

# A LOW COMPLEXITY ARCHITECTURE FOR COMPLEX DISCRETE WAVELET TRANSFORM

Swapna Banerjee\*

Bipul Das

Department of ECE, University of Illinois  
Chicago, IL - 60607, USA

Department of E & ECE  
Indian Institute of Technology  
Kharagpur, India - 721302  
e-mail: swapna@ece.iitkgp.ernet.in

## ABSTRACT

This paper presents a low complexity architecture for realization of CDWT for minimal resource environment like FPGA and SoC applications in image processing. The proposed architecture uses parameterization of the coefficients for reduced redundancy in computation. Use of CSD based multipliers along with pipelined CORDIC provided an architecture for efficient implementation in constrained environment along with high speed and efficiency. Low data path length and lower control complexity are other salient features of the architecture presented in the paper. This design was implemented on Xilinx FPGA platform. The operational frequency of the implemented circuit is 47MHz.

## 1. INTRODUCTION

CDWT - an acronym for *Complex Discrete Wavelet Transform* [1, 2, 3, 4] which uses a complex kernel for construction of the *mother wavelet* is a better choice for motion estimation and other 3-D image processing, like stereo imaging applications and preserves the properties of shift invariance and good directional selectivity.

VLSI implementation of DWT has been proposed by a number of researchers [5, 6, 7]. But very few architectures have been so far reported for CDWT [8].

The architecture for CDWT presented in [8] though promises to be efficient enough for high speed applications but is highly hardware intensive. Moreover, the use of hyperbolic CORDIC units restrict the CDWT architecture to a very limited filter order. This paper presents an architecture to provide much more flexibility to the design and also ensures efficient hardware utilization. This architecture also aims at optimization in the design of the segments for hyperbolic part of the filter. The Canonical Signed Digit (CSD) based multipliers has been used. In fact the use of CSD has provided more gain in hardware complexity, as the average number of operations reduces to  $\frac{b}{3}$ , where  $b$  is the wordlength of the machine. This architecture also uses parameterization of the CDWT coefficients thus reducing the computational overhead effectively.

## 2. THEORETICAL BACKGROUND OF CDWT

Rational valued complex kernel is used to realize the Complex Discrete Wavelet Transform (CDWT). These kernels can be modeled by even length FIR filter with approximate Gabor form, given

by [3] :

$$h(n) \approx a_0 e^{-0.5(\frac{n-n_0}{\sigma_0})^2} e^{j\omega_0(n-n_0)} \quad (1)$$

$$g(n) \approx a_1 e^{-0.5(\frac{n-n_0}{\sigma_1})^2} e^{j\omega_1(n-n_0)}$$

where  $n$  denote the samples in the sampling window. The Gaussian profile is symmetrically distributed about the point  $n_0$  in the interval  $[-D, D-1]$ , where  $D$  is the window half-length.  $a_0$  and  $a_1$  are the amplification factors of the complex kernel, which were considered 0.5 allowing a mean square error of 0.0012 and 0.0048 for the low and high pass filters respectively. This resulted in a drop of PSNR by 0.063db for 3 level of decomposition of  $512 \times 512$  Lenna image.  $\omega_0$  and  $\omega_1$  are the modulation frequencies and,  $\sigma_0$  and  $\sigma_1$  are the window standard deviations.

## 3. ARCHITECTURAL DESIGN OF CDWT FILTER

The block level architecture is shown in Fig. (1). The main architecture comprises of a RAM dedicated for storing the raw image and the low pass coefficients after each level of decomposition. The transform section performs the CDWT operation over the data fetched from the RAM. From the transform block, 8-set of coefficients,  $A^{(1,l)}, D^{(2,l)}, D^{(1,l)}, D^{(3,l)}, A^{(2,l)}, D^{(5,l)}, D^{(4,l)}, D^{(6,l)}$ , (where  $l$  denotes the transform level) are generated at each clock.

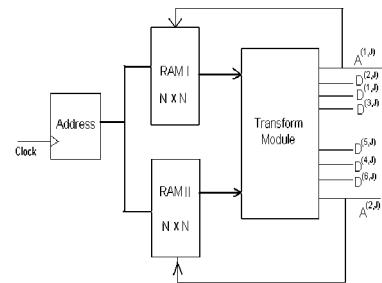


Fig. 1. Block level diagram for CDWT architecture

The two approximation coefficients ( $A^{(1,l)}, A^{(2,l)}$ ), i.e., the *LL* component of DWT, for the complex and the conjugate set of filters are stored back to the two RAMs (RAM I and RAM II). Each of the complex coefficients have two parts - the real and the imaginary. Thus the storage requirement of the whole design is  $2N^2$ .

\*Author for correspondence

The “address” section fetches four pixels from four successive locations of the RAM either along the column or along the row according to which way it is decomposing.

The general form Eqn. 1 can be written as:

$$h(n) = a_0 f_n \theta_n \text{ where } f_n = e^{-\frac{(n-n_0)^2}{2\sigma^2}} \quad (2)$$

and for low pass filter  $\theta_n = e^{j\omega_0(n-n_0)}$ , while for high pass filter  $\theta_n = e^{j\omega_1(n-n_0)}$ . For the conjugate expressions for low pass and high pass filters the expression for  $\theta_n$  are  $e^{-j\omega_0(n-n_0)}$  and  $e^{-j\omega_1(n-n_0)}$  respectively.

The factor  $f_n$  comprises of the real exponential part of the expression. The complex part on the other hand is given by  $\theta_n$ . A complex multiplication requires 4 multipliers and adders making it expensive for most DSP applications. Many number systems have been proposed so far for the reduction of multipliers of the complex multiplication [9, 10]. On the other hand CORDIC is an economic way of computation of complex numbers as long as they can be expressed in trigonometric functions [11, 12, 13, 14].

Since, for a given length filter, the real coefficients are fixed, so use of CSD constant multipliers prove to be very effective for computation of the real part. In general the input data can be considered as complex number. Actually in the first level of decomposition, the input is a real valued sample. But in the next stages the inputs are essentially complex numbers. So effectively for each multiplication two multipliers are required.

Due to the symmetric property of the Gaussian distribution the two halves about the midpoint are identical. So for a Gaussian window of width N, the correlation between the sample points are given by:  $f_0 = f_{N-1}$ ,  $f_1 = f_{N-2}$ , ...,  $f_{\frac{N}{2}-1} = f_{\frac{N}{2}}$ . For explanation, design of an 8-tap filter is considered. In general the length of the filter that can be accommodated is determined by the precision of the machine. A 16-bit machine is considered for illustration. The maximum precision of the machine is  $2^{-15} = 3.051757813 \times 10^{-5}$ . Now the lowest coefficient of the CDWT filter is determined by the exponential function of the extreme end value, i.e., at the location  $abs(\frac{N}{2} + n_0)$ , where  $n_0$  is  $-0.5$ . For a 16-bit machine the lowest exponent is given by:

$$e^{-0.5 \frac{(5-(-0.5))^2}{1.09^2}} = 1.990389683 \times 10^{-4} \quad (3)$$

Thus a 10-tap filter can be designed using a 16-bit machine.

For an 8-tap filter, the correlation between the real coefficients can be expressed as:

$$f_0 = f_7, \quad f_1 = f_6, \quad f_2 = f_5, \quad f_3 = f_4 \quad (4)$$

Thus the value  $(n - n_0)$  associated with the exponential and the circular part are  $-3.5, -2.5, -1.5, -0.5, 0.5, 1.5, 2.5, 3.5$ . The circular expression is periodic about  $2\pi$ . Again, expressing the angles within the range of 0 to  $\frac{\pi}{2}$ , the orientations  $\alpha$  for a modulation angle of  $\omega_0 = 30^\circ$  are  $15^\circ, 45^\circ, 75^\circ, -75^\circ, -45^\circ$  and  $-15^\circ$ . The multiplication of a complex number  $(x_i^r \quad x_i^i)$  is given by:

$$\begin{pmatrix} F_{re} \\ F_{im} \end{pmatrix} = A \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x_i^r \\ x_i^i \end{pmatrix} \quad (5)$$

where  $x_i^r$  and  $x_i^i$  represent the real and imaginary part of a complex number  $\mathbf{x}_i$ . Now for a 8-tap filter there will be 8 orientations. But since the total number of offered orientations are 6, so there will be strong correlation between the angles. In fact among the 6 orientation itself, three angles are just the folded version of the other

in the first and second quadrant (e.g.  $15^\circ$  and  $-15^\circ$ ). Thus realizing only three rotations, the others also can be realized. Taking the explicit value of the rotations it is seen that :

$$\phi_0 = \phi_1 = \phi_6 = \phi_7^T, \quad \phi_2 = \phi_5^T, \quad \phi_3 = \phi_4^T \quad (6)$$

where

$$\phi_n = \begin{pmatrix} \cos\alpha_n & \sin\alpha_n \\ -\sin\alpha_n & \cos\alpha_n \end{pmatrix} \quad \text{and} \quad \phi_n^T = \begin{pmatrix} \cos\alpha_n & -\sin\alpha_n \\ \sin\alpha_n & \cos\alpha_n \end{pmatrix} \quad (7)$$

and the superscript  $T$  signifies transpose of a matrix. Using the transform relation for CDWT, the parametric equation can be written as:

$$\begin{pmatrix} C_{L,i}^r \\ C_{L,i}^i \end{pmatrix} = -f_0 \begin{pmatrix} x_n^r \\ x_n^i \end{pmatrix} \phi_0 - f_0 \begin{pmatrix} x_{n-7}^r \\ x_{n-7}^i \end{pmatrix} \phi_0^T + \\ f_1 \begin{pmatrix} x_{n-1}^r \\ x_{n-1}^i \end{pmatrix} \phi_0 + f_1 \begin{pmatrix} x_{n-6}^r \\ x_{n-6}^i \end{pmatrix} \phi_0^T + \\ f_2 \begin{pmatrix} x_{n-2}^r \\ x_{n-2}^i \end{pmatrix} \phi_2 + f_2 \begin{pmatrix} x_{n-5}^r \\ x_{n-5}^i \end{pmatrix} \phi_2^T + \\ f_3 \begin{pmatrix} x_{n-3}^r \\ x_{n-3}^i \end{pmatrix} \phi_3 + f_3 \begin{pmatrix} x_{n-4}^r \\ x_{n-4}^i \end{pmatrix} \phi_3^T \quad (8)$$

where

$$\begin{aligned} \phi_0 &= \begin{pmatrix} \cos 75^\circ & \sin 75^\circ \\ -\sin 75^\circ & \cos 75^\circ \end{pmatrix} \\ \phi_2 &= \begin{pmatrix} \cos 45^\circ & \sin 45^\circ \\ -\sin 45^\circ & \cos 45^\circ \end{pmatrix} \\ \phi_3 &= \begin{pmatrix} \cos 15^\circ & \sin 15^\circ \\ -\sin 15^\circ & \cos 15^\circ \end{pmatrix} \end{aligned} \quad (9)$$

and

$$\begin{aligned} f_0 &= e^{-0.5 \frac{3.5^2}{1.09^2}}, \quad f_1 = e^{-0.5 \frac{2.5^2}{1.09^2}} \\ f_2 &= e^{-0.5 \frac{1.5^2}{1.09^2}}, \quad f_4 = e^{-0.5 \frac{0.5^2}{1.09^2}} \end{aligned} \quad (10)$$

and  $\mathbf{C}_{L,i} = \begin{pmatrix} C_{L,i}^r \\ C_{L,i}^i \end{pmatrix}$  represents the low-pass coefficient (denoted by the suffix  $L$ ) generated at the  $i^{th}$  instant. The next coefficient  $\mathbf{C}_{L,i+1}$  is given by:

$$\begin{pmatrix} C_{L,i+1}^r \\ C_{L,i+1}^i \end{pmatrix} = -f_0 \begin{pmatrix} x_{n+2}^r \\ x_{n+2}^i \end{pmatrix} \phi_0 - f_0 \begin{pmatrix} x_{n-5}^r \\ x_{n-5}^i \end{pmatrix} \phi_0^T + \\ f_1 \begin{pmatrix} x_{n+1}^r \\ x_{n+1}^i \end{pmatrix} \phi_0 + f_1 \begin{pmatrix} x_{n-4}^r \\ x_{n-4}^i \end{pmatrix} \phi_0^T + \\ f_2 \begin{pmatrix} x_n^r \\ x_n^i \end{pmatrix} \phi_2 + f_2 \begin{pmatrix} x_{n-3}^r \\ x_{n-3}^i \end{pmatrix} \phi_2^T + \\ f_3 \begin{pmatrix} x_{n-1}^r \\ x_{n-1}^i \end{pmatrix} \phi_3 + f_3 \begin{pmatrix} x_{n-2}^r \\ x_{n-2}^i \end{pmatrix} \phi_3^T \quad (11)$$

The association table between the real coefficients  $f_n$ , circular coefficient  $\phi_n$  and  $\mathbf{x}_n$  for the coefficient  $\mathbf{C}_{L,i}$  is given in Table (1). As is evident from the table the inputs  $\mathbf{x}_n$  and  $\mathbf{x}_{n-7}$  share the same set of real coefficient, namely  $-f_0$ . But these two inputs are not available to the processors in the same clock with the provided scanning pattern. Instead, the two consecutive values  $x_n$  and  $x_{n-1}$  are obtained in the same clock. Again they share the same circular rotation unit. At the same time the input  $x_{n-2}$  is multiplied with

	$\phi_0$	$\phi_2$	$\phi_3$	$\phi_3^T$	$\phi_2^T$	$\phi_0^T$
$f_0$	$\mathbf{x}_n$					
$f_1$	$\mathbf{x}_{n-1}$					
$f_2$		$\mathbf{x}_{n-2}$				
$f_3$			$\mathbf{x}_{n-3}$	$\mathbf{x}_{n-5}$	$\mathbf{x}_{n-6}$	$\mathbf{x}_{n-7}$

Table 1.  $\phi$ ,  $f_n$  and  $\mathbf{x}_n$  association table for low pass filter

$f_2$  and rotated by  $\alpha_2 = 45^\circ$ , while the input  $x_{n-3}$  is scaled by  $f_3$  and rotated by  $\alpha_3 = 15^\circ$ . But for the other four inputs,  $x_{n-4}$ ,  $x_{n-5}$ ,  $x_{n-6}$  and  $x_{n-7}$ , required for generation of the coefficient  $\mathbf{C}$ , the matrix  $\phi_n$ s need to be transposed. This is accomplished by using the matrix multiplication property. For two matrices  $A$  and  $B$ ,

$$AB^T = (A^T B)^T \quad (12)$$

So instead of transposing the  $\phi_n$  processor, the input matrix  $\mathbf{x}_n$  is transposed and multiplied with  $\phi_n$ . Then, the product  $\mathbf{x}_n^T \phi_n$  is transposed to get back the desired result. Transposition of the inputs  $\mathbf{x}_n$  can be achieved easily just by interchanging the  $x_n^r$  and  $x_n^i$ . This technique has been used for all the filter design for CDWT.

For the high pass filter (with  $\omega_1 = \frac{5\pi}{6}$ ) the parametric equation for the coefficient generated at the  $i^{th}$  instant can be formulated in the similar manner as that of low pass filter. The association table of  $\mathbf{x}_n$ ,  $f_n$  and  $\phi_n$  for the high pass filter is shown in Table (2) Similarly the association table for conjugate low and high pass filters are constructed.

	$\phi_0$	$\phi_2$	$\phi_3$	$\phi_3^T$	$\phi_2^T$	$\phi_0^T$
$f_0$						
$f_1$						
$f_2$			$-\mathbf{x}_{n-5}$			
$f_3$	$\mathbf{x}_{n-4}$				$-\mathbf{x}_{n-2}$	$\mathbf{x}_{n-3}$

Table 2.  $\phi$ ,  $f_n$  and  $\mathbf{x}_n$  association table for high pass filter

#### 4. HYPERBOLIC COEFFICIENT

The most important thing to be noted here is that, for a given filter length, the exponential parts for all the filters (low, high and the two conjugate filters) have same sample distribution (vide Table (1) and Table (2)). Thus, for a given output coefficient for the CDWT filter, the  $f_n$  to  $x_n$  correspondence is identical.

Since these are constant multiplications so a CSD based implementation of dedicated fixed multipliers is the most preferred approach, in terms of hardware cost and speed of operation. The percentage error in CSD representation of the four low pass coefficients are 0.0171, 0.3923, 0.3174 and 0.1891 respectively.

From the parametric equations of the filter, i.e., Eqn. (8) and the association Table (1) and Table (2), it is seen that the multiplication is common for all the high-pass, low-pass and both the conjugate filters. In generation of  $C_{L,i}$ ,  $C_{H,i}$ ,  $C_{CL,i}$  and  $C_{CH,i}$  the  $x_n$  and  $x_{n-7}$  will be multiplied with  $f_0$  for all the cases. Similarly  $x_{n-1}$  and  $x_{n-6}$  are multiplied with  $f_1$ . So, using this similarity the products can be generated for all the filters.

#### 5. FILTER DESIGN

Multiplication of a complex number with the filter coefficient involves two real multiplications. In the subsequent part of this paper, the CSD multipliers for accomplishing this operation are termed Complex CSD Multiplier (CCSDM). Figure (2) shows the architecture for the low-pass filter. The basic module consists of 4 set of CCSDM for generating the products of the real exponential part with the complex (or real for first level decomposition) inputs. In the first clock the products  $f_0 \mathbf{x}_n$ ,  $f_1 \mathbf{x}_{n-1}$ ,  $f_2 \mathbf{x}_{n-2}$  and

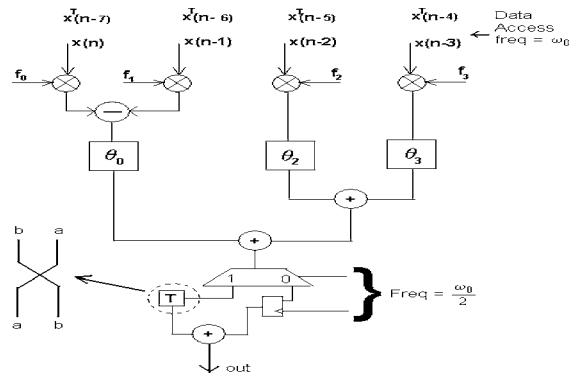


Fig. 2. Architecture for low pass filter of an 8-tap CDWT

$f_4 \mathbf{x}_{n-3}$  are computed and the first two products are subtracted to give  $-f_0 \mathbf{x}_n + f_1 \mathbf{x}_{n-1}$ . Since both the products  $-f_0 \mathbf{x}_n$  and  $f_1 \mathbf{x}_{n-1}$  undergo a rotation of  $\phi_0$  (vide Eqn. (8)), so these are added prior to rotation. This reduces the rotation overhead by one unit. The other two products  $f_2 \mathbf{x}_{n-2}$  and  $f_3 \mathbf{x}_{n-3}$  are rotated by  $\phi_2$  and  $\phi_3$  respectively and added. In the next stage partial sum of the parametric equation 8 is accomplished, as,

$$(-f_0 \mathbf{x}_n + f_1 \mathbf{x}_{n-1}) \phi_0 + f_2 \mathbf{x}_{n-2} \phi_2 + f_3 \mathbf{x}_{n-3} \phi_3$$

To generate the next part of the parametric equation,  $\phi_n$  needs to be transposed. As has been mentioned earlier, transposing the inputs instead of the rotation units and again transposing the rotated outputs give the desired result without any extra hardware overhead. Using this property it can be written:

$$\begin{aligned} & -f_0 \mathbf{x}_{n-7} \phi_0^T + f_1 \mathbf{x}_{n-6} \phi_0^T + f_2 \mathbf{x}_{n-2} \phi_2^T + f_3 \mathbf{x}_{n-3} \phi_3^T \quad (13) \\ & = (-f_0 \mathbf{x}_{n-7}^T \phi_0 + f_1 \mathbf{x}_{n-6}^T \phi_0 + f_2 \mathbf{x}_{n-2}^T \phi_2 + f_3 \mathbf{x}_{n-3}^T \phi_3)^T \end{aligned}$$

So the term,  $(-f_0 \mathbf{x}_{n-7}^T \phi_0 + f_1 \mathbf{x}_{n-6}^T \phi_0 + f_2 \mathbf{x}_{n-2}^T \phi_2 + f_3 \mathbf{x}_{n-3}^T \phi_3)$ , needs to be transposed to get the LHS of Eqn. (13)

In the next clock, the two outputs - (a) from the register and (b) from the transpose block are added. Thus, this register operates at half the frequency of the data accession block.

The circuit for high pass filter is shown in Fig. 3. The conjugate low pass and conjugate high pass filters are realized similarly. The rotation units are designed using CORDIC as the basic processing elements [8, 14].

#### 6. IMPLEMENTATION

The architecture has been implemented on Xilinx FPGA XV50FG256 package. However, this architecture is generic in

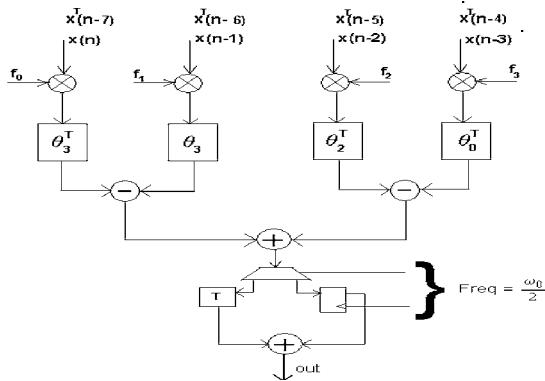


Fig. 3. Architecture for high pass filter of an 8-tap CDWT

terms of the platform of use. For 8-bit data a 16-bit machine has been used to prevent the overflow and underflow.

The resource required for the design of each of the low-pass and high pass filters in FPGA are as follows:

Number of Slices:	549 out of 768	71%
Slice Flip Flops:	514	
4 input LUTs:	872 (4 used as a route-thru)	
Number of bonded IOBs:	128 out of 176	72%
Number of GCLKs:	1 out of 4	25%
Number of GCLKIOBs:	1 out of 4	25%
Total equivalent gate count for design:	15,137	
Additional JTAG gate count for IOBs:	6,192	

For resource optimization in FPGA the design has been optimized using the relative locking constraints of the components. The register and the demultiplexer at the output stages operate at half the frequency of the global clock. This constraint is required since the addition at the last stage occurs after every two clocks (refer to Fig. (2), Fig. (3)). The operating frequency of the implemented design is  $\approx 41$  MHz.

Table (3) shows comparison between the two architectures proposed in the dissertation

Architecture	No. of proc.	No. of mult.	No. of CORDIC	Lat.	Throughput
LPF [8]	N	NIL	N	$\log_2(N)$	$O(1)$
HPF (Proposed)	$O(\frac{N}{2})$	$\frac{N}{2}$	$\frac{N}{2}-1$	$2\log_2(N)$	$O(2)$

Table 3. Comparison between hyperbolic CORDIC and CSD multiplier based architectures

It is evident from the comparison that the CORDIC based architecture is suitable for applications which require faster processing. On the other hand the proposed architecture requires much lower processor area (order of half) compared to the first architecture. The low hardware complexity and simple data scheduling makes this architecture a better candidate for low power applications.

The critical path of the proposed architecture is lower leading to high operating frequency. ASIC implementation of this design will lead to better resource optimization and timing performance compared to its FPGA counterpart.

## 7. REFERENCES

- [1] H.Pan, "General stereo image matching using symmetric complex wavelets," in *Proc SPIE: Wavelet Applications in Signal and Image Processing, VI*, Denver, August 1996, vol. 2825.
- [2] J.Magarey and A.Dick, "Multiresolution stereo image matching using complex wavelets," in *Proceedings of the 14<sup>th</sup> International Conference on Pattern Recognition*, 1998, vol. 1, pp. 4-7.
- [3] J.Magarey and N.G.Dick, "An improved motion estimation algorithm using a complex-valued wavelet transform," in *Proc. IEEE International Conference on Image Processing*, September 1996, pp. 969-972.
- [4] J. Magarey and N.G.Kingsbury, "Motion estimation using a complex-valued wavelet transform," *IEEE Trans. on Signal Processing (special issue on Wavelets and Filter Banks)*, vol. 46 (4), pp. 1069-1084, April 1998.
- [5] K.K. Parhi and T. Nishitani, "VLSI architectures for discrete wavelet transforms," *IEEE Trans. on VLSI Systems*, vol. 1(2), pp. 191-202, June 1993.
- [6] M.H.Lee J.S.Chang and J.Y.Park, "A High Speed VLSI Architecture for Discrete Wavelet Transform for MPEG-4," in *Proc. IEEE Int. Symposium on Circuits and Systems*, USA, 1997, vol. ISCAS97, V, pp. 623-627.
- [7] P-C. Wu and L-G. Chen, "An Efficient Architecture for Two-Dimensional Discrete Wavelet Transform," *IEEE Trans. on Circuits, Syst. for Video Tech.*, vol. 11 (4), pp. 536-545, April 2001.
- [8] B. Das and Swapna Banerjee, "A CORDIC Based Array Architecture for Complex Discrete Wavelet Transform," in *Proc. ACM Int. Great Lakes Symp. on VLSI*, Purdue, IN, USA, Mar. 22 - 23 2001, pp. 79-84.
- [9] Y.-N. Chang and K.K. Parhi, "High-performance digit-serial complex-number multiplier-accumulator," in *Proc. IEEE International Conference on Computer Design*, Austin, Texas, 1998.
- [10] George W. Reitwiesner, "Binary arithmetic," *Advances in Computers*, vol. 1, pp. 231-308, 1960.
- [11] Y.H.Hu, "CORDIC-based VLSI architecture for digital signal processing," *IEEE Signal Processing Magazine*, pp. 16-35, July 1992.
- [12] A.S. Dhar, *CORDIC based array architectures for electro-medical signal processing*, Ph.D. thesis, Indian Institute of Technology, December 1994.
- [13] K. Maharatna, *CORDIC based signal processors for biomedical application*, Ph.D. thesis, Jadavpur University, Calcutta, India, 2002.
- [14] Bipul Das and Swapna Banerjee, "A Unified CORDIC-based Chip to Realize DFT/DHT/DCT/DST," *IEE Proceedings - Computers and Digital Techniques*, vol. 149(4), pp. 121-127, July 2002.