K. Yamuna, C. Chandrasekhar / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 4, Jul-Aug 2013, pp.1772-1777

Design And Implementation Of Efficient Lifting Based Dwt Architecture Using Wallace Tree Multiplier For Compression

K. Yamuna*, C. Chandrasekhar**
*(M. Tech (VLSI), SVCET, CHITTOOR, A.P., INDIA
** (HOD, Dept. of ECE, SVCET, CHITTOOR, A.P, INDIA

ABSTRACT

Demand for high speed and low power architecture for DWT computation have led to design of novel algorithms and architecture. In this paper we design, model and implement a hardware efficient, high speed and power efficient DWT architecture based on modified lifting scheme algorithm. The design is interfaced with SIPO and PISO to reduce the number of I/O lines on the FPGA. The design is implemented on Spartan III device and is compared with lifting scheme logic. The proposed design operates at frequency of 520.738 MHz and consumes power less than 0.103W. The pre-synthesis and post-synthesis results are verified and suitable test vectors are used in verifying the functionality of the design. The design is suitable for real time data processing.

Keywords **-** Lifting scheme, low power, high speed, FPGA implementation

I. INTRODUCTION

Discrete wavelet transforms (DWT) decomposes image into multiple sub bands of low and high frequency components. Encoding of sub band components leads to compression of image. DWT along with encoding technique represents image information with less number of bits achieving image compression. Image compression finds application in every discipline such as entertainment, medical, defense, commercial and industrial domains. The core of image compression unit is DWT. Other image processing techniques such as image enhancement, image restoration and image filtering also requires DWT and Inverse DWT for transformations. DWT-IDWT is one of the prominent transformation techniques that are widely used in signal processing and communication applications. DWT-IDWT computes or transforms signal into multiple resolution sub bands. DWT is computationally very intensive and consumes power due to large number of mathematical operations. Latency and throughput are other major limitations of DWT as there are multiple levels of hierarchy. DWT has traditionally been implemented by convolution. Digit serial or parallel representation of input data further decides the architecture complexity. Such an implementation demands a large number of computations and a large storage that are not desirable for either high-speed or low-power applications. Recently, a lifting-based scheme that often requires far fewer computations has been proposed for the DWT. The main feature of the lifting based DWT scheme is to break up the high pass and low pass filters into a sequence of upper and lower triangular matrices and convert the filter implementation into banded matrix multiplications. Since DWT requires intensive computations, several architectural solutions using special purpose parallel processor have been proposed, in order to meet the real time requirement in many applications. The solutions include parallel filter architecture, SIMD linear array architecture, SIMD multigrain architecture, 2-D block based architecture, and the AWARE's wavelet transform processor (WTP). Several versions of lifting scheme architecture have been compared and reported in literature. In terms of hardware complexity, the folded architecture is the simplest and the DSP-based architecture is the most complex. All other architectures have comparable hardware complexity and primarily differ in the number of registers and multiplexor circuitry. The control complexity of the architecture is very simple. In contrast, the number of switches, multiplexors and control signals used in the architectures of is quite large. The control complexity of the remaining architectures is moderate. In terms of timing performance, the architectures are all pipelined, with the architectures having the highest throughput (1/Tm). The architecture has fewer cycles since it is RPA based, but its clock period is higher. The architecture in has the lowest computation delay.

DWT is recommended by JPEG2000 standards as it supports features like progressive transmission, higher compression and region of interest encoding schemes. Convolution based DWT or FIR filter bank based DWT architectures occupy large area as they require more number of multipliers and adders, thus making the computations complex and time consuming. Mobile phones and other similar hand held devices that support image//video applications demand high speed and low power architectures with reduced memory size for DWT processing. There are several architectures discussed in literature to perform lifting based approach for 2-D DWT is to apply the 1-D DWT row-wise which produces L and H sub bands and then process these sub-bands column-wise to get LL, LH, HL and HH coefficients.

Several architectures like direct mapped, folded, and flipping for single level and multi-level DWT have been proposed to implement 1-D lifting DWT. Much architecture that implement the Two-Dimensional separable Forward (2D-DWT) and Inverse DWT (2D-IDWT) in order to be applied on 2D signals have been presented in the past. These architectures are consisting of filters for performing the 1D-DWT and memory units for storing the results of the transformation. Due to the fact that streaming multimedia applications - in which the DWT is present - are characterized by high throughput requirements, this imposes the need for optimizing the design of the filters in terms of speed. Moreover, portable multimedia devices require low power consumption for increasing the battery lifetime and this can be achieved by minimizing the storage size and number of memory accesses. Low power DWT architectures based on pipelining and parallel processing has been discussed, in their work low power is achieved by modifying the architecture to reduce number of computations the design was implemented on FPGA. Many of the low power techniques reported for DWT propose modifications in the architecture level to reduce power dissipation. Power reduction can be accomplished at various levels of abstraction starting from architecture level to circuit level. Power reduction at the sub system level or at the circuit level can be accomplished when ASIC design of DWT architecture is performed. Many of the work reported in literature have restricted to FPGA implementation. In this paper, in order to demonstrate the dynamic power reduction techniques at various levels of abstraction, DWT architecture is considered as a test case for illustration. ASIC design of DWT architecture optimizing dynamic power reduction using 65nm TSMC libraries is performed

II. WALLACE TREE MULTIPLIER

A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers. The benefit of the Wallace tree is that there are only $O(\log n)$ reduction layers, and each layer has $O(1)$ propagation delay. As making the partial products is $O(1)$ and the final addition is $O(\log n)$, the multiplication is only $O(\log n)$, not much slower than addition (however, much more expensive in the gate count). Naively adding partial products with regular adders would require $O(\log^2 n)$ time. From a complexity theoretic perspective, the Wallace tree algorithm puts multiplication in the class NC1.

These computations only consider gate delays and don't deal with wire delays, which can also be very substantial.

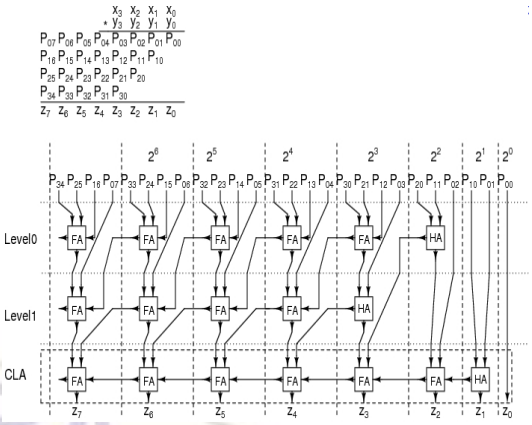

Fig.1.Wallace tree multiplier

The Wallace tree can be also represented by a tree of 3/2 or 4/2 adders. It is sometimes combined with Booth encoding.

III. LIFTING SCHEME

The Lifting Scheme is a well-known method for constructing bi-orthogonal wavelets. The main difference with the classical construction is that it does not rely on the Fourier transform. The lifting scheme is an efficient implementation of a wavelet transform algorithm. It was primarily developed as a method to improve wavelet transform, and then it was extended to a generic method to create so-called second-generation wavelets. Second-generation wavelets are much more flexible and powerful than the first generation wavelets. The lifting scheme is an implementation of the filtering operations at each level [6]. The figure 3 represents the classical and lifting based implementations of DWT.

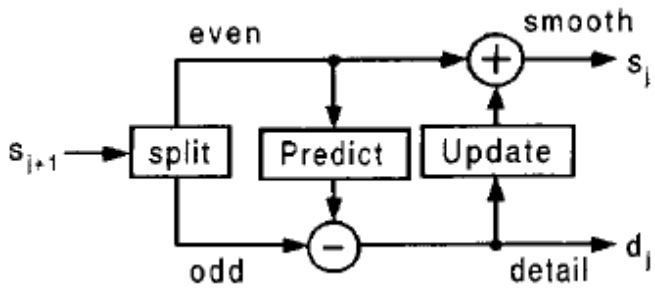

Fig.2.Lifting scheme

Lifting Scheme consists of three steps: SPLIT, PREDICT and UPDATE, as shown in the figure 2.

SPLIT: In this step, the data is divided into ODD and EVEN elements.

PREDICT: The PREDICT step uses a function that approximates the data set. The differences between the approximation and the actual data, replace the odd elements of the data set. The even elements are left unchanged and become the input for the next step in the transform. The PREDICT step, where the odd

value is "predicted" from the even value is described by the equation

Oddj+1, i = oddj, i – P (evenj, i)

UPDATE: The UPDATE step replaces the even elements with an average. These results in a smoother input for the next step of the wavelet transform. The odd elements also represent an approximation of the original data set, which allows filters to be constructed. The UPDATE phase follows the PREDICT phase. The original values of the odd elements have been overwritten by the difference between the odd element and its even "predictor". So in calculating an average the UPDATE phase must operate on the differences that are stored in the odd elements:

Evenj+1, i = evenj, i + **U** (oddj+1, i)

The equations for the lifting based implementation of the bi-orthogonal wavelet are:

Predict P1: di1 = α (x2i + x2i+2) +x2i+1 .. (1)

Update U1: ai1 = β (di1+di-11) + x2i ….. (2)

Predict P2: di2 = γ (ai1 + ai+1) +x2i+1 …..(3)

Update U2: ai2 = δ (di2+di-12) + ai1 .. (4)

Scale G1: ai = ζ ai2 …... (5)

Scale G1: di = di2/ζ …….. (6)

The figure 4 shows the lifting scheme architecture to realize the equations shown above.

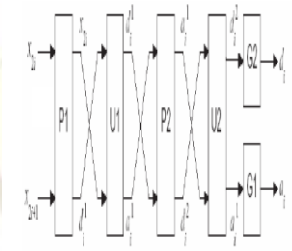

Fig.3.Lifting Scheme Architecture

The input data x is first split into even and odd samples and each of the samples are taken through predict and update stages as per the architecture shown above. As the data moves from first stage to the last stage, data switching occurs at the input and output of every stage. Every stage consists of multipliers and adders. For the given set of Predict and Update stages, assuming the value of **i = 0,** the equation can be finalized.

IV. MODIFIED LIFTING SCHEME

By re-arranging all the values and the constant co-efficient, the final equation can be derived.

ai= (3 * γ.β.δ.ζ + δ.ζ + β.ζ) [α (x0 + x2) + x1 + α (x0 + x-2) + x-1] + ζ.δ.β.γ [α(x2 + x4) + x3 + α(x-2 + x-4) + x-3] + ζ.δ.γ (x0 + x2 + x0 + x-2) + ζ * x0 …… (7)

di= 1/ζ [(2 * γ.β +1){ α (x0 + x2) + x1} + γ.β { α (x0 + x-2) + x-1 + α(x2 + x4) + x3} +γ(x0 + x2)] ……(8)

Being a dedicated DWT core for JPEG 2000, the filter coefficients are fixed. The filter coefficients are: α = 1.586134342, β = 0.05298011854, γ = 0.8829110762, δ =0.4435068522, ζ = 1.149604398. By substituting the above values in the modified equation, the coefficient values obtained then are also decimals, by multiplying them with constants they form integers as: 1 * 32 = 57, 2 * 256 = 6, 3 * 64 = 30, 4 * 32 = 35, 5 * 256 = 12, 6 * 32 = 26, 7 * 32 = 50.

Thus the above integers are the values of the underlined coefficients in above equations. From the equations it is observed that there are common lifting coefficients to compute ai and di coefficients and there are input terms. The architecture realised by the above equations considering the constant coefficients is shown in the figure 4.

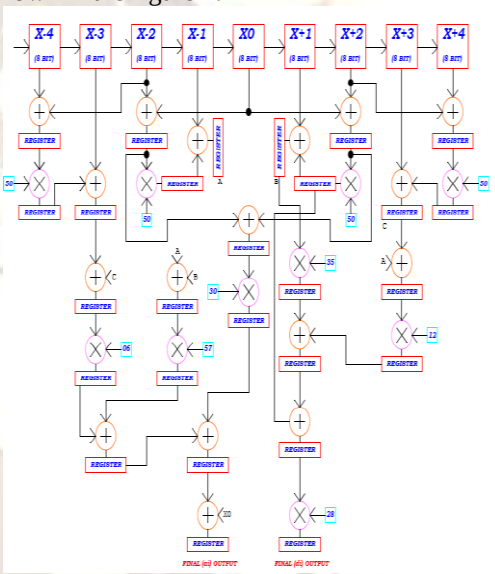

Figure 4 Modified Lifting Scheme Architecture for DWT

Table 1 Estimation of Power, Delay and Area of sub blocks

Sl No	Sub Block	Power (W)	Area(no of slices,utilization of slice%)	Delay(nS)
1)	Adder	0.142	5(1%)	3.369
2)	Multiplier(50)	0.007	9 (1%)	1.999
3)	Multiplier(30)	0.037	7(1%)	0.881
4)	Multiplier(12)	0.037	6(1%)	0.881
5)	Multiplier(35)	0.037	13(1%)	0.881
6)	Multiplier(57)	0.037	11(1%)	0.881
7)	Multiplier(6)	0.037	6(0%)	0.881
8)	Multiplier(26)	0.037	7(1%)	0.881
9)	PISO	0.011	5(1%)	1.216
10)	SIPO	0.037	36(1%)	0.915
11)	PIPO	0.037	1(0%)	0.916

V. HDL MODELING AND FPGA IMPLEMENTATION

The top level module or block of the DWT architecture is shown in the figure 5. The figure 5 explains the input and output ports. The input ports are **clk, en, piso_load, rst and ser_in** and the output ports are **ai** and **di**. The input 9 samples each of 8 bit signed data is entered into the design through the **ser_in** input. The **rst** signal is used to reset the design when the signal is high. When the **en** signal is high, loading of the input data in all the 9- 8 bit registers for 280 clock cycles is done. The **piso_load** signal is used to take the output at **ai** and **di,** and this signal is kept high for 8 clock cycles as the 8 bit is to be taken out through the single line. The HDL models of the sub-block can be understood from the internal hardware of the RTL schematic shown in the figure 5. The figure 5 represents the schematic of the DWT architecture where all the sub blocks can be viewed. Thus the sub blocks are modelled in such a way that the multipliers used are the IP cores from the XILINX library, and the adder that is designed for 8 bit signed addition is instantiated wherever necessary. The simulation of the top level module is shown. Where the intermediate signals gives the performance of the sub blocks in the total simulation.

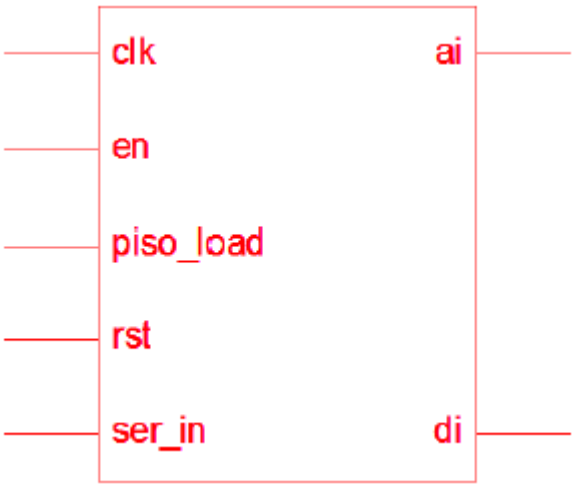

Figure 5 Top-level DWT architecture

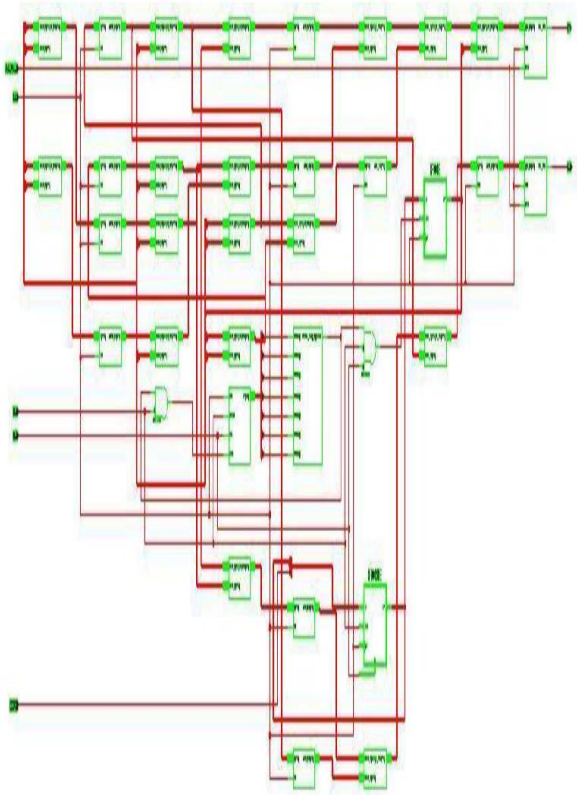

Fig. 6 .RTL schematic of DWT showing sub-block

The figure 5 **explains the integration of the sub blocks** in the main top level architecture. Initially the sub blocks are designed by considering the DWT equation, the multiplier used in the design is a **Wallace Tree multiplier** as it is faster than any other for the application required. For the present design, the constant coefficient multipliers are taken as a **IP core** from the XILINX library for different coefficients. The **adder** is 8-bit signed operator designed or modelled in the HDL and instantiated where it is necessary. The registers that are used in the design covers all the types SIPO, PIPO, PISO. SIPO at the initial stage while giving the inputs, PIPO while performing the operations intermediately, and the PISO at the output stage to take the outputs serially i.e. one bit for 8 clocks, as the required is two outputs of 8 bits taking serially. From the figure 9, the top level ports are shown; the serial input data is given in a random way. This is loaded in the registers (SIPO), when enable signal is high, after 72 clock cycles the enable is made low, and for four clock cycles the operation is performed and the output is taken when the piso_load signal is high for 8 clock cycles as the output is taken for 8 bit. Thus the same procedure follows for 8 (load)+ 4 (operation) + 8 (output) = 22 clocks. To program a single device using iMPACT, all needed is a bitstream file. To program several devices in a daisy chain configuration, or to program devices using a PROM, iMPACT is used to create a PROM file. iMPACT

accepts any number of bitstream and creates one or more PROM files containing one or more daisy chain configuration

Fig.7.simulation results for the DWT Architecture

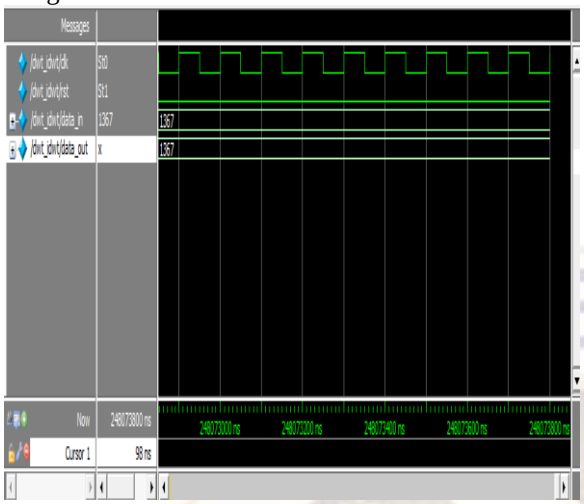

TABLE 2 **S**ynthesis results of modified lifting based DWT

Parameters	**Lifting based DWT**	**Modified Lifting based DWT with low power**
Area(sq. mm)(LUT's)	2179(7%)	835(3%)
Power(W)	0.103	0.103
Operating frequency(max) MHz	151.814	520.738

VI. CONCLUSION

In this work a modified lifting based DWT architecture is proposed, designed, modeled and verified. The design is modeled using HDL and is implemented on FPGA. The interfaces required for data processing are also designed and is used to synchronize the data transfer operation. The HDL models and simulation of the sub blocks have been done to model the top-level design architecture. The test-bench to verify the functionality and performance of the sub modules and the top level architecture have been done. Implemented the design on FPGA and verified and debugged through the Chip-Scope. The Pre and Post Synthesis have been done and compared. The design can be further optimized for video signal processing.

REFERENCES

[1] N. Jayant and P. Noll, Digital Coding of Waveforms: Principles and Applications to Speech and Video. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[2] C. Diou, L. Torres, and M. Robert, "A wavelet core for video processing," presented at the IEEE Int. Conf. Image Process., Sept. 2000.

[3] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," Proc. IEEE, vol. 81, pp. 1385–1422, Oct. 1993.

[4] B. Zovko-Cihlar, S. Grgic, and D. Modric, "Coding techniques in multimedia communications," in Proc. 2nd Int. Workshop Image and Signal Processing, IWISP'95, Budapest, Hungary, 1995, pp. 24–32.

[5] Digital Compression and Coding of Continuous Tone Still Images, ISO/IEC IS 10918, 1991.

[6] I. Daubechies, Ten Lectures on Wavelets. Philadelphia, PA: SIAM, 1992.

[7] S. Mallat, "A theory of multiresolutio

[8] n signal decomposition: The wavelet representation," IEEE Trans. Pattern Anal. Machine Intell., vol. 11, pp. 674–693, July 1989.

[9] M. Nagabushanam, Cyril Prasanna Raj P, S. Ramachandran, Design and Implementation of Parallel and Pipelined Distributive Arithmetic Based Discrete Wavelet Transform IP Core, European Journal of Scientific Research ISSN 1450-216X Vol.35 No.3 (2009), pp.378-392.

[10] M. Vishwanath, R. Owens, and M. J. Irwin, "VLSI architectures for the discrete wavelet transform," IEEE Trans. Circuits Syst. II, vol. 42, pp. 305–316, May 1995.

[11] J. S. Fridman and E. S. Manolakos, "Discrete wavelet transform: Data dependence analysis and synthesis of distributed memory and control array.

AUTHORS BIBILIOGRAPHY

1.

K.YAMUNA, student of M.Tech (VLSI Design), sri venkateswara college of engineering &Technology, Chittoor
Email.id:Yamuna.k712@gmail.com

2. Under the guidance of C.CHANDRA SEKHAR,HOD &Professor, Dept.of ECE
Email.id:uma_sekhar2000@gmail.com