

Memory Efficient and Low power VLSI architecture for 2-D Lifting based DWT with Dual data Scan Technique

A.D.Darji, A.N.Chandorkar, and S.N.Merchant

Abstract—The lifting scheme reduces the computational complexity for computing Discrete Wavelet Transform (DWT) compared to convolution. 2-D DWT is widely used frequency domain transform for various multimedia applications. Due to battery operated handheld devices for multimedia application need is arise to design low power yet high speed and area efficient chip for 2-D DWT. We have proposed a high performance and memory efficient architecture with parallel scanning method for 2-D DWT using 5/3 Lifting wavelet and done chip level implementation using 180nm UMC standard cell library. This architecture is composed with two 1-D DWT units and a Transpose Unit (TU). Proposed parallel scanning reduces not only of on-chip line buffer but enhances through put as well compared to other line based scanning. Proposed 2-D DWT architecture utilizes only $2N$ size buffer for $N \times N$ sized image, which is low compare to $3.5N$ usual requirement for to implement 5/3 Lifting wavelet. Designed TU operates at half clock rate which reduces power and its design is independent of size of input image. Instead of shifter we propose Hardwired Scaling Unit (HSU) for coefficient multiplication in order to save dynamic power. This architecture is first synthesized using Xilinx ISE 10.1 and is implemented on Virtex-II PRO XC2VP30 FPGA and then compile RTL with UMC 180 nm standard cell library for ASIC (Application Specific Integrated Circuit) implementation. This design is compared for power, speed and area with existed architectures.

Keywords—DWT, Dual Scan Architecture, Lifting, Low Power, VLSI

Manuscript received May 31, 2011. This work was supported by the Ministry of information and technology, Government of India through SMDP-II project. A.D.Darji is deputed at Indian Institute of technology Bombay, 4300076 INDIA and working as assistant professor at S.V.National Institute of Technology Surat 395007 INDIA (phone: 91-261-2201743; e-mail: anand@ee.iitb.ac.in).

Second author Dr. A.N. Chandorkar is with Electrical Engineering Department, Indian Institute of Technology Bombay, Powai ,INDIA (e-mail: anc@ee.iitb.ac.in).

Prof. S.N.Merchant is with the Electrical Engineering Department, Indian Institute of Technology Bombay, Powai, INDIA (e-mail: merchant@ee.iitb.ac.in).

I. INTRODUCTION

Multi-resolution representation of signals using DWT has been very effectively used in many signal processing applications like compression, watermarking, data streaming etc. JPEG 2000 uses DWT which provide features like progressive image transmission, ease of image compression and region of interest coding, etc. Conventionally DWT has been implemented using convolution or FIR filter bank structures which require large number of arithmetic computations and huge memories. Hand held image/video applications demand high speed, low memories and low power, hence traditional approach of image/video transforms do not work. So, in this paper we are proposing 2-D DWT architecture based on lifting scheme [1] and its chip level implementation. There are several architectures discussed in literature to perform lifting based DWT. General approach for 2-D DWT is to apply the 1-D DWT row-wise which produces L and H sub-bands and then process these sub-bands column-wise to get LL, LH, HL and HH coefficients. Several architectures like direct mapped [2], folded [3], and flipping [4] for single level and multi-level DWT have been proposed to implement 1-D lifting DWT. To use 1-D DWT to get 2-D DWT is required memory buffer because we cannot process columns till all rows are available in memory buffer. N^2 size frame memory is required and hence it is usually off chip to DWT processor. This leads to more power consumption in data access with external memory. Second approach is to start filtering column as soon as sufficient numbers of rows have been filtered which deals with intelligent memory management in 2-D DWT architectures. Size of internal buffer memory depends on different memory scanning techniques like line-based, block-based and stripe-based scan have been proposed [5] they may be overlapped or non-overlapped type. In proposed architecture we have used non overlapped stripe based scan to optimize speed, power and memory requirement. Lifting based DWT architecture with predicts and update step is described in [6] was slow and was modified by introducing parallel processing and pipelining [7]. High performance and memory efficient pipelined architecture for 5/3 and 9/7

2-D DWT for JPEG 2000 codec was proposed in [8]. An efficient high speed 2-D DWT architecture using two horizontal and one vertical filter was proposed by [9] with reduced latency with great amount. Proposed architecture uses one parallel vertical filter and one parallel horizontal filter module. Memory scanning technique adopted provides 100% hardware utilization. In [12] memory efficient pipelined architecture is proposed and ASIC implementation is also done. Rest of this Paper is organized as follows. In section II discussions on lifting scheme is included. In section III, proposed architecture is discussed along with different design issues. Performance comparison of hardware and time complexity is discussed in section IV. Result of implementation of proposed architecture is summarized in section V and conclusion is given in section VI.

II. LIFTING SCHEME

2-D DWT coefficient calculation through convolution method required intense processing. Hence lifting scheme is used for DWT in many embedded applications. Lifting scheme uses spatial domain to construct a wavelet and has three steps: split, predict and update as shown in Fig.1 Principle used is to breakup poly-phase matrices for wavelet filters in to sequences of upper and lower triangular matrices and convert the filter implementation in to banded matrices multiplications.

Using split phase the input stream is divided into odd and even samples. The predict value is calculated from the present and past odd sample values depending on the filter size. The high frequency component or the detailing coefficients can be calculated by subtracting the predicted value (output of P stage) from the even sample. Now, update value is calculated from the present and past predicted values which also depend on the filter size. The low frequency components or the smooth coefficients can be calculated by adding the updated value (output of U stage) to the odd sample.

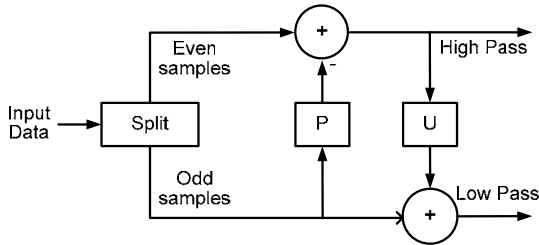


Fig.1. Lifting Scheme

These steps can be again applied on the coefficients recursively to get multilevel coefficients. The high pass $g(z)$ and low pass $h(z)$ filters can be represented as (1) and (2). For 9/7 filter dual lifting steps are used.

$$g(z) = \sum_{i=0}^{J-1} g_i z^{-i} \quad (1)$$

$$h(z) = \sum_{i=0}^{J-1} h_i z^{-i} \quad (2)$$

Expression (1) and (2) can be split into even and odd as (3) and (4).

$$g(z) = g_e(z^2) + z^{-1}g_o(z^2) \quad (3)$$

$$h(z) = h_e(z^2) + z^{-1}h_o(z^2) \quad (4)$$

These filters are represented in a poly-phase matrix as shown in (5)

$$P(z) = \begin{bmatrix} h_e(z) & h_o(z) \\ g_e(z) & g_o(z) \end{bmatrix} \quad (5)$$

$$= \prod_{i=1}^m \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}$$

Since the lifting scheme applies for constructing bi-orthogonal wavelets, symmetrical extension can always be used to calculate the lifting scheme. In this work zero extension is used to reduce latency. For 5/3 filter, there is only one scaling and lifting step which can be depicted in terms of pixel values in (6),(7),(8),(9).

(1) Splitting Step:

$$e_i = x_{2i+1} \quad (6)$$

$$o_i = x_{2i} \quad (7)$$

(2) Lifting Step:

$$o_i^1 = o_i^0 + \alpha \times (e_i^0 + e_{i+1}^0) \rightarrow (\text{predictor}) \quad (8)$$

$$e_i^1 = e_i^0 + \beta \times (o_{i-1}^1 + o_i^1) \rightarrow (\text{updater}) \quad (9)$$

Where, x is the input pixel; e and o represent even and odd pixels/coefficients; α and β are multiplication factors and are $\frac{1}{2}$ and $\frac{1}{4}$ respectively for 5/3 Lifting filter.

III. PROPOSED ARCHITECTURE

A. Scanning

The data flow of 5/3 lifting scheme for 1-D DWT is shown in Fig. 2. The symbols α and β are scaling coefficients of the wavelet. The X_o and X_e are the odd and even pixels of an image or video frame. The data is scanned as described in the Fig. 3, where two pixel values are read

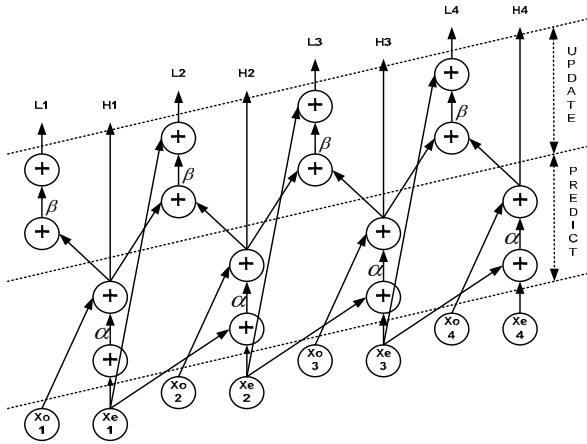


Fig.2. Data Flow Graph for 1D DWT

from the dual port RAM/ROM of the row in a single clock and 1-D DWT architecture processes them in the next clock. This leads to four coefficients required to start column operation for 2-D DWT operation. Thus provides 2-D coefficients at only three clocks latency.

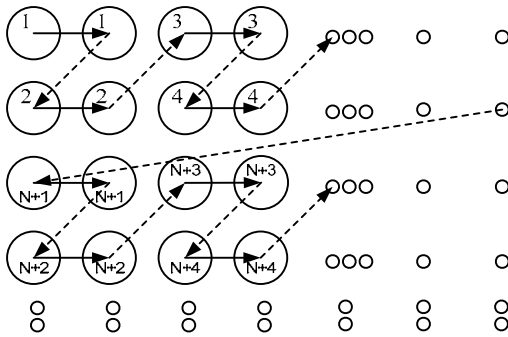


Fig.3. Proposed Scanning

B. 2-D DWT Architecture

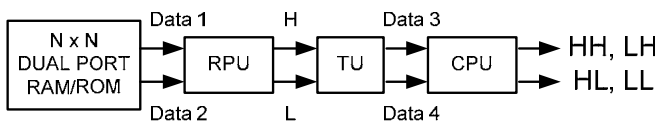


Fig.4. Proposed 2D Architecture

Proposed 2-D DWT architecture is as depicted in Fig.4 has been used for FPGA and ASIC implementation. Image or video frame is read from dual port ROM/RAM with dual scan method discussed in earlier section and given to Row Processing Unit (RPU) to calculate 1-D DWT. These 1-D coefficients are given to TU. The TU manages the column

wise input to Column Processing Unit (CPU) and calculates 2-DDWT coefficients.

C. Row processing Unit

The proposed architecture shown in Fig.5 takes two inputs and gives two outputs per cycle. Data1 and Data2 are the odd and even input samples given to hardware in single clock for 100 % hardware utilization. This architecture is very simple design as compared to other architectures suggested in [6] and [10] which have complex control path to achieve 100% hardware utilization. Simple control path further reduces power and core area. Usually 2-D DWT architecture has to wait for one complete row to be processed to start column processing and has requirement of line buffer with size N for image size $N \times N$ [11]. Proposed architecture has an inherit property to process two rows at alternate clocks which gives required 1-D coefficients to start column processing simultaneously to get 2-D DWT coefficients. This dual scan not only saves line buffer but reduces latency. Transpose Unit (TU) is responsible for sequencing coefficient available from RPU to CPU. Further, pipeline stages can be added to reduce the critical path delay from $4T_a$ to $2T_a$ at the cost of latency and register count, where T_a is the adder delay.

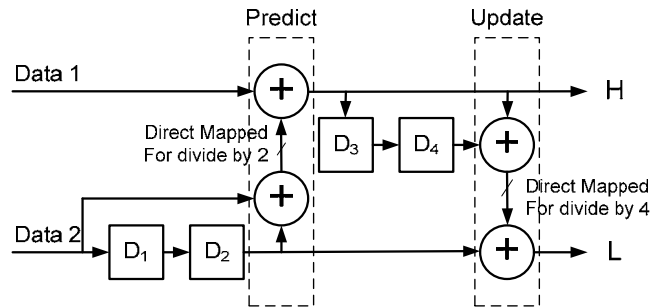


Fig.5. Proposed Row Processing Unit

DWT with 5/3 lifting needs divide by two and divide by four operations as shown in (8) and (9). This can be designed with shift registers and switching of shift register gives rise to dynamic power. Apart from this issue shift register also need large area. So we have proposed simple and power efficient Hardwired Scaling Unit (HSU) designed using hardwired connections as shown in the Fig.6 . This arrangement allows us to remove the multipliers or shifters which give rise to critical path and latency.

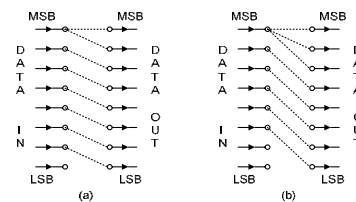


Fig.6. Proposed Hardwired Scaling Unit

D. Transpose unit

The incoming data is given column wise to perform the 2-D DWT. It is managed through novel power efficient architecture is two inputs and two outputs which are managed through dual port memory.

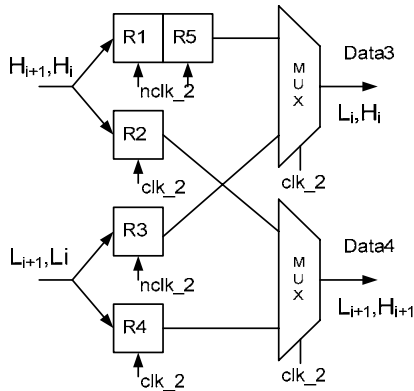


Fig.7. Proposed Transpose Unit

TU architecture. This is design using only five registers and two multiplexers as shown in the Fig.7. Here, clk_2 and $nclk_2$ are half rate clock and inverted half rate clock respectively. At every clock there are two inputs, high and low 1-D wavelet coefficients to TU and two rearranged outputs from TU as shown in Fig.8.

The CPU architecture has buffer size N (Number of row pixels) can be seen in Fig.9. Buffers are shift registers whose data flow is similar to FIFO. These buffers are initialized with zero for zero extension technique is adopted for boundary treatment. The outputs Data3 and Data4 from the TU are given to CPU to calculate 2-D DWT coefficients. Input and output sequence of coefficients for DWT is shown in Fig. 8.

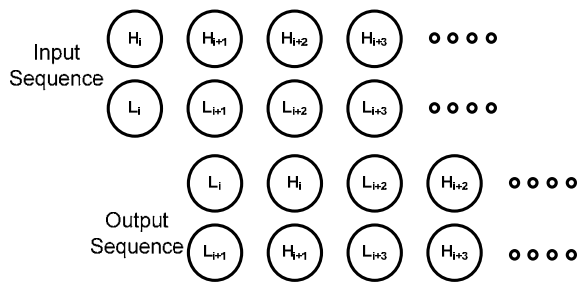


Fig.8. Data Flow in Transpose Unit

E. Column processing unit

The performance of proposed architecture is compared to the architectures proposed by others in Table II for single level 2-D DWT. Proposed architecture has very low memory requirement and produces 2-D coefficients at a latency of only 3 cycles. This way lot of parallelism is introduced to save clocks as shown in Table I. The required

buffer size is also less than other architectures. The proposed architecture does not require any multiplier which gives an additional advantage in power consumption. The throughput of the

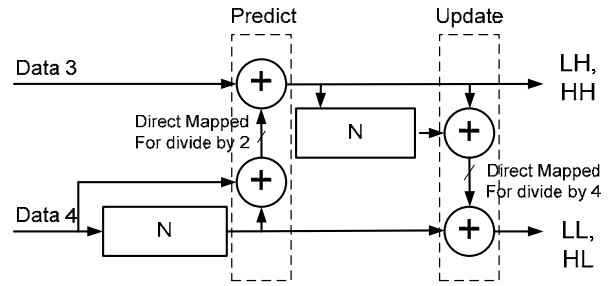


Fig.9. Proposed Column Processing Unit

IV. PERFORMANCE ANALYSIS AND COMPARISON

TABLE I
DATA SEQUENCE OF RPU AND CPU

Clk	Input	1D DWT Output	2D DWT Output
1	$X_{1,1} ; X_{1,2}$		
2	$X_{2,1} ; X_{2,2}$	$L_{1,1} ; H_{1,2}$	
3	$X_{1,3} ; X_{1,4}$	$L_{2,1} ; H_{2,2}$	
4	$X_{2,3} ; X_{2,4}$	$L_{1,3} ; H_{1,4}$	$LL_{1,1} ; LH_{1,2}$
5	$X_{1,5} ; X_{1,6}$	$L_{2,3} ; H_{2,4}$	$HL_{2,1} ; HH_{2,2}$
6	$X_{2,5} ; X_{2,6}$	$L_{1,5} ; H_{1,6}$	$LL_{1,3} ; LH_{1,4}$
7	$X_{1,7} ; X_{1,8}$	$L_{2,5} ; H_{2,6}$	$HL_{2,3} ; HH_{2,4}$
..

V. IMPLEMENTATION RESULTS

The design is coded in VHDL and synthesized using Xilinx ISE 10.1 using Virtex-II PRO FPGA as target device. Complete 2-D DWT core uses 251 slices out of 13696 and 477 LUTs out of 27392 and operates on maximum frequencies 174.03 MHz. The High Definition (HD) video format has resolution 1920x1080 (2073600 pixels per Frame). Hence for a rate of 30 frames per second refresh requires 62208000 read cycles. This requires 62.208 MHz operating frequency and proposed architecture supports this. The simulation waveforms are shown in Fig.10. The proposed architecture is also implemented using UMC 180 nm technology with clock frequency is 100 MHz for ASIC development. Synthesized netlist is shown in Fig. 11. The total cell area and estimated power is calculated with synopsys design vision after synthesis. The dynamic power is observed as 67.84 mW at 100 MHz working frequency and cell area is 0.165 mm². Proposed architecture has low power compare to Lai *et al.*[12] due to optimized

switching activities and intelligent buffer management. Chip layout view under cadence encounter is shown in Fig. 12

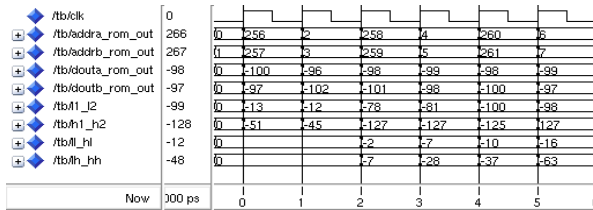


Fig.10. Simulation waveforms of 2D DWT

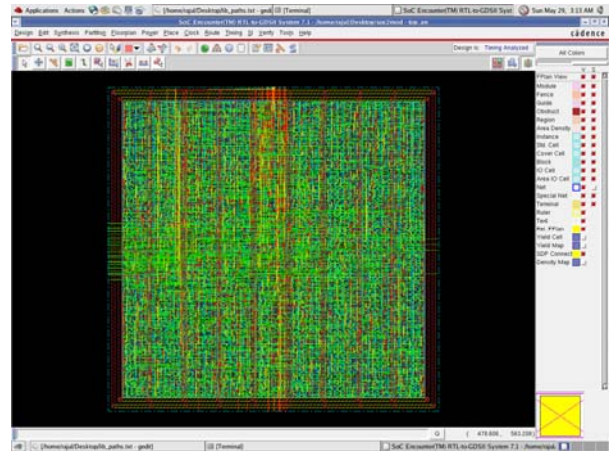


Fig. 12 Chip Layout View in SOC encounter

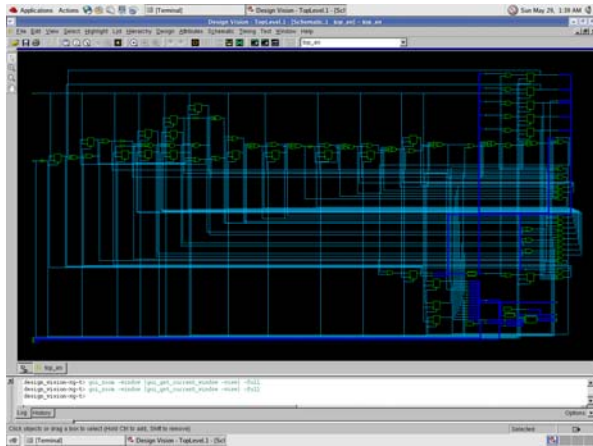


Fig.11. Gate level net-list Synopsys Design Vision

VI. CONCLUSION

Novel architecture for 2-D DWT based on 5/3 lifting scheme is proposed. Architecture uses one row processor unit and column processor unit working parallel and pipeline mode to achieve low latency and small line buffers. Performance analysis for the architecture and comparison results with the other works shows that proposed 2-D architectures are efficient in terms of throughput, output latency, control complexity, on-chip buffers and power etc. Architecture is also implemented on FPGA and compiled with UMC 180 nm standard cell library to estimate speed and area for ASIC implementation.

TABLE II
COMPARISON OF HARDWARE AND TIME COMPLEXITIES OF PROPOSED 2-D 5/3 LIFTING ARCHITECTURE WITH EXISTING STRUCTURES

Architecture	Mult/Shifters	Adders	Memory		Computing Time	Output Latency	Hardware Utilization	Throughput
			On-Chip	Off-Chip				
Wu [8]	16	16	5N	N ² /4	N ² /2	2N	100%	1i/p,1o/p
Andra [7]	4	8	N ² +4N	0	N ² /2	2N	100%	2i/p,2o/p
Liao [10]	4	8	4N	0	N ²	2N	50% -66.7%	1i/p,1o/p
C.Xiong [9]	8	16	3.5N	N ² /4	N ² /2	N/2	100%	4i/p,4o/p
Lai[12]	5	8	2N	N ² /4	N+N ² /2	3	100%	2i/p,2o/p
Proposed	0	8	2N	N ² /4	N ² /2	3	100%	2i/p,2o/p

REFERENCES

- [1] I. Daubechies and W. Sweldens, "Factoring Wavelet transforms into Lifting Schemes," *The J. of Fourier Analysis and Applications*, vol. 4, 1, pp. 247–269, 1998
- [2] C.C. Liu, Y.H. Shiau, and J.M. Jou, "Design and Implementation of a Progressive Image Coding Chip Based on the Lifted Wavelet Transform," in *Proc. of the 11th VLSI Design/CAD Symposium*, Taiwan, 2000.
- [3] C.J Lian, K.F. Chen, H.H. Chen, and L.G. Chen, "Lifting Based Discrete Wavelet Transform Architecture for JPEG 2000," in *IEEE International Symposium on Circuits and Systems*, Sydney, Australia, pp. 445–448, 2001
- [4] C.T. Huang, P.C. Tseng, and L.G. Chen, "Flipping Structure: An Efficient VLSI Architecture for Lifting-Based Discrete Wavelet Transform," in *IEEE Transactions on Signal Processing*, pp. 1080–1089, 2004
- [5] C.-T. Huang, P.-C. Tseng, and L.-G. Chen, "Memory Analysis and Architecture for Two-Dimensional Discrete Wavelet Transform," in *Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2004, pp. 13–16
- [6] M. Ferretti and D. Rizzo, "A parallel architecture for the 2-D discrete wavelet transform with integer lifting scheme," *J. VLSI Signal Processing*, vol. 28, pp. 165–185, July 2001.
- [7] K. Andra, C. Chakrabati, and T. Acharya, "A VLSI architecture for lifting-based forward and inverse wavelet transform," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 966–977, April 2002.
- [8] Bing-Fei Wu and Chung-Fu Lin, "A High-Performance and Memory-Efficient Pipeline Architecture for the 5/3 and 9/7 Discrete Wavelet Transform of JPEG2000 Codec," *IEEE Trans. on circuit and systems for video Technology*, vol. 15, no. 12, pp. 1615–1627, December 2005
- [9] C.Xiong, J.Tian, and J.liu, "Efficient Architecture for Two-Dimensional Discrete Wavelet Transform Using Lifting Scheme," *IEEE Trans. on Image Process.*, vol. 16, no. 3, pp. 607-614, March 2007.
- [10] Hongyu Liao, Mrinal Kr. Mandal, "Efficient Architecture for 1-D and 2-D Lifting-Based Wavelet Transform," *IEEE Trans. on Signal Processing*, vol. 52, no. 5, pp. 1315-1326, May 2004
- [11] Jose Oliver, "On the Design of Fast Wavelet Transform Algorithms With Low Memory Requirements," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, no. 2, pp. 237-248, February 2008
- [12] Yeong-Kang Lai, Lien-Fei Chen and Yui-Chih shih, "A High-performance and Memory-Efficient VLSI Architecture with Parallel Scanning method for 2-D Lifting-Based Discrete Wavelet Transform" *IEEE Transaction on Consumer Electronics*, vol. 55, No. 2, May 2009