Embedded Reconfigurable Array Fabrics for Efficient Implementation of Image Compression Techniques

Sajid Baloch[1,2], Tughrul Arslan[1,2], Adrian Stoica[1,3]

1: School of Electronics & Engineering. University of Edinburgh, King's Buildings, Mayfield Rd, EH9 3JL, UK
2: Institute for System Level Integration, The Alba Campus, The Alba Centre, Livingston, EH54 7EG, UK)
3: NASA, Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109, USA

ABSTRACT

The discrete wavelet Transform (DWT), as defined by the Image Compression Standard JPEG-2000, is one of the most time-consuming computations which cannot be efficiently executed on current hardware architectures. This paper presents and compares a number of new, different architectures for domain-specific arrays to efficiently implement various DWT algorithms. A number of different algorithms are mapped to demonstrate the flexibility of these new embedded configurable SoC architectures and their ability to support different implementations having different performance characteristics. Our results demonstrate up to 59 percent improvement to the previous work in literature.

I. INTRODUCTION

Multimedia processing on hand-held devices, such as mobile phones, requires significant computational power to execute specific computations such as Discrete Wavelet Transform (DWT). Digital Signal Processors (DSPs) provide a hardware solution but due to high operating frequency, they results in high power consumption. The other solution is a dedicated hardwired logic. This solution reduces the power consumption considerably at the expense of reducing flexibility in the hardware. The specification for such algorithms do consistently change over period of time, hence there is a need for a flexible architecture that could accommodate such changes and integrate within a system-on-chip (SoC) design overflow. A possible solution could be Field Programmable Gate Arrays (FPGA) which provides a high-flexibility and low-cost but at the expense of increased power consumption and large integration costs.

The domain-specific arrays are less flexible than generic FPGA but in case of domain-specific computations, these are more efficient over generic FPGA. Furthermore, image compression techniques such as DWT have a number of different algorithms each with different advantages terms of quality, power-consumption and processing time. Reconfigurable arrays provide an efficient platform for mapping a number of these possible implementations and switch between them dynamically depending on overall system requirements depending on overall system requirements.

There are two types of DWT; for example 5/3 for loss-less image compression and 9/7 which is not loss-less. Both, 5/3 and 9/7 DWT computations can be computed through different algorithms which are based on either convolution, lifting or integer based DWT [4][10]. Each has their own merits and limitation depending upon quality and application.

In this paper, we present a number of different DWT new reconfigurable array architectures. The architectures are novel in the sense that currently there are no reconfigurable fabrics which can efficiently implement different DWT algorithms such as Lifting and Integer based DWT etc. The implementations have different characteristics in terms of array usage (Area usage) and power consumption which are explored in this paper. The paper demonstrates the ability of the new reconfigurable System-on-Chip to support these implementations which are well suited for high-flexibility demanding applications such as JPEG-

Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06)
0-7695-2614-4/06 $20.00 © 2006 **IEEE**

2000. The novelty of the architecture is that the reconfigurable arrays are flexible enough to incorporate different DWT algorithms efficiently yet high performance with approximately 20% reduction in power consumption than previously introduced work by the authors [1][2].

II. RECONFIGURABLE SoC PLATFORM AND DOMAIN-SPECIFIC ARRAYS

The proposed overall system contains processors and digital Signal Processors (DSP) along-with a number of embedded reconfigurable arrays (RAs) as shown in Figure-1. Each RA can be specific to a particular computation such as DWT etc. The system also allows a provision of a combined array that may target multiple computations. The RA can be easily incorporated in to SoC design-flow as a synthesizable soft-core.

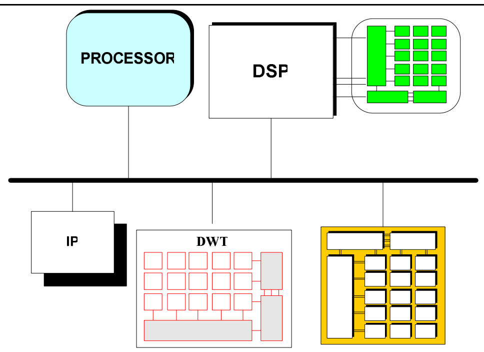

Figure 1: Reconfigurable System-on-Chip [1]

The RA itself consists of programmable clusters which are interconnected through configurable switches [1]. The clusters define the functionality of the array. Each RA is heterogeneous and contains different types of clusters, with each cluster specific to one operation. The clusters that constitute the RA can be chosen at design-time according to requirements and constraints. The domain and the degree of flexibility for a RA can be set through the choice of the clusters and interconnects used. The RA is provided as a soft-core which can be simulated, synthesized and routed as a normal ASIC core. This allows RAs to be incorporated into design flow of the full SoC, making the design and verification of the system easier.

In addition, the RA can be configured by a processor or a DSP. This can be done dynamically to allow the adjustment of the RA's operation at run-time. The data from and to the array is read and fed by the processor through on-chip bus such as AMBA bus.

A reconfigurable array computes the 1-D DWT. The output of DWT array is stored in the main memory and then transposed to be fed into the same RA to get the 2-D DWT output. The reconfigurable array is reused in order to get the 2-D DWT output. Figure-2 illustrates the process. The controller can be a dedicated hardware or can be implemented in software.

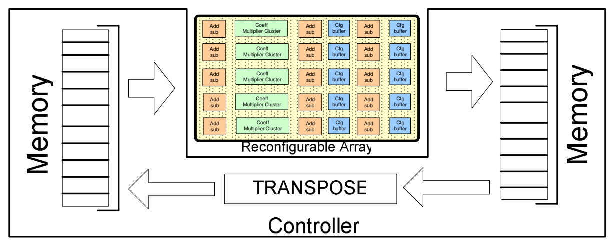

Figure 2: 2-D DWT Mechanism

III. RECONFIGURABLE ARRAYS FOR DWT

The clusters for the reconfigurable arrays targeting DWT computations were designed for different DWT algorithms including 5/3 and 9/7, with varying performance, speed and area requirements. The following clusters were identified as common reusable blocks and arranged in an array as shown in Figure-3: the clusters are designed with the consideration that the RA can be used fully/partly for any general purpose computations when it is not being used in DWT domain. This feature makes it quite attractive choice for handheld multimedia applications.

A. Add-subtract Cluster

The add-subtract cluster can be configured as:

i) Parallel, digit-serial or bit-serial adder/subtractor
ii) Performs A-B and B-A operation

Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06)
0-7695-2614-4/06 $20.00 © 2006 IEEE

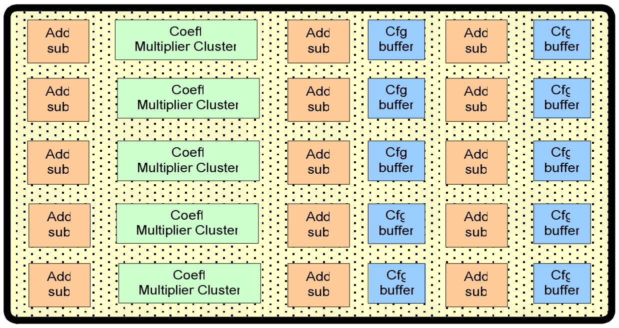

Figure 3: Reconfigurable Array with Cluster for DWT Computations.

The basic module is 8-bits wide, three modules are grouped into a cluster and configurable switches are provided between them to support cascading to get wider bit ranges (up to 24-bits). Even wider bit ranges are possible for different operations by cascading multiple clusters through a mesh interconnect. The different configurations of Add-subtract clusters can be selected through configuration bits reserved for the cluster. The precision of the calculation can be selected by configuring the cluster depending upon how many modules are required (basic module is 8-bits wide) 16 bits are required for the loss-less calculation based upon 5/3 which can be achieved by incorporating two basic modules through configuration switches. Flexibility of selecting the bit-widths, makes the architecture more versatile. Moreover the operation of the cluster can also be reconfigured through the configuration bits which allow the complete reusability of the cluster and gives more flexibility to incorporate different types of DWT algorithms for implementation.

B. Coefficient Multiplier Cluster

Filter coefficients with in the DWT [4][10] are multiplied through reconfigurable coefficient multiplier clusters. The floating point coefficients (for 9/7 lifting based DWT) are implemented through canonical-signed-digit (CSD) form [9].

A cluster has programmable shifter blocks, adder blocks and a multiplexer to accommodate a wide range of coefficients. The cluster can handle up to 24-bit operation to facilitate required precision. Filter coefficients for 9/7 lifting based DWT are incorporated through CSD and explained through the Table-1.

Table-1 9/7 Filter coefficients in CSD form

	Value	12-Bit CSD Representation
Alpha	1.586134342	$2^{1}-2^{-1}+2^{-3}-2^{-5}-2^{-7}+2^{-12}=$ 1.5861816
Beta	0.052980118	$2^{-4}-2^{-7}-2^{-9}+2^{-12}=$ 0.0529785
Gamma	0.882911076	$2^{0}-2^{-3}+2^{-7}=$ 0.8828125
Delta	0.443506852	$2^{-1}-2^{-4}+2^{-7}-2^{-9}+2^{-12}=$ 0.4436035

Figure-4 illustrates the internal structure of the coefficient multiplier cluster. The cluster has three internal sub-module. These are

- Cfg-Shifter Module
- add-Sub Module
- Multiplexer Module

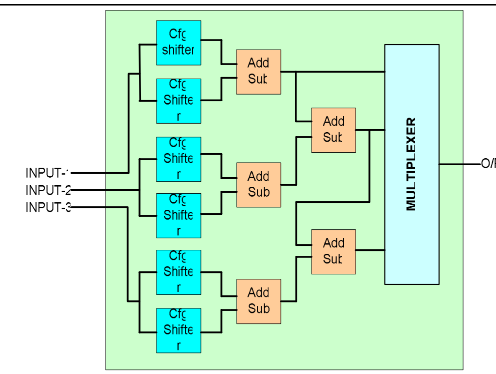

Figure-4 Coefficient Multiplier Cluster

Cfg-Shifter performs the multiplication and division depending upon the algorithm. It can be configured to multiply or divide by any even integer value between 2 and 32. add-sub is explained in section IIIa. The multiplexer is configured to select one of its inputs based upon the DWT algorithm. All internal modules are interconnected through programmable switches to incorporate different multiplications and divisions.

C. Configurable Buffer Cluster

The cluster can be programmed for different bit-widths. It has normalizing functionality as well which can be configured depending upon the DWT filter type.

The elements of the array are interconnected through symmetrical configurable switches. Sixteen 4-bit tracks and sixteen-1bit tracks are provided for both data and control lines in contrast to the previously introduced RA [1][2]. Connection boxes (C-Boxes) connect the pins of the cluster to the tracks, and the switch boxes (S-Boxes) connect together the intersection of the tracks [11]. The connection of clusters with tracks is illustrated in Figure-5.

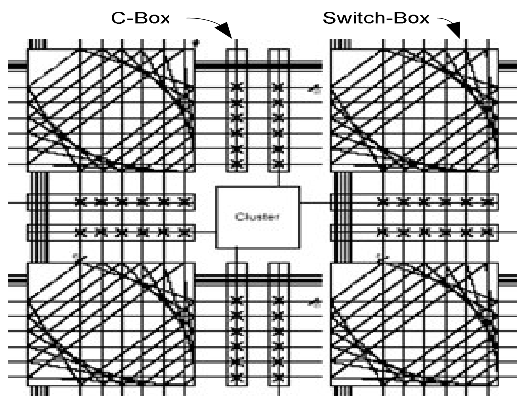

Figure-5 Cluster with S-Boxes and C-Boxes

The different clusters are arranged in the array to keep the configuration bits and channel track width minimum while maintaining the flexibility of the newly designed array.

Two types of DWT algorithms for 5/3 and 9/7 were implemented on our proposed array as discussed in later sections. The hardware/cluster utilisation of different DWT algorithms is presented in table-1.

The RA has been specially tailored in terms of size due to the enhanced functionality of the clusters. This helps in reducing the overall power consumption of the improved customized domain-specific array. Results are included in later section.

The choice of different cluster's position in the array is inspired by the study of different DWT algorithms and their implementations, for example; normalization factor or previously calculated value is required at the end of one calculation cycle and that's a why configuration buffer is placed at the end of each row of the array (Figure-3). Addition/subtraction operations are required frequently in computations and due to this fact; the cluster is distributed all over the array.

Table-1 Comparison of DWT algorithms based upon utilization of proposed array

	Cfig Coeffi-Multiplier	Cfig Buffer Cluster	Cig Add-Subtract Cluster
DWT Architecture [20] Implemented on			
5/3 lifting based DWT	2	1	5
9/7 lifting based DWT	2	1	5
Lian etal [13]			
5/3 lifting based DWT	2	8	4
9/7 lifting based DWT	2	8	4
Lian etal [13]			
5/3 lifting based DWT	5	2	8
9/7 lifting based DWT	5	2	8

IV. DWT IMPLEMNTATIONS

A. Implementation-1: 5/3 and 9/7 DWT

5/3 lifting based DWT has many advantages over other DWT algorithms. The 5/3 helps to achieve lossless image compression and has short filter length for both low-pass and high-pass filters as compared to other JPEG2000 specified DWT filters i-e Daubechies 9/7 filter [9]. A 5/3 Filter has only one set of lifting step compared to 9/7, which has two.

5/3 lifting based DWT, are computed through Equation-1 [3].Y(2n) is even and Y(2n+1) is an odd term [6].

$$\tilde{L}P : Y(2n) = X_{ext}(2n) + \left[\frac{Y_{ext}(2n-1) + Y_{ext}(2n+1) + 2}{4} \right] \quad (1)$$
$$\tilde{H}P : Y(2n+1) = X_{ext}(2n+1) - \left[\frac{X_{ext}(2n) + X_{ext}(2n+2)}{2} \right]$$

The implementation is efficient and quite unique as it requires a few numbers of elements as compared to [13]. The implementation takes three inputs in a fashion shown in the Figure-6.

The implementation [2] has 100% reusability in terms of implementing 5/3 and 9/7 lifting based DWT. The proposed implementation uses only one buffer and gives output/DWT coefficients on every clock cycle while [7][9] requires 6 pipeline and 4 pipeline registers and initially requires more than one clock cycle to give the first output.

This hardware realization as shown in Figure-6, was used to bench mark performance of our proposed array against standard FPGAs. 5/3 and 9/7 lifting based DWT algorithms were implemented on this architecture. Result when Image of Lenna 512x512 is processed through proposed Architecture (array configured in 5/3 DWT mode) is shown in Fig-7. The output from the proposed architecture was processed through MATLAB and original picture was reconstructed as shown in Figure-8.

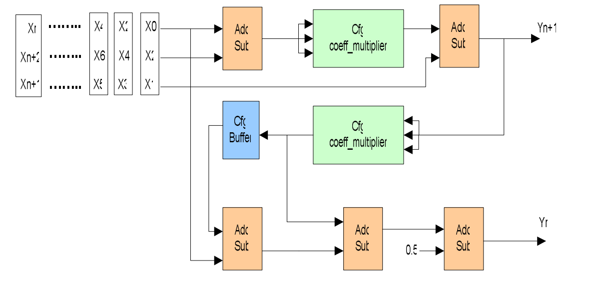

Figure 6: Realization of DWT on Reconfigurable Array

The coefficient multiplier cluster is configured in such a way that it performs the required coefficient multiplication.

Figure 7: DWT processing through Proposed Reconfigurable Array

The unused modules in the proposed array are kept ‘off ‘with the help of configuration bits. This helps to reduce the overall power consumption. The important feature of this implementation is that the configuration for the RA is same for the 5/3 and 9/7 lifting based DWT. This allows the array to be dynamically be reconfigured for both 5/3 and 9/7.

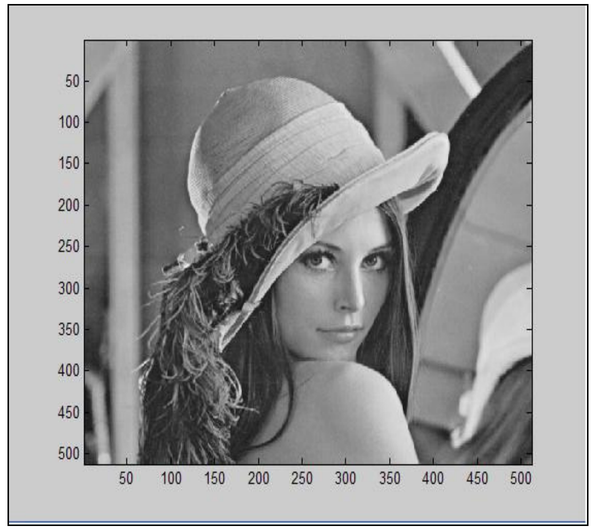

Figure 8: Reconstructed Image of Lena

B. DWT Implementation-2

Reference-[7][9] introduced architecture for implementation of 5/3 and 9/7 lifting based DWT. The algorithm uses multipliers to compute DWT. The architecture [7][9] is efficient in terms of reusability but not efficient in terms of power consumption as it employs multipliers. The authors implemented the algorithm on the newly designed arrays by using canonic-signed-digit [9] which is a technique, based on shifts and additions. This along with the array architecture helps to achieve 65% power saving over reference [7][9] architecture. Hardware realization of [7][9] in terms of our RA is shown in Figure-9.

C. DWT Implementation-3 : Integer Fast DWT

Integer based DWT is an efficient approach that is based on computing power-of-two wavelet coefficients which are derived directly from the roots of the half-band filter [8].

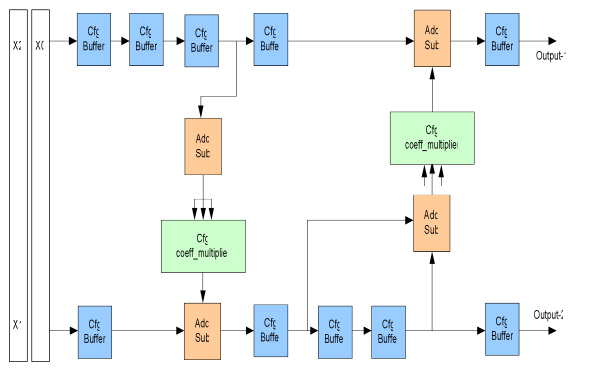

Figure 9: Realization of DWT Architecture [13] in terms of Reconfigurable Array

This technique is quite efficient and fast as it does not use any multipliers but it can only compute 9/7 DWT. The DWT is computed through following equation.

$$A = N*C*S*L*I^T \quad (2)$$

'A' denotes the outputs of two channels filter bank system and other definitions of different components of equation-2 are:

$$A = \begin{bmatrix} a_0(n) \\ a_1(n) \end{bmatrix} \quad C = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad N = \begin{bmatrix} 2^{-8} & 0 \\ 0 & 2^{-7} \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad S = \begin{bmatrix} s_4 \\ s_3 \\ s_2 \\ s_1 \\ s_0 \end{bmatrix} = \begin{bmatrix} 0 & -2^2 & 0 & 2^2 & 2^4 \\ 0 & 0 & 0 & 2^6 & 2^7 \\ 2^3 & 0 & -2^4 & 0 & 2^3 \\ -1 & 0 & -2^2 & 0 & 2 \\ 0 & -2^3 & -2^3 & 2^3 & 0 \end{bmatrix}$$

$$I = \begin{bmatrix} x(n-4) & . & . & x(n-1) & x(n) & . & . & x(n+4) \end{bmatrix}$$

The new customized array is flexible enough to incorporate even Integer Fast DWT as well and implementation of the algorithm in terms of the proposed array is shown in Figure-10:

V. COMPARISON OF IMPLEMNTATIONS

All implementations are carried out on commercial Xilinx Vitex-E FPGA [4] and on our proposed RAs. The performance in terms of overall power consumption and maximum operating frequency is shown in Tables-2, 3 and 4. All these systems use 0.18µm CMOS technology and run at 1.8V. The values are measured for single frame of the Lenna's image 128x128.

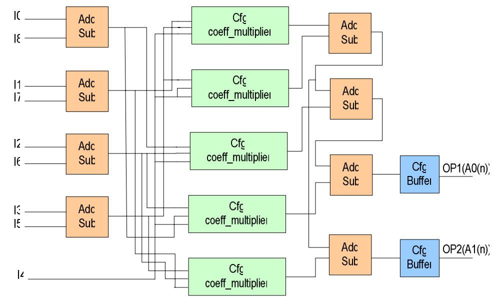

Figure 10: 9/7 Integer Fast DWT Implementation on Reconfigurable Array

When modules and clusters are un-configured and if they have no activity at their inputs, they exhibit only static power consumption and no dynamic power. In the case of un-configured C-boxes, some switching power is dissipated when the outputs of the cluster connected to the C-box are switching.

Figure-11 shows the area overhead used to make the hardware reconfigurable. The add-subtract cluster occupies only 6% of the total area while the C- and S-boxes occupy 50% and 44% respectively. As can be seen from the graph these area values include the area occupied by the configuration registers, which represents a large percentage of the area of the boxes. The total area can be reduced considerably if the flexibility of the C- and S-boxes is reduced.

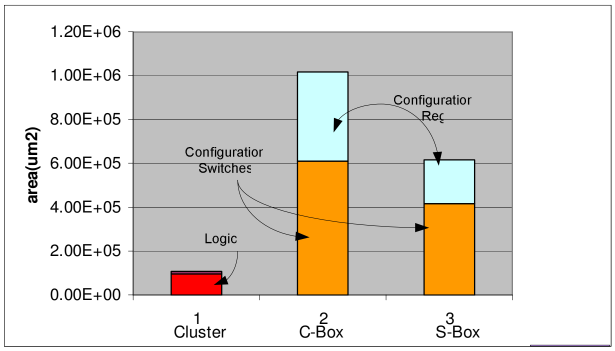

Figure-11 Area Distribution of Add-Subtract cluster of the Reconfigurable Array

Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06)
0-7695-2614-4/06 $20.00 © 2006 IEEE

It is quite evident from the results that the proposed array is efficient in speed and power consumption for all the DWT implementations. This is because the clusters are customized for one type of computations as compared to generic small clusters in case of generic FPGAs. The customization of clusters for one domain and arrangement of clusters, help to reduce the number of clusters required for the desired operation; C-boxes and S-Boxes (which are main cause of power consumption [1]).

The proposed array incorporates multiplication through a number of addition and shifting operations [9] which help to achieve further power savings. When compared with previously introduced architecture by the authors [1][2].

The improved array is approximately 20% more power efficient due to improved cluster design and by reduction in s–boxes and c-boxes due to efficient cluster placement in RA. The proposed reconfigurable arrays are approximately 70% power efficient and on average 50% speed efficient in all the above mentioned implementations of DWT.

Table-2 Performance Evaluation of Implementation-2

IMPLEMENTATION - 1	Power Consumption (mw)	Max Frequency (MHz)
Xilinx Virtex-E		
5/3 lifting based DWT	9.98	98
9/7 lifting based DWT	21.48	73
Proposed RA		
5/3 lifting based DWT	4.96	123
9/7 lifting based DWT	9.38	123

Table-3 Performance Evaluation of Implementation-3

IMPLEMENTATION – 2	Power Consumption (mw)	Max Frequency (MHz)
Xilinx Virtex-E		
5/3 lifting based DWT	23.10	83
9/7 lifting based DWT	40.68	66
Proposed RA		
5/3 lifting based DWT	11.64	110
9/7 lifting based DWT	20.68	110

Table-4 Performance Evaluation of Implementation-4

IMPLEMENTATION - 3	Power Consumption (mw)	Max Frequency (MHz)
Xilinx Virtex-E		
9/7 lifting based DWT	12.5	73
Proposed RA		
9/7 lifting based DWT	7.02	123

VI. CONCLUSION

This paper has described a number of new customized domain-specific reconfigurable arrays for the implementation of different DWT algorithms. The implementations have different advantages in terms of power consumption and time needed to complete these computations as well as quality and precision of the output. The results demonstrate the flexibility provided by the arrays in allowing the mapping of a range of different DWT algorithms and maintaining performance advantage.

The new flexible arrays provide a better alternate to generic FPGA solutions for complex algorithms which are continuously updated, such as JPEG-2000, within a SoC platform.

REFERENCES

1. Baloch, S.; Ahmed, I.; Arslan, T.; Stoica, A.; “Low power domain-specific reconfigurable array for discrete wavelet transforms targeting multimedia applications” Field Programmable Logic and Applications, 2005. International Conference Aug. 24-26, 2005 Page(s):618 - 621

2: Baloch, S.; Ahmed, I.; Arslan, T.; “Domain-specific reconfigurable array targeting discrete wavelet transform for system-on-chip applications” Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International conference 4-8 April 2005 Page(s):4 pp.

3. ISO/IEC15444-1,”An information Technology-JPEG-2000 image coding system-Part 1: Core design-System http://www.jpeg.org/JPEG2000.html

4. I. Daubechies and W. Sweldens, ”Factoring wavelet transform into lifting steps, journal of Fourier analysis and applications, vol. 4, pp. 247-269, 1998.

5. Xilinx, The Programmable Logic Data Book¸ Xilinx Inc., 2001

6. Tan, K.C.B.; Arslan, T.;” Low power embedded extension algorithm for lifting based Discrete wavelet transform in JPEG2000” Electronics Letters , Volume: 37

7. C-J Lian, K-F Chen, H-H Chen and L-G Chen, “Lifting based discrete wavelet transform architecture for JPEG-2000”, IEEE international symposium on circuits and systems, 2001

8. Dang, P.P.; Chau, P.M.; “Integer fast wavelet transform and its VLSI implementation for low power applications” Signal Processing Systems, 2002. (SIPS'02). IEEE Workshop on, 16-18 Oct. 2002 Pages:93 – 98

9. C-J Lian, C. Chakrabarti and T Acharya, “A VLSI architecture for lifting based forward and inverse wavelet transforms”, IEEE transaction on signal processing, Vol. 50

10 M. Anthoni, M Barlaud, P Mathieu, I Daubechies, ”Image coding using wavelet transform”, IEEE transaction on image processing, Col. 1, 1992

11 Rose J., Brown S., “Flexibility of interconnection structures for field-programmable gate arrays”, Solid-State Circuits, IEEE Journal of , Vol.26, Iss.3, 1990, Pgs: 277- 282