Multiplier-less and Modified SQRT CSA based Discrete Wavelet Transform

Sharad Kumar Mehra, Jitendra Jain
B.I.S.T Bhopal

Abstract - —In this paper, we present a new efficient distributed arithmetic (NEDA) formulation of the computation of 1-D discrete wavelet transform (DWT) using 9/7 filters, and mapped that into bit parallel for high-speed and low hardware implementations, respectively. We demonstrate that NEDA is a very efficient architecture with adders as the main component and free of ROM, multiplication, and subtraction. The bit-parallel structure has 100% hardware utilization efficiency. Compared with the existing multiplier-less structures, the proposed structures offer significantly higher throughput rate and involve less area-delay product.

Keywords: -Discrete Wavelet Transform (DWT), NEDA, Xilinx Simulation.

I. INTRODUCTION

The discrete wavelet transform (DWT) has been widely used in many areas of science and engineering, e.g., signal and image processing, bio-informatics, geophysics, and meteorology etc. for the applications involving compression and analysis of various forms of data. The well-known image coding standards, namely, MPEG-4 and JPEG2000 have adopted DWT as the transform coder due to its remarkable advantages over the other transforms. For lossy compression, Daubechies 9/7 orthogonal filter is used as the default wavelet filter in JPEG 2000. Efficient implementation of DWT using 9/7 filters in resource-constrained hand-held devices with capability for real-time processing of the computation- intensive multimedia applications is, therefore, a necessary challenge. Multiplier-less hardware implementation approach provides a kind of solution to this problem due to its scope for lower hardware-complexity and higher throughput of computation.

Several designs have been proposed for the multiplier-less implementation of DWT based on the principle of distributed arithmetic (DA) [1]–[3]. The structure of distributes the bits of the fixed coefficients instead of the bits of input samples. Consequently, the adder-complexity of the structure of depends on the DA-matrix of the fixed coefficients [2]. Martina et al [4] have approximated the 9/7 filter coefficients and expressed the 9/7 filter outputs in terms of 5/3 filter outputs. By that approach, they have significantly reduced the adder-complexity of the 9/7 DWT. Gourav et al [5] have suggested an LUT-less DA-based design for the implementation of 1-D DWT. They have eliminated the ROM.

II. MATHEMATICAL DERIVATION OF NEDA

Let us consider the following sum of products [6]:

$$D = \sum_{k=1}^{L} B_k \times C_k \tag{1}$$

Where B_k are fixed coefficients and they C_k are the input data words. Equation (1) can also be written in the form of a matrix product as:

$$D = \begin{bmatrix} B_1 & B_2 & \dots & B_L \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ . \\ C_L \end{bmatrix} \tag{2}$$

Both B_k and C_k are in two's complement format. The two's complement representation of B_k may be expressed as

$$B_k = -B_k^M 2^M + \sum_{i=N}^{M-1} B_k^i 2^i \tag{3}$$

Where $B_k^i = 0$ or 1, and $i =$ N, N+1… M and B_k^M is the sign bit and B_k^N is the least significant bit (LSB).

Equation (3) can be expressed in matrix form as:

$$B_k = \begin{bmatrix} 2^N & 2^{N+1} & \dots & 2^M \end{bmatrix} \begin{bmatrix} B_k^N \\ B_k^{N+1} \\ . \\ -B_k^M \end{bmatrix} \tag{4}$$

Similarly C_k can be represented in two's complemented format as:

$$C_k = -C_k^X 2^X + \sum_{i=W}^{X-1} C_k^i 2^i \qquad (5)$$

Where $C_k^i = 0$ or 1, and $i =$ W, W+1, …,X and C_k^M is the sign bit and C_k^N is the least significant bit (LSB).

Now on combining equations (1) and (3), we get-

$$D = -(D^M .2^M) + \sum_{i=N}^{M-1} (D^i .2^i) \qquad (6)$$Where

$$D^i = \sum_{k=1}^{L} B_k^i C_k \ , \ i = \text{N, N+1… M}$$

III. PROPOSED ARCHITECTURE

In this paper, we have proposed a multiplier-less architecture for 9/7 wavelet Filter by using NEDA. The filter coefficients of 9/7 wavelet filter are given in table1. We multiply the filter coefficients by 100 for simplification. The mathematical calculation for high pass output is explained by an example.

Table 1**:** Filter Coefficients Of 9/7 Wavelet Filter.

	Coefficients	**Multiplied by 100**	**6 bit binary representation with 2's complement of negative no.**
h(0)	0.60294901823636	60	111100
h(1)	0.26686441184287	26	011010
h(2)	-0.07822326652899	-7	001001
h(3)	-0.01686411844287	-1	000011
h(4)	0.02674875741081	2	000010
g(0)	0.5575435262285	55	110111
g(1)	-0.29563588155713	-29	100011
g(2)	-0.02877176311425	-2	000110
g(3)	0.04563588155713	4	000100

Where h(0), h(1),… h(4) are the Low pass filter coefficients and g(0),g(1)…g(3) are the High pass filter coefficients.

If we take the high pass coefficients g(0),g(1),g(2) and g(3), and multiply by r(1),r(2),r(3) and r(4) then we get the High pass output Y_H of the 9/7 filter as [6]:

$$Y_H = \begin{bmatrix} g(0) & g(1) & g(2) & g(3) \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ r(4) \end{bmatrix}$$

(7)

Where r(1)=x(1)+x(n-6), r(2)=x(n-1)+x(n-5), r(3)=x(n-2)+x(n-4),r(4)=x(n-3).

Let r(1)=1, r(2)=2,r(3)=3,r(4)=4 then

$$Y_H = \begin{bmatrix} 55 & -29 & -2 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = 7 \qquad (8)$$

Now if we implement this with NEDA then

$$Y_H = \begin{bmatrix} 55 & -29 & -2 & 4 \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ r(4) \end{bmatrix} \qquad (9)$$

$$Y_H = \begin{bmatrix} 110111 & 100011 & 000110 & 000100 \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ r(4) \end{bmatrix}$$

Now we can make the DA matrix by the filter coefficients as

$$\begin{bmatrix} B_k \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \qquad (10)$$

And thus

$$Y_H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ r(4) \end{bmatrix} = \begin{bmatrix} r(1)+r(2) \\ r(1)+r(2)+r(3) \\ r(1)+r(3)+r(4) \\ 0 \\ r(1) \\ r(1)+r(2) \end{bmatrix} \qquad (11)$$

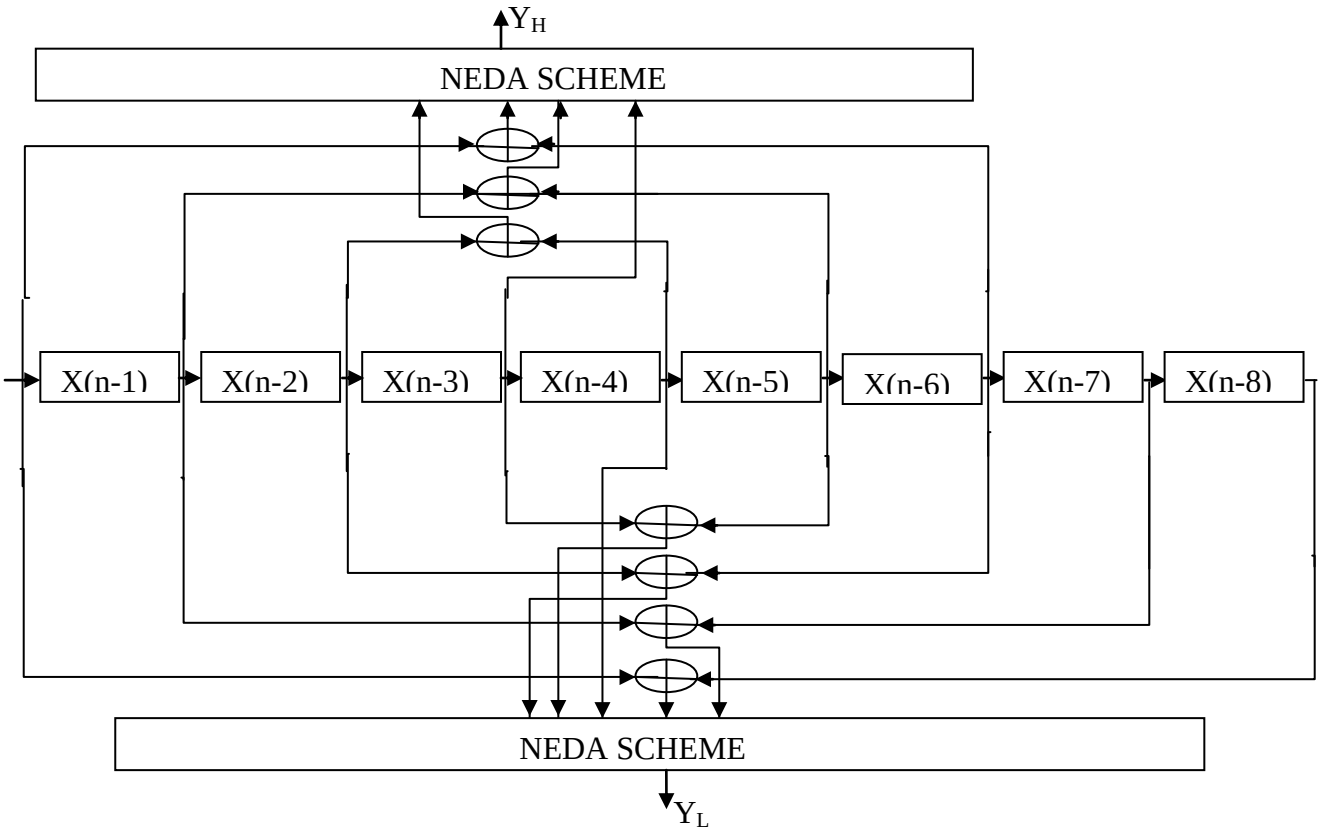

Figure 1: Proposed Multiplier-Less 9/7 Wavelet Filter Using NEDA Scheme

IV. SYSTEM MODEL

Following the NEDA architecture in Fig. 1, the configuration for computing Y_H is illustrated in Fig. 2 (a). As can be observed from this example, NEDA eliminates totally the encoder logic required in Booth Multiplier for two's complement manipulation.

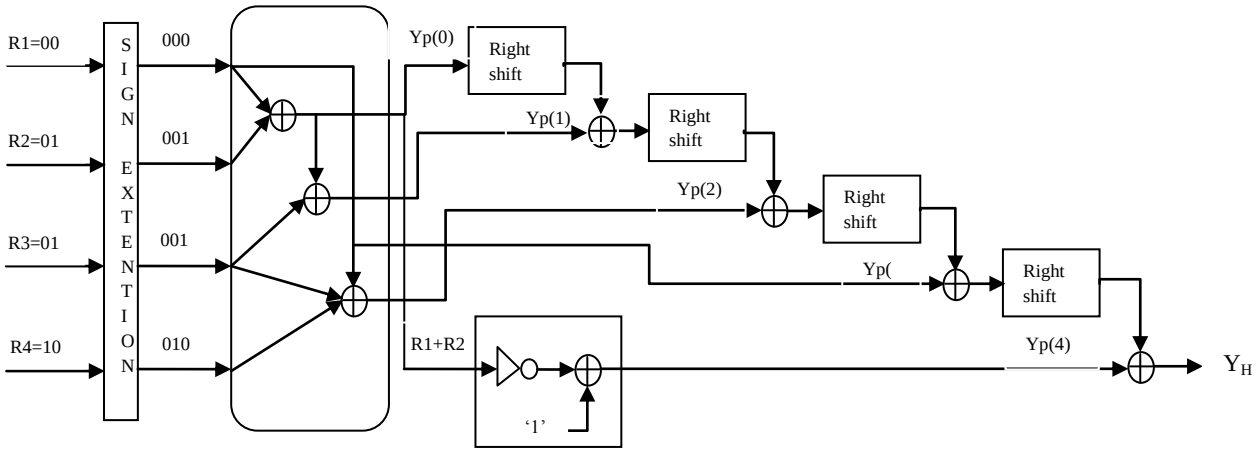

Figure 2 (a): Mathematical calculation of the NEDA Scheme of the High-pass Wavelet Filter Output

16-B SQRT CSA USING BEC:

The structure of the proposed 16-Bmodified CSA using BEC for RCA with $C_i = 1$ is shown in Figure 4 In the proposed architecture we have replaced RCA with a BEC for $C_i = 1$.The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure.

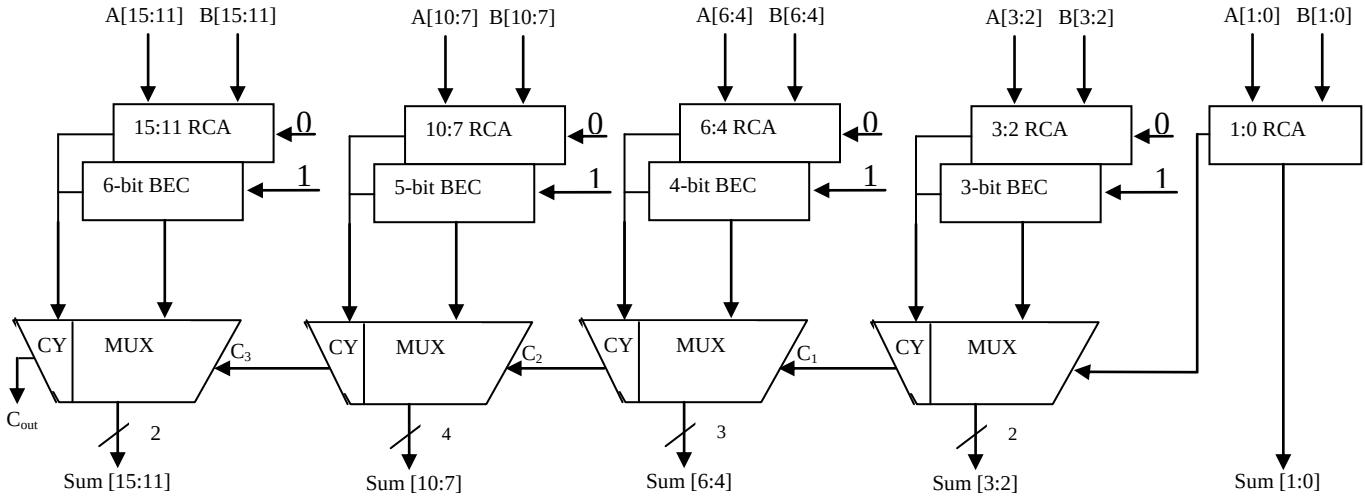

Figure 3: Block Diagram of Regular 16-bit SQRT CSLA using RCA with BEC

The CSA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. However, the CSA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input $C_i = 0$ and $C_i = 1$, then the final sum and carry are selected by the multiplexers (mux).

V. RESULT AND SIMULATION

In the proposed architecture, calculate the high-pass and low-pass wavelet filter output using NEDA scheme. NEDA does not require ROM.

Furthermore, only one type of operations-addition, take place during the intermediate stages of computation, greatly simplifying hardware design. What needs special care is the sign output from the adder array, which is simply taking two's complement. In the above example, invert-and-add-1 is all one needs to convert "0011" to "1101= Y_{P4}

Proposed structure consists only 29 adders, zero mux and 27 registers. In the proposed architecture is better than other architecture in shown the Table 2.

Implementation the Longa et al. [5] and proposed architecture has been captured by VHDL and the functionality is verified by RTL and gate level simulation.

Table 2:Comparison Of Proposed With Existing Architecture

Structure	Adder	MUX	REG	CP
Alam *et al.* [1]	43	0	9	$6T_A$
Cao *et al* [2]	27	0	9	$6T_A$
Martina *et al* [4]	19	8	9	$6T_A$
Longa *et al* [5]	35	40	9	$6T_A$
Proposed	29	0	27	$6T_A$

We functionally verified each unit presented in this paper including three method and 1_D and 2_D design. We have been found from the results shown in Table 2 that number of slices used is same in case of method 1 and method 2 which is less than slices used in method 3. So we designed 1_D DWT and 2_D DWT multiplier using SQRT Carry select adder (CSA) whose device utilization summary is given inTable.2.

Table 3: Comparisons result for 1_D and 2_D DWT design

Design	Method	No. of slices	No. of 4 input LUTs	MCPD (ns)
1_D DWT	Multiplier and ripple adder	187	317	38.977
	NEDA technique and ripple adder	149	249	33.359
	NEDA technique and SQRT CSA	139	243	31.573
	NEDA technique and Modified SQRT CSA	119	204	29.457
2_D DWT	Multiplier and ripple adder	783	1428	53.865
	NEDA technique and ripple adder	697	1218	51.837
	NEDA technique and SQRT CSA	639	1196	50.801
	NEDA technique and Modified SQRT CSA	589	1057	47.254

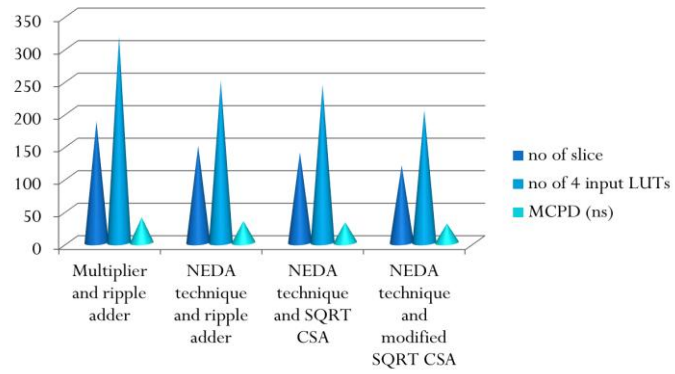

Figure-4: Bar graph of all the parameter in NEDA based 1-D DWT

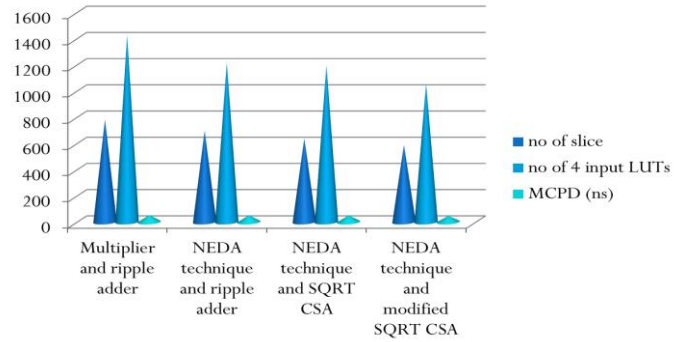

Figure-5: Bar graph of all the parameter in NEDA based 2-D DWT

VI. CONCLUSION

We propose a novel distributed arithmetic paradigm named NEDA for VLSI implementation of DSP algorithms involving inner product of vectors. Mathematical proof is given for the validity of the NEDA scheme. We demonstrate that NEDA is a very efficient architecture with adders as the main component and free of ROM, multiplication, and subtraction. For the adder array, a systematic approach is introduced to remove the potential redundancy so that minimum additions are necessary. NEDA is an accuracy preserving scheme and capable of maintaining a satisfactory performance even at low DA precision.

REFERENCES

[1] Ms. S.Manjui and Mr. V. Sornagopae, "An Efficient SQRT Architecture of Carry Select Adder Design by Common Boolean Logic", 978-1-4673-5301-4/13/$31.00 ©2013 IEEE

[2] B. Ramkumar and Harish M Kittur, "Low-Power and Area-Efficient Carry Select Adder", IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 20, NO. 2, FEBRUARY 2012.

[3] Gaurav Tewari, Santu Sardar and K. A. Babu, "High-Speed & Memory Efficient 2-D DWT on Xilinx Spartan3A DSP using scalable Polyphase Structure with DA for JPEG2000 Standard", 978-1-4244-8679-3/11/$26.00 ©2011 IEEE.

[4] Milad Ghantous and Magdy Bayoumi, "P2E-DWT: A Parallel and Pipelined Efficient VLSI Architecture of 2-D Discrete Wavelet Transform", 978-1-4244-9474-3/11/$26.00 ©2011 IEEE.

[5] M. Alam, C. A. Rahman, and G. Jullian, "Efficient distributed arithmetic based DWT architectures for multimedia applications," in *Proc. IEEE Workshop on SoC for real-time applications*, pp. 333 336, 2003.

[6] X. Cao, Q. Xie, C. Peng, Q. Wang and D. Yu, "An efficient VLSI implementation of distributed architecture for DWT," in *Proc. IEEE Workshop on Multimedia and Signal Process.*, pp. 364-367, 2006.

[7] M. Martina, and G. Masera, "Low-complexity, efficient 9/7 wavelet filters VLSI implementation," *IEEE Trans. on Circuits and Syst. II, Express Brief* vol. 53, no. 11, pp. 1289-1293, Nov. 2006.

[8] M. Martina, and G. Masera, "Multiplierless, folded 9/7-5/3 wavelet VLSI architecture," *IEEE Trans. on Circuits and syst. II, Express Brief* vol. 54, no. 9, pp. 770-774, Sep. 2007.

[9] Gaurav Tewari, Santu Sardar, K. A. Babu, " High-Speed & Memory Efficient 2-D DWT on Xilinx Spartan3A DSP using scalable Polyphase Structure with DA for JPEG2000 Standard," 978-1-4244-8679-3/11/$26.00 ©2011 IEEE.

[10] A. M. Shams, A. Chidanandan, W.Pan and M. A. Bayoumi, "NEDA: A Low-Power High-Performance DCT Architecture," IEEE Transactions on signal processing, vol. 54, no. 3, march 2006.

[11] Archana Chidanandan and Magdy Bayoumi, "AREA-EFFICIENT NEDA ARCHITECTURE FOR THE 1-D DCT/IDCT," ICASSP 2006.