Indian Journal of Science and Technology, Vol 8(21), DOI: 10.17485/ijst/2015/v8i21/79151, September 2015

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Low Complexity DWT Architecture Implementation for Feature Extraction using Different Multipliers

Tanay Mayankbhai Modi*, **Patel Hardik Anilkumar and John Sahaya Rani Alex**

School of Electronics Engineering Department, VIT University, Chennai Campus, Chennai - 600127,
Tamil Nadu, India; tanaymodi2007@gmail.com, patelhardik.anilkumar2013@vit.ac.in, jsranialex@vit.ac.in

Abstract

Background/Objectives: Discrete wavelet Transform (DWT) is substantially applied in many Digital Signal Processing (DSP) applications. Multiplication of the coefficient is the key factor for complexity in FIR filters. **Methods/Statistical Analysis:** This paper presents different multiplication techniques used to reduce the complexity in DWT such as Constant Shift Method (CSM), Vedic multiplication and Binary Signed Sub-coefficient (BSS) method. CSM technique uses Shift and Add unit followed with Mux to select appropriate output. BSS technique uses signed sub-coefficients, which reduces the multiplexer size. Vedic multiplication is famous for its low complexity architecture of bit-bit multiplication. Partial product and BCSE methods are used for designing the architectures. The coefficients are represented in IEEE 754 floating point half precision standard. Software simulation is performed in ModelSim and the designs are synthesized in Cadence RC tool for 180 nm technology. **Findings:** The CSM method gives high speed operation because of its reduced number of adders. The Vedic method provides low power and high speed operation. The BSS method uses Multiplexers and signed bit as control switch, which has a great impact over area requirement. The area requirement is reduced by 6% in BSS technique. Vedic multiplier gives about 23% power reduction compared to conventional multiplier. Both CSM and BSS techniques can be called Reconfigurable architectures as they can be hardwired and the mux output depends on the coefficients only. **Conclusion/Improvements:** CSM technique is gives the perfect balance for power reduction and area efficiency. Vedic technique gives immense reduction in power consumption but the area overhead is increased.

Keywords: Binary Signed Sub-coefficients, Constant Shift Method, Daubechies (db4) DWT, Discrete Wavelet Transform (DWT), Vedic Multiplication

1. Introduction

Recently, a great amount of work is proposed to provide a decent speech recognition system. It should give continuous speech recognition despite the noisy ambience. Gaining statistical information about the signal is called feature extraction. A resemblance to Heisenberg's Uncertainty hypothesis, time domain signals have a wide frequency band and signals with narrow frequency band have a large time period in time domain. Fourier transform is not sufficient for speech, which is non-stationary signal.

Several systems have been suggested to improve speech recognition. Linear Predictive Coding (LPC) and Mel-Frequency Cepstrum Coefficients (MFCC) are some of the orthodox techniques for feature extraction of speech signal. Linear Predictive Cepstral Coefficients (LPCC) and their regression coefficients were designed from them. MFCC is based on filter bank approach. In this method, filters are spread out in a shape same as human ears. These methods have high noise sensitivity. Yet, humans can isolate the speech in composite acoustic atmosphere. This fact has encouraged scientists to develop more robust speech feature extraction architecture.

Wavelets are used to reduce the influence of noise intrusion in feature extraction using DWT as shown in Figure 1. Mallat[1] proposed the DWT algorithm to extract the data from the signal. Filters are one of the most broadly used signal processing techniques. Wavelets can be recognized by duplication of filters with rescaling. It performs a multi-resolution signal analysis, in which locality is adjustable in both time and frequency domains. Correlation between local maxima of high frequency components are

*Author for correspondence

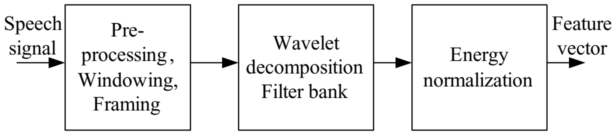
Speech signal
Pre-processing, Windowing, Framing
Wavelet decomposition Filter bank
Energy normalization
Feature vector

Figure 1. DWT based feature extraction block diagram.

extracted from multi-scale wavelet decomposition of the signal. DWT can be modelled by diverse choice of wavelet families[2] and basis functions. Haar wavelet is the modest among all DWT wavelet families. Daubechies (db4) wavelet is so far the most popular DWT family. The selection of DWT family is up to the application. As application varies, the mother wavelet also changes. Conventional DWT includes decomposition of Finite Impulse Response (FIR) filters. Because of the tedious calculations required in the DWT technique, many studies are proposed to project an innovative swift and low-power DWT[3–5].

The crucial roadblock in any DSP implementation is the coefficient multiplication. In traditional way, the technique used for multiplication is add/shift method, which is the most power consuming and tedious work. In non-reconfigurable systems, the coefficients are constant, due to which a series of adders and shifters are required. To reduce the complexity, some techniques are proposed such as Canonical Signed Digit[6] (CSD) and Binary Common Sub-expression Elimination[7] (BCSE).

Other approaches are focused on the implementation of FIR filter used in DWT by partitioning the filter coefficients into fixed groups (sub-coefficients) and compute the products of input depending on the coefficients. Multiplexers are used to select the proper output from the coefficient multiplication.

Vedic Mathematics[8] is a primitive system for a fast and effective way of computations. It contains 16 different equations named as "Sutras". The use of Vedic mathematics is to do faster calculations such as multiplication, division, squaring, square root, cubing and cube root. It can also be used for applications which need linear and non-linear differential equations, trigonometry, matrices, exponential and logarithm equations.

In the next section, the architecture of DWT is discussed. The BCSE[9–12] method is analyzed in Section 3. The Constant Shift Method (CSM) is proposed in Section 4. Section 5 represents Vedic multiplication. Binary Signed Sub-coefficient (BSS) method is discussed in Section 6. The results are compared in Section 7. Conclusion is given in the final section.

2. Discrete Wavelet Transform (DWT)

The initial proposition about Discrete Wavelet Transform (DWT) was given by Mallat[1]. The conventional DWT is also called Mallat-tree decomposition or Mallat algorithm. It consists low-pass and high-pass filter bank of the discrete time-domain signal. As shown in Figure 2, the sequence x[n] denotes input signal, where n is an integer. G0 and H0 represents low pass filter and high pass filter respectively. Each decomposition level gives two output signals such as coarse approximations a[n] and detail information d[n]. The low pass filter produces coarse approximations a[n] is produced by the low pass filter. The detail information d[n] of the signal is given by the high pass filter. The filter banks applied in the Forward Wavelet Transform is named Analysis filters, while the Inverse Wavelet Transform uses the filter banks as Synthesis filters.

There are numerous ways to construct a filter bank[4] such as Direct Form, Lattice structure, Lifting Scheme and Poly-phase. A filter bank fundamentally consists of a set of low pass filter and high pass filter, along with decimators or expanders.

2.1 Direct Form Structure

The Direct Form structure of DWT for analysis filter has low pass and high pass filter. Decimators are given at the end of the filter outputs. The synthesis filter has reverse steps as up-samplers at first and then followed by the low pass and high pass filters. Figure 3 describes both analysis and synthesis filters.

2.2 Daubechies (db4) DWT

Daubechies wavelet filters were first introduced by Ingrid Daubechies. They are also called Maxflat filters because of their flat frequency response at 0 and π. Daubechies wavelets are proposed by set of filters. The process contains convolution of input signal and filter coefficients

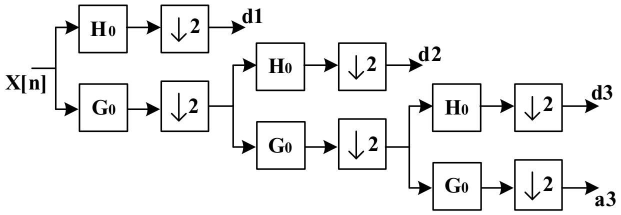

Figure 2. 3-level decomposition forward DWT.

and input. This filter is broadly famous for the orthogonal properties. They provides perfect reconstruction of the input signal at the end of synthesis filter bank. The coefficients are shown in Table 1.

3. Study of BCSE Method

To review the BCSE method in FIR filter, the partial product method is essential. N-tap FIR filter is taken which is described by Equation (1).

$$y[n]=\sum_{k=0}^{N-1} x[n]*h[k-n] \qquad (1)$$

Where h[k] is the kth coefficient. The partial product method is represented by splitting the coefficient to sub-coefficients and multiplying the input by sub-coefficients. The result is determined by shift and add operations.

If the coefficients are considered w-bits, which further are split into k sub-coefficients with m bit word-length, where m=w/k. Then h[i] coefficient is written by Equation (2).

$$h[i]=d_{i,0}+d_{i,1}*2^{m}+d_{i,2}*2^{2m}+\ldots+d_{i,k-1}*2^{(k-1)m}$$

$$h[i]=\sum_{j=0}^{k-1} d_{i,k}*2^{mj} \qquad (2)$$

Where di,j is the jth sub-coefficient of the ith coefficient. Using representation for filter coefficients, the FIR filter transform function can be written as given in Equation (3) and (4).

$$y[n]=\sum_{i=0}^{N-1}\sum_{j=0}^{k-1} d_{i,j}*2^{mj}*x[n-1] \qquad (3)$$

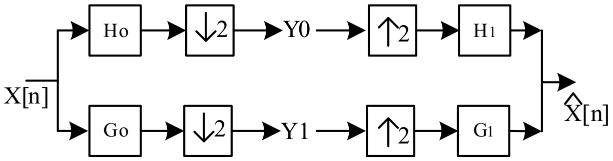

Figure 3. Analysis and synthesis filter bank of DWT.

Table 1. Filter coefficients for Daubechies (DB4) WT

1. Tap	2. Low Pass Filter	3. High Pass Filter
0	0.4829629	0.1294095
1	0.8365163	0.2241439
2	0.2241439	-0.8365163
3	-0.1294095	0.4829629

$$y[n]=\sum_{i=0}^{N-1}\sum_{j=0}^{k-1}\left(d_{i,j}*x[n-1]\right)*2^{mj} \qquad (4)$$

As given in the equation, y[n] can be represented by sum of partial products of input data. The operations can be easily executed by shift and add method because of the reduced bits of the coefficient. In includes set of pre-computed partial products. The block is named as Process Element (PE).

The redundant binary common sub-expressions (BCS) are focused in BCSE method. An n-bit binary coefficients can form $2^n-(n+1)$ BCSs. For example, 3-bit sub-coefficient have [011], [101], [110] and [111] BCSs. If x is an input, then the BCSs can be expressed as: [101] = $x + 2^{-2}x$, [110] = $x + 2^{-1}x$, [011] = $2^{-1}x + 2^{-2}x$, [111] = $x + 2^{-1}x + 2^{-2}x$. The other BCSs like [001], [010] and [100] don't require any adder or shifter. The direct realization can infer 5 adders but the reuse of output can reduce the adders to 3.for example, [011] can be written as [011] = $2^{-1}x + 2^{-2}x = 2^{-1}(x + 2^{-1}x)$. The shift and add architecture is shown in Figure 4.

The total adders required for n-bit coefficient is 2^{n-1}-1. In SDR receivers, the coefficients are needed to be changed according to the frequency which requires reconfigurability.

4. Constant Shift Method (CSM)

The Constant Shift Method (CSM) deals in taking the output from shift and add block and multiplexing it to get the correct output depending on the coefficients. In this method, the coefficients are already stored in LUT. They are partitioned into groups of sub-coefficients and used as the control signals for the multiplexers. If the coefficients are grouped in 3-bits, then the n-bit coefficient is divided into [n/3] groups. The numbers of multiplexers required are the absolute value of [n/3] or the next integer. A shown in Figure 5, the method is explained with 11 bits.

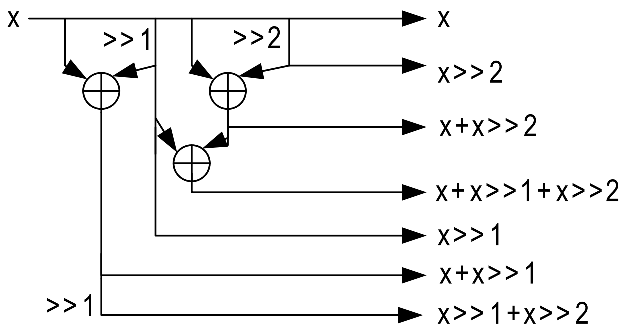

Figure 4. Shift and add unit.

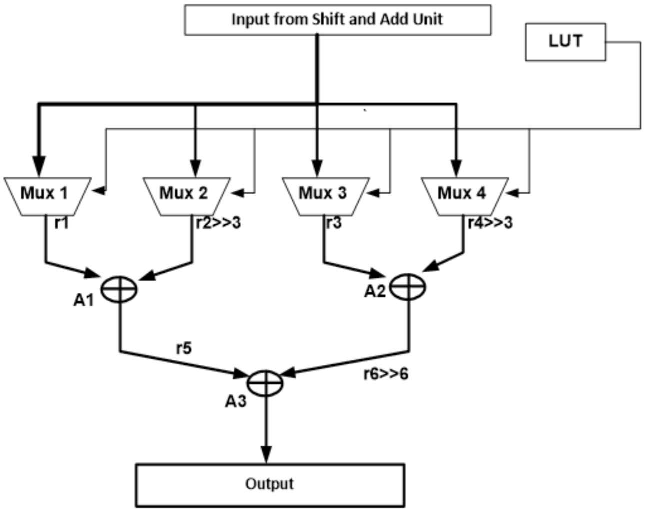

Figure 5. CSM architecture for 11-bit multiplication.

The coefficient h=1.1111111111. This is the worst-case coefficient that can occur. It needs the most number of adders and shifters due to no nonzero bits. So the output y can be written as shown in Equation (5).

$$y = x + 2^{-1}x + 2^{-2}x + \ldots + 2^{-9}x + 2^{-10}x \tag{5}$$

By partitioning the coefficient into group of 3-bit,

$$y = \left(x + 2^{-1}x + 2^{-2}x\right) + 2^{-3} * \left(x + 2^{-1}x + 2^{-2}x\right) + 2^{-6} * \left(x + 2^{-1}x + 2^{-2}x\right) + 2^{-9}\left(x + 2^{-1}x\right) \tag{6}$$

$$y = \left(x + 2^{-1}x + 2^{-2}x\right) + 2^{-3} * (x + 2^{-1}x + 2^{-2} x + 2^{-3} * (x + 2^{-1}x + 2^{-2}x + 2^{-3}(x + 2^{-1}x))) \tag{7}$$

As we can see that $x+2^{-1}x$ and $x+2^{-1}x+2^{-2}x$ can be obtained by the shift and add unit. The equation can be computed by three 8:1 multiplexers, one 4:1 multiplexer and some shifter and adders. The final shifter will do 2^{-3}, 2^{-6} and 2^{-9} shift. By partitioning,

$$y = r1 + 2^{-3}r2 + 2^{-6}r3 + 2^{-9}r4 \tag{8}$$

$$y = r1 + 2^{-3}r2 + 2^{-6}\left(r3 + 2^{-3}r4\right) \tag{9}$$

$$y = r5 + 2^{-6}r6 \tag{10}$$

Where r1, r2, r3 and r4 are the output of the multiplexers while r5 and r6 are taken as ($r1 + 2^{-3}r2$) and ($r3 + 2^{-3}r4$). As the shifts are constants and don't change irrespective of the coefficients, the architecture can be hardwired.

5. Vedic Multiplication

Vedic Mathematics includes a set of 16 Sutras. They are derived from the ancient Vedas. The Sutras are used to do complex calculations by converting them into simple equations. They are used to solve trigonometry, arithmetic, logarithmic problems.

5.1 Urdhva Tiryagbhyam

Urdhva Tiryagbhyam is one of the 16 Sutras available in Vedic Mathematics. It uses Vertical and Cross-wise bit multiplication.

In this method, all the partial products of the input data are produced, followed by the addition. Generated carry is propagated to the next bit adder as shown in Figure 6. The example of the method is discussed below.

Steps to be followed are given below:

- Generate all the partial products by vertical and cross-wise multiplication.
- Add the partial products according to the bit positions and pass the generated carry for the next addition.
- Repeat the step ii for all bits.

All the partial products can be generated simultaneously. This process gives net delay as the delay of adders only. Due to the bit-multiplication method, the time complexity and power consumption is less compared to conventional shift and add method.

6. Binary Signed Sub-Coefficient Method

In the proposed technique, the coefficients are represented in signed form. So for the conventional partitioning,

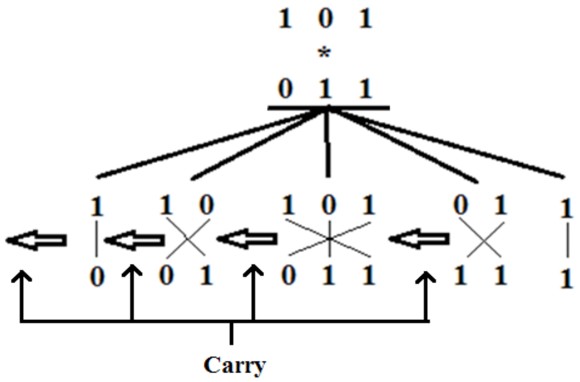

Figure 6. Vedic Multiplication.

the n-bit word can have values from 0 to 2^n-1. But in the BSS method, each sub-coefficient has a signed digit, due to which the range of n-bit word is -2^{n-1} to +2^{n-1}-1. This method gives relaxation in subtraction operation by reducing multiplexer blocks from 2^{n-1}:1. For it h 4-bit sub-coefficient word, if the value is greater than 8, then the MSB is removed and 1 is added in the (i-1)th sub-coefficient and so on. For example, if the coefficient is hi=12'h49A then 9 and A are greater than 8 (23), they are converted into BSS form.

$$h_i = 4*2^8 + 9*2^4 + A \tag{11}$$

$$h_i = 4*2^8 + \left(2^4 - 7\right)*2^4 + (1*2^4 - 6) \tag{12}$$

$$\mathrm{h_i} = 5*2^8 - 6*2^4 - 6 \tag{13}$$

So, the final coefficient will be h_i=12h56'6'.

The sign is saved in the MSB of all sub-coefficients. This technique reduces the word length from 4 to 3 and mux size for 15:1 to 8:1. As shown in Figure 7, coefficients are given to BSS decoder which computes sub-coefficients and add/sub control bits. The sign digit work as a control signal for the add/sub unit as switch between addition and subtraction.

7. Results and Discussion

In this section, results of power consumption and area requirement for all architectures are shown. To represent the coefficients, IEEE 754 floating point half precision standard is proposed. The coefficients have 16 bits, in which 10 LSB bits represent mantissa, the MSB is sign bit and 5 intermediate bits gives exponents.

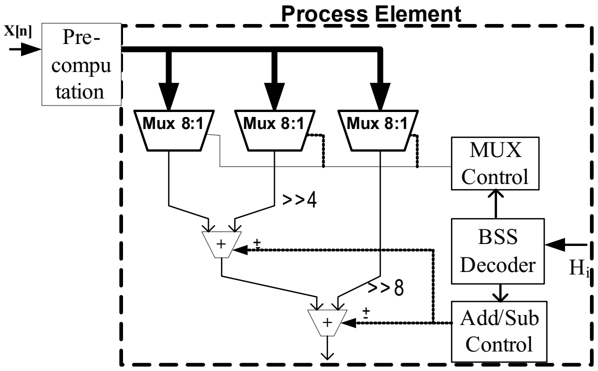

Figure 7. 4-bit BSS representation for 11-bit coefficient.

Power, timing and area constraints are compared for different multipliers. Simulation is done in ModelSim and synthesized in Cadence RC tool for 180 nm technology.

Figure 8 shows the results of power dissipation comparison between different multipliers. As shown in it, Vedic multiplier is the least power hungry module among all.

As shown in Figure 9, cell area required by Vedic multiplier is the most among all as it contains great amount of AND gates. The CSM technique has the least area requirement due to the constant block arrangement.

In Figure 10, power dissipation in DWT architecture with different multiplication modules is shown. The DWT with Vedic multiplier built in it gives tremendous reduction in power compared to the conventional DWT.

Figure 11 gives the insight about the area requirement in 3-level decomposition DWT with different multipliers

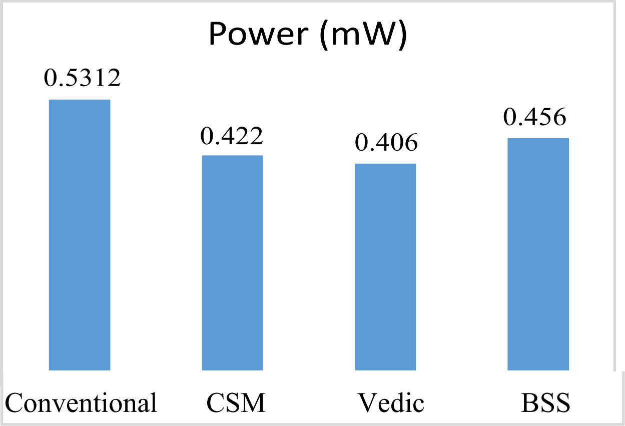

Figure 8. Power comparison between different multiplication techniques.

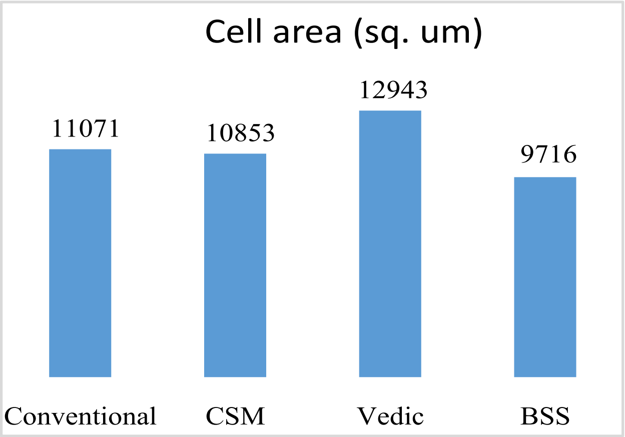

Figure 9. Cell area comparison between different multipliers.

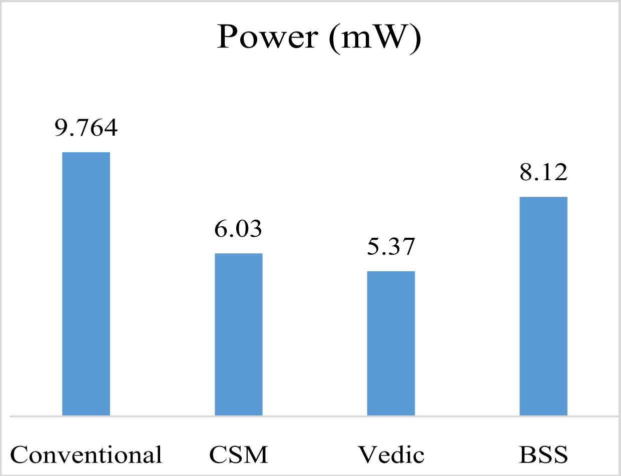

Figure 10. Comparison of power dissipation between 3-level decomposition DWT with different multipliers.

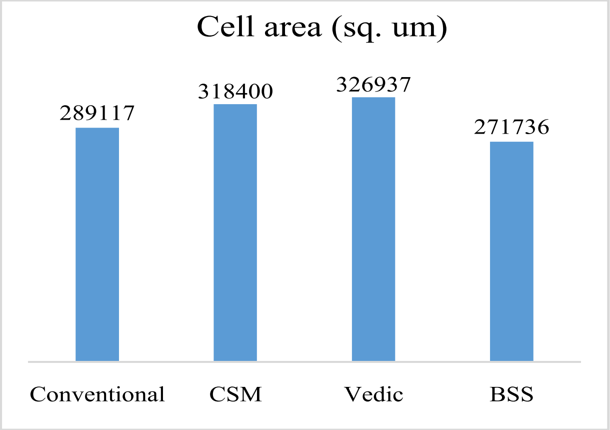

Figure 11. Cell area comparison between DWT with different multipliers.

in it. The BSS technique has the least cell area because of MUX arrangement instead of registers.

For future work, the energy normalization can be implemented to design full speech signal feature extraction.

8. Conclusion

An efficient DWT architecture is proposed for speech signal feature extraction. Different techniques are discussed for designing an efficient multiplier unit which is used to design DWT. Partial product and BCSE methods are used for designing the architectures. The CSM method gives high speed operation. The Vedic method provides low power and high speed operation. The BSS method uses MUXs which has a great impact over area requirement. The area requirement is reduced by 6% in BSS technique. Vedic multiplier gives about 23% power reduction compared to conventional multiplier. DWT architecture designed with Vedic multiplier produces about 45% power efficiency. As CSM technique is gives the perfect balance for power reduction and area efficiency, it gives the moderate design for overall performance. For low power approach, DWT designed with Vedic multiplier is advised to implement. For low-complexity approach, DWT designed with BSS technique should be used. For generalized structure, DWT with CSM technique is best among all of the architectures.

9. References

1. Mallat S. A theory for multi-resolution signal decomposition: The wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1989 Jul; 11(7): 674–93.
2. Weeks M, Bayoumi M. Discrete Wavelet Transform: Architectures, Design and Performance Issues. Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology. 2003 Sep; 35(2):155–78.
3. Chilo J, Lindblad T. Hardware implementation of 1D wavelet transform on an FPGA for infrasound signal classification. IEEE Transactions on Nuclear Science. 2008 Feb; 55(1):9–13.
4. Souani C. Optimized VLSI designs of wavelet transform architecture. International Conference on ICM Proceedings Microelectronics; Tunis, Tunisia. 2004 Dec 6-8. p. 558–63.
5. Motra AS, Bora PK, Chakrabarti I. An Efficient Hardware Implementation of DWT and IDWT. Conference on Convergent Technologies for Asia-Pacific Region; Bangalore, India. 2003 Oct. p. 95–9.
6. Abbaszadeh A, Sadeghipour KD. A New Hardware Efficient Reconfigurable FIR Filter Architecture Suitable For FPGA Applications. IEEE International Conference on Digital Signal Processing; Corfu. 2011 Jul 6-8. p. 1–4.
7. Mahesh R, Vinod AP. Reconfigurable Low Complexity FIR Filters for Software radio receivers. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications; Helsinki, Finland. 2006 Sep 11-14. p. 1–5.
8. Ramalatha M, Dayalan KD, Dharani P, Priya SD. High speed energy efficient ALU design using Vedic multiplication techniques. IEEE International Conference on Advances in Computational Tools for Engineering Applications; Zouk Mosbeh, Lebanon. 2009 Jul 15-17. p. 600–3.

9. Hartley RI. Subexpression sharing in filters using canonic signed digit multipliers. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. 1996 Oct; 43(10):677–88.
10. Pasko R, Schaumont P, Derudder V, Vernalde S, Durackova D. A new algorithm for elimination of common subexpressions. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems; 1999 Jan; 18(1):58–68.
11. Martinez-Peiro M, Boemo EI, Wanhammar L. Design of high-speed multiplier less filters using a nonrecursive. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. 2002 Mar; 49(3):196–203.
12. Mahesh R, Vinod AP. A new common Subexpression elimination algorithm for realizing low complexity higher order digital filters. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. 2008 Feb; 49(3):196–203.