DCT) helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT is similar to the discrete Fourier transform: it transforms a signal or image from the spatial domain to the frequency domain.

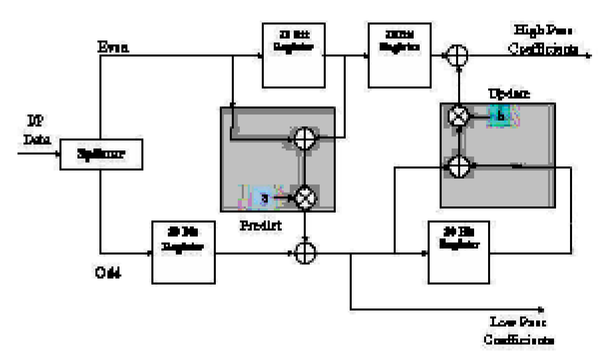


C. Discrete Wavelet Transform (DWT)

The discrete wavelet transform (DWT) refers to wavelet transforms for which the wavelets are discretely sampled. A transform which localizes a function both in space and scaling and has some desirable properties compared to the Fourier transform. The transform is based on a wavelet matrix, which can be computed more quickly than the analogous Fourier matrix. Most notably, the discrete wavelet transform is used for signal coding, where the properties of the transform are exploited to represent a discrete signal in a more redundant form, often as a preconditioning for data compression. The discrete wavelet transform has a huge number of applications in Science, Engineering, Mathematics and Computer Science.

Wavelet compression is a form of data compression well suited for image compression (sometimes also video compression and audio compression). In this the values a and b is taken as 1 for digital design purpose. If we take those values as 0 then it became Zero response. The splitter is used to split the values and the 20 bit registers are used to store the values and transfer the values. The adders are used to add the original and delay values of the register discrete cosine transform, had been used. First a wavelet transform is applied. This produces as many coefficients as there are pixels in the image (i.e.: there is no compression yet since it is only a transform). These coefficients can then be compressed more easily because the information is statistically concentrated in just a few coefficients. Image data in as little space as possible in a file. A certain loss of quality is accepted (lossy Compression).

Figure 4: Proposed Architecture



SIMILARITIES BETWEEN FOURIER AND WAVELET TRANSFORM

The fast Fourier transform (FFT) and the discrete wavelet transform (DWT) are both linear operations that generate a data structure that contains log, n segments of various lengths, usually filling and transforming it into a different data vector of length 2^n. The mathematical properties of the matrices involved in the transforms are similar as well. The inverse transform matrix for both the FFT and the DWT is the transpose of the original. As a result, both transforms can be viewed as a rotation in function space to a different domain. For the FFT, this new domain contains basis functions that are sines and cosines. For the wavelet transform, this new domain contains more complicated basis functions called wavelets, mother wavelets, or analyzing wavelets. Both transforms have another similarity. The basic functions are localized in frequency, making mathematical tools such as power spectra (how much power is contained in a frequency interval) and scale grams (to be defined later) useful at picking out frequencies and calculating power distributions.

A. Dissimilarities between Fourier and Wavelet Transform

The most interesting dissimilarity between these two kinds of transforms is that individual wavelet functions are *localized in space.* Fourier sine and cosine functions are not. This localization feature, along with wavelets' localization of frequency, makes many functions and operators using wavelets "sparse" when transformed into the wavelet domain. This sparseness, in turn, results in a number of useful applications such as data compression, detecting features in images, and removing noise from time series.

B. Applications of DWT

Generally, an approximation to DWT is used for data compression if signal is already sampled, and the CWT for signal analysis. Thus, DWT approximation is commonly used in engineering and computer science, and the CWT in scientific research. One use of wavelet approximation is in data compression. Like some other transforms, wavelet transforms can be used to transform data and then encode the transformed data, resulting in effective compression. For example, JPEG 2000 is an image compression standard that uses bi-orthogonal wavelets. A related use is that of smoothing/denoising data based on wavelet coefficient thresholding, also called wavelet shrinkage. By adaptively thresholding the wavelet coefficients that correspond to undesired frequency components smoothing and/or denoising operations can be performed. Other applied fields that are making use of wavelets include astronomy, acoustics, nuclear engineering, sub-band coding, signal and image processing, neurophysiology, music, magnetic resonance imaging, speech discrimination, optics, fractals, turbulence, earthquake-prediction, radar, human vision, and pure mathematics applications such as solving partial differential equations.

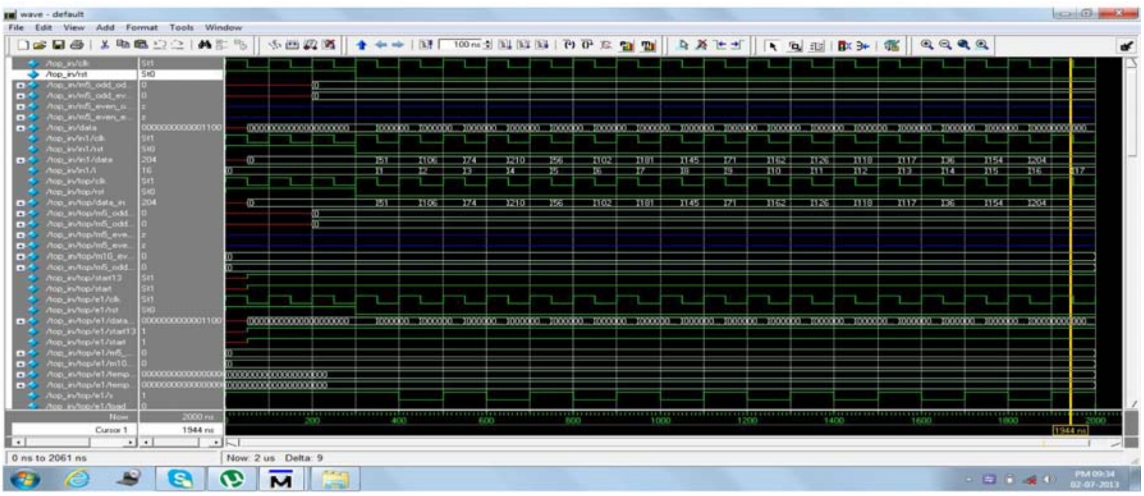
C. Wavelets for image compression

Wavelet transform exploits both the spatial and frequency correlation of data by dilations (or contractions) and translations of mother wavelet on the input data. It supports the multi-resolution analysis of data i.e. it can be applied to different scales according to the details required, which allows progressive transmission and zooming of the image without the need of extra storage. Another encouraging feature of wavelet transform is its symmetric nature that is both the forward and the inverse transform has the same complexity, building fast compression and decompression routines. Its characteristics well suited for image compression include the ability to take into account of Human Visual System's (HVS) characteristics, very good energy compaction capabilities, robustness under transmission, high compression ratio etc.

Wavelet transform divides the information of an image into approximation and detail sub-signals. The approximation sub-signal shows the general trend of pixel values and other three detail sub-signals show the vertical, horizontal and diagonal details or changes in the images. If these details are very small (threshold) then they can be set to zero without significantly changing the image. The greater the number of zeros the greater the compression ratio. If the energy retained (amount of information retained by an image after compression and decompression) is 100% then the compression is lossless as the image can be reconstructed exactly. This occurs when the threshold value is set to zero, meaning that the details have not been changed.

SIMULATION RESULTS

Figure 5: Top Module



20-bit register : 13

Comparators : 6

3-bit comparator greater : 1

3-bit comparator lessequal : 2

4-bit comparator greater : 1

4-bit comparator lessequal : 2

Total : 16.917ns (11.928ns logic, 4.990ns route)

(70.5% logic, 29.5% route)

Total memory usage is 198724 kilobytes, Comparison of Proposed and existing architectures:

Table 1: Comparision Table

Device Utilization Summary (Estimated values)				
Logic Utilization	Proposed method	Existing method	Utilization	
			Proposed	Existing
Number of Slices	532	2852	18%	27%
Number of Slice Flip Flops	543	1059	51%	59%
Number of 4 input LUTs	672	4989	13%	18%
Number of bonded IOBs	94	130	72%	79%
Number of GCLKs	1	8	4%	5%

CONCLUSION

In this paper, separable pipeline architecture for fast computation of the 2-D DWT with a less memory and low latency is proposed. The low latency and less memory is achieved by proper designing of two 1-D DWT filtering processes and also efficiently transferring the data between the two 1-D DWT architectures. This architecture is simulated, synthesized and implemented by VERILOG language using XILINX ISE Tool. From the LL output it can conclude that the error occurance is less in memory oriented applications.

REFERENCES

1. Alam M, Badawy W, Dimitrov V, and Jullien G (2005), "An efficient architecture for a lifted 2-D biorthogonal DWT," *J. VLSI Signal Process.*, Vol. 40, pp. 333–342.
2. Chakrabarti C and Vishwanath M (1995), "Efficient realizations single chip implementations to mapping on SIMD array computers," *IEEE Trans. Signal Process.*, Vol. 43, No. 3, pp. 759–771.
3. Guevorkian D, Liuha P, Launiainen A, and Lappalainen V (2005), "Architectures for Discrete Wavelet Transforms," US 6976046, Dec. 13.
4. Huang C T, Tseng P C, and Chen L G (2004), "Memory Analysis and Architecture for Two-Dimensional Discrete Wavelet Transform," in Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. V13–V16.
5. Liao H Y, Mandal M K, and Cockburn B F (2004), "Efficient architectures for 1-D and 2-D lifting-based wavelet transforms," *IEEE Trans. Signal Process.*, Vol. 52, No. 5, pp. 1315–1326.
6. Tseng P C, Huang C T, and Chen L G (2002), "Generic RAM Based Architecture for Two-Dimensional Discrete Wavelet Transform with Line-Based Method," in Proceedings of the Asia-Pacific Conference on Circuits and Systems, pp. 363–366.

7. Vishwanath M, Owens R, and Irwin M J (1995), "VLSI architectures for the discrete wavelet transform," *IEEE Trans. Circuits Syst. II, Analog. Digit. Signal Process.*, Vol. 42, No. 5, pp. 305–316.

8. Vishwanath M, Owens R, and Irwin M J (1995), "VLSI architectures for the discrete wavelet transform," *IEEE Trans. Circuits Syst. II, Analog. Digit. Signal Process.*, Vol. 42, No. 5, pp. 305–316.

9. Wu P C and Chen L G (2001), "An efficient architecture for two-dimensional discrete wavelet transform," *IEEE Trans. Circuits Syst. Video Technol*., Vol. 11, pp. 536–545.

10. Yu C and Chen S J (1997), "VLSI implementation of 2-D discrete wavelet transform for real-time video signal processing," *IEEE Trans. Consum. Electron.*, Vol. 43, No. 4, pp. 1270–79.