Vol 05, Article 08401, August 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

IMPLEMENTATION OF 2-D TWO LEVEL DWT VLSI ARCHITECTURE

KANCHETI SIVAPRIYA, VENKATASAICHAND NANDANAVANAM
Department of Electronics & Communication Engineering, QIS College, Ongole
sivapriya0192@gmail.com, saichandnandanavanam@gmail.com

ABSTRACT

Images are to be transmitted without loss of information. That can be achieved by transforming using Discrete Wavelet Transform (DWT). The discrete wavelet transform (DWT) is being increasingly used for image coding. This is due to the fact that DWT supports features like progressive image transmission (by quality, by resolution), ease of transformed image manipulation, region of interest coding, etc. Hence, there is a need of design efficient & fast architecture for DWT. This paper is introducing an efficient architecture to enhance speed of DWT Computation. The Discrete Wavelet Transform (DWT) is based on time-scale representation, which provides efficient multi-resolution. The introduced architecture increases levels of DWT architecture to achieve lower computational complexity and reduced memory. As the DWT traditionally been implemented by convolution which demands both a large number of computations and a large storage features that are not desirable for either high-speed or low-power applications. This paper describes Lossless 2-D DWT (Discrete Wavelet Transform) using folded Scheme Architecture to reduce computational overheads. The behavior of designed DWT architecture is modeled using the Verilog HDL and functionality could be verified using the Modelsim simulation tool.

Index Terms–Discrete wavelet transform, very-large-scale integration (VLSI), folded architecture, single-input/single-output

1. INTRODUCTION

The discrete wavelet transform (DWT) has been widely used for image coding.. The existing architectures for implementing the DWT are mainly classified into two categories: 1) convolution based and 2) FIR based [2]–[3]. Since the lifting-based architectures have advantages over the computation complexity and memory requirement, more attention is paid on the lifting based ones. The Jou *et al.* proposed architecture for directly implementing the lifting scheme. Based on this direct architecture, Lian *et al* proposed a folded architecture to increase the hardware utilization. Unfortunately, these architectures have limitations on the critical path latency and memory requirement. The flipping structure can reduce the critical path latency by eliminating the multipliers on the path from the input node to the computation node without hardware overhead.

To construct efficient single input/single output (SISO) and TITO VLSI architecture based on folded scheme, which meets the high processing speed requirement with controlled increase of hardware cost and simple control signals. High processing speed can be achieved when multiple row data samples are processed simultaneously. And time multiplexing technique is adopted to control the increase of the hardware cost.

Furthermore, the control signals are simple, since the regular architecture is a combination of simple single-input/single-output (SISO) modules and two-input/two-output (TITO) modules. It provides a variety of hardware implementations to meet different processing speed requirements the rapid progress of VLSI design technologies, many processors based on audio and image signal processing have been developed recently. The two-dimensional discrete wavelet transform (2-D DWT) plays a major role in theJPEG-2000 images compression standard. Presently, research on the DWT is attracting a great deal of attention. In addition to audio and image compression the DWT has important applications in many areas, such as computer graphics, numerical analysis, radar target distinguishing and so forth. The architecture of the 2-D DWT is mainly composed of the multirate filters.

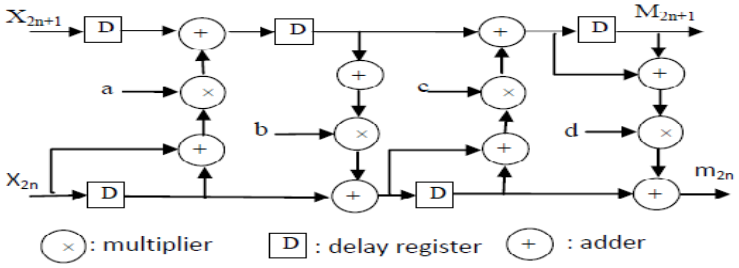


Fig.1.Two-input/two-output lifting architecture of the CDF97

Because extensive computation is involved in the practical applications, e.g., digital cameras, high efficiency and low-cost hardware is indispensable. At present, many VLSI architectures for the 2-D DWT have been proposed for folded scheme to meet the requirements of real-time processing.

Vol 05, Article 08401, August 2014 International Journal of VLSI and Embedded Systems-IJVES
http://ijves.com ISSN: 2249 – 6556

However, because the filtering operations are required in both the horizontal and vertical directions, designing a highly efficient architecture at a low cost is difficult. Lewis and Knowles used the four-tap Daubechies filter to design a 2-D DWT architecture. Parhi and Nishitani proposed two architectures that combine the word-parallel and digital-serial methodologies. Chakrabarti and Vishwanath presented the non-separable architecture and the SIMD array architecture. Vishwanath *et al.* employed two systolic array filters and two parallel filters to implement the 2-D DWT. The modified version uses four parallel filters as reported in and . Chuang and Chen proposed a parallel pipelined VLSI array architecture for the2-D DWT. Chen and Bayoumi [18] presented scalable systolic array architecture. Other 2-D DWT architectures have been reported Among the various architectures, the best-known design for the 2-D DWT is the parallel filter architecture .The design of the parallel filter architecture is based on the modified recursive pyramid algorithm (MRPA), which intersperses the computation of the second and following levels among the computation of the first level. The MRPA is feasible for the 1-DDWT architecture, but is not suitable for the 2-D DWT, because the hardware utilization is inefficient and a complicated control circuit results from the interleaving data flow. Therefore, in this paper, we propose a new VLSI architecture for the folded scheme 2-D DWT. The advantages of the proposed architecture are the100% hardware utilization, fast computing time, regular dataflow, and low control complexity. Additionally, because of the regular structure, the proposed architecture can easily be scaled with the filter length and the 2-D DWT level.

2. LINE-BASED 2-D WAVELET TRANSFORM

Image data is usually acquired in a serial manner. For example, a very common way to acquire image data is to scan an image one line at a time. Throughout this paper, we will assume our system operates with this line-by-line acquisition. Given this, our objective in this section will be to design a 2-DWT that requires *storing a minimum total number of lines*. The assumption is that images are stored in memory only while they are used to generate output coefficients, and they are released from memory when no longer needed. Obviously, performing a 1-D WT on a single line can be done without significant memory. However, in order to implement the separable 2-D transform the next step is to perform column filtering and here memory utilization can become a concern. For example, a completely separable implementation would require thus memory sizes of the order of the image size will be required.

3. LIFTING AND FOLDED SCHEME OF DWT

Lifting scheme is a relatively old method to construct wavelet bases, which was first introduced by Sweldens in 1990s [4]. This scheme is called the second-generation wavelet, which leads to a fast in-place implementation of the DWT. According to [4], any DWT of perfect reconstruction can be decomposed into a finite sequence of lifting steps. This decomposition corresponds to a factorization for the poly-phase matrix of the target wavelet filter into a sequence of alternating upper and lower triangular matrices and a constant diagonal matrix, which can be expressed as follows

$$h(z)=h_e(z^2)+z^{-1}h_0(z^2)$$
$$g(z)=g_e(z^2)+z^{-1}g_0(z^2)$$

$$p(z)=\begin{bmatrix} he(z) & ge(z) \\ ho(z) & go(z) \end{bmatrix}$$

$$=\prod^{m}_{i+1}\begin{bmatrix} 1 & si(z) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ ti(z) & 1 \end{bmatrix}\begin{bmatrix} K_0 & 0 \\ 0 & K_1 \end{bmatrix}$$

Where h(z) and g(z) are the low-pass and high-pass analysis filters, respectively. Equation (1) is the poly-phase decomposition and P(z) is the poly-phase matrix. For example, the (9, 7) filter (CDF97) adopted in JPEG2000 can be decomposed into four lifting stages as follows

$$P(z)=\begin{bmatrix} 1 & a(1+z^{-1}) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ b(1+z) & 1 \end{bmatrix}\begin{bmatrix} 1 & c(1+z^{-1}) \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ d(1+z) & 1 \end{bmatrix}\begin{bmatrix} K_0 & 0 \\ 0 & K_1 \end{bmatrix}$$

Where a, b, c, and d are the lifting coefficients, and K_0, K_1 are the scale normalization coefficients. The scale normalization coefficients can be implemented together with the quantization, if image compression is performed [7]. Thus, we focus on the implementation of the lifting stages in this brief paper. The odd (even) indexed data samples are represented by x_{2n+1} (x_{2n}). The intermediate values computed during lifting steps are

denoted as m^k_{2n+1} and m^k_{2n} (k = 0, 1, 2, 3), and the high- and low-frequency coefficients are expressed as the

Vol 05, Article 08401, August 2014 International Journal of VLSI and Embedded Systems-IJVES
http://ijves.com ISSN: 2249 – 6556

sequence m_{2n+1} and m_{2n}, respectively. With these mathematical notations, the implementation of the CDF97 can be rewritten as follows:

$$m_{2n+1}^{0} = x_{2n+1}$$

$$m_{2n}^{0} = x_{2n}$$

$$m_{2n+1}^{1} = m_{2n+1}^{0} + a(m_{2n}^{0} + m_{2n+2}^{0})$$

$$m_{2n}^{1} = m_{2n}^{0}$$

$$m_{2n+1}^{2} = m_{2n+1}^{1}$$

$$m_{2n}^{2} = m_{2n}^{1} + b(m_{2n-1}^{1} + m_{2n+1}^{1})$$

$$m_{2n+1}^{3} = m_{2n+1}^{2} + c(m_{2n}^{2} + m_{2n+2}^{2})$$

$$m_{2n}^{3} = m_{2n}^{2}$$

$$m_{2n+1} = m_{2n+1}^{3}$$

$$m_{2n} = m_{2n}^{3} + d(m_{2n-1}^{3} + m_{2n+1}^{3})$$

Based on (3), the two-input/two-output lifting architecture of the CDF97 is shown in Fig. 1.

3.1 Architecture for the Horizontal Filtering along the Rows (M = 8)

First, CDF97 is applied to the row dimension, which is a 2D DWT. The architecture for the horizontal filtering along the rows consists of eight SISO modules, as shown in Fig. 2. The input data flow is shown in Fig. 3.The elements from each row are processed by one SISO module. We can accordingly get output data flow of the architecture for the horizontal filtering, as shown in Fig. 4, where mi,j denotes the computation results after we apply CDF97 to the row dimension.

Many SISO architectures for the 2D DWT are proposed folded scheme. An efficient SISO architecture is proposed in by employing the fold technique. Therefore, we adopt it in our SISO modules, which consists of two multipliers, four adders, and ten registers.

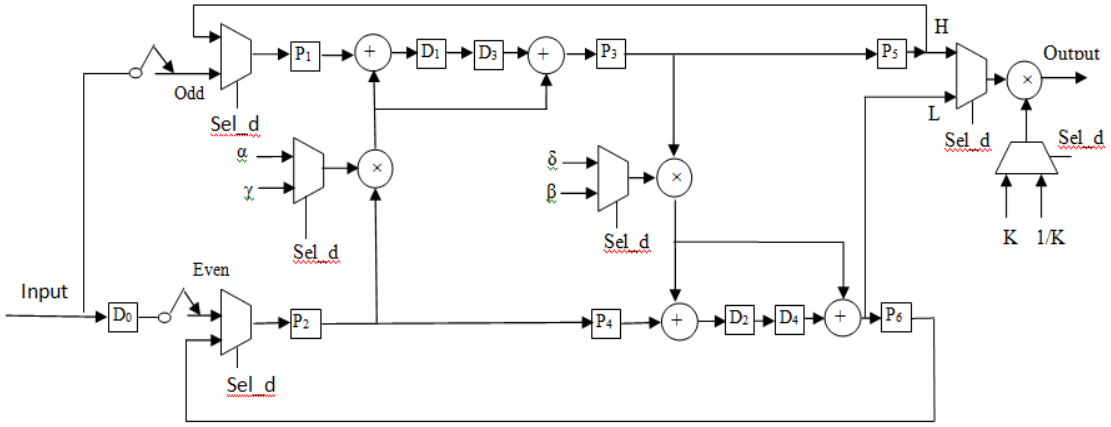


Fig. 2.Proposed Folded Architecture

3.2 Architecture for the Vertical Filtering along the Columns (M = 8)

Second, CDF97 is applied to the column dimension. Eight elements (for example, $m_{0,0}$, $m_{1,0}$ . . .$m_{7,0}$) from each column arrive simultaneously. One solution is to design an eight-input/eight-output architecture by directly mapping, which can process eight elements per clock cycle. However, it will lead to a complex architecture, which has complicated control signals and cannot be easily extended for its irregularity. For example, a four-input/four output can process four elements from each column per clock cycle. The eight-input/eight-output architecture will be more complicated.

The problem can be solved by folded technique, which converts the possible complex architecture into a combination of some simple TITO modules. The proposed architecture for the vertical filtering is shown in Fig. 6. It consists of four TITO modules, some multiplexers (mux), and line delay registers (LD). Each TITO module

Vol 05, Article 08401, August 2014 International Journal of VLSI and Embedded Systems-IJVES
http://ijves.com ISSN: 2249 – 6556

is designed to receive two elements simultaneously. The LD is composed of some delay registers, which are used to temporarily store the elements. Port A (Port B, Port C ....Port H) is used as a symbol to demonstrate how the mux works. The elements, which have been delayed by the LD, will be selected by the mux and sent to TITO modules at a proper time.

The number of delay registers in LD (LD1, LD2, LD3) is determined by the rule: The time interval arriving at the same TITO module for every two elements from one column is constant. The time interval is measured by the number of clock cycles (ccs).We assume that the length of the image is N pixels, the throughput rate of proposed architecture is M pixels/ccs, and the computation time interval between the first pair and the second pair of the elements from one column is ccs. Therefore architecture by directly mapping (3) is shown in Fig. 5,

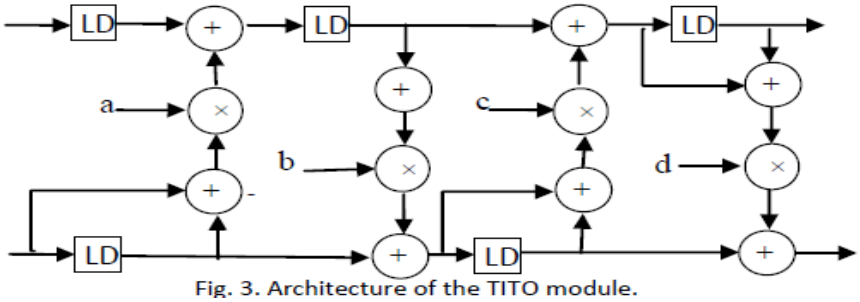


Fig. 3. Architecture of the TITO module.

the number of the input in the architecture for the vertical filtering is M, then the computing time interval between the first pair and the (M / 2 + 1)th pair(such as $m_{0,0}$, $m_{1,0}$ and $m_{8,0}$, $m_{9,0}$) is M / 2 ×Δ ccs. On the other hand, from Figs. 4 and 6, the time interval between the first pair and the (M / 2 + 1)th pair arriving at the same TITO module is equal to

$$\frac{N\ pixels \times M}{M\frac{pixels}{ccs}} = N\ css$$

Therefore, the following formula should be satisfied:

$$\frac{M}{2} \times \Delta = N$$

By solving (5), we have that

$$\Delta = \frac{2N}{M}.$$

The selected time interval for the mux from one input port to another is equal to ccs. Because every two elements from one column will arrive at the same TITO module with a fixed time interval, the architecture of the TITO module can be constructed by replacing the delay registers used in Fig. 1 with corresponding LD4. The architecture of the TITO module is shown in Fig. 3. The LD4 is used to synchronize the intermediate results of the vertical filtering, in which the number of delay registers is equal to ccs. An example is given to show the data flow in a TITO module (PORT E). P1 (P2, . . . P10), as shown in Fig. 3, is the intermediate node of the architecture of TITO module, which is used to show how the vertical filtering is processed. The data flow is shown in Table 1, where mm_{ij} denote the computation results after CDF97 is applied to the column dimension.

The subscript i and j represent the coordinates of the results. And mm_{ij}^{k} (k =0, 1, 2, 3) denote the intermediate results of the vertical filtering, which are calculated as in (3). For example, $mm_{1,0}^{1}$, can be calculated from $mm_{0,0}^{0}$, $mm_{1,0}^{0}$ and $mm_{2,0}^{0}$ of the same column. Based on (3), Figs. 8 and 9, and Table 1, it can clearly show how the computation results are calculated.

4. TITO VLSI Architecture for 2-D Folded-Based DWT

An efficient multi-input/multi-output VLSI architecture (MIMOA) is constructed as shown in Fig.4, which meets the high processing speed requirement with controlled increase of hardware cost and simple control signals. High processing speed can be achieved when multiple row data samples are processed simultaneously. And folded technique is adopted to control the increase of the hardware cost for the MIMOA. Furthermore, the control signals are simple, since the regular architecture is a combination of simple single input/ single-output (SISO) modules and two-input/two-output (TITO) modules. It provides a variety of hardware implementations to meet different processing speed requirements by selecting different throughput rates.

Vol 05, Article 08401, August 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

If the particular application requires higher processing speed, then throughput rate M will be larger than 8. The proposed SISO &TITO(folded) modules, M/2 TITO modules, some multiplexers, and delay registers, which is shown in Fig.8. "2N/M D" in Fig. 8 represents that the number of delay registers is 2N/M. The number of registers required in one SISO module is 10. Thus, the total number of registers required in MIMOA is given by
= M x 10 + 2 [2N/M + 4N/M +.....+ [(M-2) N]/M + [M/ 2] x 5 x [2N/ M] = 10M + N (M-2)/2 + 5N (4) = 10M+ (NM/2) +4N

The number of the required adders is equal to
4M + (M/2) x 8 = 8M
The number of the required multipliers is equal to
2M + (M/2) x 4 = 4M (6)

5. RESULTS

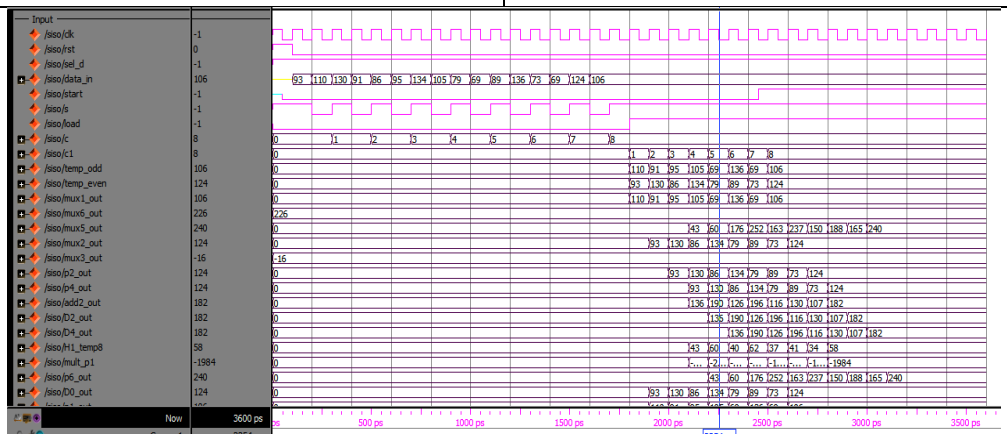
This section discusses about the functional verification of the Architecture Folded DWT. The behaviour of folded DWT architecture is described in Verilog HDL & simulation has been done by using Modelsim. The input & output of the various functional blocks of the designed architecture is listed below.

SISO:

The table 1 describes the inputs & outputs of SISO respectively

Table1.Inputs to SISO

INPUT DATA	OUTPUT DATA
SISO Inputs = 93 110 130 91 86 95 134 105 79 69 89 136 73 59 124 106	SISO Outputs = 124 148 151 187 127 163 107 196 166 184 184 183



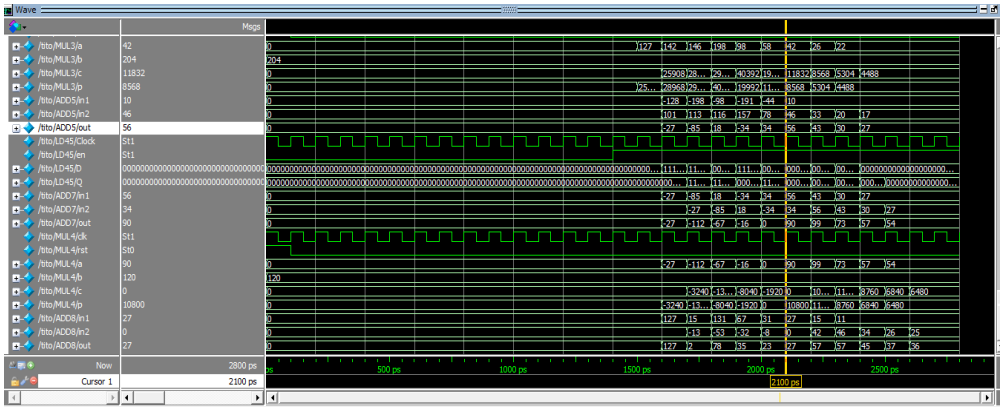
Tito Results:

This section discusses about the functional verification of the system. The table 2 describes the inputs & outputs of TITO respectively

Table2. Inputs to TITO

INPUT DATA	OUTPUT DATA
TITO Inputs = 124 148 151 187 127 163 107 196 166 184 184 183	TITO Outputs = 46 -109 -55 -37 -22 -18 -17 148 188 180 151 186 204 213 215 216

Vol 05, Article 08401, August 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556



6.CONCLUSION

In this brief, we have proposed a novel EFA for the DWT. We have given a new formula for the folded algorithm**.** Then, by employing the of SISO techniques, the conventional data flow of the lifting-based DWT is converted to a parallel one, resulting in the OA with repeatable property. Based on this property, the proposed EFA is derived from the by further employing the fold technique.

The FPGA implementation results show that the proposed EFA possesses short critical path latency and achieves high hardware utilization. Performance comparisons indicate that our EFA provides an efficient alternative in trade-off among the critical path latency, hardware cost, and control complexity.

REFERENCES

[1].C. Christopoulos, A. Skodras, and T. Ebrahimi, "TheJPEG2000 Still Image Coding System: An Overview"IEEE Trans. Consumer Electronics, vol. 46,no. 4, pp.1103-1127, Nov. 2000**.**

[2].C. Chrysafis and A. Ortega, "Line-Based, Reduced Memory, Wavelet Image Compression," IEEE Trans. Image Processing. vol. 9, no. 3, pp. 378-389, Mar.2000**.**

[3].P. Wu and L. Chen, "An Efficient Architecture for Two-Dimensional Discrete Wavelet Transform," IEEE Trans. Circuits and Systems for Video Technology, vol. 11, no. 4, pp. 536-545, Apr. 2001.

[4].Daubechies and W. Sweldens, "Factoring Wavelet Transforms intoLifting Steps," J. Fourier Analysis and Applications, vol. 4, no. 3, pp. 247-269, 1998.

[5].M. Grangetto, E. Magli, M. Martina, and G. Olmo, "Optimization and Implementation of the Integer Wavelet Transform for Image Coding," IEEE Trans. Image Processing, vol. 11, no. 6, pp. 596-604, June 2002.

[6].H. Liao, M.K. Mandal, and B.F. Cockburn, "Efficient Architectures for 1Dand 2D Lifting-Based Wavelet Transform," IEEE Trans. Signal Processing, vol. 52, no. 5, pp. 1315-1326, May 2004.

[7].C.T. Huang, P.C. Tseng, and L.G. Chen, "Flipping Structure: An Efficient VLSI Architecture for Lifting Based Discrete Wavelet Transform," IEEE Trans. Signal Pr C.T. Huang, P.C. Tseng, and L.G. Chen, "Flipping Structure: An Efficient VLSI Architecture for Lifting Based Discrete Wavelet Transform," IEEE Trans. Signal Processing, vol. 52, no. 4, pp. 1080-1089, Apr. 2004.

[8].T. Acharya and C. Chakrabarti, "A survey on lifting-based discrete wavelet transform architectures," J. VLSI Signal Process., vol. 42, no. 3,pp. 321–339, Mar. 2006.

[9].C. Chakrabarti and M. Vishwanath, "Efficient realizations of discrete and continuous wavelet transforms: From single chip implementations to mapping on SIMD array computers," IEEE Trans. Signal Process., vol. 43, no. 3, pp. 759–771, Mar. 1995.

[10]. M. Vishwanath, R.M. Owens, and M. J. Irwin, "VLSI architectures for the discrete wavelet transform,"IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol.42, no. 5, pp. 305–316, May 1995

AUTHOR BIOGRAPHY

Kancheti Sivapriya was born on 10th Aug 1991 at Narasaraopet, India. She received her, B.Tech in Electronics & Communication Engineering from EVM College of Engineering, Narasaraopet, affiliated to JNTU, Kakinada, AP, India and Pursuing M.Tech in DECS at Qis College of Engineering, Ongole, affiliated to JNTU, Kakinada, AP, India. Her areas of interest are Digital Vlsi design and Vlsi image processing.

Venkata Saichand Nandanavanam received his B. Tech in Electronics & Communication Engineering from PBR Viswodaya Institute of science and technology Kavali, AP, India. And M. Tech degree in VLSI Design from Amrita Unversity, Coimbatore, Tamilnadu, India. He is currently working as an Asst.prof in Qis College of Engineering and technology Ongole, AP, India. And His work experience is 5 years. His areas of interest are Digital Vlsi design, VLSI Imageprocessing and Signal processing, Neural networks.