VOL. 10, NO. 1, JANUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SPEED EFFICIENT VLSI DESIGN OF LIFTING BASED 2D DWT ARCHITECTURE USING VEDIC MATHEMATICS

S. SenthilKumar[1], R. Radhakrishnan[2]
[1]Department of Electronics and Communication Engineering, SVS College of Engineering, Coimbatore, Tamil Nadu, India
[2]Vidya Mandhir Institute of Technology, Erode, Tamil Nadu, India
E-Mail: sentheeyan@yahoo.co.in

ABSTRACT

This paper presents VLSI architecture for lifting based 2D DWT architecture with reduced delay. The proposed structure offers high speed and high area efficiency. Fast computation is achieved by replacing conventional multiplier units of DWT architecture with Vedic multiplier. Three sutras of Vedic multiplication are employed to reduce logic shifting operations of multiplier units and so high speed is made possible. The computation techniques of three DWT structures have been compared to prove high performance of proposed 2D DWT architecture. The proposed 2D DWT architecture is modelled and implemented using XILINX ISE 9.1.

Keywords: image compression, discrete wavelet transform (DWT), lifting scheme, high speed, vedic multiplication.

INTRODUCTION

Many efficient image compression techniques have been developed to address the needs of multimedia and internet applications. Traditionally, image compression adopts discrete cosine transform (DCT) in most of the applications. However, the compression method that adopts DCT has several shortcomings such as obvious blocking artifact and bad subjective quality when the images are restored by the method with the high compression ratios. Image compression is one of the most visible applications of wavelets. So, much of the research activities in image coding have been focused on the DWT in recent times. DWT can provide significant compression ratios without great loss of visual quality than the previous techniques such as Discrete Cosine Transform (DCT) and Discrete Fourier Transform (DFT). The DWT compression phase is mainly divided into three sequential steps: (1) Discrete Wavelet Transform, (2) Quantization and (3) Entropy coding. After preprocessing, each component is independently analyzed by an appropriate discrete wavelet transform [1]. The source image is divided into rectangular non-overlapping blocks. These blocks are known as image tiles and these tiles are independently compressed as per JPEG 2000 standard. The wavelet transformation is performed on each tile independently and so tiling reduces memory requirements [2]. Basic lifting scheme approach has been proposed to implement 9/7 DWT filter by using straight forward transformation [3].

LIFTING SCHEME AND VEDIC MATHEMATICS

In Lifting scheme, same architecture can be used for both forward and inverse transform. The inverse transform goes from right to the left, inversing the normalized coefficients and changes positive sign to negative. Figure-1 shows block diagram of lifting scheme [4]. The basic idea behind the lifting scheme is very simple. The correlation in the data is used to remove redundancy. In the split phase, data splits into odd samples and even samples as shown in Figure-1. Because of the assumed smoothness of the data, it is predicted that the odd samples have a value that is closely related to their neighboring even samples. In the prediction phase, N even samples are used to predict the value of a neighboring odd sample. Odd sample can be replaced with the difference between the odd sample and its prediction. In the similar way, even samples can also be transformed.

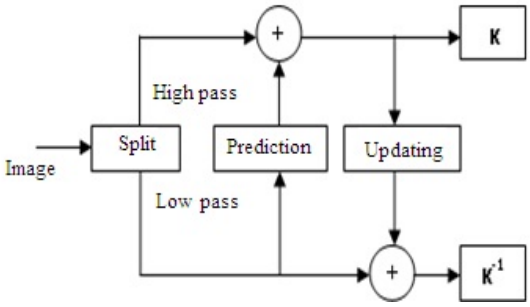


Figure-1. Block diagram of lifting scheme.

In case of images, intensity (mean of the samples) has to be kept constant throughout different levels. The third step (update phase) updates the even samples using the newly calculated odd samples and so desired property can be preserved. The lifting scheme algorithm of 9/7 filter can be performed as follows [5]:

- Split step: The original signal is split into odd and even samples.
- Lifting step: This step is executed as N number of sub-steps depending on the type of the filter where the odd and even samples are filtered by the prediction and update filters.
- Scaling normalization step: After N lifting steps, scaling coefficients K and K^{-1} are applied respectively to the odd and even samples so as to obtain the low pass sub-band and high pass sub-band.

Vedic mathematics is mainly based on 16 Sutras dealing with various branches of mathematics like arithmetic, algebra and geometry. They are namely 1) Urdhva-Tiryagbyham*, 2) Nikhilam Navatashcaramam Dashatah*, 2.a) *Anurupyena* (upa sutra), 3) Ekadhikina purvena,* 4) Paraavartya Yojayet, 5) Shunyam Saamyasamuccaye, 6) (Anurupye) Shunyamanyat, 7) Sankalanavyavakalanabhyam, 8) Puranapuranabyham, 9) Chalana-Kalanabyham, 10) Yaavadunam, 11) Vyashtisamanstih 12) Shesanyankena Charamena, 13) Sopaantyadvayamantyam, 14) Ekanyunena Purvena 15) Gunitasamuchyah and 16) Gunakasamuchyah.

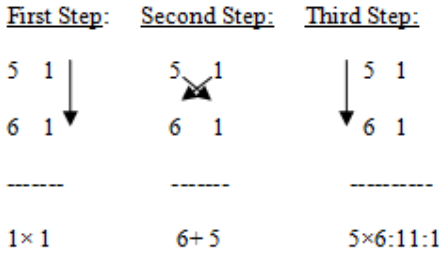
Among these sutras, urdhuva tiryaghyam, nikhilam and anurupyena sutras (first three sutras) can be employed for fast multiplication [6]. Fast multiplication is very much essential to make lifting scheme efficient. Multiplier is one of the arithmetic building blocks that consumes the maximum area and also constitutes delay. Vedic multiplication reduces area by using less number of logic elements and increases computation speed by reducing logical shifting operations.

VEDIC MULTIPLICATION

Urdhva tiryagbhyam

The basic sutras and upa sutras in the vedic mathematics helps to do almost all the numeric computations in simple way and fast manner. The first proposed multiplication sutra is Urdhva Tiryagbhyam. Urdhva-tiryagbhyam is a formula applicable to all cases of multiplication [7]. It means "vertically and cross wise".

Example: Find the product of 51×61. The procedure of multiplication is as follows:



Which gives 3111.

Nikhilam Sutra

'All from nine and last from ten' is the literal meaning of Nikhilam Sutra. It finds out the compliment of the large number from its nearest base to perform the multiplication operation on it. Hence the complexity of the multiplication is lesser when the number is larger.

Example: Find the product of 502 x 603.

The nearest base for the above two numbers are 500. Both the numbers are greater than the base. Subtract base (500) from given number to calculate deficiency.
502-500=+002
603-500=+103 and, align numbers as follows,

502 +002
603 +103

Left hand side is 605 | (+ 206) is right hand side. The left hand side has got by adding diagonally on any one side and the right hand side is the result of multiplying the deficiencies. (605/2) | (+206) since we have fixed base 500 and 500 is 1000/2. So, we will get 302.5|206. The left hand side has a decimal point, so carry out that half to the R.H.S from L.H.S.

Now, Right hand side becomes 206+500=706. 302|706. Therefore, the answer of the given product (502 x 603) is equal to 3, 02,706.

Anurupyena

The literal meaning of 'Anurupyena' (upa- sutra) is 'proportionality' or 'similarly'. This third technique is highly useful to find products of two numbers if both of them are near common bases like 50, 60 and 300 (multiples of ten).

Example: Find the product of 45×44. In this case, the nearest higher multiples of 10 are 50.

Steps:

a) Find nearest higher multiples of ten of given numbers. i.e., working base is 100/2=50.
b) Subtract each number from 50 and write the difference against each number on right side as follows:

45 -05
44 -06
---

c) Do cross-subtraction and write the answer in the left side. Multiply the differences and write the product in the right side of the answer.

d) The base is 100 / 2 = 50, 39 in the answer represent 39X50. Hence divide 39 by 2 because 50 = 100 / 2. Thus 39÷2 gives 19, where 19 is quotient and 1 is remainder. Reminder 1 gives one 50 and makes the left hand side of the answer 30+50=80. Right hand side 19 and left hand side 80 together make the answer 1, 980.

Figure-2 depicts structure of 1D lifting scheme for 9/7 filter or sub-cell. D-register is delay register and P-

Register is pipeline register. Delay registers are employed for synchronization and pipeline registers are used for improving processing time. The fractional values of lifting constants are listed in the Table-1. Constants α, β, γ and δ are filter coefficients and K is scale normalization factor. Predictor and updater stages have been merged to reduce critical timing delay.

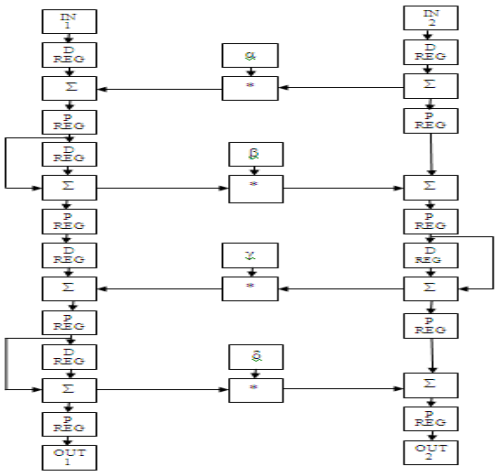


Figure-2. Structure of 1D lifting scheme for 9/7 filter.

Table-1. Lifting constants and their fractional values.

Lifting constants	Fractional values
α	-1.58613
β	0.05298
γ	0.88291
δ	0.44350

The basic processing block of lifting scheme includes one multiplier and two adders. Discrete wavelet transformations can be done by applying the following steps to the entire input. i_0 is an index of the first coefficient of the input and i_1 is an index of the coefficient immediately following coefficient.

$$Y_{2n+1} = X_{2n+1} + \alpha \times (X_{2n} + X_{2n+2}) \quad i_0 - 3 \le 2n+1 < i_1 + 3 \tag{1}$$

$$Y_{2n} = X_{2n} + \beta \times (Y_{2n-1} + Y_{2n+1}) \quad i_0 - 2 \le 2n < i_1 + 2 \tag{2}$$

$$Y_{2n+1} = Y_{2n+1} + \gamma \times (Y_{2n} + Y_{2n+2}) \quad i_0 - 1 \le 2n+1 < i_1 + 1 \tag{3}$$

$$Y_{2n} = Y_{2n} + \delta \times (Y_{2n-1} + Y_{2n+1}) \quad i_0 \le 2n < i_1 \tag{4}$$

$$Y_{2n+1} = -\mathrm{K} \times Y_{2n+1} \quad i_0 \le 2n+1 < i_1 \tag{5}$$

$$Y_{2n} = Y_{2n} / \mathrm{K} \quad i_0 \le 2n < i_1 \tag{6}$$

Desired lifting step can be implemented by providing the appropriate lifting constant to the multiplier. Accuracy of the lifting constants determines the width of the multiplier. Area occupation and speed of multiplier are considered to be the main drawback of this type of implementation. This drawback can be eliminated by employing Vedic sutras in multiplication

RESULTS AND PERFORMANCE COMPARISON

The conventional shift and add multiplier units of DWT architecture have been replaced by multipliers using sutras like urdhuva tiryaghyam, nikhilam and anurupyena (first three sutras*) so as to reduce shift delay and to improve the speed of performance. Proposed architecture (DWT Structure-1) employs urdhuva tiryaghyam sutra, DWT structure-2 employs nikhilam sutra and DWT structure-3 employs anurupyena sutra respectively. Three DWT structures are modeled in VHDL and the results are tabulated and compared in Table-2.

Area efficiency and speed efficiency of three DWT structures have been compared in Figures 3 and 4. Critical path delay of DWT structure is calculated by using T_m and T_a (T_m is the delay time of multiplier and T_a is the delay time of adder of the DWT structure). Delay time of multiplier (T_m) is further reduced in proposed technique.

The proposed 2D architecture using lifting scheme is verified using Model-sim with VHDL code and carried out on SPATRAN 3 development kit with target device XC3S5000. The maximum frequency at which the design works is 25.998 M Hz. Figure-5 shows the synthesis report of proposed architecture using DWT structure-1 with summary of major device utilization in terms of LUT's and slices. Figure-6 shows time report of proposed structure of 2D DWT architecture.

Table-2. Performance comparison of Three DWT structures.

Parameters	DWT Structure-1	DWT Structure- 2	DWT Structure- 3
No. of Slices	1477	3123	5764
No. of Slices Flip-flops	435	456	949
No. of 4 input LUTs	2856	6066	11062
Minimum Period(ns)	38.464	40.005	53.655
Maximum frequency (MHz)	25.998	24.997	18.638

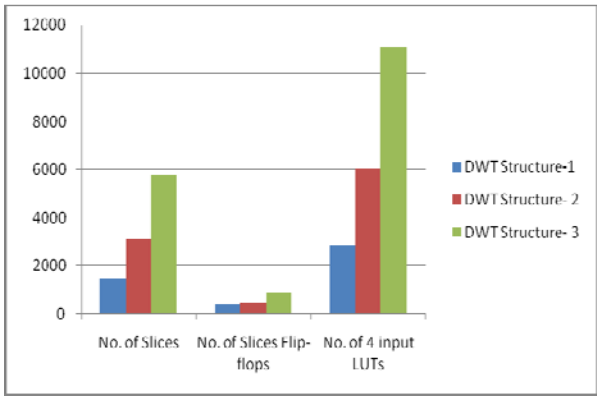


Figure-3. Area comparison.

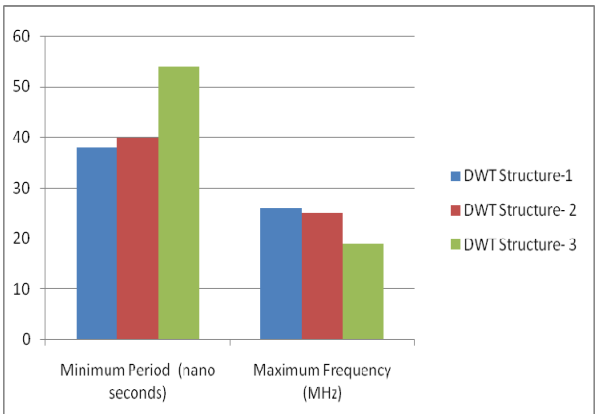


Figure-4. Speed comparison.

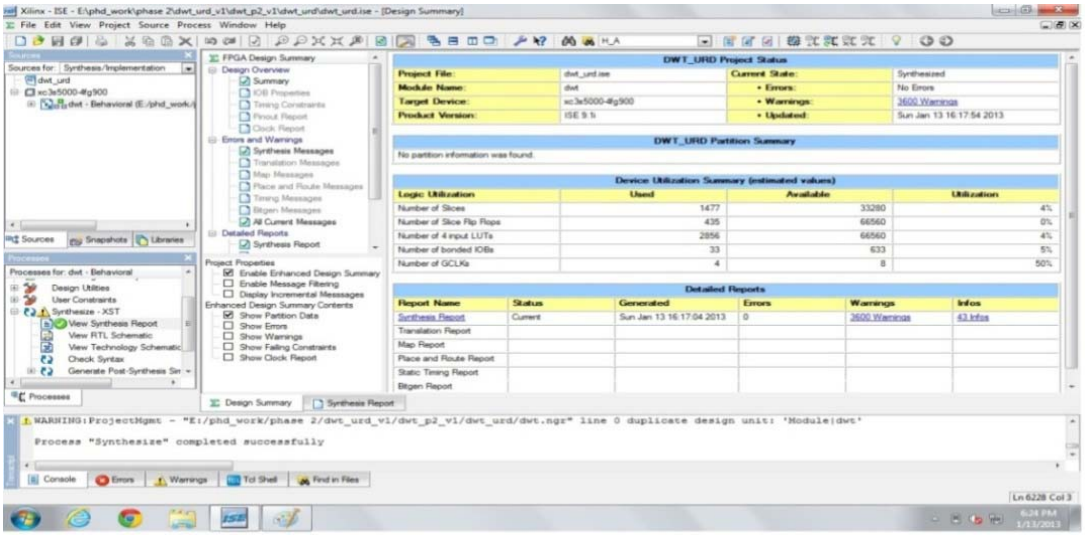


Figure-5. Synthesis report.

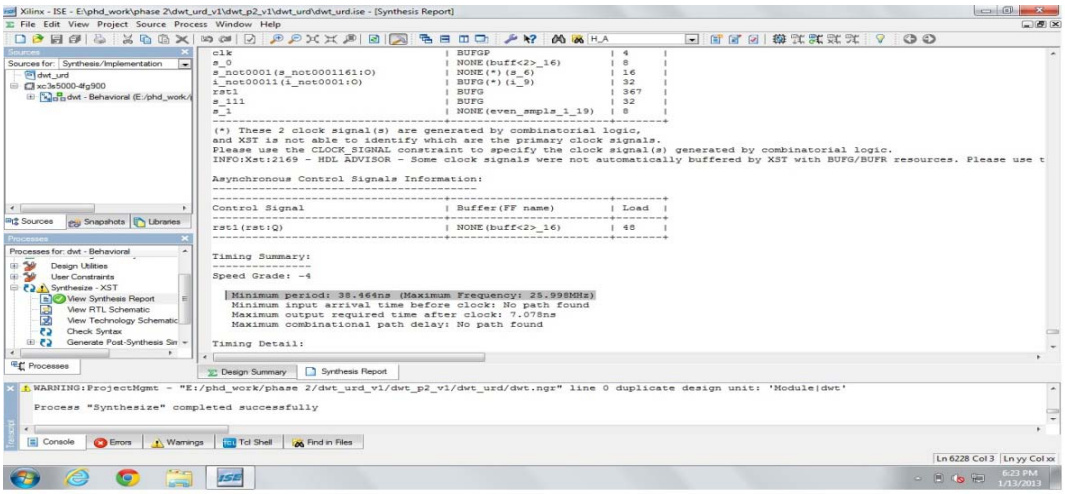


Figure-6. Time report.

CONCLUSIONS

This paper proposes speed efficient and area-efficient VLSI implementation of lifting based 2 D DWT architecture by using 9/7 filter. Maximum HUE has been achieved by using pipeline structures. Area efficiency and speed have been enhanced by employing fast computation technique. Fast computation is achieved in the proposed architecture by using urdhuva tiryaghyam sutra for multiplication. The high performance of proposed structure has been proved in the experimental result. The proposed 2 D-DWT architecture has been carried out on SPATRAN 3 development kit with maximum frequency of 25.998 M Hz.

REFERENCES

[1] Mountassar Maamoun, Abderrahmane Namane, Mehdi Neggazi, Rachid Beguenane, Abdelhamid Meraghni and Daoud Berkani. 2009. VLSI Design for High-Speed Image Computing Using Fast Convolution-Based Discrete Wavelet Transform. Proc. World Congress on Engineering. Vol. 1, WCE 2009, July 1 - 3, London, U.K.

[2] M. Nagabushanam, Cyril Prasanna Raj P, S. Ramachandran. 2009. Design and Implementation of Parallel and Pipelined Distributive Arithmetic Based Discrete Wavelet Transform IP Core. European Journal of Scientific Research. 35(3).

[3] ISO/IEC FCD15444-1: 2000 V1.0. 2001. JPEG 2000 Image Coding System.

[4] B. F. Wu, C.-F. Lin. 2005. High-performance Memory-efficient Pipeline Architecture for 5/3 and 9/7 Discrete Wavelet transform of JPEG 2000 codec. IEEE Trans. Circuits Syst. Video Technol. 15(12): 1615-1628.

[5] Basant K. Mohanty and Pramod Kumar Meher. 2011. Memory Efficiency Modular VLSI Architecture for High throughput and Low Latency Implementation of Multilevel Lifting 2 D DWT. IEEE Trans. Signal Process. 59(5).

[6] Leonard Gibson Moses S and Thilagar M. 2010. VLSI implementation of high speed DSP algorithms using Vedic mathematics. Int. Journal of Computer communication and Information System (IJCCIS). 2(1). ISSN: 0976–1349, July - Dec.

[7] http://vedicmathematics.blogspot.in.

[8] J.M. Jou, Y.H. Shiau and C .C .Liu. 2001. Efficient VLSI architecture for the bi-orthogonal wavelet transform by filter bank and lifting scheme. In Proc. IEEEISCAS. 2: 529-532.

[9] C.J Lian, K. F. Chen, H. H. Chen and L.G. Chen. 2001. Lifting Based Discrete Wavelet Transform Architecture for JPEG2000. In IEEE Int. Symposium on Circuits and Systems, Sydney, Australia. pp. 445-448.

[10] H. Liao, M.K. Mandal and B.F. Cockburn. 2002. Novel architecture for Lifting-Based Discrete Wavelet Transform. In Electronics Letters. 38(18): 1010-1012.

[11] G. Shi, W. Liu and L. Zhang. 2009. An efficient folded architecture for lifting based discrete wavelet transform. IEEE Trans. Circuits Syst, ii, Exp. Briefs. 15(4): 290-294.

[12] C.T. Huang, P.C. Tseng and L.G. Chen. 2004. Flipping structure: An efficient VLSI architecture for lifting based discrete wavelet transform. IEEE Trans. Signal Process. 52(4): 1080-1089.