

Reconfigurable co-processor for high performance Discrete Wavelet Transform

Kalyan Mohanta

Department of CSE

B P Poddar Institute of Management and Technology

Kolkata, India

kalyanmohanta@yahoo.com

Abstract—Wavelet transforms have proven to be useful tool for several signal processing applications, including image and video compressions, image segmentation, speech synthesis and telecommunication. With the continuous increase in the use of internet and wireless devices, wavelet transform become more popular over traditional Fourier and cosine transforms in DSP applications especially for embedded multimedia applications. Designers are trying to develop more computation and energy-efficient VLSI architectures for discrete wavelet transform (DWT) so that it can be mapped into application specific DSP processors or into FPGA based reconfigurable coprocessors for embedded electronic devices. Several VLSI architectures have been proposed for computing 1-D and 2-D DWT which range from SIMD arrays to folded architectures such as systolic arrays and parallel filters. This paper proposes an efficient architecture for DWT computation based on systolic array model and its configuration as IP core in FPGA based reconfigurable coprocessors. The proposed architecture may be applicable for computation intensive DSP applications for mobile devices.

Keywords-Discrete Wavelet Transform (DWT), Systolic Array Architecture, Field Programmable Gate Arrays (FPGA)

I. INTRODUCTION

Discrete Wavelet Transform (DWT) becoming increasingly popular due to its computational simplicity and its usability in a variety of signal processing applications. DWT becoming more popular over traditional fourier and cosine transforms (DFT, DCT) in DSP applications mainly because of its capability of multi-resolution signal analysis with localization in both time and frequency and better performance factors. DWT can be viewed as a multi resolution decomposition of a signal into its components in different frequency bands and analyzing each component with a resolution matched to its scale. There are several proposed VLSI architectures for DWT which range from SIMD arrays to folded architectures such as systolic arrays and parallel filters. Present application trends of DWT in consumer electronics and real time embedded devices (for image, speech and video processing) demands optimized architectures both in terms of computational time and silicon area needed.

FPGA technology already proved that it has the potential of design high performance embedded systems at low cost and less time to market and can be reconfigured for adopting new refined protocol or design specification. Electronic devices using reconfigurable IP core also provides flexibility to run multiple data and operation intensive applications simultaneously through dynamic reconfiguration of the core. The present work is motivated by current trend of wavelet transform applications in smaller bandwidth signal (image, speech, and video) processing and transmission for low power embedded devices. In this paper, the major objective is to provide an efficient architecture for DWT computation based on systolic array model and how it can be configured as which can be used as IP cores for FPGA based embedded signal processing devices.

The rest of the paper is organized as follows. In section II Discrete Wavelet Transform (DWT) theory is described briefly. Section III describes different existing architectures for DWT computation or related works. The overall architecture is presented in section IV. The analyses of the results and its performance have been also studied in this section. Section V presents the methodology for mapping the architecture into FPGA based reconfigurable systems. Finally the conclusion is drawn in section VI.

II. DISCRETE WAVELET TRANSFORM

Discrete Wavelet Transform (DWT) [17] [1], decomposes a finite time domain input signal into its components in different frequency bands (usually in high frequency and low frequency components) by applying a set of orthogonal basis functions which are defined by a recursive difference equation (1) [2] [10] [19].

$$\varphi(x) = \sum_{k=0}^{M-1} C_k \varphi(2x - k) \tag{1}$$

The range of the summation is determined by the specified number of nonzero coefficients M. The value of the coefficients determined by constraints of orthogonality and normalization in (2) [2] [10].

$$\int \varphi(x)\varphi(x - k) dx = 0 \tag{2}$$

Basically, in 1-D DWT [18] [20], the input signal is passed through a series of high pass filters to analyze high frequencies and through a series of low pass filters to analyze low frequencies. Filters of different cutoff frequencies are used to analyze the signal at different resolutions. Let x[n] be the original signal, it is first passed through a half-band high pass filter g[n] and a low pass filter h[n]. After the filtering one half of the samples can be eliminated according to the Nyquist's sampling theorem. The signal can therefore be sub sampled by 2, simply by discarding every second sample. This constitutes one level of decomposition and can be mathematically expressed as follows:

$$y_{high}[k] = \sum_n x[k]g[2k - n] \tag{3}$$

$$y_{low}[k] = \sum_n x[k]h[2n - k] \tag{4}$$

Where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the high pass and low pass filters respectively after sub sampling by 2. The above process can be repeated for further decomposition. The outputs of high pass and low pass filters are called DWT coefficients and these coefficients can be processed in reverse order to reconstruct the original image or signal which is called inverse discrete wavelet transform (IDWT) [17]. The signals at every level are up sampled by two, passed through the synthesis filters $g^*[n]$ and $h^*[n]$ (high pass and low pass respectively), and then added. The analysis and synthesis filters are identical to each other, except for a time reversal. Therefore, for each layer the reconstruction formula becomes

$$x[n] = \sum_n (y_{high}[k]g^*[-n + 2k] + y_{low}[k]h^*[-n + 2k]) \tag{5}$$

To ensure the above IDWT and DWT relationship the following orthogonality condition on the filters $H(\omega)$ and $G(\omega)$ must hold:

$$|H(\omega)|^2 + |G(\omega)|^2 = 1 \tag{6}$$

Where $H(\omega) = \sum_n h[n]e^{-jn\omega}$ and $G(\omega) = \sum_n g[n]e^{-jn\omega}$. An example of such $H(\omega)$ and $G(\omega)$ can be shown as $H(\omega) = \frac{1}{2} + \frac{1}{2}e^{-j\omega}$ and $G(\omega) = \frac{1}{2} - \frac{1}{2}e^{-j\omega}$

This is also known as the Haar wavelet filters. The process of computing 1-D DWT by successive low pass and high pass filtering of the discrete time domain signal is known as Mallat's Pyramid algorithm [1] [19] (based on octave band decomposition of signal frequency). The corresponding wavelet decomposition tree for an input data sequence x[n] is shown in Fig 1.

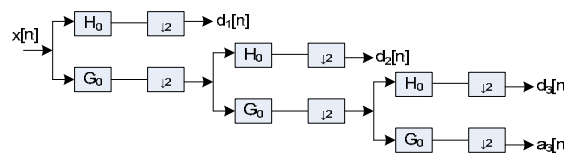


Figure 1. Three level wavelet pyramid (n=8) where G0 and H0 denotes low and high pass filters

Here, at each level, the high pass filter produces detail information d[n], while the low pass filter associated with scaling function produces coarse approximations, a[n]. At each level of decomposition, the half-band filters produce signals spanning only half the frequency band which doubles the frequency resolution as the uncertainty in frequency is reduced by half. DWT based Image analysis uses a short window at high frequency data and a long window at low spatial frequency data which makes DWT more accurate in analyzing image signals in

different spatial frequency and thus can represent more accurately both smooth and dynamic regions of an image. This is the main advantage of DWT over DCT [20] [21] which follows block based decomposition of signals. In the next section describes different existing architectures for 1-D DWT computation.

III. RELATED WORKS

Previous works attempted to implement the DWT architectures where the computation follows standard Mallat's algorithm [1] [3]. Mainly those architectures are based on SIMD model and a few based on systolic array model. SIMD architectures implement filter bank structure based on the recursive application of the two channel sub-band decomposition. The filter bank structure for data size N is implemented on that type of architecture consisting of a linear array of N processors with reconfigurable interconnection. For the first octave computation, all N processors are active. For the second octave computation, the N processor array is reconfigured to a $N/2$ processor array with processors $P(2^j)$ being octave, $0 \leq j \leq \lfloor N/2 \rfloor$. In general, for the m th octave computation, the N processor array is reconfigured to an array of size $N/2^{m-1}$ with processors $P(2^{m-1}j)$ being octave, where $0 \leq j \leq \lfloor N/2^{m-1} \rfloor$. All the outputs of an octave are computed at the same time. The weights are broadcast and the partial results move from one processor to another. Since the time to compute an output is L (levels), the computation time for computing J octaves is LJ and the period is also LJ . The period can be reduced to L in a multi-grid architecture of size $2N-1$ (There are $N/2^i$ processors in level i , $(0 \leq i \leq \log_2 N)$). So, SIMD based architectures to implement 1D DWT are optimal with respect to time as the array size is proportional to the size of data sequence but the interconnection network is complex for the implementation which increases the silicon area needed hence inefficient to map it into reconfigurable architectures. Chakraborty and Viswanathan [4] presented several architectures implementing 1-D and 2-D DWT based on SIMD model.

Systolic array based architectures implement DWT filtering process by incorporating an array of processing elements (PE) each of which stores the filter co-efficients hence achieving high data locality which removes the need of communicating the co-efficients through the array.

Each PE has MAC computation capabilities, reads from its own column memory, and writes to the column memory two PEs to the right. There is a single Control Unit which controls the rightmost processing element through a series of control signals. Since each PE performs the same calculations as its right neighbor one clock cycle behind it, these control signals can propagate from the rightmost PE leftwards through the systolic array.

Here, with decimation, it is not required to perform every second calculation of each filtering stage which reduces the computational complexity by a factor of 2 in each stage. So, this architecture has the advantages in terms of silicon area needed as single chip implementation can be done but requires large computational time as involvement of control unit need to set wavelet filter coefficients in memory and decimated data need to be recomputed. Grazeszczak et al. [3] proposed a systolic array for computing 1-D DWT. Pan and Park [5] proposed a modified DWT architecture based on systolic array model. Souni, Abid and Tourki [14] presented a 1-D DWT architecture using parallel filters. Wu and Chen [15] proposed a 2-D architecture that employs folding techniques.

IV. PROPOSED ARCHITECTURE

Through the above survey, we have discussed standard methods of DWT computation along with its existing architectural support and the performance factors. In this portion, we have proposed a unified efficient architecture for 1-D DWT computation which is based on multiple systolic arrays. First, the 1-D DWT computation process has been described. Basically, in 1-D DWT, N element input signal vector is multiplied/convoluted by the transform matrix of size $N \times N$ (which is pre-defined for different standards i.e. Haar transform, Daubchie's transform) to get $N/2$ wavelet coefficients and $N/2$ averages. Next phase uses $N/2 \times N/2$ transform matrix containing only averages to generate $N/4$ wavelet coefficients and $N/4$ average from previous $N/2$ wavelet co-efficients. This process continues for $\log_2 N$ times to calculate the 1-D transformed vector. This process is described in example 1.

Example 1: Consider 4 element input signal vector $S[i]$. Thus a 4×4 transform matrix is needed which generates a new vector consist of two wavelet coefficients (c_0, c_1) and two average values (a_0, a_1). At the next level of computation those average values are used for further transformation.

$$\begin{bmatrix} t_{00} & t_{01} & t_{02} & t_{03} \\ t_{10} & t_{11} & t_{12} & t_{13} \\ t_{20} & t_{21} & t_{22} & t_{23} \\ t_{30} & t_{31} & t_{32} & t_{33} \end{bmatrix} \cdot \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \\ a_1 \\ c_1 \end{bmatrix} \Rightarrow \begin{bmatrix} a_0 \\ a_1 \\ c_0 \\ c_1 \end{bmatrix} \quad (7a)$$

$$\begin{bmatrix} t_{00} & t_{01} \\ t_{10} & t_{11} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} c_2 \\ a_2 \end{bmatrix} \quad (7b)$$

In the proposed architectural construct for the 1-D DWT computation, a set of systolic arrays is used which improves performance of computation and also reduces hardware complexity. Architecture consists of set of processing elements (PEs), data selector (MUX), delay elements (Z-1s) and output buffer.

A. Processing Element Structure

Each Processing Element (PE) has the capability of MAC (multiply and accumulation) computation, storing of result in internal register and of routing data to the next PE and to next level data selector. For data routing, each PE has four I/O lines. Elements of each row vector of transform matrix are fed through the top input and the signal vector through the left input line to each PE serially. Each PE performs the MAC computation and transmits result either to next level data selector through bottom output or to output buffer through bottom output according to the routing logic. Through the right output line, each PE passes the signal vector elements to its right neighbor delayed by one clock. All the operations data processing and sifting through different PE is controlled by a synchronized clock.

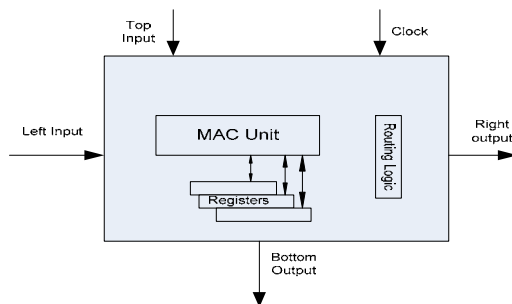


Figure. 1. Processing Element (PE) Structure

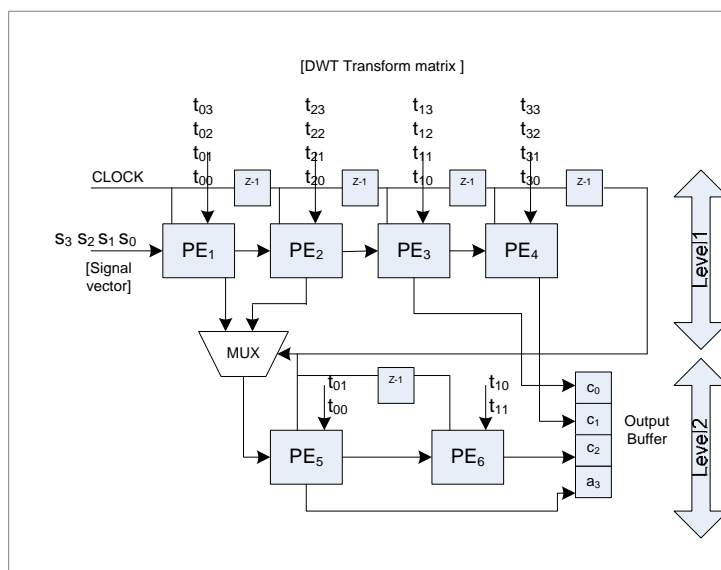


Figure 2. Proposed Architecture for 1-D DWT Computation

B. DWT Architecture (4- point 1-D DWT computation)

The proposed architecture for 1-D DWT computation is shown in figure 5. Here, the transform matrix is divided row-wise and passed to the PEs in parallel through top input lines. Signal vector is passed to each PE through left input lines and propagated serially by one clock delay. Delay elements are used because, the computation starts in each PE delayed by one clock to previous PE. Finally, the output buffer holds the transformed data (represents the frequency components of the signal and image in terms of wavelet co-efficients and averages). Architecture can be generalized for N point 1-D DWT computation by incorporating $\log_2 N$ levels and corresponding hardware. The signal flow graph (SFG) and processing states for 4-point 1-D DWT computation is described in Fig. 6 and Fig. 7 respectively.

C. Result and Performance Analysis

The simulation of the proposed architecture has been carried out in Xilinx Modelsim [23] [26] simulator using Verilog HDL [24][25] and test the results along with performance parameters. One such experimental result is shown in example 2. In the simulation process, initially the behavioral models of basic computational blocks (MAC module, Data Selector and Delay elements) are defined; then the routing network for data processing and forwarding in main computation module has been implemented. The synchronization is achieved by system clock. The computational correctness and performance parameters with various set of test-bench have been tested.

Example 2: Considered a 4-point signal vector and 4x4 transform matrix as input.

Signal Vector: $S [0..3] = [8 \ 4 \ 4 \ 4]$

Transform Matrix:

$$\begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0.5 & -0.5 \end{bmatrix}$$

Output Vector : $c[0..3] = [2 \ 0 \ 1 \ 5]$, where $c[0], c[1], c[2]$ are computed wavelet coefficients and $c[3]$ which is basically average value (a2).

Simulation is done using 20ms clock. All the operations within a PE are performed at every positive edge of clock. Process start at simulation time 10ms, get the first wavelet coefficient at 70ms, computation completed at 130ms. So 120ms needed for computation which means 7 clock cycles required for 4-point DWT computation which incorporates 6 processing elements in two levels yielding 66.67% of hardware utilization. Similarly, we have also done the simulation for 8-point computation which requires 15 clock cycles incorporating 14 processing elements in four levels yielding 60% hardware utilization. Hardware utilization can be increased by interleaving the operations of alternate PE for large point computation or by doubling no of MAC module in each PE. The generalized results with computational time and hardware utilization are given next.

Hardware Complexity: 1-D Haar DWT computation requires $\log_2 N$ number of computational levels for N-element signal vector (where N is a power of 2) in the proposed architecture. Each level i , incorporates $\frac{N}{2^{i-1}}$

number of processing elements (PE). So total number of PE for N-point 1-D DWT computation equals

$$N \left(1 + \frac{1}{2} + \frac{1}{2^2} + \dots + \frac{1}{2^{i-1}} \right) = 2N - 2 \approx O(N) \quad (8)$$

[Where $i = \log_2 N$].

Time Complexity: For N element signal vector, number of computational level (L) = $\log_2 N$. In each level, computation is performed separately through individual Systolic array of PE's in a pipelined fashion. The last level consists of two PEs. The first level computation terminates at $(2N - 1)^{\text{th}}$ clock, and within that time all other computations are completed by lower level processing elements. So total number of clock required for N-point 1-D DWT computation equals

$$N + (N - 1) = 2N - 1 \approx O(N) \quad (9)$$

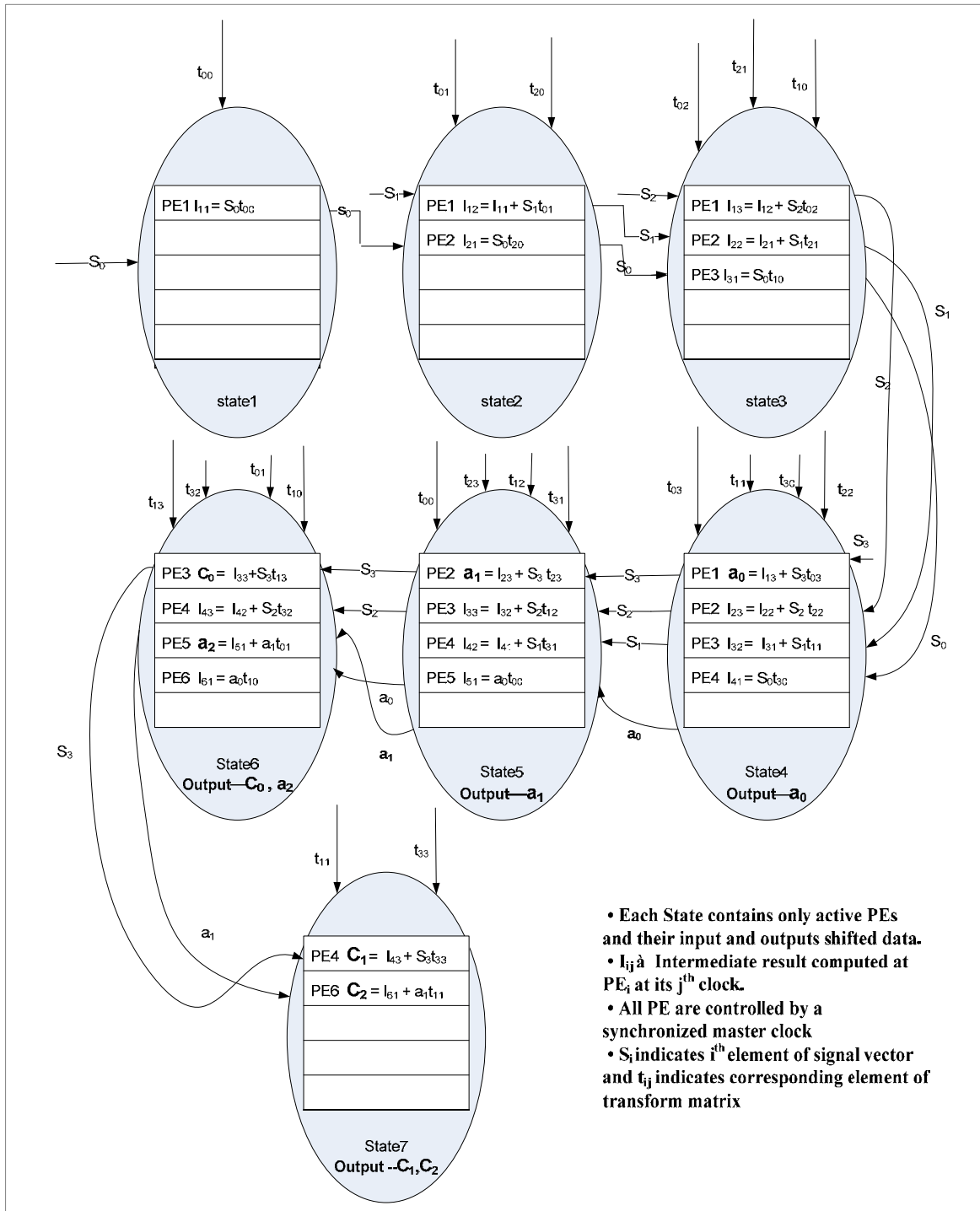


Figure 3. Signal Flow Graph of 4-point 1-D DWT Computation

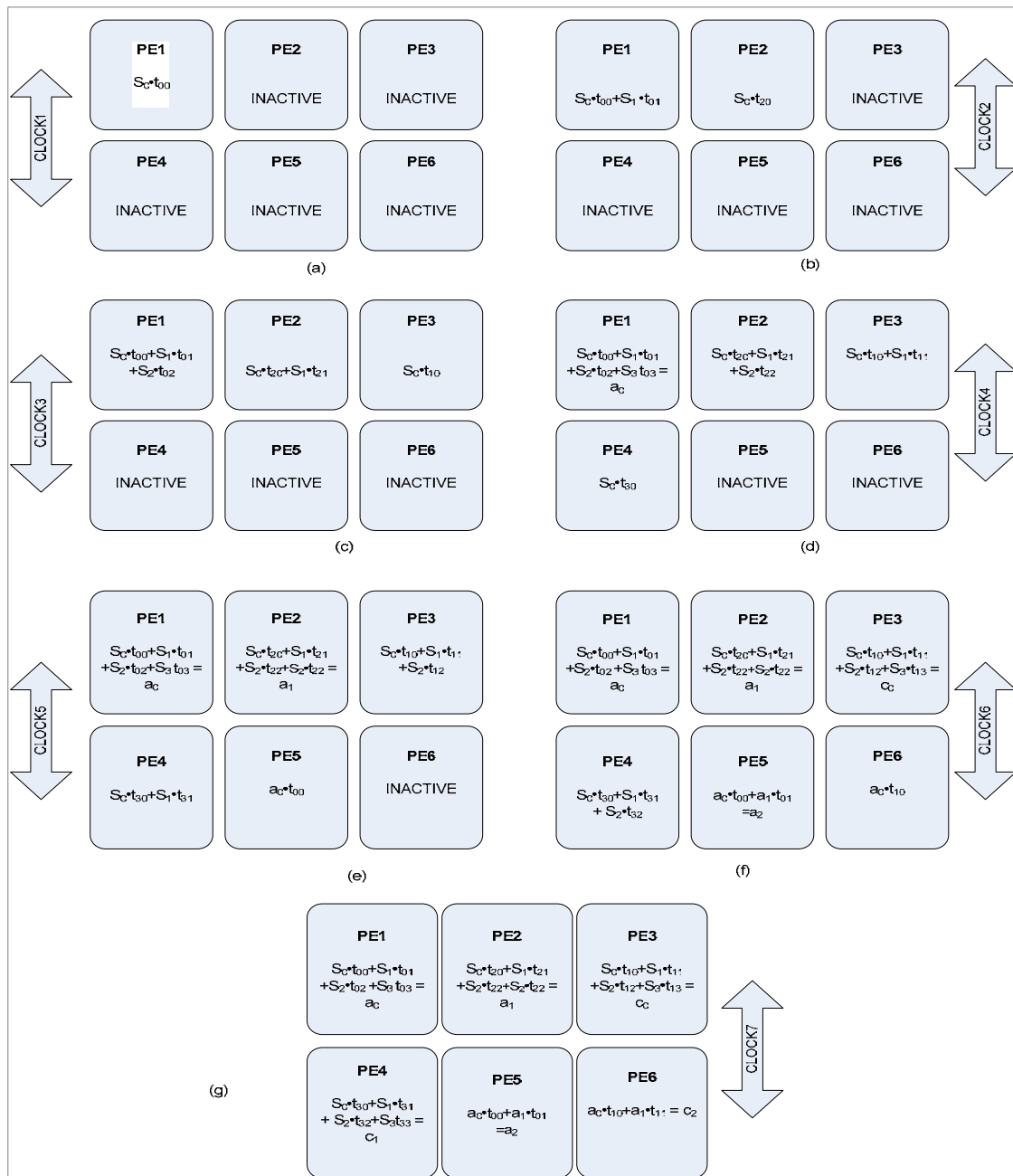


Figure 4. Processing States of 4-point 1-D DWT Computation

V. DWT ARCHITECTURE AS RECONFIGURABLE CO-PROCESSOR

Complete software implementation of the DWT appears to be performance bottleneck in real time systems where energy, space and time deadline are major factors. Therefore, hardware acceleration of the DWT has become an important issue. Structurally programmable circuits like FPGA offer attractive solution by providing a rapid design and prototyping platform where new designs can be replaced by reconfiguration within seconds. The computing core of an FPGA consists, in general of a highly complex re-configurable matrix of logic IC, registers, RAM and routing resources [7] [8] [9]. These resources can be used for performing logical and arithmetic operations, for variable storage and transfer data between different parts of the system.

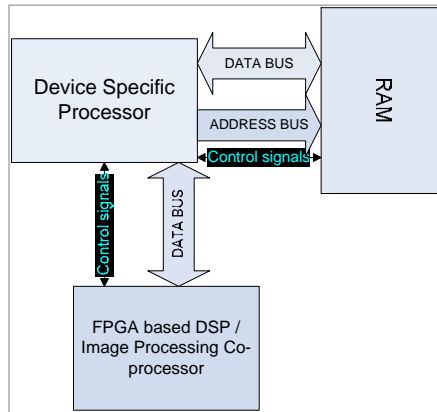


Figure 5. Reconfigurable Co-processor using Vertex FPGA

Further, FPGA based systems can be reprogrammed to achieve different functionalities without incurring non-engineering costs typically associated with custom IC fabrication hence can be configured as application specific co-processor in different real time systems.

The proposed DWT architecture can be implemented as a primitive module or as an image processing co-processor for application specific devices especially in hand-held portable embedded devices or multimedia wireless devices. It requires generating configuration bit stream from the HDL implementation of the architecture which defines the IP Core for the architecture. External RAM can be used to store the configuration bit stream which enables to configure a FPGA under the control of the processor in the device which will then perform as co-processor for dedicated DWT computations. Such device/system architecture is described in Fig. 8 where the device specific processor uses a Virtex FPGA as DSP or image processing co-processor for complex computation intensive DSP operations as for example DWT operations.

The main advantage of FPGA based implementation of the DWT architecture is, since there is no centralized control and sequential instructions to process in this type of implementation, typically thousands of instructions can be performed in parallel on an FPGA during one clock cycle which accelerates the computation.

Application specific IP Core is also upgradeable and reconfigurable for multiple applications by loading the modified configuration bit stream via USB or device specific I/O Port. Such a device architecture where FPGA is used with EPROM/FLASH as reconfigurable coprocessor needs to support SelectMap mode for loading configuration data into FPGA from EPROM with extra hardware support.

In FPGA based implementation there is no centralized control and sequential instructions to process, typically thousands of instructions can be performed in parallel during one clock cycle which accelerates the computation. Block diagram of DWT architecture as FPGA based reconfigurable co-processor with SelectMap mode support is shown in Fig. 7. Configuration and startup sequences of FPGA for performing DWT are controlled by specified control signals shown in Table 1.

TABLE I.

Control Signal & Data Bus	Description
CCLK	Clock input.
PROG	This input signal reset the internal configuration logic and re-initializes internal configuration memory of FPGA.
DONE	This output signal indicates the completion of configuration and the beginning of the startup sequence.
INIT	Hold off configuration initialization and indicate when CRC error occurs in configuration data.
WRITE	This Input signal must be asserted and held throughout the loading of data.
BUSY	This Output signal indicates whether the current byte is being loaded or ignored.
CS	Chip Select signal.
M2 M1 M0	Mode pins For SelectMAP configuration bits must be <110>.
D7...D0	8 bit bidirectional data bus.

Hence, using FPGA based architectures different complex DSP modules (i.e. DWT, DCT) can be fused as a co-processor in the system so that multiple functions can be performed by sharing common hardware resources extracted via similarity exploitation and tolerable routing complexity which also improves overall hardware utilization.

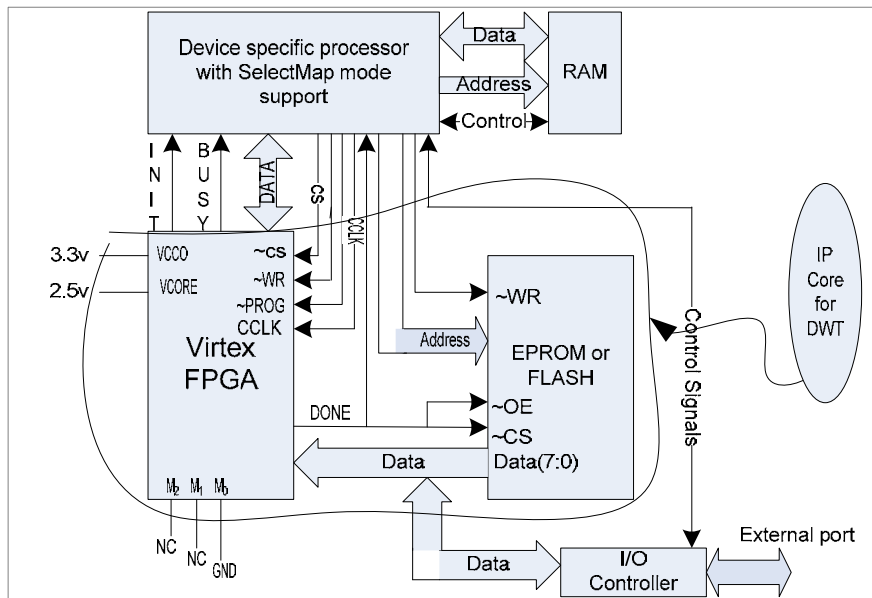


Figure 6. Reconfigurable Co-processor using Vertex FPGA in SelectMap mode

VI. CONCLUSION

This paper proposes a modified systolic array architecture designed to implement 1-D Haar DWT which proves its efficiency in terms of time and hardware requirements both of which are $O(N)$ for N point computation. The key feature of the architecture is scalability, practical I/O requirements, area requirement is independent of the input sequence length. A utilization of 60% can be achieved using this computation model as in first $N-1$ clocks, at least one PE is remaining inactive which can be increased by doubling the number of MAC units and adding extra registers in each PE or by interleaved computation. Systolic routing network for the proposed architecture seems to be practically attractive, due to its low component count and simplicity in design. A simplified model to map this architecture into FPGA based reconfigurable co-processor or DSP processors especially for embedded signal processing applications has also been presented. The future scope of work is to implement the architecture as an IP core for reconfigurable co-processor or embedded DSP processors by providing a set of libraries with a suite of primitive building blocks covering all indivisible components of the architecture so that it can be dynamically re-designed based on different signal processing application which can also help the application programmers. This architecture will be useful in computation intensive DSP applications (e.g., image compression), specially for low resource portable multimedia devices.

REFERENCES

- [1] S.G.Mallat, "A Theory for Multi-resolution signal decomposition: The wavelet representation", IEEE Trans. Pattern Anal. Mach. Intell. 11(7), pp 674-693, July 1989.
- [2] Daubechies, "The wavelet transform, time frequency, localization and signal analysis", IEEE Trans. Inform. Theory 36 (9), pp 961-1005, September 1990.
- [3] Grazeszczak, M.K.Mandal, S.Panchanathan and T. Yeap, "VLSI implementation of discrete wavelet transform", IEEE Trans, VLSI syst., vol.4, pp. 421-433, Dec. 1996.
- [4] C.Chakrabarty and M.Vishwanathan, "Efficient realizations of the discrete and continuous wavelet transform: From single chip Implementation to mapping on SIMD array computers", IEEE Trans. Signal processing, vol.43, pp. 759-769, Dec. 1996.
- [5] Sung Bum Pan and Rae-Hang Park, "Systolic Array architectures for Computation of the discrete wavelet transform", Journal of Visual Communication and Image Representation, Volume 14, Issue 3, pp 217-231, September 2003.
- [6] M.Nibouche, A.Bouridane and O.Nibouche, "Rapid prototyping of orthogonal Discrete Wavelet Transform on FPGA", IEEE International Conference on Electronics, Circuits and Systems, Malta, Sep 2001
- [7] R.Tesseier, W.Burison, "Reconfigurable Computing for Digital Signal Processing – A survey", Journal of VLSI Signal Processing, 2000

- [8] C. Carmichael, "Configuring Virtex FPGAs from parallel EPROMs with a CPLD", Xapp 137 March1, 1999, XILINX.
- [9] T.Acharya, "A High speed Reconfigurable Integrated Architecture for DWT", IEEE Global Telecommunications Conference, 1997.
- [10] K. K. Parhi, "VLSI Digital Signal Processing Systems-Design and Implementation", John Willey & Sons, 1999.
- [11] G. Sassatelli, L. Torires, P. Besoit, J. Gil, C. Diou, G. Cambon, "Highly scaleable Dynamically Reconfigurable systolic ring architecture for DSP application", Design, Automation and Test in Europe Conference and Exhibition(DATE) 2002.
- [12] T.C.Denk and K.K.Parhi, "VLSI Architecture for Lattice Structure Based Ortho-normal Discrete Wavelet Transform", IEEE Transactions on Circuits and Systems-II, Vol. 44, no.2, pp. 129-132, 1997.
- [13] K.Andra, C. Chakrabarti, and T.Acharya, "A VLSI architecture for lifting-based forward and inverse wavelet transform", IEEE Trans. Signal Processing, vol.50, pp. 966-977, Apr.2002
- [14] C.Souani, M. Abid, K. Tourki and R. Tourki, "VLSI design of 1-D DWT architecture with parallel filters", Integration, the VLSI Journal, Volume 29, Issue 2, pp 181-207 ,September 2000.
- [15] P.Wu and L. Chen, "An efficient architectures for Two-Dimensional discrete wavelet transform", IEEE trans,Circuits Syst. Video Tech., vol. 11, pp. 536-544, Apr. 2001.
- [16] B. Wu, C. Lin, "A high-performance and memory-efficient pipeline architecture for the Legal5/3 and 9/7 discrete wavelet transform of JPEG2000 Codec", IEEE Trans. Circuits Systems Video Tech. 15(12), pp 1615-1628, December 2005.
- [17] I. Daubchies, Ten Lectures on wavelets, SIAM, Philadelphia, PA, 1992.
- [18] C.Hereley, M.Vetterli, "Wavelet and recursive filter banks", IEEE Trans. Signal Process. 41(8), pp 2536-2556, 1993.
- [19] M.Vishwanathan, "The recursive pyramid algorithm for the discrete wavelet transform", IEEE Trans. Signal Process. 42(3), pp 673-676, 1994.
- [20] N.Ahmed, T.Natarajan, K.R.Rao, "Discrete Cosine transform", IEEE Trans. On computer, vol. 23, no. 1, pp 90-93, January 1987.
- [21] Yuk-Hee Chan, Lap Pui Chan and Wam-Chi Siu, "Efficient implementation of Discrete Cosine Transform using Recursive Filter structure", IEEE Trans. Circuits and Systems. for Video Technology, vol.4, no. 6, December 1994.
- [22] M.T.Sun, L.Wu and M.L.Liou, "A concurrent architecture for VLSI implementation of DCT", IEEE Trans. Circuits and Systems, vol. 34, pp 992-994, 1987.
- [23] Web reference - <http://www.xilinx.com/support/swmanuals/xilinx9/index.htm>
- [24] Web reference <http://verilog.net/doc.htm>
- [25] Web reference <http://www.eg.bucknell.edu/~cs320/1995-fall/verilog-manual.html>
- [26] <http://www.model.com/resources/student-edition/>

AUTHORS PROFILE



Kalyan Mohanta (M'78) received B. Tech. in Computer Science & Technology from Kalyani Govt. Engineering College, and M. E. in CSE from West Bengal University of Technology. He is presently in B. P. Poddar Institute of Management & Technology. His research interest include Embedded System, Parallel and Distributed computing.