Research Journal of Applied Sciences, Engineering and Technology 7(22): 4706-4713, 2014
DOI:10.19026/rjaset.7.855
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: November 07, 2013 Accepted: December 09, 2013 Published: June 10, 2014

Research Article

Reconfigurable and Reusable Mutual Module Based Parameterization Approach for DWT and FFT Algorithm

[1]J. Venkatesh and [2]R. Rani Hemamalini
[1]St. Peter`s University,
[2]Department of ECE, St. Peter`s College of Engineering and Technology, Avadi, Chennai-600054, India

Abstract: Goal of this research article is to study and the analysis of a parameterization technique called Common Operator for an embedded digital signal processing system and implementation of that embedded system using a special architecture to achieve better performance. This can be realized by efficient hardware design with reduced area and low-power dissipation which is a significant challenge for embedded systems, particularly in portable devices. The challenge is even more pronounced when DWT and FFT with large transform lengths need to be realized in embedded hardware for several embedded signal processing applications. When it comes to miniaturization or reuse or sharing of the hardware, two Parameterization approaches, viz; the Common Function approach and the Common Operator approach are the effective ones. But in this study, it is proposed a reconfigurable FFT (Fast Fourier Transform) operator using the common operator approach. This operator can be reconfigured to switch from an operator dedicated to compute the FFT to an operator which computes the FFT in order to perform DWT. Contribution of this study is to Minimize the waste of resources by modifying slightly the FFT butterfly to support the DWT computation and to reduce the resources allocated to the reconfiguration, Simplify the reconfiguration between the two algorithms, by separating the calculation of the Real and Imaginary parts of FFT modules.

Keywords: DWT-lifting, FFT-butterfly, mutual module, parameterization

INTRODUCTION

Currently, 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, the parameterization techniques have been introduced Naoues *et al*. (2010, 2011a), 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 study, 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 particular function is implemented and

Corresponding Author: J. Venkatesh, St. Peter`s University, Avadi, Chennai-600054, India

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

The work presented in this study will be to share two most important and functionally different algorithms implemented in wireless communication systems viz., FFT and DWT algorithms. Where, FFT algorithm is usually utilized for decoding or multicarrier modulation in OFDM along with DWT algorithm preferred in thresholding (wavelet thresholding) or in Multicarrier Modulation in OFDM (MCM-OFDM). In this study, i have developed a Mutual Module based parameterization approach for DWT and FFT algorithm. The two distinct algorithms sharing the implementation module under the parameterization context are FFT and DWT algorithms. If the processed data and functions performed by the DWT and FFT algorithms are compared, they seem to be totally unlike. Therefore, a common function can't be recognized and hence in the common operator based parameterization approach for DWT and FFT algorithm is considered to enhance the design of the system which can perform both FFT and DWT operation by executing the one common architecture for FFT and DWT algorithm instead of executing both individually. The system proposed in this study will be a reconfigurable cell that can be reused to perform the operations of both DWT and FFT algorithm. Contribution of this article is to minimize the waste of resources by modifying slightly the FFT butterfly architecture to support the DWT computation architecture and to reduce the resources allocated to the reconfiguration cell. The reconfiguration between the two algorithms is simplified by separating the complex valued FFT architecture into Real and Imaginary parts of the FFT module. The Reconfigurable Mutual Module based parameterization approach for DWT and FFT algorithm design alike the existing common operator techniques will provide effective results in terms of Area required and power consumed by the traditional FFT and DWT architectures.

LITERATURE REVIEW

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, Naoues *et al*. (2011a) presented a common butterfly for the FFT and Viterbi algorithms. They investigated where reuse and power consumption is traded against throughput. Performance comparisons with similar works are also discussed. Al-Ghouwayel *et al*. (2012) 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. Gul *et al*. (2007) presented a 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 study illustrated two scenarios of multi-standard radio system designs. Furthermore their approach presented the scheduling issues.

Naoues *et al*. (2010) 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. Naoues *et al*. (2011b) 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.

Alaus *et al*. (2010) elaborated the Common Operator (CO) technique, which defined a multi standard terminal, based on a limited set of Common Operators. Their approach enhanced the re-configurability 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 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%.

Takala and Konsta (2006) 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 are applicable. Alaus *et al*. (2009) 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.

Motivation of research: Recently, the parameterization techniques have been introduced (Naoues *et al*., 2010, 2011a), 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 literature there have lot of works presenting Parameterization approach, called the common operator Technique for Viterbi and FFT algorithm. This approach was to design the two distinct algorithms sharing the implementation module under the parameterization context are FFT and Viterbi algorithms. And mean while the work presented by Naoues *et al*. (2011a) motivated me to think further over implementation of common operator and hence i have decided to enhance the design in common operator based parameterization approach for DWT and FFT algorithm. The work in this study will be to design a reconfigurable cell that can be reused across these functions of DWT and FFT algorithm. Contribution of this study is to Minimize the waste of resources by modifying slightly the FFT butterfly to support the DWT computation and to reduce the resources allocated to the reconfiguration, Simplify the reconfiguration between the two algorithms, by separating the calculation of the Real and Imaginary parts of FFT modules.

The Reconfigurable Mutual Module based parameterization approach for DWT and FFT algorithm design alike the existing common operator techniques will provide effective results. The Reconfigurable Mutual Module based architecture proposed will be designed in Verilog and implemented will be synthesized in Xilinx ISE tool.

PROPOSED METHODOLOGY

DWT and FFT algorithms appear to be entirely different in their approach, interms 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.

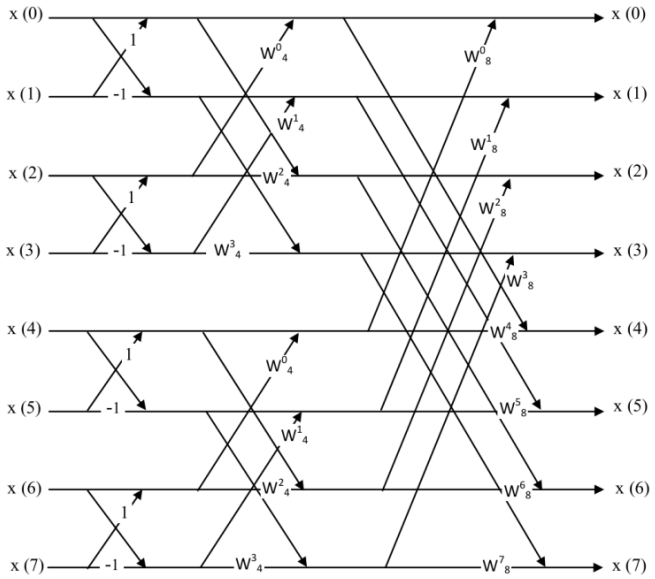

Fig. 1: N-point radix-2 FFT architecture constituted by several butterfly elements

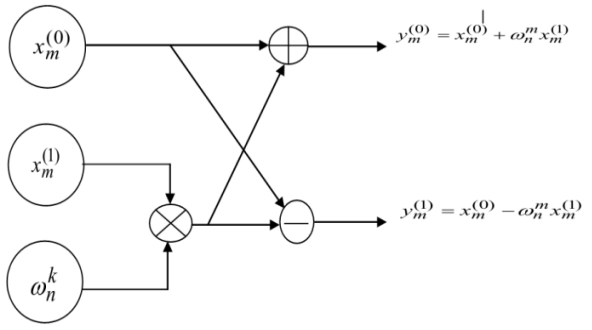

Fig. 2: A 2-point radix-2 FFT butterfly element

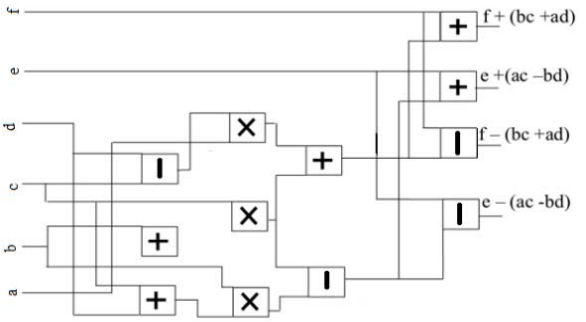

Fig. 3: A 2-point FFT butterfly realization (Al Ghouwayel *et al*., 2012)

FFT algorithm, its review and proposed approach: DFT is one of the best transform of its kind. DFT performs Fourier Anlysis. DFT to a great deal is computed using FFT. This is because the computation time consumed by DFT to perform the arithmetic operations is more while the same arithmetic operations performed by FFT consumes less computational time:

$$X_l = \sum_{m=0}^{N-1} x_m . e^{-j\left(\frac{2.\pi}{N}.m\right)}, l = 0,...,N-1, \text{where } x_0,...x_{N-1} \quad (1)$$

are complex numbers

Of many algorithms presented for FFT. Cooley and Tukey (1965) algorithm is most significantly utilized to decompose the transform by size *N*/r at every phase for preferred radix-2. The decomposed transform constitute as smaller elements called Butterfly structure/element. The combination of all these small elements for FFT computation is called as FFT butterfly of size 'r'. The name butterfly is derived from the structure resemblance of the processing element of the Radix FFT architecture as in Fig. 1.

In this study, radix-2 cooley-tukey FFT algorithm is presented. The N-point FFT can be presented as 2^n-point FFT size and to do so a radix-2 cooley-tukey algorithm is utilized. A Radix-r cooley-tukey algorithm based N-FFT algorithm is expressed in Eq. (1). Therefore the N-point FFT by adopting radix-2 cooley-tukey algorithm requires results of only two-N/2 Fourier transforms, thereby divide and conquer scheme is applied to the input signal to obtain even and odd elements. This is expressed in Eq. (2). The sketch below in Fig. 2 represents the butterfly element of the radix-2 cooley-tukey FFT algorithm. The mathematical computation of FFT butterfly structure resembles as in Fig. 2:

$$\left.\begin{aligned} X_m &= \sum_{q=0}^{N-1} x_q * e^{-j\left(\frac{2.\pi}{N}.q.m\right)} \\ &= \sum_{p=0}^{(N/2)-1} x_{2p} * e^{-j\left(\frac{2.\pi}{N}.2p.m\right)} + \sum_{p=0}^{(N/2)-1} x_{(2p+1)} * e^{-j\left(\frac{2.\pi}{N}.(2.p.+1).m\right)} \end{aligned}\right\} \quad (2)$$

The schematic symbol presented for butterfly element manages to draw the performance required for a Radix-2 FFT implementation. From the butterfly element presented in Fig. 2, it is observed that each of the processing elements of Radix-2 FFT is constituted by an adder, a subtract and a multiplier. This add-subtract and multiply unit performs mathematical computation on the complex values. The mathematical model of complex valued FFT operation can be expressed in general form as in Eq. (3). The general form of Eq. (3) Naoues *et al*. (2010) defines real and imaginary parameters of the butterfly element. Input to the butterfly element is expressed as below:

$$\left.\begin{aligned} x_m^{[1]} &= A + jB \\ x_m^{[0]} &= E + jF \\ w^m n &= C + jD \end{aligned}\right\} \Rightarrow complex\ -FFT\ -inputs \quad (3)$$

Output of the butterfly element for the complex input parameters is expressed as:

$$\left.\begin{aligned} y_m^{[0]} &= \left(x_m^{[0]} + x_m^{[1]} * w^m{}_n\right) = E + (CA - DB) + j.[F + (DA + CB)] \\ y_m^{[1]} &= \left(x_m^{[0]} - x_m^{[1]} * w^m{}_n\right) = E - (CA - DB) + j.[F - (DA + CB)] \end{aligned}\right\} \quad (4)$$

To down play the computational complexity for designing the butterfly element Eq. (4) can be Rearranged (Naoues *et al*., 2010). In Eq. (4) the mathematical representation of bitterfly element is recorde, which when modeled will require 6 add-subtract units and 4-multipliers. Thus by rearranging Eq. (4) as in Eq. (5), we reduce the model complexity and thus requires less multipliers than the previous representation. The rearranged expression helps minimize architecture/ hardware complexity by reducing multipliers in the architecture. According to Eq. (5) it will require 9 add-subtract. This is sketched in Fig. 3:

$$\left.\begin{aligned} y_m^{[0]} &= [E + (C.(A + B) - B.(C + D))] + j.[F + (C.(A + B) + A.(D - C))] \\ y_m^{[1]} &= [E - (C.(A + B) - B.(C + D))] + j.[F - (C.(A + B) + A.(D - C))] \end{aligned}\right\} \quad (5)$$

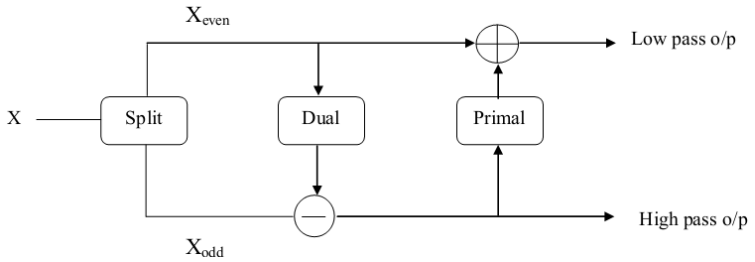

Fig. 4: Lifting scheme based DWT block schematic

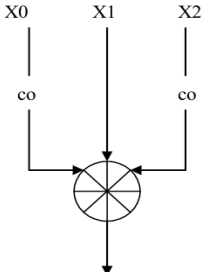

Fig. 5: Conventional representation of computational unit (processing element) for the lifting based DWT

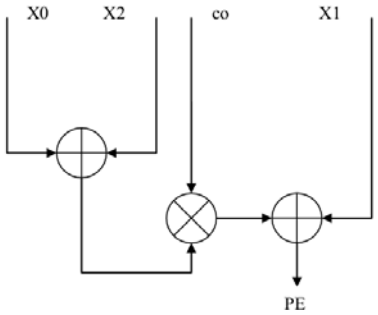

Fig. 6: Proposed graphical representation of the MCU for the lifting based DWT

DWT algorithm, its review and proposed approach: Interpretation of a signal in time-frequency domain is practiced by Discrete Wavelet Transform (DWT). DWT is preferred because it is implemented using a simple and effective approach of tree structured filter banks. In this study, since we are utilizing the lifting DWT, this part of the section reviews lifting DWT. To design the tree structured two channel filter banks, lifting scheme is preferred because of its fast and efficient approach. Primal and Dual lifting are the two performance measure of lifting scheme along with splitting operation. The primal lifting step is also referred to as Update operation, while the Dual lifting step is referred to as Predict operation.

Characteristics of lifting DWT:

- Factorizing a polyphase matrix into a sequence of upper and lower triangular matrices and a constant diagonal matrix
- Two performance measures viz; Predict (Primal) and Update (Dual) lifting steps
- To execute DWT with lifting technique a couple of arithmetic operations are required
- Normalization of the Dual and Primal lifting outcomes
- With lifting technique it is easier to supervise the boundary extension:

$$P(z)=\left\{\prod_{l=1}^{m}\begin{bmatrix}1 & 0\\ -h_l(z^{-1}) & 1\end{bmatrix}\begin{bmatrix}1 & -i_l(z^{-1})\\ 0 & 1\end{bmatrix}\right\}\begin{bmatrix}\frac{1}{k} & 0\\ 0 & k\end{bmatrix} \quad (6)$$

The lifting based DWT schematic presented in Fig. 4 defines the mathematical model of the Eq. (6) and also describes the computation techniques having three lifting steps respectively. The Lifting element/Processing element presented in Fig. 4 is usually constitutes by Multiply Accumulate unit (MAC), but here we refer it as Main Computation Unit (MCU) which is defined similarly by Tseng *et al*. (2002). Mathematical expression for the same is recorded in Eq. (7) to (10), while the model design for the MCU is sketched in Fig. 5 and processing element based on MCU is sketched in Fig. 6. The complete Data Flow Hardware design of the Lifting DWT for 9/7 filter is represented in Fig. 7:

$$\left.\begin{aligned} P_n &= X1 + \alpha\left[X0 + X2\right] \Rightarrow \textit{predict}\\ U_n &= X2 + \beta\left[P_0 + P_1\right] \Rightarrow \textit{update}\end{aligned}\right\} \quad (7)$$

$$\left.\begin{aligned} Y_H &= P_n + \gamma\left[U_n + U_n\right] \Rightarrow \textit{highpass}\\ Y_L &= U_n + \delta\left[Y_{H(n-1)} + Y_{H(n)}\right] \Rightarrow \textit{lowpass}\end{aligned}\right\} \quad (8)$$

$$\left.\begin{aligned} Y_H &= K * Y_H\\ Y_L &= \left(\frac{1}{K}\right) * Y_L\end{aligned}\right\} \Rightarrow \textit{scaling} \quad (9)$$

$$R = X1 + co*\left[X0 + X2\right] \Rightarrow MCU \quad (10)$$

The processing element initially, adds the 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 Eq. (10).

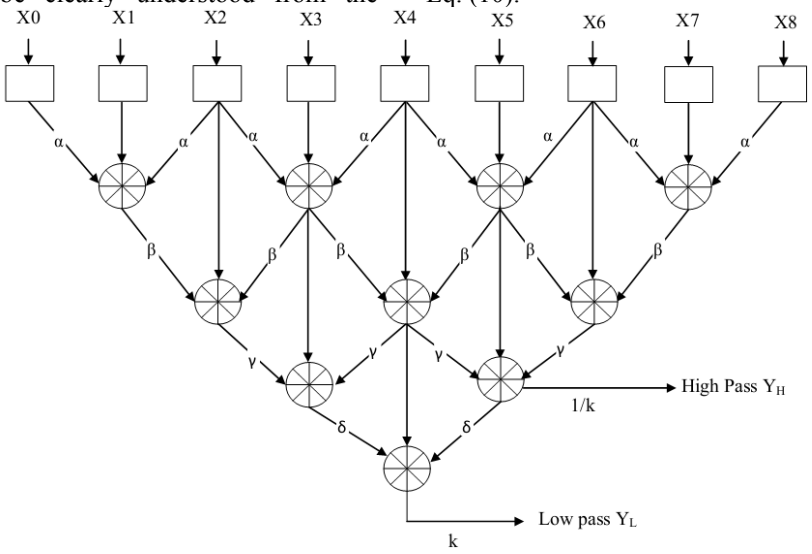

Fig. 7: Data flow diagram for the (9/7) lifting based DWT

Throughout this section we reviewed the FFT and DWT structure for its realization on FPGA platform. Also, we outline a approach to realize FFT and DWT as a common operator under parameterization technique. Here, we have configured the processing elements in such a way that, it meets the common operator requirements. We call it as mutual module; this is because the processing element of the FFT (radix-2 cooley-tukey algorithm) shares its processing elements or configuration to perform DWT operation (lifting DWT processing element). The mutual module is discussed in the next part of this section.

Reconfigurable and reusable mutual module for DWT and FFT algorithms: Now in this part we discuss the implementation of the two algorithms DWT and FFT adopting parameterization approach. Mutual Module for DWT and FFT is configured using parameterization based common operator technique. It is successfully executed by dividing the FFT outputs as real part of a pair of output and imaginary pair results. Now, the DWT processing element i.e., MCU output is either detail coefficient or coarse coefficient. Therefore, we have pinned down this as expression for better understanding:

$$FFT\_butterfly\_element\_o/p-1=[E+(C.(A+B)-B.(C+D))]+ j.[F+(C.(A+B)+A.(D-C))]$$

$$FFT\_butterfly\_element\_o/p-2=[E-(C.(A+B)-B.(C+D))]+ j.[F-(C.(A+B)+A.(D-C))]$$

$$Actually\_FFT\_butterfly\_o/p=[real]+j[imaginary]$$

$$real\_part=\left[E\pm\left(C.(A+B)-\underline{B.(C+D)}\right)\right], and$$

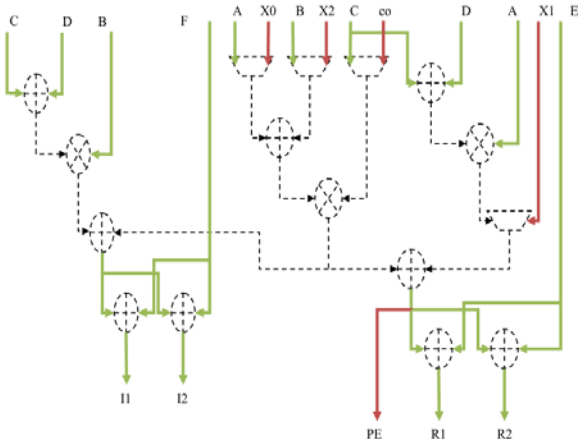

Fig. 8: Mutual module for DWT and FFT algorithms

$$imaginary\_part=\left[F\pm\left(C.(A+B)+\underline{A.(D-C)}\right)\right]$$

The underlined parameters are uncommon in real part and imaginary part, while C (A+B) is common and this can be utilized as common operator:

$$where\_MCU\_o/p\Rightarrow MCU=X1+co*[X0+X2]$$

Therefore, MCU is realized by either Real apart or Imaginary part of butterfly element:

$$=[E\pm(C.(A+B)+B.(C+D))]or[F\pm(C.(A+B)+A.(D-C))]$$

$$\text{Comparing}\rightarrow[(A(D-C)+C.(A+B))]or$$
$$[(B.(C+D)+C.(A+B))]=[X1+(co*(X0+X2))]$$

Now, the mutual module representation will be configured as in Fig. 8.

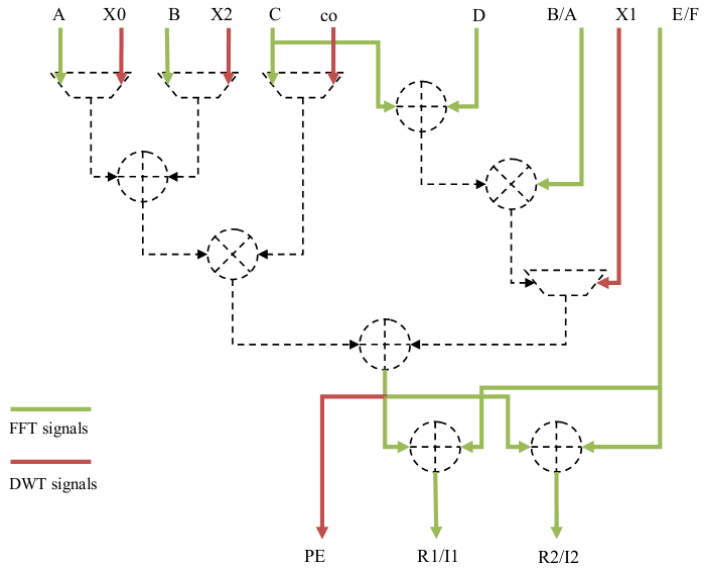

Fig. 9: Proposed reconfigurable and reusable mutual module for DWT and FFT algorithms

If we compare the DWT and FFT processing element expressions they are realized by simple multiplication and addition operations. Thus, we can say that DWT can be accommodated in FFT. Thereby, DWT is accommodated in FFT module to configure a Mutual Module. This configuration of mutual module reduces the power consumption and area required by the two individual systems viz; FFT and DWT on the chip. While, our implementation can be further extended to reconfigure and reuse of the available mutual module hardware. Again, we pin-down this reusable and reconfigurable mutual module by realizing DWT in half part (real or imaginary part) execution of the radix-2 cooley-tukey FFT algorithm i.e., each module viz; MCU of DWT, Real part of FFT, Imaginary part of FFT is executed in individual clock cycle. Thus, making the implementation low on throughput rate but reduces area and power consumption. The mathematical expression for reconfigurable and reusable mutual module is presented below.

From the Fig. 8 we can analyze that the Mutual module can be still reconfigured by reusing the real part of presented radix-2 FFT butterfly element as the imaginary part in next clock cycle:

$$\text{Comparing} \rightarrow C.(A+B) = co*(X0 \quad + X2)$$
$$and \rightarrow A.(D-C) or B.(C+D) = X1$$

Here we reuse B. (C + D) _as_A (D - C) in next clock. Where B. (C + D) contribute to real part of FFT and A (D - C) contributes to imaginary part of FFT. Thus, the reuse and reconfiguration technique is represented in Fig. 9.

The lifting DWT and radix-2 FFT butterfly are selected by a select line for the multiplexers while the real and imaginary part of fft are executed in 2 clock cycles. Therefore FFT execution require 2 clock cycles.

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 (Table 1 to 4).

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 300 mW. And in case of 8-point radix-2 FFT architecture utilizes 108 adder/subtractor and 36 multipliers in return it consumes 315 mW 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 81 slices and power consumed is 247 mW. Therefore, the presented approach proves to be effective and less complex implementation.

Table 1: Device utilization table of proposed reconfigurable and reusable mutual model

Device utilization summary				
Slice logic utilization	Used	Available	Utilization (%)	Note (s)
Number of slice LUTs	64	19,200	1	
Number used as logic	63	19,200	1	
Number using 06 output only	49			
Number using 05 output only	14			
Number used as exclusive route-thru	1			
Number of route-thrus	15	38,400	1	
Number using 06 output only	15			
Slice logic distribution				
Number of occupied slice	17	4,800	1	
Number of LUT flip flop pairs used	64			
Number with an unused flip flop	64	64	100	
Number with an unused LUT	0	64	0	
Number of fully used LUT.FF pairs	0	64	0	

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 06 output only	208			
Number using 05 output only	1			
Number using 05 and 06	3			
Number of route-thrus	4	38,400	1	
Number using 06 output only	1			
Number using 05 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	64	19,200	1	
Number used as flip flops	64			
Number of slice LUTs	176	19,200	1	
Number used as logic	176	19,200	1	
Number using 06 output only	176			
Slice logic distribution				
Number of occupied slices	52	4,800	1	
Number of LUT flip flop pairs used	176			
Number with an unused flip flop	112	176	63	
Number with an unused LUT	0	176	0	
Number of fully used LUT.FF pairs	64	176	36	
Number of unique control sets	1			

Table 4: Logic comparison table

Parameters	9/7 2D-DWT	8-point FFT	Proposed mutual module
Adder/subtractor	60	108	5
Multiplers	36	36	2
Mux	-	-	4
Total area	274	228	81
Total power@100 MHz	300 mW	315 mW	247 mW
Total power@200 MHz	318 mW	387 mW	250 mW

CONCLUSION

In this study Reconfigurable and reusable common operator based parameterization technique is approached which takes advantage of the mutual modules present in the two distinct algorithms popularly used in the Embedded Digital Signal Processing systems applications. A common operator for the DWT and FFT algorithms is proposed. The Reconfigurable and Reusable Mutual module approach presents considerably low hardware complexity, reduced area and power consumption. This approach is performed over 9/7 DWT and 8-point FFT system under test achieves almost 60% of reduction in power consumption and for the same system under test the area reduced is 84%.

REFERENCES

Alaus, L., D. Noguet and J. Palicot, 2010. A common operator bank to resolve scheduling issue on a complexity optimized SDR terminal. Proceeding of 6th Advanced International Conference on Telecommunications (AICT, 2010), pp: 142-146.

Alaus, L., J. Palicot, C. Roland, Y. Louet and D. Noguet, 2009. Promising technique of parameterization for reconfigurable radio, the common operators technique: Fundamentals and examples. J. Signal Process. Sys., 62(2): 173-185.

Al Ghouwayel, A.C., A. Haj-Ali, Z. El-Bazzal and Y. Loüet, 2012. Towards a triple mode common operator FFT for software radio systems. Proceeding of 19th International Conference on Telecommunication (ICT, 2012), pp: 1-6.

Cooley, J.W. and J.W. Tukey, 1965. An algorithm for the machine calculation of complex Fourier series. Math. Comput., 19(90): 297-301.

Gul, S.T., C. Moy and J. Palicot, 2007. Two scenarios of flexible Multi-standard architecture designs using a multi-granularity exploration. Proceeding of the IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, pp: 1-5.

Naoues, M., L. Alaus and D. Noguet, 2010. A common operator for FFT and viterbi algorithms. Proceeding of the 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, pp: 309-313.

Naoues, M., L. Alaus, D. Noguet and Y. Louët, 2011b. A common operator for FFT and FEC decoding. Microprocess. Microsy., 35(8): 708-715.

Naoues, M., D. Noguet, Y. Louët, K. Grati and A. Ghazel, 2011a. An efficient flexible common operator for FFT and viterbi algorithms. Proceeding of IEEE 73rd Vehicular Technology Conference (VTC, Spring), pp: 1-5.

Takala, J. and P. Konsta, 2006. Scalable FFT processors and pipelined butterfly units. J. VLSI Sig. Proc. Syst., 43(2-3): 113-123.

Tseng, P.C., C.T. Huang and L.G. Chen, 2002. VLSI implementation of shape-adaptive discrete wavelet transform. Proceeding of SPIE 4671, Visual Communications and Image Processing, pp: 655.