

# An Efficient Area and Power 2D-DWT Lifting using Radix-8 Modified Booth Algorithm

S.Angel Latha Mary<sup>1</sup>, N.Dharani<sup>2</sup>

<sup>1</sup>Professor, Karpagam College of Engineering ,Coimbatore, India

<sup>2</sup>Asst. Professor, Info Institute of Engineering,Coimbatore, India

Email: xavierangellatha@gmail.com, dharumuthu@gmail.com

Abstract-Image compression is the application of Data compression on digital images. Two-Dimensional (2-D) discrete wavelet transform (DWT) is widely used in image and video compression. Lifting based and Convolution based designs have been suggested for efficient VLSI implementation of 2D-DWT. This paper presents an efficient implementation of high speed multiplier using the shift and adds method, Radix-8 modified Booth multiplier algorithm. The booth radix-8 multiplier reduces the number of the partial products, when compared to radix-4 booth multiplier. As a result of which they occupy lesser space as compared to the serial multiplier. Efficient area and low power 2D-DWT lifting is implemented. This is very important criteria because in the fabrication of chips and high performance system requires components which are as small as possible.

Keywords- Discrete Wavelet Transform, radix-8 multiplier, Discrete Cosine Transform

## I. INTRODUCTION

With rapid advance in the development of multimedia and communication techniques numerous services such as video conference, HDTV, video on demand, worldwide web (www) and distance learning, etc., plays an important role in modern life. All of these services have a common feature that is a great amount of image data, which is transmitted in a bandwidth limited communication channel. Since the bandwidth of the communication channel is limited, the compression and decompression techniques of the image data become very important for some of the real-time applications. In the recent years, the Discrete Wavelet Transform (DWT) has become a powerful tool in many areas, such as image compression and analysis, texture discrimination, pattern recognition and so on. Due to the localization property both in time and frequency domain, wavelet transform is more efficient than the STFT or Cosine Transform in de-correlating the time domain signal correlation. It has been proved that the wavelet based image compression scheme has better performance than DCT based, especially in very low bit rate. Some audio applications also use wavelets to handle the audio compression problem.

Wavelets decompose the signal at one level of approximation and detail signal at the next level. Thus the subsequent levels can add more details to the information content. This transform is very efficient for multi-resolution decomposition of signals. The wavelet transform, based on time-scale representation, provides an alternative to time-frequency representation based signal processing and the perfect reconstruction property of the analysis and synthesis wavelets along with the absence of perceptual degradation at the block boundaries favor use of the wavelet transform in video/image coding applications. The

large application-domain of wavelet makes the study of their implementation in VLSI is very important. In general, DWT is a computationally very intensive process and hence very slow, when using general purpose computing system. To make it suitable for real-time application, it is essential to develop special purpose custom VLSI chips for DWT. A digital image, or "bitmap", consists of a grid of dots, or "pixels", with each pixel defined by a numeric value that gives its color. The term data compression refers to the process of reducing the amount of data required to represent a given quantity of information. Now, a particular piece of information may contain some portion, which is not important and can be comfortably removed. Image Compression is an important component of the solutions available for creating image file sizes of manageable and transmittable dimensions. Platform portability and performance are important in the selection of the compression/decompression technique to be employed.

The Two-Dimensional DWT (2D-DWT) is a multi level decomposition technique. It converts images from spatial domain to frequency domain. One-level of wavelet decomposition produces four filtered and sub-sampled images, referred to as sub bands. DWT process a data on variable time-frequency plane that matches progressively the lower frequency components to coarser time resolutions and the high-frequency components to finer time resolutions, thus achieving a multiresolution analysis. The Discrete Wavelet Transform has become powerful tool in a wide range of applications including image/video processing, numerical analysis and telecommunication. The advantage of DWT over existing transforms, such as Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT), is that the DWT performs a multi resolution analysis of a signal with localization in both time and frequency domain. The need for Radix 8 Modified Booth Multiplier over Radix 4 Modified Booth Multiplier is given in chapter 2. The Lifting DWT is discussed in chapter 3. Modified booth multiplier present in chapter 4. Various simulation results and comparison of methods is shown in chapter 5. The chapter 6 deals with conclusion and future enhancements.

## II. BACKGROUND STUDY

Rajesh K. Yadav, S.P. Gangwar and Harsh V. Singh give an introduction to image processing techniques and its applications [1]. An efficient VLSI architecture of a high speed, low power 2-D Discrete Wavelet Transform computing. The proposed architecture, based on new and fast lifting scheme approach for (9, 7) filter in DWT, reduces the hardware complexity and memory accesses. Moreover, it has the ability of performing progressive computations by minimizing the buffering between

the decomposition levels. The system [2] is fully compatible with JPEG2000 standard. Our designs were realized in VHDL language and optimized in terms of throughput and memory requirements. The implementations are completely parameterized with respect to the size of the input image and the number of decomposition levels. The proposed architecture is verified by simulation and successfully implemented in a Cyclone II and Stratix III FPGAs, and the estimated frequency of operation is 350 MHz. The resulting computing rate is up to 48 frames (4096x2160) per second with 24 bpp. The architecture has regular structure, simple control flow, small embedded buffers and low power consumption. High-efficient lifting-based architectures for the 5/3 discrete wavelet transform (DWT) [3] are proposed. The proposed parallel and pipelined architecture consists of a horizontal filter (HF) and a vertical filter (VF). The system delays of the proposed architectures are reduced. Filter coefficients of the biorthogonal 5/3 wavelet low-pass filter are quantized before implementation in the high-speed computation hardware. In the proposed architecture, all multiplications are performed using less shifts and additions. The proposed architecture is 100% hardware utilization and ultra low-power. The architecture has regular structure, simple control flow, high throughput and high scalability.

A new approach for designing and implementing lifting-based VLSI architectures for two dimensional discrete wavelet transform (2-D DWT) is introduced. As a result, two high performance VLSI architectures that perform 2-D DWT for lossless 5/3 and lossy 9/7 filters [4] are proposed. In addition, the architectures implement symmetric extension for boundary treatment. First, two pipelined architectures consisting of two stages, the row and column processors stages, were developed for 5/3 and 9/7 filters. The internal memory between the row and the column processors is reduced to a few registers. Second, in order to speedup the computation, fully pipelined data path architectures for row and column processors were separately developed for each 5/3 and 9/7 filters that can be incorporated into the two architectures developed in the first part. The role of the compression is to reduce bandwidth requirements for transmission & memory requirements for storage of all forms of data as it would not be practical to put images, audio, video alone on websites without compression. The medical community has many applications in image compression often involving various types of diagnostic imaging. The use of wavelet transform is now well established due to its multi resolution & scaling property. Among the various techniques we have used the lifting scheme as the lifting scheme allows perfect reconstruction by its structure. The system is fully compatible with JPEG 2000 standard. The most important property of this concept seems to be the possibility of simple and fast applications into FPGA chip. It requires fewer operations and provides in-place computation of the wavelet coefficients. This paper presents a method which implements 3-D lifting [5] wavelet by FPGA. This architecture has an efficient pipeline structure to implement high-throughput processing without any on-chip memory/first in first out access. The proposed VLSI architecture is more efficient than the previous proposed architectures in terms of memory access, hardware regularity and simplicity and throughput.

Architecture for the lifting-based two-dimensional discrete wavelet transform is presented. The architecture is easily scaled to accommodate different numbers of lifting steps. The

architecture has regular data flow and low control complexity, and achieves 100% hardware utilization. Symmetric Extension [6] of the image to be transformed is handled in a way that does not require additional computations or clock cycles. The architecture is investigated in terms of hardware parameters such as memory size and number of ports, number of memory accesses, latency, and throughput. The proposed architecture achieves higher throughput and uses less embedded memory than architectures based on convolutional filter banks.

The design and implementation of Booth multiplier using VHDL [7]. This compares the power consumption and delay of radix 2 and modified radix 4 Booth multipliers. Experimental results demonstrate that the modified radix 4 Booth multiplier has 22.9% power reduction than the conventional radix 2 Booth Multiplier. Implementation of radix-4 Modified Booth Multiplier [8] and this implementation is compared with Radix-2 Booth Multiplier. Modified Booth's algorithm employs both addition and subtraction and also treats positive and negative operands uniformly. No special actions are required for negative numbers. In this Paper, we investigate the method of implementing the Parallel MAC with the smallest possible delay. Parallel MAC is frequently used in digital signal processing and video/graphics applications. A new architecture of multiplier - and accumulator (MAC) for high speed arithmetic by combining multiplication with accumulation and devising a carry look ahead adder (CLA), the performance is improved. Modified Booth multiplication algorithm [8] is designed using high speed adder. High speed adder is used to speed up the operation of Multiplication. Designing of this algorithm is done by using VHDL and simulated using Xilinx ISE 9.1i software has been used and implemented on FPGA xc3s50-5pq208. Low power consumption and smaller area are some of the most important criteria for the fabrication of DSP systems and high performance systems. Optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. In our project we try to determine the best solution to this problem by comparing a few multipliers.

This project presents an efficient implementation of high speed multiplier using the shift and adds method, Radix-4 modified Booth multiplier algorithm. The parallel multipliers like radix 2 and radix 4 modified booth multiplier does the computations using lesser adders and lesser iterative steps. As a result of which they occupy lesser space as compared to the serial multiplier. This is very important criteria because in the fabrication of chips and high performance system requires components which are as small as possible. A new architecture, namely, Multiplier-and accumulator (MAC) based Radix-4 Booth Multiplication Algorithm [9] for high-speed arithmetic logics have been proposed and implemented on Xilinx FPGA device. By combining multiplication with accumulation and devising a hybrid type adder the performance was improved. The modified booth encoder will reduce the number of partial products generated by a factor of 2. Fast multipliers are essential parts of digital signal processing systems. The speed of multiply operation is of great importance in digital signal processing as well as in the general purpose processors. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of addition generates a partial product. Parallel MAC

is frequently used in digital signal processing and video/graphics applications. The MAC provides high speed multiplication and multiplication with accumulative addition. This paper presents a combined process of multiplication and accumulation based on radix-4 & radix-8 booth encodings [10]. In this Paper, we investigate the method of implementing the Parallel MAC with the smallest possible delay. Enhancing the speed of operation of the parallel MAC is a major design issue. This has been achieved by developing a new Radix-5 Kogge stone adder for parallel MAC.

A 21\*21-bit, 2s complement binary numbers, radix-8 multiplier unit design for specific purpose is presented. Specific purpose means that the multiplicand belongs to a previously known set of numbers which are stored in a memory, so it can be used to make a modification in the radix-8 multiplication algorithm. We can modify the previous adder so that it runs faster. A full-custom layout-level design and electrical simulation of the multiplier unit has been done. Results have been compared to what we could get applying a more conventional radix-4 multiplier architecture [11]. Carry Save Adder (CSA) is core of multipliers. The delay of conventional Carry Save Adders constitutes high carry propagation delay and this delay reduces the overall performance of the radix-8 booth [12] encoded multiplier. This paper proposes a simple and efficient approach to reduce the maximum delay of carry propagation in final stage of multiplier. The radix-8 Booth encoding reduces the number of partial products to  $\lceil n/3 \rceil + 1$ , which is more aggressive than Radix-4 booth encoding. The speed of multiplier increases and power is also increased.

### III. IMPLEMENTATION

The lifting scheme which entirely relies on the spatial domain has many advantages compared to filter bank structure, such as lower area, power consumption and computational complexity. The lifting scheme can be easily implemented by hardware due to its significantly reduced computations. Lifting has other advantages, such as "in-place" computation of the DWT; integer-to-integer wavelet transforms which are useful for lossless coding. The lifting scheme has been developed as a flexible tool suitable for constructing the second generation wavelets. It is composed of three basic operation stages: split, predict and update. A multiplier has two stages. In the first stage, the partial products are generated by the Booth encoder and the partial product generator (PPG), and are summed by compressors. In the second stage, the two final products are added to form the final product through a final adder.

#### RADIX-8

The number of subsequent calculation stages can be decreased by enhancing the parallelism operation. So, one of the solutions of realizing high speed multiplier is to enhance parallelism operation. The Radix-4 Booth multiplier is the modified version of the conventional version of the Booth algorithm (Radix-2), which has two drawbacks. They are:

- (i) Which is inconvenient in designing parallel multipliers because the number of add subtract operations and the number of shift operations becomes variable.
- (ii) When there are isolated 1's, the algorithm becomes inefficient. These problems are overcome by using

modified Radix-8 Booth multiplier. The Booth algorithm which scans the strings of three bits is given below:

- a) If necessary to ensure that  $n$  is even, then the sign bit 1 position is extending.
- b) A 0 bit is appended to the right of the LSB of the multiplier.
- c) Each partial products will be 0,  $+M$ ,  $-M$ ,  $+2M$ ,  $-2M$ ,  $+3M$ ,  $-3M$ ,  $+4M$  or  $-4M$ .

Generation of Radix 2 and Radix 8 multiplication (referred to as a hard multiple, since it cannot be obtained via simple shifting and complementing of the multiplicand) generally requires some kind of carry propagate adder to produce. Generated carry propagate adder may increase the latency, mainly due to the long wires that are required for propagating carries from the less significant to more significant bits. High-speed modulo multipliers using Booth encoding for partial product generation have been proposed. The Booth encoding technique reduces the number of partial products to be generated and accumulated, thereby minimizing the associated hardware. The radix-4 Booth encoding is most prevalent as all modulo-reduced partial products can be generated by mere shifting and negation. Greater savings in area and dynamic power dissipation are feasible for large word-length multipliers by increasing the radix beyond four.

In the radix-8 Booth encoding, the number of partial products is reduced by two-thirds. However this reduction in the number of partial products comes at the expense of increased complexity in their generation. A Digital multiplier is the fundamental component in general purpose microprocessor and in DSP. Compared with many other arithmetic operations multiplication is time consuming and power hungry. Thus enhancing the performance of the circuit and reducing the power dissipation are the most important design challenges for all applications in which multiplier unit dominates the system performance and power dissipation.

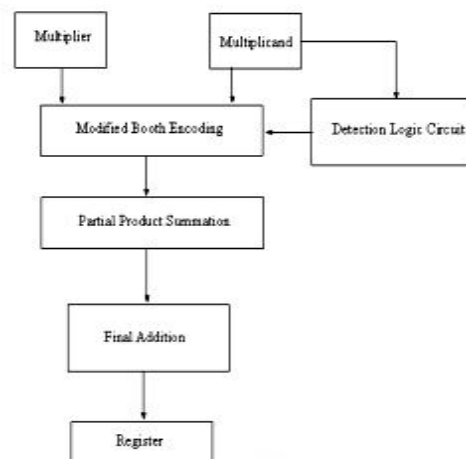


Fig 1 Block Diagram of Radix-8 MBA

The one most effective way to increase the speed of a multiplier is to reduce the number of the partial products. The number of partial products can be reduced with a higher radix Booth encoder, but the number of hard multiples that are costly to generate and which increases simultaneously. To increase the speed of operation and performance, nowadays many parallel

MAC architectures have been proposed. The technique Parallelism in obtaining partial products is the most common technique used in the above implemented architecture. There are two common approaches that make use of parallelism to enhance the multiplication performance. The difference between the two is that the latest one carries out the accumulation by feeding back the final CSA (Carry Save Adder) output rather than the final adder results which we are obtaining. The rest of the paper is organized as follows. Section second, in which an introduction to the general MAC is given along with basic MAC algorithms. Section third, in which the entire process of parallel MAC based on radix-8 booth encodings is explained. In section four which shows implementation result and the characteristics of parallel MAC based on both of the booth encodings. At last, the conclusion will be given in section five in which provides summary of our proposed approach and discuss scope of future extensions.

#### IV. RESULTS AND DISCUSSIONS

2D DWT Lifting using radix4 algorithm and 2D DWT Lifting using radix8 algorithm are coded using VHDL and simulated using Modelsim 10.0b simulator. The simulation waveform for 2D DWT Lifting using radix4 algorithm is shown in Fig.2

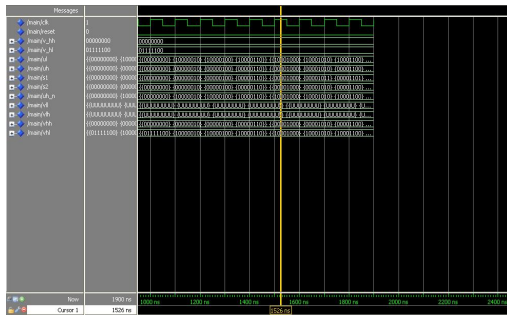


Fig. 2 2D DWT Lifting using Radix4 algorithm

The 2D DWT Lifting using radix4 algorithm, 2D DWT Lifting using radix8 algorithm are coded using VHDL and results are synthesized on FPGA (Device Spartan3) using Xilinx ISE 9.1i.

The device utilization summary for 2D DWT Lifting using radix4 algorithm is shown in Fig.3.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	72	13,824	1%
Number of 4-input LUTs	561	13,824	6%
Logic Distribution			
Number of occupied Slices	576	6,912	7%
Number of Slices containing only related logic	576	576	100%
Number of Slices containing unrelated logic	0	576	0%
Total Number of 4 input LUTs	561	13,824	6%
Number of bonded IOBs	33	325	10%
IOB Flip Flops	16		
Number of GCLUTs	1	4	25%
Number of GCLKIOBs	1	4	25%
Total equivalent gate count for design	2,157		
Additional JTAG gate count for IOBs	1,632		

Fig.3 Device utilization summary for 2D DWT Lifting using Radix4 algorithm

The device utilization summary for 2D DWT Lifting using radix8 algorithm is shown in Fig.4.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	96	13,824	1%
Number of 4-input LUTs	643	13,824	4%
Logic Distribution			
Number of occupied Slices	360	6,912	5%
Number of Slices containing only related logic	360	360	100%
Number of Slices containing unrelated logic	0	360	0%
Total Number of 4 input LUTs	643	13,824	4%
Number of bonded IOBs	33	325	10%
IOB Flip Flops	16		
Number of GCLUTs	1	4	25%
Number of GCLKIOBs	1	4	25%
Total equivalent gate count for design	4,811		
Additional JTAG gate count for IOBs	1,632		

Fig 4 Device utilization summary for 2D DWT Lifting using Radix8 algorithm

The power report for the 2D DWT Lifting using radix4 algorithm is shown in Fig.5.

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		
Vccint 1.80V:	26	46
Vcco33 3.30V:	2	7
Clocks:	10	17
Inputs:	1	2
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Fig.5.4 Power reports for the 2D DWT Lifting using radix4 algorithm

The power report for the 2D DWT Lifting using radix8 algorithm is shown in Fig 6

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		
Vccint 1.80V:	22	39
Vcco33 3.30V:	2	7
Clocks:	6	12
Inputs:	0	1
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Fig.6 Power reports for the 2D DWT Lifting using radix8 algorithm

#### V. CONCLUSION AND FUTURE SCOPE

This paper implements a 2D-DWT lifting using radix8 modified booth algorithm to improve the area efficiently and it consumes less power when compare to 2D DWT lifting using radix4 modified booth algorithm. 2D DWT lifting using radix4 modified booth algorithm and 2D DWT lifting using radix8 modified booth algorithm are coded using VHDL, simulated using Modelsim 10.0b and synthesized in Spartan 3 FPGA using Xilinx ISE 9.1i. The work can be further extended with the design of mathematically optimized distributed arithmetic (DA) technique.

#### REFERENCES

- [1] C. Cheng and K. K. Parhi, 'High-speed VLSI implementation of 2D discrete wavelet transform,' *IEEE Transaction Signal Process*, Volume 56, Number 1, Page 393-403, January 2008.
- [2] C.-T. Huang, P.-C. Tseng and L.-G. Chen, 'Flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform,' *IEEE Transaction Signal Process*, Volume 52, Number 4, Page 1080–1089, April 2004.
- [3] B. K Y.-K. Lai, L.-F. Lien, and Y.-C. Shih, 'A high performance and memory-efficient VLSI architecture with parallel scanning method for 2D lifting-based discrete wavelet transform,' *IEEE Transaction*

*Consumption Electronics*, Volume 55, Number 2, Page 400–407, May 2009.

- [4] B.K. Mohanty, A. Mahajan, and P. K. Meher, 'Area- and power-efficient architecture for high-throughput implementation of lifting 2-D DWT,' *IEEE Transaction Circuits System II, Briefs*, Volume 59, Number 7, Page 434–438, July 2012.
- [5] B. K. Mohanty and P. K. Meher, 'Memory-efficient high-speed convolution-based generic structure for multilevel 2-D DWT,' *IEEE Transaction Circuits System Video Technology*, Volume 23, Number 2, Page 353–363, February 2013.
- [6] B.-F. Wu and C.-F. Chung, 'A high performance and memory-efficient pipeline architecture for the 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec,' *IEEE Transaction Circuits System Video Technology*, Volume 15, Number 12, Page 1615–1628, December 2005.
- [7] G.-M. Shi, W.-F. Liu, L. Zhang, and F. Li, 'An efficient folded architecture for lifting-based discrete wavelet transform,' *IEEE Transaction Circuits System II, Briefs*, volume 56, Number 4, Page 290–294, April 2009.
- [8] I. Daubechies and W. Sweldens, 'Factoring wavelet transforms into lifting steps,' *J. Fourier Analysis and its Applications*, Volume 4, Number 3, Page 247–269, 1998.
- [9] S.G. Mallat, 'A theory for multiresolution signal decomposition: The wavelet representation,' *IEEE Transaction Pattern Analysis Machines Intelligent*, Volume 11, Number 7, Page 674–693, July 1989.
- [10] W. Zhang, Z. Jiang, Z. Gao, and Y. Liu, 'An efficient VLSI architecture for lifting-based discrete wavelet transform,' *IEEE Transaction Circuits System II*, Volume 59, Number 3, Page 158–162, March 2012.
- [11] X. Tian, L. Wu, Y.-H. Tan, and J.W. Tian, 'Efficient multi-input / multi-output VLSI architecture for two-dimensional lifting-based discrete wavelet transform,' *IEEE Transaction Computer*, Volume 60, Number 8, Page 1207–1211, August 2011.
- [12] Y.K. Lai, L.F. Lien, and Y.C. Shih, 'A high-performance and memory efficient VLSI architecture with parallel scanning method for 2-D lifting based discrete wavelet transform,' *IEEE Transaction Consumer Electronics*, Volume 55, Number 2, Page 400–407, May 2009.