

SYSTOLIC ARRAY ARCHITECTURES FOR DISCRETE WAVELET TRANSFORM: A SURVEY

G. Nagendra Babu¹, Ganapathi Hegde² and Pukh Raj Vaya³

Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham, India
E-mail: ¹nagendra417@yahoo.com, ²ganapathi_hegde@blr.amrita.edu, ³pr_vaya@blr.amrita.edu

Abstract

Demand for High Speed & Low Power Architecture for Image/Video Compression Algorithms are increasing with scaling in VLSI Technology many Architectures in the Discrete Wavelet Transform (DWT) System have been proposed. This Paper surveys the different designed DWT's using Systolic Array Architectures and the Architectures are classified based on the application whether it is 1-D, 2-D or 3-D. This paper presents the overview of the architectures based on latency, number of MAC's, memory used, hardware efficiency etc. and this paper will give an insight to the reader on advantages and disadvantages of the design that are to be used in various applications.

Keywords:

Systolic Array Architecture, DWT, Image and Video Processing

1. INTRODUCTION

Wavelet Transform has proved the extreme use of them in the image, video processing, and speech analysis. DWT is discrete in time and scale, means that DWT coefficients have floating point values but the time and scale values used to index these coefficients are integers. Wavelet Transform is favored over other coding transforms because of its attractive characteristics like wavelet transform decomposes a nonstationary signal into a set of multiscaled small wavelets which are easier to code and is also more flexible such that it can be easily adapted to human visual system, Lower Aliasing, Inherent Scalability. A DWT can be 1-D, 2-D, 3-D, 4-D etc depending upon the signals dimension. A 2-D DWT is extensively used for still image coding/ compression, 3-D DWT for video applications, 4-D DWT is used for the light field compression and so on. Besides these DWT have major applications in variety of fields such as signal processing, digital communications, numerical analysis, computer graphics, radar target distinguishing, fractal analysis, texture discrimination and many more.

Wavelets are a special kind of functions which exhibits oscillatory behavior for a short period of time and then die out. In wavelets we use a single function and its dilations and translations are used to generate a set of orthonormal basis functions to represent a signal. Most of the wavelets used in DWT are fractal in nature. In a general purpose computing system implementation of DWT is computationally intensive process so it is essential to develop special purpose custom VLSI architectures for DWT exploiting the underlying data parallelism to yield high throughput and high data rate. The DWT can be implemented either by non-separable direct approach or by separable indirect approach. Direct approach involves less clock cycles latency, more computation time, extra amount of hardware to achieve the same throughput when compared with the separable approach. Separable approach i.e. row-column method requires huge

memory to save the intermediate coefficients which are obtained during transposition. The non-separable approach does not require any transposition but requires more number of multipliers and Accumulators (MAC's).

A Signal decomposed by using DWT into one or more levels is called as octaves. In Analysis side Low pass and High pass filters are used. The Low pass which applies the scaling function produces the approximation (average) signal where as the high pass filter which applies the wavelet function produces the detailed signal information. During the late seventies and early eighties subband (scale) coding and Multiresolution Analysis (MRA) or Pyramidal coding was developed. Both the scale and resolution are very important notations in DWT. Scale are related to size of signal while resolution is linked to the amount of details present in the signal. In MRA the average signal from one level (1 Octave) is sent to another level of filters which again produces the average and detail of the signal. The detailed signals are discarded and at the same time higher octave average signals can also be discarded because they can be re-computed during the inverse transformation. The following figure shows the analysis and synthesis of a 1- dimensional 1- octave DWT and Inverse DWT.

2-Dimensional is the application of the 1-Dimensional in Horizontal and Vertical directions which is only for separable cases. In Separable case the 2-Dimensional can also be extended to the 3-Dimensional also. This paper is organized as follows Architectural considerations are analyzed in the following section, followed by 1-D DWT architecture in section 3, section 4 deals with 2-D DWT and section 5 deals with 3-D issues. Finally section 6 summarizes all of the above described sections.

2. ARCHITECTURE CONSIDERATIONS

The DWT Architectures cost and performance is influenced by several factors such as memory, control, area, latency and their impacts are different for different applications.

2.1 DESIGN ISSUES

The basic serial i.e. systolic filter uses L MAC's and L wavelet coefficients. Each MAC performs 1 Addition and 1 Multiplication so that latency will be 1/L where as parallel filter has L multipliers for each wavelet coefficient most of the architectures are fixed point representation which makes multiplication faster and requires less silicon area. Wavelet transform does not need the floating point representation due to small coefficients. For N input in the DWT computation there are not exactly N/2 filter outputs this can be overcome by using zero padding. Due to similarity between DWT and IDWT the same hardware can compute both functions with few modifications. Therefore most designs have only DWT, some

designs such as Wavelet Transform Processor (WTP) have built in DWT and IDWT.

2.2 AREA

An Efficient design will minimize the area. In DWT the multiplier, adder will add to the Architectures size. Area is expressed in terms of λ^n for $n = 2$. The number of multipliers, adders is affected depending upon whether the serial or parallel filter used. The number of octaves and wavelet coefficient has relation with the number of MAC's.

2.3 CONTROL

The choice of control is very important in the design. The different control types available are centralized, flow control and pre-stored. The centralized is easy to implement but it increases the area. If the control is built within the Processing Element (PE) it is called pre-stored where as in flow control the control signal flows from PE to PE.

2.4 MEMORY

Input Values and partial computations which are generated must be stored within the hardware. The distributed architecture has local storage on each processor. The different types of storage units available are Systolic, RAM based, Mux based etc. Systolic storage unit is similar to that of the Mux based unit except the busses and tri-state buffers are used in place of multiplexers. This allows more multiplexers to be added without changing the size of the multiplexers. The semi systolic unit has a long bus. In RAM based storage unit a large RAM replaces all of the storage cells minimizing the storage make control more complex, decreases the scalability and regularity. Mux based unit has array of storage cells that form Serial In Parallel (SIPO) Queue.

3. ARCHITECTURES FOR 1-D DWT

The first 1-D architecture was proposed in the year 1977, which is an integrated systolic architecture. This architecture is unique in the sense that the same architecture is used for the forward and Inverse DWT by selecting some suitable control signals. Altogether of 5 PE's are required for implementing this type of architecture. Since DWT, IDWT are not similar very few multiplexers and control signals are used to integrate these two computations. Systolic Architecture based on the separable approach method is presented in [1]. To store the generated intermediate results he used much large number of multiplexers. He used the 4-tap Daubechies filter. This architecture does not use multipliers which save the area and increases the speed. Since it has no multipliers the multiplication operation is performed by shifting the data either to left or right and then adding. The disadvantage of this architecture is it has low precision and the design cannot be altered for different wavelet. A simple systolic algorithm suitable for high speed VLSI Implementation is proposed in [2]. See Fig.1, he integrated both the forward and Inverse DWT signals into a single systolic architecture by adding some extra control unit. This Integrated

Architecture yields 100% throughput. This Integrated Architecture has five PE's which are enough for computation of DWT and IDWT. Though the interconnections for DWT and IDWT are not similar, they are integrated using some extra amount of circuitry like multiplexers and control signals. This architecture can be easily extended to the 2-D easily by cascading two 1-D modules and a transpose circuit. It works as a DWT when control signals $I_1 = I_2 = 0$ and it works as an IDWT when control signals $I_1 = I_2 = 1$.

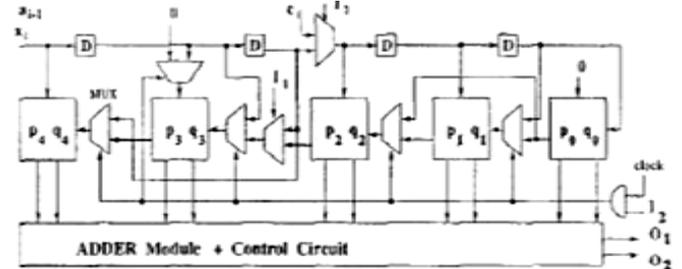


Fig.1. Combined DWT/IDWT Architecture [2]

The Systolic Architecture design requires a space representation of the algorithm called as a dependence graph (DG) is presented in [3]. Each node in DG represents a MAC operation. This DS is mapped to systolic architecture using a processor space vector (p) and a Schedule vector (s) only 2 MAC operations are performed at each node in the superimposed DG. Based on the DG he proposed the three systolic architectures see Fig.2. They have proposed three architectures here only the third architecture is presented which have hardware efficiency close to 100%; it has the simple routing and control unit.

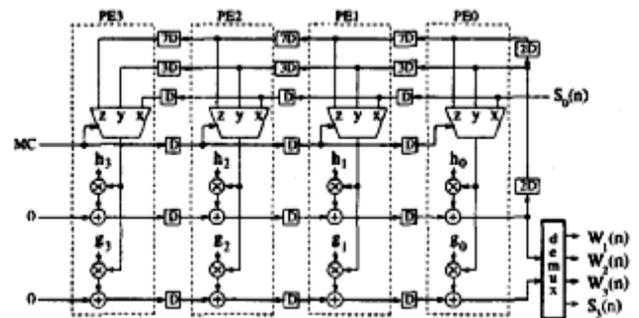


Fig.2. Fully Systolic Three-level DWT Architecture [3]

Systolic Architecture which meets the lowest possible latency for 4 octaves DWT is presented in [4]. See Fig.3, it has an efficiency of $1-2^{-j}$ which is near to 100% when the octaves are increased. Having 4 or more octaves increases the latency due to data collision which is due to the scheduling. Each PE of this architecture has 5 memory registers each, 2 registers and 1 additional register per octave. The design needs L PE's where the L stands for number of wavelet coefficients. It has an advantage of distributed memory and control. The design is simple, modular with distributed control.

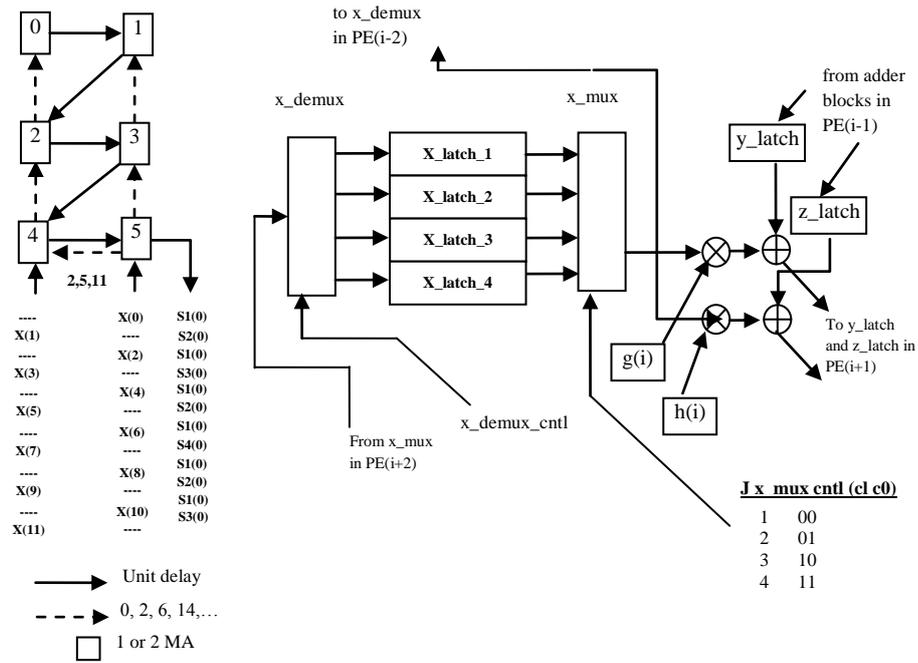


Fig.3. DWT PE array and PE architecture [4]

A non-separable architecture which computes both the low pass and high pass output sequences using the same product term is proposed in [5]. He proposed 2 types of architectures see Fig.4 which is based on block based computation. His type 1 architecture needs 4 PE arrays, 4 multiplication PE's, 4 Addition PE's. Type 2 needs 2 PE arrays, 4 multiplication PE's, 4 Addition PE's.

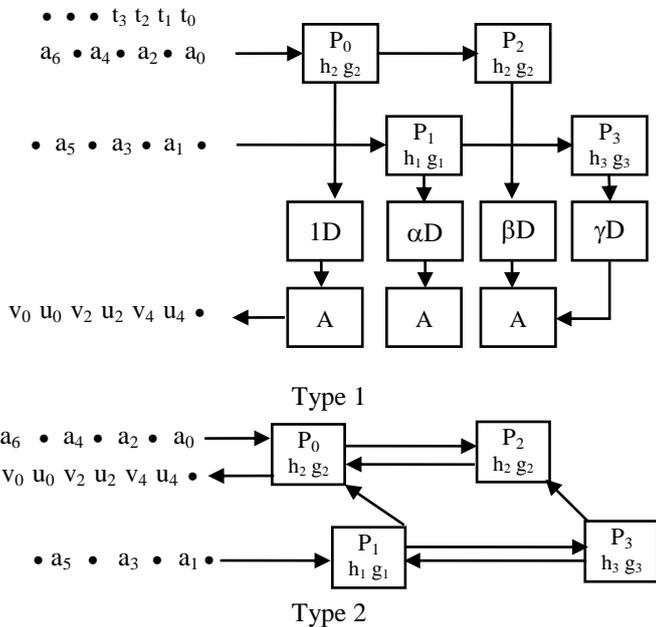


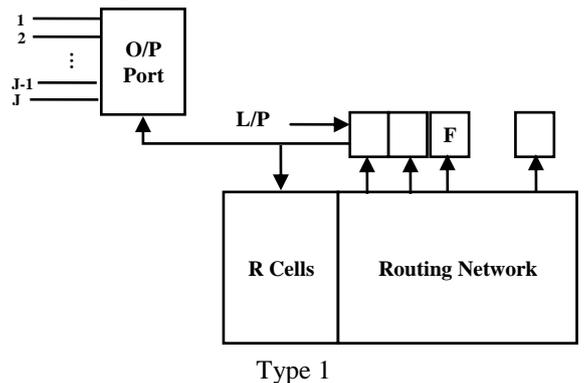
Fig.4. Systolic Architecture for 1 level Analysis Filter Type 1 and Type 2 Architectures [5]

This Architecture does not require a control unit. In Type 2 architecture the first PE array implements the first level wavelet analysis filter and the second one realizes the higher-level wavelet analysis filter. The second PE array implements the second-level wavelet filter with 50% PE utilization and the third-

level one with 25% PE utilization, and so on. Therefore, this architecture effectively uses the multiplication and addition PEs. This architecture can be extended to 2-D easily.

Three different Architectures were proposed in [6]. Out of them the first architecture which is similar to time-multiplexed architecture. It cascades linear systolic arrays in a matrix, where each row computes one octave, while each column contains a MAC for each wavelet coefficient. Inputs flow from left to right and outputs flow in opposite direction. One output can be obtained for every two clock cycles thus creates more latency. To overcome this issue, two overlapped input streams are fed into the architecture. This architecture requires large area and the processor is idle for most of the time say 67%.

The second architecture by these researchers [6] uses a register network to store the intermediate data. This architecture was designed intently to overcome the drawbacks of their previous architecture. It improved the processor utilization time thus reducing the area. The routing network has a shift register. It has the latency of $2N$ clock cycles with JL shift registers and L MAC's. Their third architecture has less area and uses RAM see Fig.9. It also has the latency and MAC's as that of their 2nd architecture but the main advantage of third architecture is it has a flow control type of control unit.



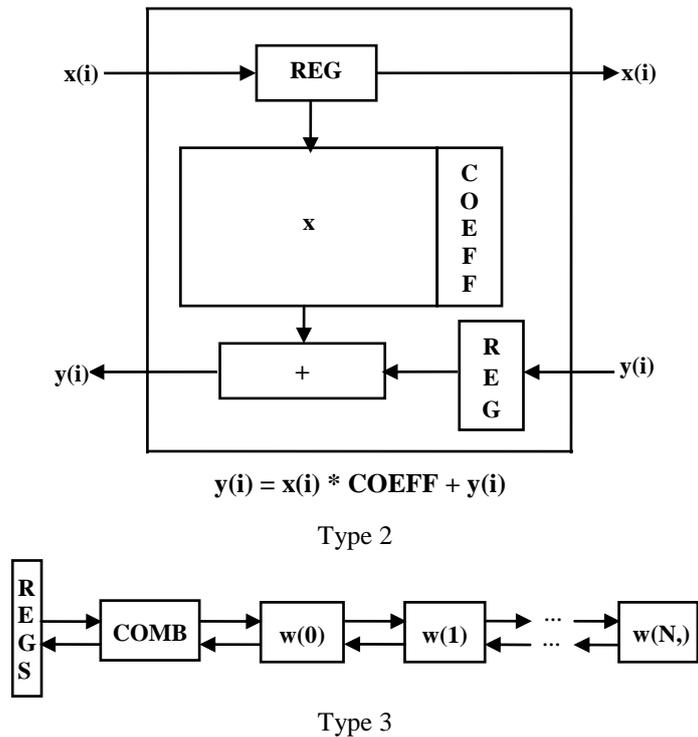


Fig.5. Overall Architectures namely type 1, type 2, type 3 [6]

The Architecture which mainly focuses on the software implementation like Modelsim, Matlab, Xilinx, Leonardo Spectrum Tools is presented in [9]. It uses the direct form FIR filters the intermediate results are stored and routed. In his decomposition algorithm the 1st level extracts the high frequency components while the higher level extracts the low frequency components. The DWT is calculated recursively as a series of convolutions and decimations of each octave. The PE in systolic array is fully pipelined and the communication edges among PE contain delay element and the whole system contains the local interconnections. This architecture uses a 6 tap non recursive FIR filter. The DWT coefficients are computed by multiply and accumulate method where partial products are computed separately and then added. The high pass and low pass coefficients are pre loaded into the design. Each cell has a single multiplier, adder and 2 registers. The architecture consists of filter cells, input delay, Baugh Wooley Multiplier and Carry Save Adder see Fig.6. The design can be easily modified to operate with filter types of higher degrees. The image is being transformed, if the transformed coefficients can be loaded from or written back to the external memory simultaneously can be a possible method for increasing the performance. This decreases the average amount of latency and therefore leads to a performance increase.

A wide variety of algorithms and architectures for computing the 1-D, 2-D DWT's was proposed in [10]. They introduced new on-line algorithms (such as RPA, MRPA) for DWT computation. The systolic array architecture implements these on-line algorithms for DWT computation in a word serial manner. This architecture is optimal under word serial model with respect to area and time.

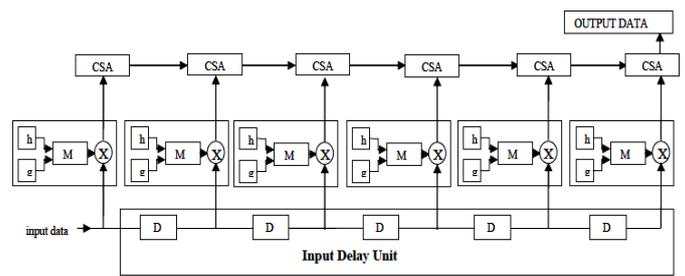


Fig.6. Systolic DWT Architecture [9]

They process the signal very fast. The word-serial model does not place any restrictions on the order of the inputs as long as they are input in a word serial manner. They proposed 2 systolic architectures, the first architecture has linear systolic array to computes both high pass and low pass outputs and a storage unit to store the inputs of a higher octave computation. The inputs are to the first octave in alternate clock cycles to one end of array while the inputs for higher order octave computations are fed in parallel from storage unit. This architecture requires a storage unit of $O(LJK)$ with a delay of $2N$ cycles which satisfies the word serial model.

The modified architecture of this is the second architecture which computes the DWT with a delay of N cycles. It consists of 2 linear arrays for computation of low pass and high pass each. The inputs are pre loaded into storage unit and then loaded parallel to the 2 systolic arrays. Area remains same as that of the previous architecture. Both these architectures can be extended for the 2-D implementation.

Table.1. Comparison of 1-D Systolic Architectures

Author	Latency	Area	Memory	MAC's	Control	Hardware Efficiency
[7]	$2N$	$O(NK)$	48reg, $2L(J+3)$	12mults, 12 adders	Simple	33%
[9]	$2N$		2reg/PE	1 adder/PE, 1mult/PE		
[1]	$O(N)$		2 reg/PE			
[7]	$2N$	$O(LK \log N)$	JL reg	L MAC's	Simple	40%
[7]	$2N$	$O(LK)$	$L(J+3)$ reg	L MAC's	Complex	
[2]				26 adders, 12mults	Complex	99%
[11]	$O(N)$		60 reg	18mults, 21 adders	Simple	
[12]	$O(N)$		49 reg	L	Simple	
[10]	$2N$	$O(LJK)$		L		
[10]	$\sim N$	$O(LJK)$		$2L$		

N = input data size, L = filter length, K = number of bits per input sample, J = total number of octaves

4. ARCHITECTURES FOR 2-D DWT

2-D is the extension of 1-D in Horizontal and vertical directions. The data gets most correlated from the transform when then the second dimension is considered. One can treat 2-D data as 1-D data and perform 1-D DWT on it but the transformation will be less effective. DWT compacts the majority of signal energy into low pass output. The resulting

approximate signal from 1 2-D transform will be $\frac{1}{4}$ the size of original whereas for 1-D it is $\frac{1}{2}$ the size of original. Therefore 2-D transform is more efficient for 2-D data than that of 1-D transform. The row inputs the data flow to one high pass and low pass filter. These filter outputs are down sampled by a factor of 2 (i.e. suppression for every 2 samples the output is discarded) by this the 1-D is said to complete for one octave. The 2-D sends each of the output to low pass and high pass again which operates along the columns. Outputs of these filters are again down sampled by a factor of 2 by this 2-D is said to complete for one octave which results in four signals each of the signal only of $\frac{1}{4}$ th the size of original input.

The architecture for computing the 2-D DWT using the recursive pyramid algorithm is proposed in [13]. It consists of systolic filters, parallel filters and a bank of registers. It requires an area of $O(NLK)$ and with a delay of $N2+N$ cycles. In [14] the hardware utilization is about 100% when it is modified for higher octaves the hardware utilization drops. The PE's use even and odd coefficients alternatively resulting in half the number of PE's that that are required in original. By this the throughput increases and hardware utilization drops. This architecture uses only 1 filter, 2 register banks. A total of 91% utilization is obtained for this architecture.

Architecture using a distributed memory and control architecture for 2-D DWT using a 6 tap filter is proposed in [15]. This architecture performs the 2-D Inverse Transformation using Space time mapping technique the dependence graph and systolic array architecture. A technique of control pipelining is followed for generating control tags. Each tag is used by PE to determine the correct state of operation. Thus the control functions are simple, global control is eliminated and occupies less area see Fig.7. Non linear transformation is applied to remove the inter octave dependencies. 2 MAC cells are used for performing filter operation here the PE's used for performing the filter operation differ in design some processing elements include the communication links for routing.

A Scalable Systolic architecture using a schematic way of mapping the data dependencies and life time chart to the architecture. Subband coding is designed mainly for finer frequency resolution particularly at lower frequencies. Each iteration doubles the frequency resolution due to halving of the low bandwidth, at each iteration; the current high pass band portion corresponds to the difference between the previous low band portion and the current band. The attractive feature of this architecture is its relative low complexity and it is independent of depth of iteration. In his architecture the average number of filtering operations are given by $8(1-4^{-m})/3$ where m is a 2-D wavelet. Since the number of low pass or high pass.

Operations are same the number of filters required are $4/3$. Thus it produces about 70% of hardware utilization. Since we know that DWT has blocking artifact problem which requires a lot of memory to store the intermediate data to solve this he used time space mapping technique. The memory requirement on Horizontal dimension is eliminated completely by moving the data around systolic array according to data dependencies but the vertical memory is still needed.

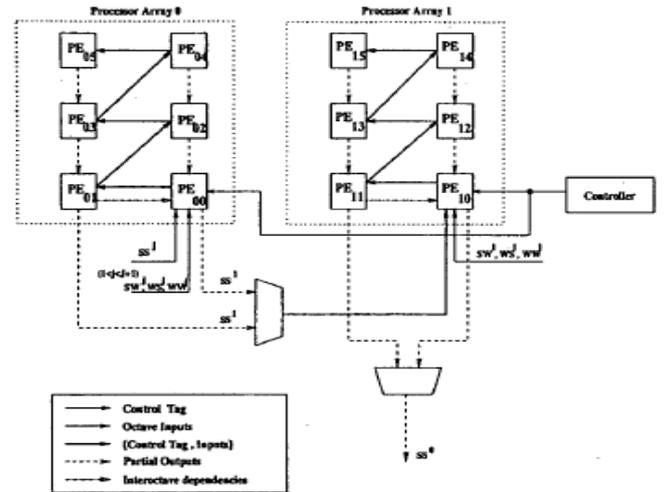


Fig.7. Systolic Array Architecture for 2-D IDWT [15]

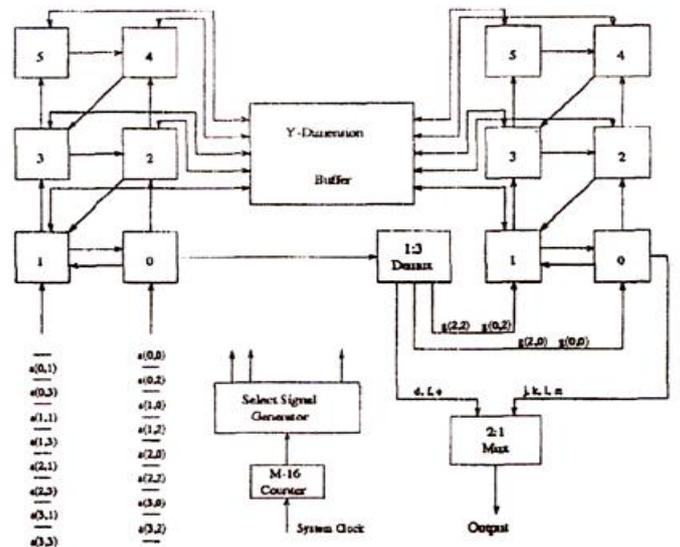


Fig.8. Systolic Array for 2-D DWT [16]

This vertical memory is incorporated internally to the PE. Each PE consists of 2 adders, 2 multipliers, accumulation latch, filter coefficient latch, data latch, mux and demux. For a block of $M \times N$ he uses $M, M/2$ latches for Horizontal and Vertical dimensions. By using a modulo-16 counter and set of mux-demux he designed a simple control unit. Some delay elements are used to resolve the conflict the between the low pass and high pass filter outputs. This architecture is shown in Fig.8.

The Pipelined Systolic Array Architecture is presented in [17] see Fig.9. In general the 2-D DWT is computed using 1-D devices in three stages. In first stage the 1-D DWT is performed on each of the N columns of the input image to obtain the intermediate matrix and in the second stage the intermediate matrix is transposed and in the third stage the N number of 1-D DWT is again performed on the transposed matrix to obtain the 2-D DWT. This architecture mainly eliminates the transposition of the generated matrix. It has 2 linear arrays; each of the linear arrays has P (Block Size) processing modules (PM). PM is again divided into PM1 and PM2. The linear array receives two blocks of sample of an input column in every clock cycle and performs

filtering operation on every alternate input sample. During a clock cycle PM1 yields 2 outputs the results produced by PM1 are propagated vertically to PM2 without a delay. PM2 also stores a pair of filter coefficients. PM2 performs the filter computation with 100% hardware utilization. The computations in PM2 are fully pipelined. The linear array receives the data in the column serial manner. This structure has an average computation time of $(N * N)/2P$. This design is suitable for high speed applications during the real time.

An (m-D) DWT architecture was presented in [19]. This architecture decomposes an image of size $N_1 \times N_2 \times N_3 \dots N_m$ in $N_m/(2^m-1)$ cycles. This architecture has little hardware complexity, simple control. His architecture contains a Ram module which is an efficient off-chip memory for storing data, multiplexer for selecting proper data for decomposition and 2^m sub band filters. Mux selects the original input data for first level decomposition only otherwise it selects the data from the RAM module see Fig.10.

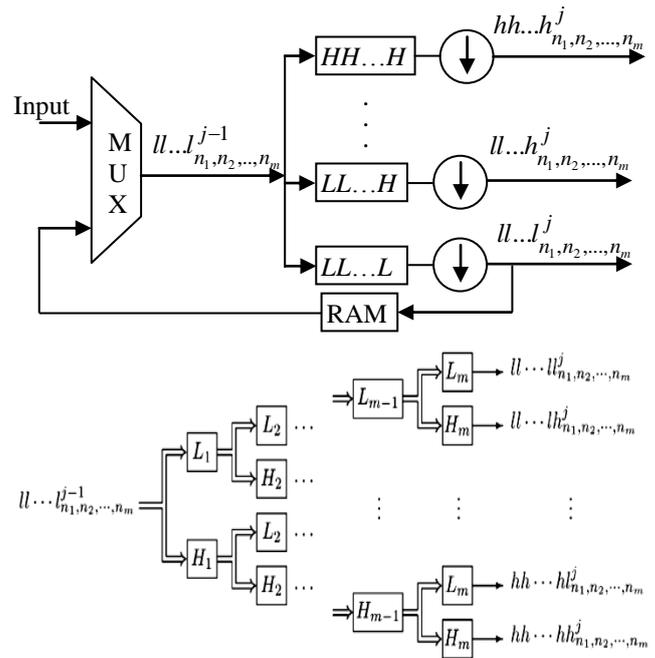


Fig.10. Architectures of m-D DWT and its subband filters [19]

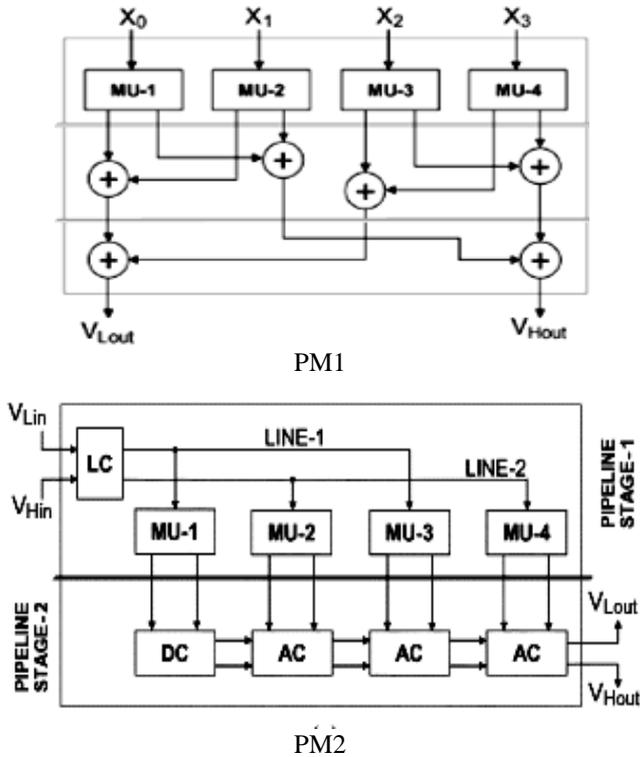


Fig.9. Internal Structure of PM1 and PM2 [17]

The latency between two data in the data stream can be avoided by using shift register. The RAM module used in this architecture is of the $(N * N)/4$. It requires latency of $(N * N)/3$, 8K multipliers, $8(K-1)$ a Computes of the 2-D DWT is done in [5] i.e. by operating the high-high, low-low, low-high, high-low components simultaneously. These architectures need some extra overhead such as memory unit or routing network and both the type 1 and type 2 architectures are having simple control unit adders and more on chip memory.

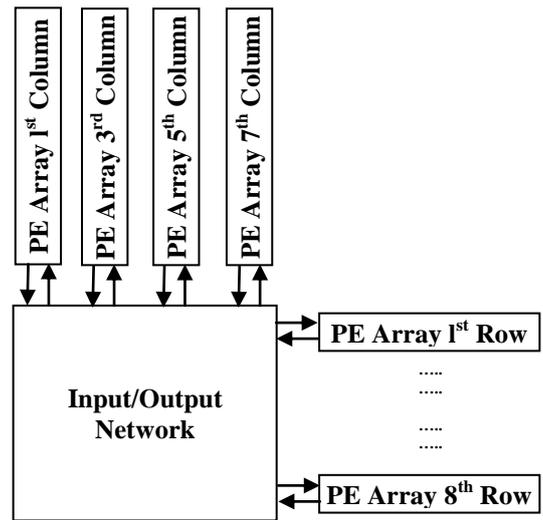


Fig.11. Systolic Architecture for a three level analysis filter (2-D DWT) [5]

Both of them differ in the total number of PE's type 1 has $(M * M)/2+M$ and type 2 has $2M$ where M stands for number of filter taps. Both the architectures are based on block based computation. The 2-D is implemented with the help of a memory unit and $3N' - 1$ -D DWT architectures where $N' = 0 < N' \leq N/2$, $N = \text{Image Size}$.

Table.2. Comparison of 2-D Systolic Architectures

Author	Latency	Area	Memory	MAC's	Control	Hardware Efficiency
[13]	$N(1+N)$	$O(NLK)$		2L,4Lmults, 4L adders		
[11]	$N(1+N)$		$N(2L-1)$	12PE's, 18 adders,	Simple	90-100%

				18 mults		
[15]				2L	Moderate	
[14]				18 adders, 18 mults	Simple	
[19]	N*N/3			8K mults, 8 (K-1) adders	Simple	
[17]	1/2N	2KN*N			Simple	99.9%

N = input data size, L = filter length, K = number of bits per input sample, J = total number of octaves

5. ARCHITECTURES FOR 3-D DWT

3-D Architectures are used for the compression of the video sequence and its decomposition is done by applying three 1-D transformations separately along the axes of video. In 2-D case x and y directions are used denoted as spatial coordinates where as for video an extra third dimension z is added for time.

The high throughput is obtained in the architecture of [19]. An throughput of $4/(T_m+T_a)$ and it requires 6K multipliers and 6(K-1) adders. It requires an on chip storage of O(MKN) and off chip storage of $O(N * N)$ and it computes the video sequence in $(MN^2)/7$ cycles. Since this architecture uses frame buffers to compute inter-frame DWT it occupies more space. To compute the N/2 PE's of 3-D DWT it requires (KN)/2 line buffers of size N.

An extension to their 2-D architecture by adding an extra module is presented in [20]. High throughput has been achieved by their architecture without any off chip memory. The intermediate coefficients generated in each cycle of computation are passed to the next stage without using any buffers. This architecture has (N/2) PE's arranged in a linear order fashion a total period of (N+K-2) cycles. The architecture has 3 stages of which each stage performs the operation of a each subcell. Each subcell is having a multiplication unit which stores the low pass and high pass filter coefficients. The values are fed into it using Serial In Parallel Out fashion. It has a latency of (K+2+logK) cycles and each subcell requires 3K multipliers, (3K-1) adders.

Table.3. Comparison of 3-D Architectures

Author	Latency	Area	Memory	MAC's	Control	Hardware Efficiency
[19]	$(M*N*N)/7$	$(KN)/2$	$(K-2)(N+2)M/4$	24Kmults, 24(K-1) adders	Moderate	
[20]	O(N)		$1.5K+N(K+1.5)$	9K mults, 9(K-1) adders	Simple	~100%

N = input data size, L= filter length, K = number of bits per input sample, J = total number of octaves

6. CONCLUSION

Architectures for 1-D, 2-D as well as for 3-D have been proposed by various authors. Each Architecture has its own advantages and disadvantages compared to the other. Some architectures was designed to overcome the drawbacks of their previous architecture, this paper gives an overview of implemented architectures and their performance. Comparison of 1-D, 2-D and 3-D gives the detailed information. The trade-

off between the area and latency determines the structure of architecture. Depending upon the type of application particular architecture can be chosen.

REFERENCES

- [1] G. Knowles, "VLSI Architecture for the Discrete Wavelet Transform", *Electronics Letters*, Vol. 26, No. 15, pp. 1184-1185, 1990.
- [2] T. Acharya, "A Systolic Architecture for Discrete Wavelet Transforms", *13th International Conference on Digital Signal Processing Proceedings*, Vol. 2, pp. 571-574, 1977.
- [3] T. C. Denk and K. K. Parhi, "Systolic VLSI architectures for 1-d discrete wavelet transforms, in Signals, Systems & Computers", *Conference Record of the Thirty-Second Asilomar Conference on Signals, Systems and Computers*, Vol. 2, pp. 1220-1224, 1998.
- [4] J. Fridman and E. S. Manolakos, "Distributed Memory and Control VLSI Architectures for the 1-D Discrete Wavelet Transform", *7th IEEE Proceedings on VLSI Signal Processing*, pp. 388-397, 1994.
- [5] Sung Bum Pan and Rae-Hong Park, "Systolic array architectures for computation of the discrete wavelet transform", *Journal of Visual Communication and Image Representation*, Vol. 14, No. 3, pp. 217-231, 2003.
- [6] M. Vishwanath, R. M. Owens and M. J. Irwin, "VLSI Architectures for the Discrete Wavelet Transform", *IEEE Transactions on Circuits and Systems- II: Analog and Digital Signal Processing*, Vol. 42, No. 5, pp. 305-316, 1995.
- [7] M. Vishwanath, R. M. Owens, and M. J. Irwin, "Discrete Wavelet Transforms in VLSI", *Proceedings of the International Conference on Application Specific Array Processors*, pp. 218-229, 1992.
- [8] S. B. Pan and R-H Park "VLSI Architectures of the 1-D and 2-D discrete wavelet transforms for JPEG 2000", *Journal of Signal Processing*, Vol. 82, No. 7, pp. 981-992, 2002.
- [9] S. Sankar Sumanth and K. A. Narayanan Kutty, "VLSI Implementation of Discrete Wavelet Transform using Systolic Array Architecture", *Advances in Computer Information Sciences and Engineering, Business Media*, pp. 467-472, 2008.
- [10] C. Chakrabarti and C. Mumford, "Efficient realizations of discrete and continuous wavelet transform: from single chip implementations to mappings on SIMD array computers", *IEEE Transactions on signal processing*, Vol. 43, No. 3, pp. 759-771, 2002.
- [11] S. Syed, M. Bayoumi and J. Limqueco, "An Integrated Discrete Wavelet Transform Array Architecture", *Proceedings of the Workshop on Computer Architecture for Machine Perception*, pp. 32-36, 1995.
- [12] A. Grzeszczak, M. K. Mandal and S. Panchanathan, "VLSI Implementation of Discrete Wavelet Transform", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 4, No. 4, pp. 421-433, 2002.
- [13] M. Vishwanath and C. Chakrabarti, "A VLSI Architecture for Real-Time Hierarchical Encoding/Decoding of Video using the Wavelet Transform", *IEEE International*

- Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 401-404, 1994.
- [14] J. Chen and M. Bayoumi, "A Scalable Systolic Array Architecture for 2-D Discrete Wavelet Transforms", *Proceedings of IEEE Workshop on VLSI Signal Processing*, Vol. III, pp. 303-312, 1995.
- [15] J. Singh, A. Antoniou and D. J. Shpak, "A Systolic Array Architecture for 2-D Inverse Wavelet Transform", *Pacific Rim Conference on Communications, Computers and Signal Processing*, pp. 193-196, 1999.
- [16] J. Chen and Magdy A. Bayoumi, "A Scalable Systolic Array Architecture for 2-D Discrete Wavelet Transform", *IEEE Signal processing Society workshop on VLSI Signal processing*, Vol. VIII, pp. 303-312, 1995.
- [17] P. K. Meher, B. K. Mohanty and J. C. Patra "Hardware-Efficient Systolic-like Modular Design for Two-Dimensional Discrete Wavelet Transform", *IEEE Transactions on Circuits and Systems-II, Express Briefs*, Vol. 55, No. 2, pp. 151-154, 2008.
- [18] B. K. Mohanty and P. K. Meher, "Systolic architecture for transposition-free VLSI Implementation of 2-D DWT", *10th IEEE International Conference on Communication Systems*, pp. 1-5, 2006.
- [19] Qionghai Dai, Xinjian Chen and Chuang Lin, "A novel VLSI architecture for multidimensional discrete wavelet transform", *IEEE Transactions on Circuit and Systems for Video Technology*, Vol. 14, No. 8, pp. 1105-1110, 2004.
- [20] P. K. Meher and B. K. Mohanty, "Concurrent Systolic Architecture for High-Throughput Implementation of 3-Dimensional Discrete Wavelet Transform", *19th IEEE International Conference Applications- specific systems, Architectures and processors*, pp. 168-172, 2008.
- [21] Stephane G. Mallat, "Multifrequency Channel Decompositions of Images and Wavelet Models", *IEEE Transactions of Acoustics, Speech and Signal Processing*, Vol. 37, No. 12, pp. 2091-2110, 1989.
- [22] K. K. Parhi and T. Nishitani, "VLSI Architectures for Discrete Wavelet Transforms", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 1, No. 2, pp. 191-202, 2002.