FPGA Implementation of Multiplierless 2D DWT Architecture for Image Compression

Divakara.S.S,
Research Scholar,
J.S.S. Research Foundation, Mysore

Cyril Prasanna Raj P
Dean(R&D), MSEC, Bangalore

Thejas M S
Assistant Professor
Coorg Institute Of Technology,Ponnampet.

Abstract: In this paper a novel architecture for DWT computation of input image of size greater than 512 x 512 is designed and implemented on FPGA. Bior4.4 or 9/7 filters coefficients are scaled to integer values and rounded-off to nearest integer, the input image samples are multiplied with the rounded-off coefficients using shift operation. The shift operation replaces multiplication operation for computation of 2D image coefficients and are computed using a pipelined architecture that is realized on XILINX FPGA. The designed architecture operates at a maximum frequency of over 231.192 MHz and consumes area less than 30% of the CLB resources on FPGA with power consumption of 0.04325mW.

Index Terms: Image compression, 2DDWT, FPGA, High speed, low power, multiplierless architecture, pipelined architecture.

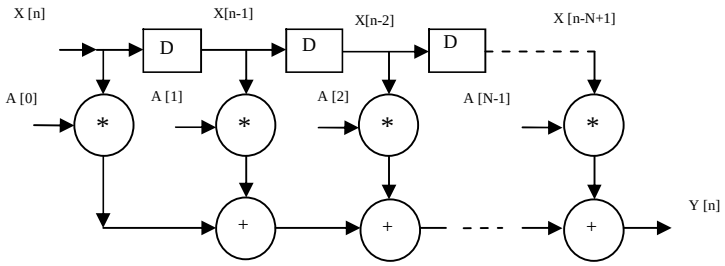
I. INTRODUCTION

The advantages of the wavelet transform over conventional transforms, such as the Fourier transform, are now well recognized. Because of its excellent localizing ability in time-frequency domain, wavelet transform is remarkable and extensively used for signal analysis, compressing and deionizing. Defining DWT, Mallat [1] provided possibility of its digital hardware or software implementation. The Discrete Wavelet Transform (DWT) performs a multi resolution signal analysis which has adjustable locality in both the space (time) and frequency domains [1]. Unlike the Fourier transform, the wavelet transform has many possible sets of basis functions. Using finite impulse response (FIR) filters and then subsampling is the classical method for implementing the DWT. Due to the large amount of computations required, there have been many research efforts to develop new rapid algorithms [2]. In 1996, Sweldens presented a lifting scheme for a fast DWT, which can be easily implemented by hardware due to significantly reduced computations [3]. Due to the intensive computations involved with this transform, the design of efficient VLSI architectures for a fast computation of the transforms have become essential, especially for real-time applications and those requiring processing of high-speed data. All the architectures in [4-10] aim at providing high-speed computation of the DWT using resource-efficient hardware.

Due to recent advances in the technology, implementation of the DWT on Field Programmable Gate Array (FPGA) and Digital Signal Processing (DSP) chips has been widely developed. The main challenges in the hardware architectures for 2-D DWT are the processing speed and the number of multipliers and adders.

DWT architecture

As shown in Figureure 1, DWT computation based on convolution algorithm is implemented using multiplication and accumulation unit as shown in Figureure 1.



Figureure 1 Convolution architecture for DWT [4]

In the conventional architecture for DWT based on convolution algorithm, the input samples are sequentially processed by multiplying the input samples with the filter coefficients A[n], and accumulating the samples to compute Y[n]. The DWT filters for Biorthogonal 9/7 wavelet consists of 9 filter coefficients and 7 filter coefficients for low -pass and high-pass respectively. For every output sample computation due to the sequential approach in the MAC operation, the latency is 9 clock cycles and 7 clock cycles for low-pass and high-pass computation. The throughput without pipeline is 9 clock cycles and 7 clock cycles respectively. For every clock cycle multiplication and addition need to be performed and hence the clock frequency of multiplier and adder should be 10 times higher (10 bit multiplier and adder) than the clock frequency of input data. In order to reduce the computation complexity and improve the throughput, a modified algorithm is proposed in this work. The rest of the paper is organized as follows: section II details about the modified architecture for 1D DWT computation.
Section III discusses about the results and discussion. Section IV describes the conclusion.

.

II. MODIFIED ARCHITECTURE FOR 1D DWT COMPUTATION

The LPF outputs are $Y_{L0} = X_0 L_0 + X_{-1} L_1 + X_2 L_2 + X_{-3} L_3 + ---- X_{-8} L_8$; $Y_{L1} = X1 L_0 + X_0 L_1 + X_{-1} L_2 + ---- X_{-7} L_8$. Similarly $Y_{L8} = X_0 L_0 + X_1 L_1 + X_2 L_2 + X_3 L_3 + X_4 L_4 + X_5 L_5 + X_6 L_6 + X_7 L_7 + X_8 L_8$. For Bior 9/7, filter coefficients are symmetric, i.e, $L_0 = L_8$, $L_1 = L_7$, $L_2 = L_6$, $L_3 = L_5$. Hence the output can be rewritten as $Y_{L8} = (X_0 + X_8) L_8 + (X_1 + X_7) L_7 + (X_2 + X_6) L_6 + (X_3 + X_5) L_5 + X_4 L_4$ which can be represented as $Y_{L8} = X^1_0 L_8 + X^1_1 L_7 + X^1_2 L_6 + X^1_3 L_5 + X^1_4 L_4$ where $X^1_0=(X_0 + X_8)$.

Based on the modified equation for Y_{L8} the modified architecture is derived as shown in Figure 2. The architecture consists of input memory, rearranged input memory, adder, intermediate register and processing unit for multiplication and accumulation, output memory.

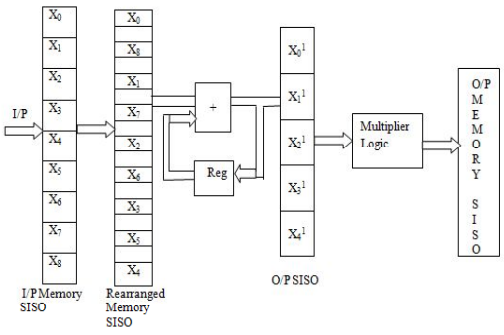


Figure.2. Modified architecture for DWT [LPF]

The input data [X_0 to X_8] is rearranged as in rearranged memory, the data is read and accumulated in the adder and stored in the O/P SISO. The samples [X^1_0 to X^1_4] are multiplied to obtain the LPF O/P. The LPF equation is written as in (1), after substituting filter coefficients.

$$Y_{L8} = X^1_0 [10] + X^1_1 [-6] + X^1_2 [-28] + X^1_3 [97] + X^1_4[218] \quad (1)$$

Above expression can be rewritten as,

$$Y_{L8} = X^1_0 [8 - 2] + X^1_1 [-8 + 2] + X^1_2 [-32 + 4] + X^1_3 [64 + 32] + X^1_4 [256 - 32]. \quad (2)$$

The filter coefficients 10, -6, -28, 97, 218 are expressed in multiples of 2, which are expanded as:

$$Y_{L8} = X^1_0 8 - X^1_0 2 - X^1_1 8 + X^1_1 2 - X^1_2 32 + X^1_2 4 + X^1_3 64 + X^1_3 32 + X^1_4 256 - X^1_4 32 \quad (3)$$

The input samples are to be multiplied by the modified filter coefficients, which can be realized by performing left shift operation as indicated in (4).

$$Y_{L8} = [X^1_0 << 3] - [X^1_0 << 1] - [X^1_1 << 3] + [X^1_1 << 1] - [X^1_2 << 5] + [X^1_2 << 2] + [X^1_3 << 6] + [X^1_3 << 5] + [X^1_4 << 8] - [X^1_4 << 5] \quad (4)$$

The filter coefficients 97 and 218 are rounded of to 96 and 224 in order to achieve left shift in multiples of 2. Based on the modified equation in (4), the LPF architecture is realized and Figure. 3 shows the architecture of proposed logic for Y_{L8} computation.

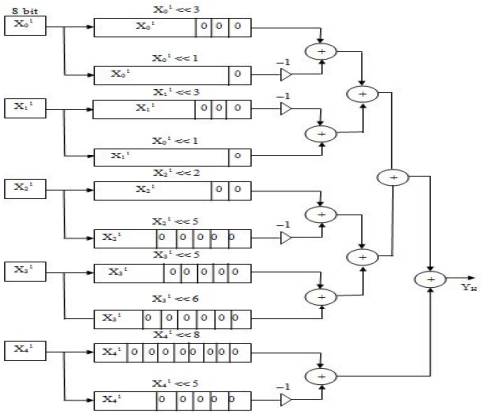


Figure.3 Multiplierless DWT LPF architecture

The left shift operations are performed by barrel shifter logic and this is faster and does not require clock. The adders are realized to operate in parallel, thus reducing delay in computation. Similarly the HPF computation is designed as in equation (5) – (9)

$$Y_{H6} = H_0 X_0 + H_1 X_1 + H_2 X_2 + H_3 X_3 + H_4 X_4 + H_5 X_5 + H_6 X_6 \quad (5)$$

$$Y_{H6} = [X_0 + X_6] H_0 + [X_1 + X_5] H_1 + [X_2 + X_4] H_2 + H_3 X_3 \quad (6)$$

$$Y_{H6} = X^1_0 H_0 + X^1_1 H_1 + X^1_2 H_2 + X^1_3 H_3 \quad (7)$$

$$Y_{H6} = X^1_0 [-17] + X^1_1 [10] + X^1_2 [107] + X^1_3 [-202]$$

$$= X^1_0 [-16 - 1] + X^1_1 [-8 - 2] + X^1_3 [128 + 64 + 8] + X^1_2 [128 - 16] \quad (8)$$

$$Y_{H6} = -[X^1_0 << 4] - [X^1_0 << 0] - [X^1_1 << 3] - [X^1_1 << 1] + [X^1_2 << 7] - [X^1_2 << 4] + [X^1_3 << 8]$$

$$+ [X^1_3 << 5] + [X^1_3 << 3] \quad (9)$$

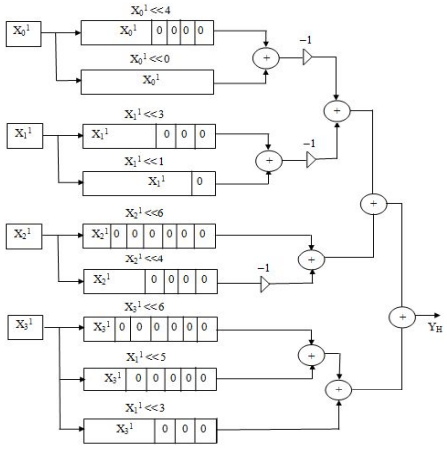


Figure.4. DWT HPF architecture without multipliers

The coefficients 107 and 202 are rounded of to 112 and 200 respectively. The modified architecture is realized based on the above equations using shifters and is shown in Figure 4. The top level architecture for 2D DWT processor is implemented using the modified 1D-DWT architecture discussed in Figure 3 and Figure 4. The 2D-DWT processor consists of input memory, output memory and three 1D-DWT processors as shown in Figure 5.

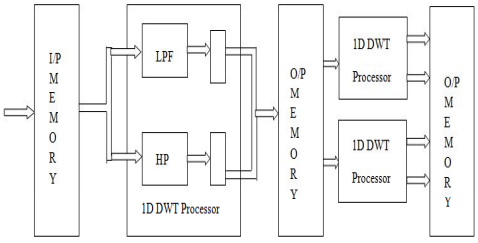


Figure.5.2D DWT architecture

HDL code for 1D DWT processor, input memory and output memory is developed and are integrated to top module. The top module is verified using test bench written in Verilog and with set of input vectors. The simulation results and synthesis results are obtained using Xilinx ISE.

III. RESULTS AND DISCUSSION

The input x_i is of 8 bits; clock and reset are the other inputs. The outputs obtained were a_i and d_i which were of 16 bit signed outputs with the consideration of scaling. The negative values of outputs were obtained in 2's complement form. In one clock cycle, the corresponding values of a_i and d_i were obtained for each value of i. Verilog code for DWT was synthesized using Xilinx with the Virtex-5 device having gate capacity 110 million gates and the RTL schematic obtained is as shown in the Figure. 6. Figure. 7 shows the 2D-DWT architecture using the proposed architecture.

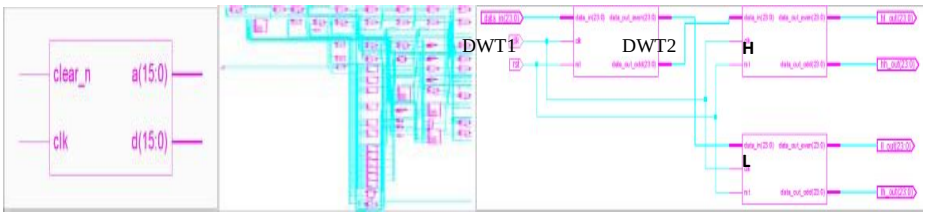


Figure.6. RT schematic of DWT

Figure.7. 2D- DWT architecture synthesis schematic

The synthesis results obtained are verified with various constraints options provided in the tool. The default options were producing best results. The area report in terms of slices, the power report and timing report have been generated and are reported in this work. A comparison of the hardware utilization and computation time of the proposed architecture with other various architectures of [4-10] for an image of 512 × 512 are reported in Table I.

The performance of the proposed architecture is now compared with various architectures in terms of the FPGA implementation results available in the literature. The FPGA implementation results for the architectures presented in [4]-[10] are listed in Table II.

From the comparison results it is demonstrated that the proposed architecture consumes very less resources, as the multipliers are replaced with shift operations, the operating frequency is increased to 231.192 MHz and power dissipation is reduced by setting the low power constraints. One of the major challenges in the design is data synchronization in DWT computing, as the shift operations are used for multiplication operation, it is mandatory to carefully design the control unit to keep track of the data output and read the data into register for further computation and hence there is need for a predesigned control logic to monitor the data flow logic.

IV. CONCLUSION

From the simulation results, it is demonstrated that minimum of 7-10 bits are required number representation for hardware realization. The DWT processor is realized using a modified algorithm that reduces the number of computation by half by exploiting t**h**e symmetric property. The filter coefficients obtained after quantization are rounded to represent the filter coefficients, so that the quantization noise does not impact the gain by more than 20 dB. Multiplication is realized using shift left operations. The modified algorithm is realized using HDL and 1D-DWT processor and 2D-DWT processor were implemented on FPGA. The results demonstrate improvement in area and speed performances without loss and hence are suitable for lossless image compression.

REFERENCES

[1] S.Mallat, "A Theory for Multiresolution Signal Decomposition: the wavelet representation," *IEEE Trans,Pattern Anal.* Mach. Intell. 11, pp 674–693(1989).
[2] T. Acharya, C. Chakrabarti, "A Survey on Lifting-based Discrete Wavelet Transform Architectures," *J.VLSI Signal Process*. 42, 321–339 (2006).
[3] I. Daubechies, W. Sweldens, "Factoring Wavelet Transform into Lifting Steps." *J. Fourier Anal*. Appl. 4, 247–269 (1998).
[4] H. Y. Liao, M. K. Mandal, and B. F. Cockburn, "Efficient architectures for 1-D and 2-Dlifting based wavelet transforms", *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp. 1315–1326, May 2004.
[5] A. Benkrid, D. Crookes, and K. Benkrid "Design and implementation of a generic 2-Dorthogonal discrete wavelet transform on an FPGA", in *Proc. IEEE 9th Symp.Field programming Custom Computing Machines (FCCM)*, , pp. 190–198. Apr. 2001.
[6] P. McCanny, S. Masud, and J. McCanny, "Design and implementation of the symmetrically extended 2-D wavelet transform", in *Proc. IEEE Int. Conf Acoustic, Speech,Signal Process. (ICASSP)*, vol. 3, pp. 3108–3111,2002.
[7] S. Raghunath and S. M. Aziz, "High speed area efficient multi-resolution 2-D 9/7 filter DWT processor", in *Proc. Int. Conf. Very Large Scale Integration (IFIP)*, , vol. 16–18, pp. 210–215.,Oct. 2006.
[8] C. Cheng and K. K. Parhi, "High-speed VLSI implementation of 2-Ddiscrete wavelet transform", *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 393–403, Jan. 2008.
[9] I. S. Uzun and A. Amira, "Rapid prototyping—framework for FPGA based discrete biorthogonal wavelet transforms implementation", *IEE Vision, Image Signal Process.*, vol.153, no. 6, pp. 721–734, Dec. 2006.
[10] Chengjun Zhang, Chunyan Wang,, "A pipeline VLSI architecture for fast computation of the 2-d discrete wavelet transform", IEEE transactions on circuits and systems, vol. 59, no. 8, pp 1775-1785 August 2012 .

TABLE 1

Expressions for Metrics of Various Architectures

Architecture	No. of multipliers	No. of adders	Storage size	Filter type	No. of clock cycle	Latency
Recursive Architecture[4]	12	16	4N	1-D(9/7)	N^2+N	T_cN^2
Generic folded[5]	6*J*(l/2)	$6J(1+log_2(l/2))$	$(l-1)N/3	1-D	N^2	T_cN^2
Symmetrically Extended[6]	L/2=L/4=L/8	2(L/2+L/4+L/8)	(L+0.5)N	1-D	$1.5N^2$	$1.5\ T_cN^2$
Parallel FDWT[7]	12	16	3N/2	1-D(9/7)	N^2	T_cN^2
Parallel Probe. 4[8]	96	240	[4N=32J+256](on chip delay units) [8N=+128(j-1)](off chip buffer)	2-D(L=4)	$N^2/12$	$T_cN^2/12$
Arch2D-II[9]	$L^2/2$	$L^2/2+L$	N/A	2-D	$2N^2/3$	$2T_cN^2/3$
Pipeline[10]	$11L^2/4$	$11\log_2(L^2/2)+9$	$3L+3L^2$(on chip delay units) $3NL/4+3N^2/128$(off chip buffer)	2-D	$N-^2/2$	$T_cN^2/2$
Proposed	0	$9N^2$	Distributed ram	2-D(9/7)	N^2	$T_cN^2/4$

* Not available

TABLE II

COMPARISON OF VARIOUS FPGA IMPLEMENTATIONS

Architecture	Image size(N)	No. of CLB Slices	RAM size(bits)	f_{max} (MHz)	Time(ms)	Area^{x}Time*	Device
Recursive Architecture[4]	512	879	10N	50	5.3	4659	XC2V250
Generic folded[5]	256	4720	10X(4K)	75	0.874	4125	Virtex 600E-8
Symmetrically Extended[6]	512	2559	17X(18K)	44.1	9	23031	XC2V500
Parallel FDWT[7]	512(J=5)	1700	3N/2	171.8	3.1	5270	Virtex 2
Parallel Probe. 4[8]	Implementation results not available						
Arch2D-II[9]	512	4348	24X(18K)	105	1.7	7392	Virtex2000E
Pipeline[10]	512(J=6)	2842	8X(18K)	135	0.97	2757	XC2VP30
proposed	512	1060	Distributed ram	231.192	0.04325	46	5vlx110tff1136

*The value of area in the calculation of area –time product is replaced by the number of CLB slices since the former is proportional to the latter.