Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MUTUAL MODULE BASED PARAMETERIZATION APPROACH FOR DWT AND FFT ALGORITHM

[1]J.VENKATESH, [2]R.RANI HEMAMALINI
[1]Research Scholar, St.Peter`s University, Avadi, Chennai -600054
[2]Prof & Head/ECE, St.Peter`s college of Engineering & Technology
Avadi, Chennai-600054
[1]E-mail: jvenkatesh0771@gmail.com

ABSTRACT

In Embedded and wireless communication systems, the use of DWT (Discrete Wavelet Transform) and FFT (Fast Fourier Transform) algorithms based modules has been significant. Of late, in a system if various module designs for distinct algorithms exist then the parameterization/reconfiguration techniques are becoming an effective implementation idea of research. And so, in this paper, parameterization technique is approached which takes advantage of the mutual modules present in the two distinct algorithms popularly used in the Embedded and wireless communication systems applications. Significance of this paper is to introduce a mutual module for the lifting DWT and FFT butterfly such that it contributes to reduction in area and power consumption. The advantage of this mutual module implementation in the parameterization approach is that the achieved power consumption is 253mW at 100MHz and the area required is 242 slices. Therefore, the power reduction for 8point FFT and 9/7DWT system under test is almost 40 % and for the same system under test the area reduced is 48%.

Keywords: *DWT-lifting, FFT-butterfly, Mutual Module, Parameterization.*

1. INTRODUCTION

Presently, embedded systems have widely spread over consumer, commercial, and military applications. Embedded systems have two fundamental characteristics viz; Reactive and Real-time. Embedded digital signal processing (EDSP) systems is one of the important applications of real-time embedded system, which uses a special architecture to achieve better performance. Moreover, embedded systems are widely used in standard applications such as wireless communication protocols demanding Orthogonal Frequency Division Multiplexing (OFDM) and radar image processing or say image processing. However, efficient hardware realization with reduced area and low-power dissipation is a significant challenge for embedded systems, in particular portable devices. The challenge is even more pronounced when DWT and FFT with large transform lengths need to be realized in embedded hardware for the application in wireless communication systems/ communication systems such as in OFDM, modulation, channel coding, equalization, thresholding /denoising etc. The common functions can be distinguished and then used to greatest advantage from the sharing of common attributes among the architectures in order to improve power efficiency and area. Hence, the goal intended to be attained is a hardware efficient mutual module for DWT and FFT architectures, stressing on compact and low-power embedded realizations.

Recently, [3], [9] parameterization technique has been introduced, which identifies the common prospects amidst the targeted standards with the objective of defining a capable operation to handle the required tasks. These operations when made a slight change to their parameters can switch from a configuration to another. With the fast and growing interest in VLSI technology, various techniques have been presented on FPGA platform to improve the performance of the embedded hardware. In this paper, a Parameterization technique is used to greatest advantage, called the Mutual module for Discrete Wavelet Transform (DWT) and Fast Fourier Transform (FFT) algorithms that can be considered to build a large range of communication standards. The fundamental concept behind the Mutual module approach is to distinguish the common elements in couple of architectures that could be to a great extent reprocessed across two or more distinct standards. Therefore, the Mutual module approach stay standard independent such that, when required the mutual module for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

particular function is implemented and executed. And with minimization in area on chip, the signal processing function will characterize the Mutual module.

The sharing of two different algorithms in wireless communication systems is with FFT and DWT algorithms considered. FFT algorithm for decoding or multicarrier modulation-OFDM along with DWT in wavelet thresholding or in multicarrier modulation-OFDM (MCM-OFDM) is used. In this paper, a Mutual Module based parameterization approach for DWT and FFT algorithm is developed . The two distinct algorithms sharing the implementation module under the parameterization context are FFT and DWT algorithms. Processed data and functions performed by the DWT and FFT algorithms are compared, but they seem to be totally unlike. Therefore, I had decided to research it under the Parameterization context, so that the computational and implementation similarities between the DWT-lifting and FFT-butterfly units can be identified. Thus, a configurable processing module (element) that can switch between the two algorithms functions is proposed. With reference to the existing research, the proposed work contributes to the design of the processing element/computational unit(MCU) for lifting based DWT and to develop an radix-2 butterfly structure for 2-point FFT system. Then the mutual module for DWT in FFT is recognized which minimizes the power consumption and area required compared to two different structure of DWT and FFT.

The paper is organized in the following way: In Section-2 the literature related to the presented work is discussed; in section-3 Problem statement is briefly reviewed. The lifting based DWT and Radix-2 FFT algorithm are reviewed in section-4. The proposed mutual module for the lifting DWT and Radix-2 FFT butterfly algorithm is discussed in section-5. Whereas, the results of the proposed implementation are presented in section-6 and in the paper is concluded in section 7.

2. LITERATURE

The study of Common Operator approach is introduced in the literature. In this section a review on recent development in Common Operator approach is presented.As, *Malek.Naoueset.al*[19]presenta common butterfly f or the FFT and Viterbi algorithms. They investigated where reuse and power consumption is traded against throughput. Performance comparisons with similar works are also discussed.*Ali Chamas. Al-Ghouwayel et.al* [20] presented a reconfigurable Triple mode Multiplier that constitutes the core of the Butterfly-based FFT. A scalable and flexible unit for the polynomial reduction needed in the Galois Fields-GF (2^m) multiplication is also proposed. An FPGA implementation of the proposed multiplier is given and the measures show a gain of 18 % in terms of performance-to-cost ratio compared to a ”Velcro” approach where two self-contained operators are implemented separately.*S.T. Gulet. al* [6] presented an method for designing flexible multi-standard radio systems. Their work consisted in exploring the design of multi-standard systems at different levels of granularity and selected the convenient level depending on each designer's needs. It consisted first in making a graph description of the multi-standard system to be designed. Then, an architectural exploration extracted the operators of a given multi-standard device. These operators were requested to have enhanced reconfiguration capabilities so that a fast reconfiguration was implementable. Their paper illustrated two scenarios of multi-standard radio system designs. Furthermore their approach presented the scheduling issues.

MalekNaoueset.al [3] presented a common structure for the FFT and Viterbi algorithms. A key benefit of exhibiting common operators was the regular architecture it brings when implemented in a Common Operator Bank (COB). This regularity made their architecture open to future function mapping and adapted to accommodate silicon technology variability through dependable design. They also discussed the Global complexity impact. *MalekNaoueset.al* [9] presented the Common Operator technique to present new common structures for the FFT and FEC decoding algorithms. A key benefit of exhibiting common operators was the regular architecture it brings when implemented in a Common Operator Bank (COB). This regularity made the architecture open to future function mapping and adapted to accommodate silicon technology variability through dependable design.

L. Alaus et.al [5] elaborated the Common Operator (CO) technique, which defined a multi standard terminal, based on a limited set of Common Operators. Their approach enhanced the reconfigurability and the scalability of the design but lead to a complex management of data dependencies and scheduling of each operator for its correct execution in the terminal. They presented a organization in bank (COB) not only to mitigate

Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the scheduling issue but also to maintain the flexibility and the optimization in their technique. The COB benefits from the property of the CO though limiting the main part of the scheduling. The COB created a scalable design, limited the scheduling and reduced the number of operators. Applied in the case of LFSR targets to a tri-standard terminal, it lowers the hardware complexity by up to 40%.

J.Takala et.al in [10] presented partial-column radix-2 FFT processors and realizations of butterfly operations. The area and power-efficiency of butterfly units to be used in their processor organization based on bit-parallel multipliers, distributed arithmetic and CORDIC were analyzed and compared. Their processor organization permits the area of the FFT implementation to be traded against the computation time. The power consumption comparisons proved that butterflies based on bit-parallel multipliers were power-efficient but have limitations on clock frequency. Butterflies based on distributed arithmetic were used for higher clock frequencies. If extremely long FFTs are needed, then CORDIC based butterflies areapplicable.*L. Alaus et.al* [7] presented parameterization as a digital radio design methodology. Two different techniques, namely common functions and common operators were considered. The second view of their work was developed and illustrated with two examples: the well-known Fast Fourier Transform (FFT) and the Reconfigurable Linear Feedback Shift Register (R-LFSR), derived from the classical Linear Feedback Shift Register (LFSR) structure.

3. PROBLEM STATEMENT

Wavelet transform is an important signal analysis tool. Implementations of the wavelet transform in the embedded systems domain are generally concentrated on specific applications like image/video processing, etc. i.e., discrete wavelet transform has beena powerful tool for compression and is being used in many real-time, multimedia and embedded signal processing applications. DWT represents the signal in time-frequency domain. This approach is generally simple and effective. The DWT is implemented using a tree-structured filter banks. A number of filter banks are cascaded to produce a multi-resolution wavelet analysis, as shown below in Figure.1.

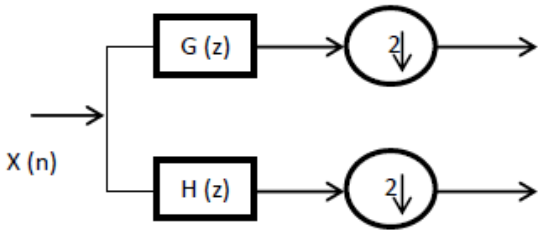

Figure.1. DWT Filter Bank Structure.

It is essential to design a high-performance FFT processor to cope with the demand of real time and low power application in number of systems viz., signal processing in embedded, communication or wireless communication systems. FFT algorithm is used to compute the discrete Fourier transform (DFT) algorithm practically. Numerous algorithms have evolved since it (FFT) came into existence; including a mathematical representation such as complex arithmetic operations. The DFT is acquired by breaking down a sequence of values into components of dissimilar frequencies. This operation is practicable directly but at the cost of time required for its computation. The Cooley–Tukey algorithm is the popularly used FFT algorithm. The FFT butterfly is represented in Figure.2.

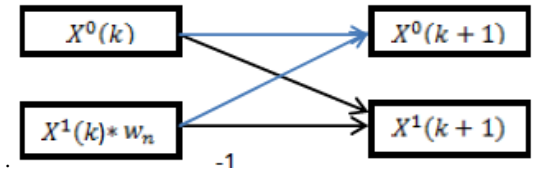

Figure.2. Butterfly Representation For FFT Unit.

The works presented in the literature areriveted on defining and utilizing the Common Operators under parameterization techniques. In this paper I have planned to utilize the concept of sharing two different algorithms under parameterization approach. The two distinct algorithms sharing the implementation module under the parameterization context are FFT and DWT algorithms. Processed data and functions performed by the DWT and FFT algorithms are compared, but they seem to be totally unlike. Therefore, I have decided to research it under the Parameterization context, so that the processing and implementation similarities between the DWT-lifting and FFT-butterfly units can be identified. Thus, a configurable processing module (element) that can switch between the two algorithms functions is proposed .i.e., the theme is to implement synthesis and simulate the Mutual module for DWT and FFT processing elements on FPGA platform using Xilinx-ise.

Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. REVIEW ON DWT AND FFT ALGORITHMS

DWT and FFT algorithms appear to be entirely different in their approach, in terms of processed data and functionality. Yet, if the system parameters are examined minutely, implementation/architectural resemblances could be identified in the Lifting and butterfly structures of DWT and FFT respectively. The DWT and FFT structures are researched in the following paragraphs for their architectural similarity.

4.1 The lifting DWT

To construct two-channel filter banks a swift and economical technique called the lifting scheme is applied. The lifting scheme (as represented in Figure.3) comprises of two computational measures: primal lifting (update) and dual lifting (predict). Its basic intention is to factor the polyphase matrix (equation (1)) of a wavelet filter into a sequence of upper and lower triangular matrices and a constant diagonal matrix [6,6,7]. Executing DWT with lifting scheme typically takes fewer multiplications and is more comfortable to supervise the boundary extension than executing DWT with convolution.

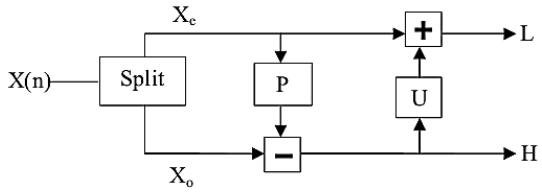

Figure.3 Lifting Scheme Based DWT Block Schematic

Executing a lifting step can be done by the following operations:

1) Split operation — splitting the input signal into disjoint components (i.e.to extract the even and odd components) this can be realized by the polyphase representation. This is also known as the lazy wavelet. The polyphase representation is composed of a delay unit and a pair of down samplers for splitting the input signal into odd and even samples.

2) Predict — the even samples are multiplied by the predict factor and then the results are added to the odd samples to generate the detailed coefficients. The predict operation is also referred to as the dual lifting step[1].

3) Update — the detailed coefficients computed by the predict step are multiplied by the update factors and then the results are added to the even samples to get the coarse coefficients. The update step is also referred to as the primal lifting step[1].

$$P(z)=\left\{\prod_{i=1}^{m}\begin{bmatrix}1 & 0\\ -t_i(z^{-1}) & 1\end{bmatrix}\begin{bmatrix}1 & -s_i(z^{-1})\\ 0 & 1\end{bmatrix}\right\}\begin{bmatrix}\frac{1}{k} & 0\\ 0 & k\end{bmatrix} \quad (1)$$

Finally, (see equation (1) normalization is applied for both the primal and dual lifting, such that the low pass and high pass sub bands are obtained respectively. This can be done by scaling the primal and dual lifting steps by k and 1/k respectively [2].

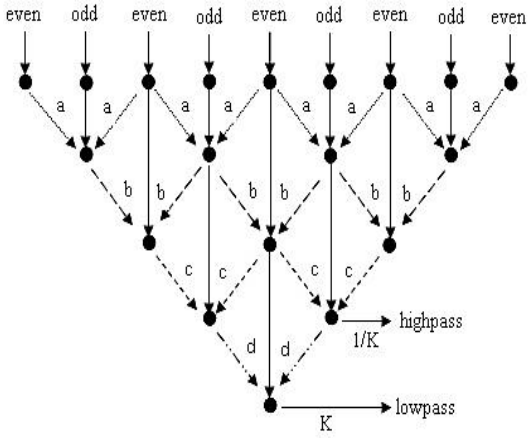

Figure.4 Data Flow Diagram For The (9/7) Lifting Based DWT.

4. 2 FFT Butterfly

The DFT is asignificant discrete transform type, used to perform Fourier analysis. Often DFT is computed by the Fast Fourier Transform (FFT). The arithmetic operations performed by the DFT consumesmore computation time. But, the computation time for execution of the same arithmetic operations by the FFT will be less. Also FFT algorithm is to a greater extent accurate than evaluating DFT definition

$$X_i=\sum_{k=0}^{N-1}x_k.e^{-j\left(\frac{2.\pi}{N}.k\right)}, i=0,...,N-1, \quad (2)$$

where $x_0,....x_{N-1}$ are complex numbers

Directly. The most common and well-known used FFT algorithm is the Cooley–Tukey algorithm which is to decomposes the transform by size *N*/r at each stage for chosen radix power-of-two, and these smaller transforms are then combined withthe

Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

structure in common is called as butterfly of size r. In this paper, the Radix-2 Cooley-Tukey FFT algorithm is considered. The data flow diagram of the processing element resembles to the butterfly like shape and hence the processing element is named "butterfly" is sketched in Figure.5 below. The radix-2 FFT algorithm in generalis practiced for the 2^n-point FFT size. From Equation (2) for N-point FFT the results of two-N/2 Fourier transform are required[3].

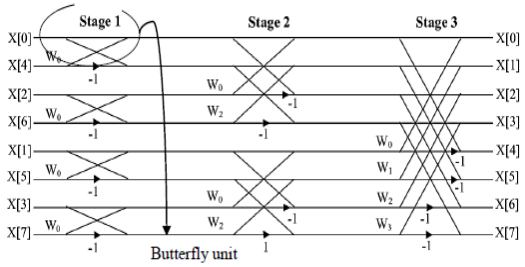

Figure.5 N-Point Radix-2 FFT Architecture Representing The Butterfly Unit

Then, an N-point transform can be reduced to two N/2-point transforms byusing the divide and conquer strategy of the input signal into even and oddelements.

$$\left.\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n * e^{-j\left(\frac{2.\pi}{N}.n.k\right)} \\ &= \sum_{m=0}^{(N/2)-1} x_{2m} * e^{-j\left(\frac{2.\pi}{N}.2m.k\right)} + \sum_{m=0}^{(N/2)-1} x_{(2m+1)} * e^{-j\left(\frac{2.\pi}{N}.(2.m.+1).k\right)} \end{aligned}\right\} \quad (3)$$

Thebutterfly structure to realize the radix-2 FFT algorithm is represented in Figure.6

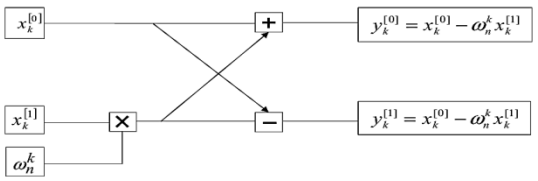

Figure.6 Represents The 2-Point Radix-2 FFT Butterfly Unit.

As discussedin this section if observed properly, a processing element for DWT with lifting scheme, is the essential operation and can be realized by one multiplication and two additions. We call this operation the Main Computation unit (MCU) [4].Where for the radix-2 FFT structure can be realized by the 2-point transform subdivision with N/2 butterflies in each stage and this implementation too requires a multiplier and a pair of adder/subtractor unit(see Figure.5 & 6).

The DWT and FFT algorithms have sound resemblances if we analyze their lifting and the butterfly structures respectively. These similitudes areexaminedin next section to figure a Mutual Module for these two algorithms.

5. PROPOSEDMUTUAL MODULEFORDWT ANDFFT ALGORITHMS

In this section, for implementation of area efficient embedded system a Mutual Moduleforthe two algorithms DWT and FFT is realized. A proposed implementation of the DWT and FFT processing elements are as follows:

5.1. DWT lifting structure

The lifting based DWT structure in Figure.3&4 definesand describes its computationmethodwith three basic operations respectively. The processing element of DWT in lifting scheme is recognized by the Main computation unit (MCU) [4].The operation of this MCU can be understood from Figure.7&8 below. The following equations (4, 5and 6) define the lifting operation of DWT and each step/stage is recognized by the defined processing element (MCU) in Figure.8 and equation (7).

$$\left.\begin{aligned} P(n) &= X_o(n) + a[X_e(n) + X_e(n+1)] \Rightarrow predict-step \\ U(n) &= X_e(n) + b[P(n-1) + P(n)] \Rightarrow update-step \end{aligned}\right\} \quad (4)$$

$$\left.\begin{aligned} Y_H(n) &= P(n) + c[U(n) + U(n+1)] \Rightarrow \\ & \qquad highpass-output \\ Y_L(n) &= U(n) + d[Y_H(n-1) + Y_H(n)] \Rightarrow \\ & \qquad lowpass-output \end{aligned}\right\} \quad (5)$$

$$\left.\begin{aligned} Y_H(n) &= K * Y_H(n) \\ Y_L(n) &= \left(\frac{1}{K}\right) * Y_L(n) \end{aligned}\right\} \Rightarrow normalised/scaled-output \quad (6)$$

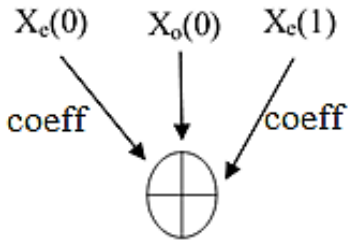

Figure.7. Conventional Representation Of Computational Unit (Processing Element) For The Lifting Based DWT.

$$R = X_o(n) + coeff * [X_e(n) + X_e(n+1)] \Rightarrow MCU \quad (7)$$

From the expression (7) above the MCU for the lifting DWT is designed as in Figure.8 below. As discussed in the previous section, the designed architecture for the MCU comprises of a pair of

Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

addition and a multiplication operation. The processing element first adds two sampled even inputs, then multiplies it with the polyphase filter coefficient and at last the odd sampled input signal is added. This can be clearly understood from the equation(7) above.

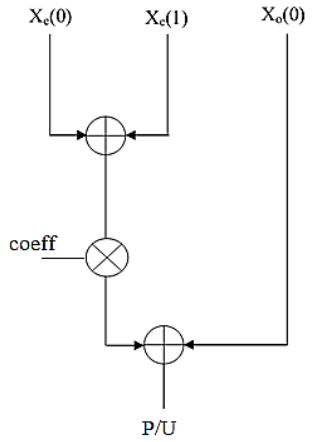

Figure.8 Proposed Graphical Representation Of The MCU For The Lifting Based DWT.

5.2. FFT butterfly structure

The FFT butterfly structure in Figure.6 defines its computation method. The butterfly unit graphically presented can derive the operations required for the radix-2 FFT implementation. As represented in Figure.6, computation of each butterfly unit (processing element) involvesan adder, a subtractor and a multiplier.

But, in practice FFT performs complex-valued operations. Therefore, the equations drawn in (8) define real and complex parameters [3] for the FFT butterfly inputs.

$$\left.\begin{aligned} x_k^{[1]} &= a + jb \\ x_k^{[0]} &= e + jf \\ w^k n &= c + jd \end{aligned}\right\} \Rightarrow complex - FFT - inputs \qquad (8)$$

By performing the operation as per the [3] butterfly representation in Figure.6, the values obtained will be in following way,

$$\left.\begin{aligned} y_k^{[0]} &= \left(x_k^{[0]} + x_k^{[1]} * w^k{}_n\right) = e + (ca - db) \\ &+ j.[f + (da + cb)] \\ y_k^{[1]} &= \left(x_k^{[0]} - x_k^{[1]} * w^k{}_n\right) = e - (ca - db) \\ &+ j.[f - (da + cb)] \end{aligned}\right\} \qquad (9)$$

But in order to minimize the computational complexity in terms of the hardware utilization, the above representation can be rewritten as in [3] equation (10) below. Since, the expression in equation (9) requires 4 multipliers and 6 adder-subtractor. The recomputed equation (10) reduces the number of the multiplier operation to three which requires more area and power consumption.

$$\left.\begin{aligned} y_k^{[0]} &= [e + (c.(a+b) - b.(c+d))] + \\ &j.[f + (c.(a+b) + a.(d-c))] \\ y_k^{[1]} &= [e - (c.(a+b) - b.(c+d))] \\ &+ j.[f - (c.(a+b) + a.(d-c))] \end{aligned}\right\} \qquad (10)$$

Thus, from the expressions above the radix-2 FFT butterfly is designed which requires 9 add-subtract and 3 multipliers in the architecture (see Figure.9).

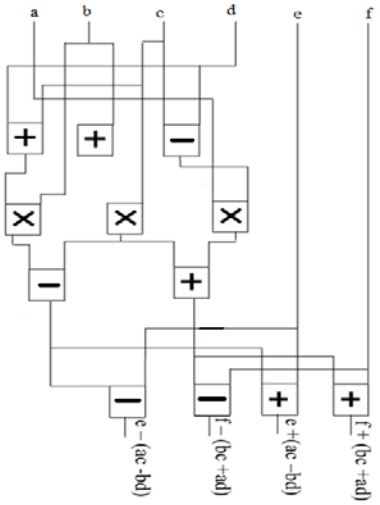

Figure.9 2-Point FFT Butterfly Implementation [3].

5.3. Mutual implementation of DWT in FFT

The DWT structure presented in the previous subsection can share its structure as the mutual part of the FFT structure. And thus in the FFT architecture, DWT is recognized and thus the hardware complexity and the area issues are considered. First we discuss the FFT implementation; the equation (10) clearly differentiates the real and imaginary part of the 2-point FFT. The DWT can be performed with either the real or the imaginary part computation of the FFT architecture as in derivation table below.

Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table: 1 Derivation Table For Mutual Operation Of Dwt In Fft Process.

Mutual operation of DWT in FFT process
$butterflypart - o / p = [e \pm (c.(a+b) - b.(c+d))] + j.[f \pm (c.(a+b) + a.(d-c))]$ $= [real] + j[imaginary]$ $\therefore real = [e \pm (c.(a+b) - b.(c+d))]$, and $imaginary = [f \pm (c.(a+b) + a.(d-c))]$
MCU Output is realized by either Real/Imaginary part of butterfly operation Therefore, MCU - O/P is realized from = $[f \pm (c.(a+b) + a.(d-c))]$ $but \Rightarrow MCU = R = X_o(n) + coeff * [X_e(n) + X_e(n+1)]$ Comparing we get $\Rightarrow [f + (c*(a+b)+0)] = [X_o(n) + (coeff * (X_e(n) + X_e(n+1)))]$ This is represented in Figure.10.

This can be done with the introduction of a couple of multiplexers at the input of the real/imaginary part and a multiplexer each for coefficient and the odd/non-multiplier part of the DWT input. For 2-point FFT complex input, a 2-point FFT output in real and imaginary form is obtained. While for DWT the predict/the update step is computed by the MCU.

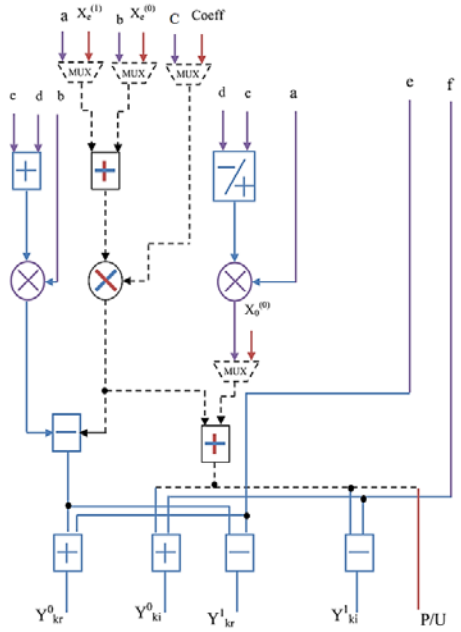

Figure.10 Proposed Mutual Module for DWT and FFT units

The lifting DWT and the FFT butterfly are differentiated or selected by a select line for the multiplexers. After the complete discussion and presentation of the DWT and FFT architectures, the Mutual Module for the same is represented in Figure.10. This Mutual Module consents to flip among two distinct executions of the DWT and FFT algorithms with the help selector.

6. RESULTS AND COMPARISON

The experimental results of our proposed method are presented below. The proposed design is simulated and synthesized using Xilinx ISE 10.1. Result for the proposed mutual module obtained are discussed and compared with the individual implementation in terms of area and power consumed. The logic utilization table for the proposed Mutual module architecture is shown in the figure below.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TABLE.1 Device Utilization Table Of Proposed Work

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	128	19,200	1%	
Number used as Flip Flops	128			
Number of Slice LUTs	175	19,200	1%	
Number used as logic	175	19,200	1%	
Number using O6 output only	160			
Number using O5 and O6	15			
Slice Logic Distribution				
Number of occupied Slices	52	4,800	1%	
Number of LUT Flip Flop pairs used	190			
Number with an unused Flip Flop	62	190	32%	
Number with an unused LUT	15	190	7%	
Number of fully used LUT-FF pairs	113	190	59%	
Number of unique control sets	3			

Table.2
Device Utilization Table Of 9/7 Dwt

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice LUTs	212	19,200	1%	
Number used as logic	212	19,200	1%	
Number using O6 output only	208			
Number using O5 output only	1			
Number using O5 and O6	3			
Number of route-thrus	4	38,400	1%	
Number using O6 output only	1			
Number using O5 output only	3			
Slice Logic Distribution				
Number of occupied Slices	62	4,800	1%	
Number of LUT Flip Flop pairs used	212			
Number with an unused Flip Flop	212	212	100%	
Number with an unused LUT	0	212	0%	
Number of fully used LUT-FF pairs	0	212	0%	

TABLE.3
Device Utilization Table Of 8-Ponit Fft

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	16	19,200	1%	
Number used as Flip Flops	16			
Number of Slice LUTs	128	19,200	1%	
Number used as logic	128	19,200	1%	
Number using O6 output only	113			
Number using O5 and O6	15			
Slice Logic Distribution				
Number of occupied Slices	37	4,800	1%	
Number of LUT Flip Flop pairs used	128			
Number with an unused Flip Flop	112	128	87%	
Number with an unused LUT	0	128	0%	
Number of fully used LUT-FF pairs	16	128	12%	
Number of unique control sets	1			

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table. 4
Logic Comparison

Sr. No.	parameters	9/7 2D-DWT	8-point FFT	Proposed Mutual Module
1	Adder/subtractor	60	108	9
2	Multiplers	36	36	3
3	Mux	-	-	4
4	Total Area	274	228	242
5	Total Power@100 MHz	300mW	315mW	253mW
6	Total power @200MHz	318mW	387mW	267mW

The comparison table above interprets the logic utilized by the respective architectures of the DWT and FFT with the proposed Mutual module. If, in an embedded system a DWT and FFT architectures of 9/7 filter and 8-point radix-2 butterfly respectively are implemented then the total area required and power consumed by this system will be very high. This can be clearly recognized from the comparison table. And hence a Mutual module is proposed which is reused to perform both the operations of FFT and DWT instead of implementing two different architectures for executing their respective functions.

The 9/7 lifting based DWT requires 60 adder/subtractor and 36 multipliers with total area of 274 slices and power consumed is 300mW. And in case of 8-point radix-2 FFT architecture utilizes 108 adder/subtractor and 36 multipliers in return it consumes 315mW power and 228 slices of area. Whereas, the proposed Mutual module for the DWT and FFT architecture utilizes 9 adder/subtractor, 3 multiplier and 4 multiplexers with total area of 242 slices and power consumed is 253mW. Therefore, the presented approach proves to be effective and less complex implementation.

7. CONCLUSION

In this paper parameterization technique is approached which takes advantage of the mutual modules present in the two distinct algorithms popularly used in the Embedded and wireless communication systems applications. A common structure for the DWT and FFT algorithms is proposed. The Mutual module approach presents considerably low hardware complexity, reduced area and power consumption. This approach performed over 8point FFT and 9/7 DWT system under test achieves almost 40 % of reduction in power consumption and for the same system under test the area reduced is 48%.

REFERENCE

[1] Chin-Fa Hsieh, Tsung-Han Tsai, Neng-Jye Hsu, Chih-Hung Lai, " A Novel, Efficient Architecture for the 1D, Lifting-Based DWT with Folded and Pipelined Schemes," ;*in Proc. JCIS*, 2006.

[2] A. Mansouri, A. Ahaitouf, and F. Abdi, "An Efficient VLSI Architectureand FPGA Implementation of High-Speed and Low Power 2-D DWT for (9, 7) Wavelet Filter," *IJCSNS International Journal of Computer Science and Network Security*, Vol.9 No.3, March 2009.

[3] MalekNaoues, Laurent Alaus, Dominique Noguet, "A Common Operator for FFT and Viterbi algorithms," Digital System Design: Architectures, Methods and Tools (DSD), *13th Euro micro Conference, IEEE*, pp.309-313, 2010.

[4] Po-Chih Tseng, Chao-Tsung Huang, and Liang-Gee Chen, "VLSI implementation of shape-adaptive discrete wavelet transform," *Visual Communications and Image ProcessingProceedings of SPIE*, vol. 4671, 2002.

[5] L. Alaus, D. Noguet, J. Palicot," A Common Operator Bank to resolve scheduling issue on a SDR Terminal," *6th Advanced international conference on telecommunications*, 2010.

[6] S.T. Gul, C. Moy, J. Palicot, "Two scenarios of flexible Multi-standard architecture designs using a multi-granularity exploration", the 18th Annual IEEE International Symposium on Personal, *Indoor and Mobile Radio Communications*, 2007.

[7] L. Alaus, J. Palicot, C. Roland, Y. Louet, D. Noguet, "Promising Technique of Parameterization For Reconfigurable Radio, the Common Operators Technique: Fundamentals and Examples" *Journal of Signal Processing Systems*, Springer, 2009.

[8] D. Noguet, G. Masera, V. Ramakrishnan, M.Belleville, D. Morche, G. Asheid, "Considering microelectronic trends in advanced wireless system design," *Advances in Electronics and Telecommunications Journal*, April 2010.

[9] MalekNaoues, Laurent Alaus, Dominique Noguet and Yves Louët, "A Common Operator for FFT and FECDecoding," *in*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Microprocessors and Microsystems Journal, vol.35, issue no.8, pp.708-715, 2011.

[10] J.Takala and K. Punkka, "Scalable FFT Processors and Pipelined Butterfly Units", *The Journal of VLSI Signal Processing, Springer Netherlands* 2006.

[11] C. Moy, J. Palicot, V. Rodriguez, D. Giri, "Optimal Determination of Common Operators for Multi-Standard Software-Defined Radio", *4th Karlsruhe Workshop on Software Radios*, March 2006.

[12] Cooley, J. W. and Tukey, J. W, "An Algorithm for the Machine Calculation of Complex Fourier Series", *Math. Computat.*, 19, 297–301, 1965.

[13] 13. J. Mitola, "The software radio architecture," *IEEE Communications Magazine*, 33, pp. 26-38, 1995.

[14] KemihKarim, KimoucheMosbah, BenslamaMalek, BenslamaAchour, "Performance of Discrete Wavelet Transform with Convolutional Code in Chaotic DS-CDMA System for Microsatellite Communication," *Geoscience and Remote Sensing Symposium, 2004.IGARSS '04. Proceedings, 2004 IEEE International*, Volume: 1.

[15] L. Biard, D. Noguet,"An adaptable architecture for the FFT algorithm", *the 7th International Symposium on Wireless Personal Multimedia Communications*, 2004.

[16] F. Jondral, "Software Defined Radio: Enabling technologies", *Wiley* 2002.

[17] W. Tuttlebee, "Software Defined Radio – Baseband Technology for 3G Handsets and Basestations," *Communications Engineer* vol. 2, no. 2, pp. 46-47.

[18] F. Jondral, "Parameter Controlled Software Defined Radio," *Software Defined Radio Technical Conference*, San Diego, Nov 2002.

[19] Malek. Naoues, Dominique. Noguet, Yves. Loüet, Khaled. Grati,Adel.Ghazel, "An Efficient Flexible Common Operator for F FT andViterbi Algorithms," *IEEE 73rd Vehicular Technology Conference* (*VTC Spring)*, pp. 1-5, 2011.

[20] Ali Chamas. Al-Ghouwayel, Amin.Haj-Ali, Zouhair.El-Bazzal andYves.Loüet, "Towards a Triple Mode common operator FFT forsoftware radio systems,"*19th International conference on telecommunication, IEEE*, pp.1-6, 2012.