Design and FPGA implementation of lifting scheme for 2D-DWT using wavepipelining

G. SEETHARAMAN, B. VENKATARAMANI AND G. LAKSHMINARAYANAN
Department of Electronics and Communication Engineering
National Institute of Technology, Tiruchirappalli, INDIA

Abstract: In this paper, a hybrid scheme is proposed for the implementation of two level 2D-DWT using lifting scheme. Two filter banks consisting of two filters each are used for the implementation of both horizontal and vertical filters. Each of the filter banks are implemented using the lifting scheme. The individual lifting blocks are implemented using wavepipelining and are interconnected with registers. An automation procedure is proposed for tuning the parameters of the wavepipelined circuit. The multipliers in the lifting blocks are implemented using Baugh-Wooley constant coefficient multiplier scheme (BW-KCM). For verifying the efficacy of the scheme proposed, both one level and two level 2-D DWT schemes for sub images of size 32x32 are implemented on Xilinx XC2S150 device. The results for the hybrid scheme are compared with that obtained using non-pipelined and pipelined approaches. For the one level 2D DWT, the hybrid scheme requires the same area but is faster than nonpipelined scheme by a factor of 1.4. The pipelined scheme using the pipelined BW-PKCM is faster than the hybrid scheme by a factor of 1.2 at the cost of increase in the no. of registers by a factor of 2.73. The delay-power product is lower for hybrid scheme by a factor of 2 than the pipelined scheme. The two level 2D DWT for both pipelined and non pipelined schemes are implemented. The implementation of hybrid scheme for 2 level 2 D DWT is under progress. The technique proposed in this paper is also applicable for ASICs and FPGAs from other vendors.

Keywords: Wavepipelining, pipelining, FPGA, Slices, LUT, interconnect, Self test, 2D-DWT.

1 Introduction

FPGAs have been used as “glue logic” between off the shelf components and as replacements for ASICs in first generation products. Recently, however, FPGAs have become so dense and fast that they have evolved into the central processors of powerful reconfigurable computing systems. The increased performance available with FPGAs make them a good candidate for implementation of area as well as speed intensive image processing systems. The FPGAs are also proposed as coprocessors to augment the power of the traditional P-DSPs, offloading the computation intensive function for image processing such as DCT, DWT and freeing the processor for other functions [1].

For image processing applications, in addition to Discrete Cosine Transform (DCT), wavelet transform is also increasingly used. It is now a part of the JPEG 2000 standard for still image coding. DWT has also been used extensively in biomedical signal processing (see [2], [3]). The VLSI implementation of image encoders with 2D DWT has been addressed in number of previous works. In Ritter [4], FPGA (XC4085XLA) based implementation of 2D DWT using lifting scheme and compression using EZT algorithm is reported. For storing the image and the transform, an external RAM is used in this work. The image block encoder proposed in [5], uses block RAMs in FPGAs for storing the sub image and its transform. As DWT is not a block transform, the image is split into overlapping blocks of sub images for the 2D DWT computation.

For the implementation of the 2D DWT, lifting scheme proposed by Swelden [6] is quite suited for implementation in ASICs and FPGAs. In [5], a new multiplier algorithm denoted as Baugh-Wooley pipelined Constant Coefficient Multiplier (BW-PKCM) which combines the KCM with Baugh-Wooley multiplication algorithm is proposed and used for the study and comparison of DAA and lifting schemes. From the implementation results, it is concluded that for the lifting scheme, the method using BW- PKCM is both area and speed efficient. The lifting scheme with BW-PKCM requires almost the same area and has the same speed as that of the DAA scheme.

In this paper, a hybrid scheme using both wavepipelining and pipelining is considered for the scheme proposed in [5]. Wave pipelined circuit dispenses with the need for registers for storing the intermediate results and instead uses the inherent

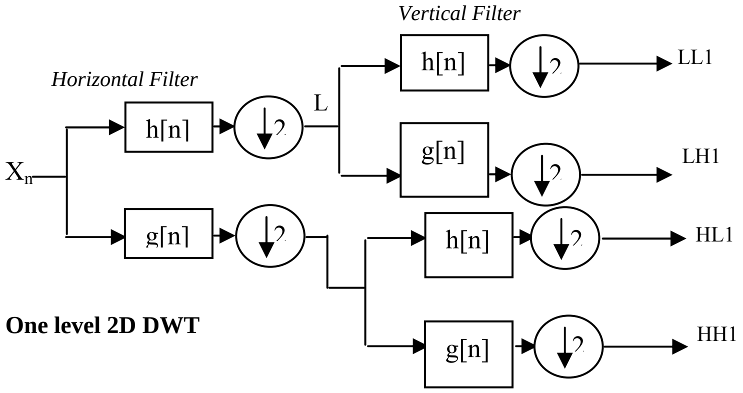

Fig.1 One level 2D DWT

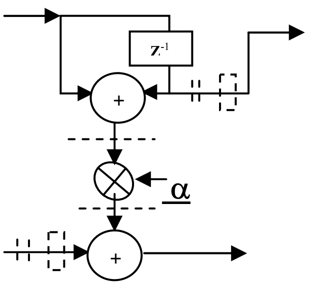

Fig. 2(a) α - Block

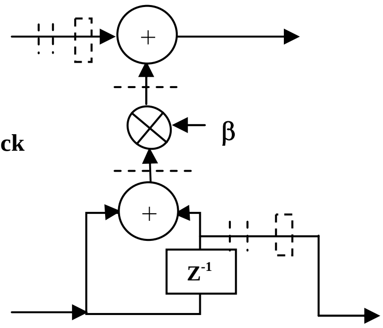

Fig. 2(b) β Block

capacitance at the input to the various combinatorial blocks. This results in lower power dissipation at the cost of speed. The hybrid scheme is aimed at combining the advantages of both pipelining and wavepipelining.

The organization of the rest of the paper is as follows. In section 2, a review of the lifting scheme and the application of the BW-PKCM is considered. In section 3, an overview of the technique used for automation of the wavepipelined circuits in general and the lifting blocks in particular, is given. In section 4, the overlap technique for processing the image using the sub images is presented. Section 5 discusses the implementation scheme for two level 2D DWT. Section 6 discusses the implementation results and conclusions based on the implementations of the 2D DWT scheme using Xilinx Spartan II XC2S150PQ208-5 device are presented in section 7.

2 Review of previous work on 2D DWT

2D discrete wavelet transform may be computed using filter banks as shown in Fig.1. The input samples x(n) are passed through the 2 stages of analysis filters. They are first processed by the low pass (h[n]) and high pass (g[n]) horizontal filters and are sub sampled by two. Subsequently, the outputs (L1, H1) are processed by low pass and high pass vertical filters. The lifting scheme [6] uses a poly-phase structure for the analysis filter.

In the lifting scheme, the odd and even input samples are processed by 5 lifting blocks (α, β, γ, δ, ξ_1 /ξ_2) in cascade. ξ_1, ξ_2 are scaling blocks. Details of α and β blocks are shown in Fig. 2(a) and Fig. 2(b). γ and δ blocks are obtained by replacing the constants α, β with γ, δ. In [5], the following modifications are proposed in the lifting scheme:

- In Fig. 2(a), since the output from one block is fed as input to the next block, the maximum rate at which the input can be fed to the system depends on the sum of the delays in all the four stages. The speed is increased in [5], by introducing pipelining at the points indicated by dotted lines in Fig. 2(a). In this case, the input rate is determined by the largest delay among all the four blocks.
- The delay in the individual stages is reduced further by using Constant Coefficient multiplier (KCM).

KCM uses a ROM for finding the product of a constant and a variable. The variable is fed as address to the ROM which contains the products corresponding to all possible combinations of the operands. When the ROM is implemented using 4 input Look Up Tables (LUTs), a no. of stages of LUTs and adders are required to find the product. For example a 12x12 bit KCM requires one ROM stage consisting of three 16X16 ROMs and two stages of 16 bit adders. The speed of the KCM can be increased by introducing the pipelining registers at the outputs of ROMs and adders.

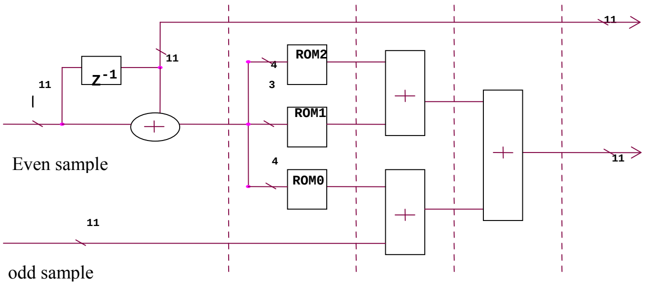

Fig. 3 α Block using BW-PKCM

The content of the ROM corresponding to multiplication of signed numbers can be computed using three approaches: (i) Assuming unsigned multiplication and 2's complement blocks (resulting multiplier is referred to as conventional 2's complement multiplier (C2CM)) (ii)Using sign extension (iii) Baugh Wooley multiplier. The PKCM using the BW content is referred to as BW-PKCM in [5] and is shown to be superior compared to the other two approaches. Hence, only this multiplier is considered for wavepipelining in this paper.

The detailed diagram of the α block implemented using BW-PKCM is shown in Fig. 3. The same scheme can be adopted for the β,γ,δ, ξ_1, ξ_2 blocks.

3 Design of wavepipelined lifting blocks on FPGA

3.1 Review of prior work on wavepipelining:

The idea of wave-pipelining or maximal rate pipelining was first formalized in [7]. Recently, this concept has been a subject of renewed interest as technology and design techniques have enabled the effective implementation of wave-pipelining in integrated circuits. The concept of wave-pipelining has been described in a number of previous works [8], [9], [10]. To illustrate this concept, graphical representation [8] of the data flow through combinational logic is used. Figures 4(a) and 4(b) show the conventional single stage system and its associated timing diagram. The combinational logic is surrounded by edge triggered input and output registers. At the beginning of each clock cycle, data is initiated into the logic block at the input register. Due to the differences in the circuit path lengths and other factors, data delay through the combinational logic will vary. In Fig.4 (b), the shaded regions bounded by the maximum and minimum delays through the logic (*Dmax* and *Dmin*) depict the flow of data through the combinational logic and the variations in the logic block with time. The nonshaded areas depict the stable duration of the logic. In the conventional system, the output register is clocked in the nonshaded region and the minimum clock period, *Tclk* is chosen to be greater than *Dmax*. In the wave-pipelined (WP) system, the clock period is chosen to be (*Dmax*-*Dmin*) + clocking overheads such as set up time, hold time etc. To ensure correct operation, the clock to the output register should be delayed so that the active clock edge occurs in the stable period. Moreover, to maximize the frequency of operation of the wave-pipelined system, the difference (*Dmax*-*Dmin*) is minimized by equalizing the path delays.

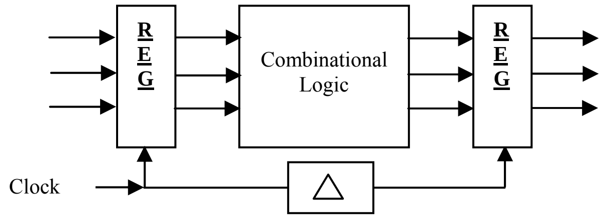

Fig.4 (a) Data flow through combinational logic circuit

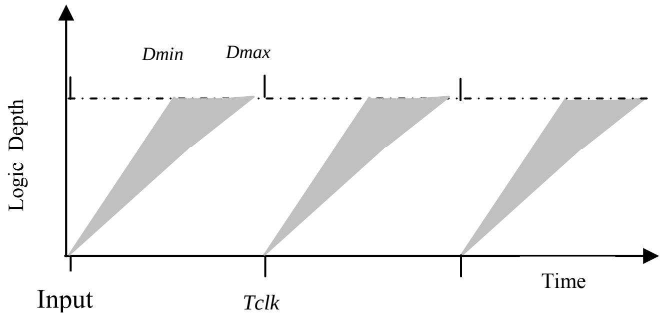

Fig.4 (b) Temporal/spatial diagram of combinational logic circuits

As the shaded region increases with increase in the logic depth, the operating clock frequency should be reduced to ensure correct operation. An alternative technique to avoid decreasing the clock frequency is pipelining. However, the need for additional registers, increases the area, power, latency and clock routing complexity. On the other hand, variation of *Dmax* and *Dmin* due to various factors such as difference in rise and fall times, variations due to process, environment and voltage changes, make the delay equalization, a challenging task in wave-pipelined systems. Several methods need to be adopted to achieve the equalization of path delays. Several methods need to be adapted to achieve the equalization of path delays. Algorithms to automatically equalize the delays in combinational logic circuit is reported in [11]. In D.Ghosh et al [12], an 8x8b wavepipelined multiplier is implemented on NPCPL and an algorithm is adapted to bring the shortest path delay equal to longest path delay. The feasibility of wavepipelining in LUT-based FPGAs has been demonstrated in [13]. In this paper, in order to analyze the feasibility of LUT-based wavepipelines, the array multiplier proposed by H.Guild [14] has been selected as case-study for the experiments. In Lakshminarayanan [15], the issues such as optimization of multiplier architectures for FPGA based wavepipelined circuits, and maximization of the operating frequency of the wavepipelined circuits are addressed.

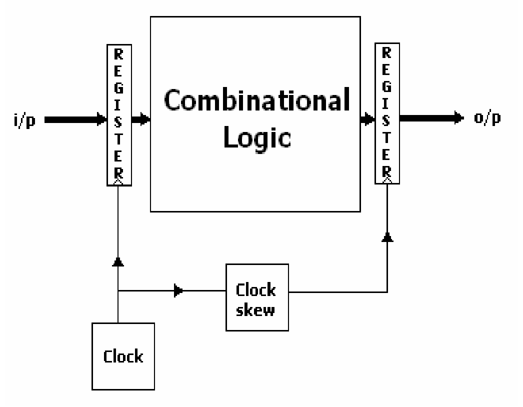

Fig. 5 Wavepipelined circuit

3.2 Design of self tuning scheme

In this section an automated approach for wavepipelined circuit is proposed. By adjusting the latching instant at the output register to lie in the stable period, the wavepipelined circuit can be made to work properly. But, for large logic depths, there may not be any stable period. Hence adjusting the latching instant by itself may not be adequate for storing the correct result at the output register. For such cases, the clock period has to be increased to increase the stable period. Equalization of path delays, adjustment of the clock period and clock skew are the three tasks carried out for maximizing the operating speed of the wavepipelined circuit. All the three tasks require the delays to be measured and altered if required.

FPGA editor may be used for measuring and altering the delays for Xilinx FPGAs. Hence wavepipelined circuit can be implemented using the Xilinx FPGAs. The wavepipelined circuit designed using the FPGA editor may be tested using simulation. However the simulation is inadequate for testing due to the difference between the actual delays and the delays calculated by the FPGA editor. This is because, the FPGA editor considers only the worst case delays and the actual delays may be significantly different due to fabrication variations. This difference becomes important as the logic depth of the circuit increases. Hence the design has to be downloaded to the actual FPGA and its operation has to be checked by feeding the test data.

Fig. 5 shows a typical wavepipelined circuit along with the input, output registers and clocking circuit which includes the clock generation and clock skewing circuit. Automation of the three tasks mentioned above are considered in [16]. The equalization of the path delays cannot be automated as the commercially available synthesis tools do not support the specification of interconnect delays. However the inequality in path delay can be minimized by proper choice of the CLBs used for mapping the design through the user constraint file submitted to place & route tool. (See for example, the structure organizer utility proposed in [17]). The clock skew and clock period can be controlled by adopting programmability. Hence a suboptimal but automated technique is proposed for the design of FPGA based wavepipelined circuits in [16]. The clock period of this circuit can be made to be lower than Dmax and hence the suboptimal approach increases the operating frequency of the combinational logic circuit. In other words, a combinational logic circuit can be operated at a higher frequency than that reported by the commercially available synthesis tools.

The automation can be carried out by including two blocks to the basic wavepipelined circuit: A FSM and a self test circuit. The finite state machine (FSM) systematically varies the clock skew and clock period till the wavepipelined circuit operates satisfactorily. The self test circuit is used for testing the correctness of the operation. The testing time can be minimized by using the optimal test vector set and a signature analyzer.

The block diagram of a self tuned wavepipelined circuit is given in Fig. 6. This is obtained by including the test vector ROM, signature analyzer, programmable clock and clock skew generators and FSM blocks to the circuit given in Fig. 5. The signature analyzer consists of a PRBS based signature generator and a comparator. The FSM block generates the control signal to choose between the normal mode and the self test mode and this is applied to the select input of multiplexer. In

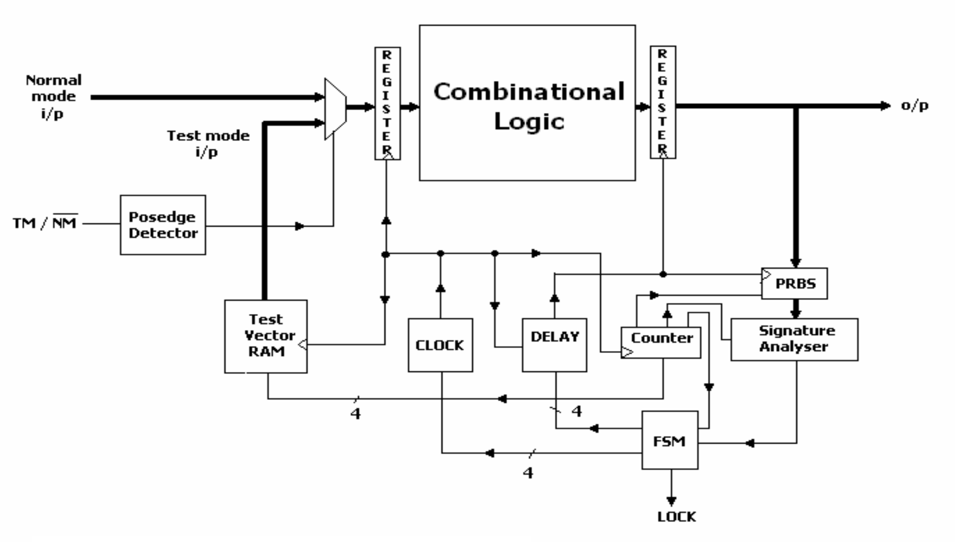

Fig.6 Self tuned Wavepipelined circuit

the self test mode, the FSM gradually increases the clock frequency and clock skew and for each set, it applies the test inputs, reads the signature analyzer-comparator output and progresses with the testing till the frequency at which the multiplier works for at least 3 or more consecutive skew values is found. The operating skew value is chosen to be the middle value so that the multiplier would reliably work even if the delays change due to environmental conditions.

The α block shown in Fig. 3 can be wavepipelined and tuned using the circuit shown in Fig. 6. Similar circuits may be used for the β, γ, δ,.ξ (ξ_1, ξ_2) blocks. However, the self tuning blocks need not be replicated. α. block may be tuned using the circuit given in Fig. 6. After determining the optimum value of clock period, β block may be tuned using the circuit given in Fig. 6. In this case the output of the α block is assumed to be fed as input to the β block. After determining the optimum value of clock period for β block, γ block is tuned assuming the input to originate from the cascaded output of α, β blocks. Similarly, δ and ξ blocks are tuned. Then the blocks (α,β,γ,δ,ξ) are interconnected using pipelining. The clock frequency should be chosen to be the smallest frequency at which all the 5 blocks function individually. For proper operation of the hybrid WP-P BW-KCM, the skew value of each individual block is to be retuned. This is carried out after downloading the design onto the device.

5 Implementation of two level 2D DWT using block RAMs

The input data for which the 2D DWT is to be computed is assumed to be split into even and odd streams and are asssumed to be stored in even and odd block RAMs respectively. In order to minimize the area required for implementation, a single horizontal filter and two vertical filters may be reused to compute the multilevel 2D DWT. The block diagram of 2 level DWT is given in Fig.7. The two level 2D DWT has 8 sub bands as shown in Fig.8. Block RAMs E1, O1 contain the even and odd streams of the initial data to be transformed. Block RAMs E2E/E2O, E3, E4, E5 denote the output of 1 level 2D DWT. The even and odd numbered coefficients of LL1 component are stored in two block RAMs E2E and E2O and are used as inputs for the 2nd level DWT. The outputs of the 2nd level DWT are stored in block RAMs E6, E7, E8 and E9. The output of the horizontal filter are stored in 4 block RAMs E10, O10, E11, O11. If LL2, the low pass band corresponding to 2 level DWT alone is required, only one demultiplexer and 7 block RAMs (E1, E2E, O1, E2O, E10, O10, E3) are required. For the purpose of verification, only LL2 is computed and compared for the different schemes of computation of two level 2D DWT. For the computation of LL1 component of 1 level 2D DWT, only block RAMs E1, O1, E2E, E2O, E10, O10 are used.

Table 1: Area and Speed performance of 9/7 bi-orthogonal filters with 11x8 multipliers using the various schemes

Multiplier (BW-KCM)	Slices	Speed (MHz)	No. of Registers	Power (mw)	Delay-Power Product (ns-mw)
Non pipelined	253	57.3	176	93.71	1639.93
Pipelined	453	149.18	803	351.31	2353.78
Hybrid WP-P	253	75.75	176	119.72	1580.30

Table 2: Area and Speed performance of one level forward 2D DWT for 32x32 sub images

Lifting scheme	Slices used	Speed (MHz)	No. of registers	Power (mw)	Delay-Power Product (ns-mw)
Non pipelined	836	54.45	611	111.50	2048.26
Pipelined	1110	87.54	1670	346.05	3951.89
Hybrid WP-P	836	75.75	611	150.74	1989.77

6 Implementation results

The implementation results of the 2D DWT computed using three multiplication schemes: Hybrid WP-P BW-KCM, non pipelined BW-KCM and BW-PKCM are given here. In order to test the functionality and for the purpose of comparison, implementation of 9/7 horizontal filters on Spartan II device XC2S150PQ208 –5 is considered first. For the implementation on Spartan II, the lifting multiplier constants (α, β, γ, δ, ξ_1, ξ_2) are assumed to be of 8 bits each and the input samples are assumed to be of 11 bits. The lifting scheme is implemented using the three multiplication schemes. For all these filters, the synthesis and timing simulation was carried out in Xilinx foundation series and the simulation results obtained for all the three cases were found to be matching. The area as well as speed corresponding to the three schemes are tabulated in table 1.

From Table 1, it may be concluded that for the filter, the method using hybrid WP-P BW-KCM is faster than non pipelined BW-KCM by a factor of 1.32 and requires the same area. The pipelined BW-PKCM is in turn faster than the hybrid WP-P BW-KCM by a factor of 1.97 and this is achieved with the increase in the number of registers by a factor of 4.6. The delay-power product for the hybrid WP-P BW-KCM is reduced by 50% compared to the pipelined BW-KCM.

The implementation of the forward 2D DWT for image block of size 32x32 is carried out for lifting scheme using the three multiplication schemes. (since the block RAM is having a size of 512x8 on the device XC2S150PQ208-5, image size is limited to 32x32). For storing, the image input, outputs of the horizontal filter and the outputs of the vertical filters, the block RAMs are configured suitably. The image is loaded into the block RAMs through the UCF of the implementation tool. The one level 2D DWT is computed using the above schemes and the results are tabulated in Table 2. From Table 2, it may be concluded that for the lifting scheme, the method using hybrid WP-P BW-KCM is faster than non pipelined BW-KCM by a factor of 1.4 and requires the same area. The pipelined BW-PKCM is in turn faster than the hybrid WP-P BW-KCM by a factor of 1.2 and this is achieved with the increase in the number of registers by a factor of 2.73. The delay-power product is lower for hybrid WP-P BW-KCM by a factor of 2 than the pipelined BW-KCM.

7 CONCLUSION:

In this paper, a novel technique for implementation of the 9/7 bi-orthogonal filters using hybrid WP-P KCM with Baugh-Wooley multiplication algorithm is proposed. This method is applicable for both ASIC and FPGA based implementation of lifting scheme. For verifying the efficacy of the scheme proposed, implementation of 1 level 2-D DWT of sub images of size 32x32 is considered on Xilinx XC2S150 device From the implementation results, it is verified that hybrid WP-P BW-KCM achieves better speed than the the non pielined scheme with the same the same area. The delay-power product is lower for hybrid WP-P BW-KCM by a factor of 2 than the pipelined BW-KCM.

The two level 2D DWT for both pipelined and non pipelined schemes are also implemented. The implementation of hybrid scheme for 2 level 2 D DWT is under progress.

The results presented in this paper can be used for the design of custom block which speed up the computation of JPEG 2000 encoder using system on a chip kit. Some details of this approach is given in [17].

References:

[1] Bruce A. D., J. Ross B., Böhm A. P. W., Charles R., and Monica C., "Accelerated Image Processing on FPGAs", IEEE Transactions On Image Processing, VOL. 12, NO. 12, DEC 2003 1543-1551

[2] D.Gibson, M.Spann, S.I. Woolley, " Wavelet based Region of interest encoder for the compression of Angiogram Video Sequences", IEEE trans. On Information technology in biomedicine, Vol. 8, No. 2, June 2004, pp 103-113

[3] N, Aydin, F. Marvasti, and H.S. Markus, "Embolic Doppler Ultrasound Signal detection using discrete wavelet transform ",IEEE trans. On Information technology in biomedicine, Vol. 8, No. 2, June 2004, pp 182-190

[4] Jörg Ritter, Paul Molitor, "A pipelined architecture for partitioned DWT based lossy image compression using FPGA's", Proc IEEE conf. FPGA 2001, pp 201-206

[5] G.Lakshminarayanan, B.Venkataramani, J.Senthil Kumar, A.K. Md, Yousuf, G.Sriram, "Design and FPGA implementation of image block encoders with 2D-DWT", Proc. IEEE region 10 (TENCON) conf. 2003, Bangalore, Oct 15-17, 2003, Vol III, pp 1015-1019

[6] I.Daubechies, W.Sweldens, "F*actoring Wavelet Transforms into Lifting Steps",* Journal of Fourier Analysis and Applications, Vol 4,Nr 3,pp 247-269, 1998.

[7] L.W.Cotton, "Maximum-rate Pipelining Systems", *Proc. Spring Joint Computer Conference AFIPS*, pp. 581-586, 1969.

[8] C. Gray, W. Liu and R. Cavin, *Wave Pipelining: Theory and Implementation*, Kluwer Academic Publishers, 1993.

[9] W. P. Burleson, M.Ciesielski, F. Klass and Liu, **"**Wave-pipelining: a tutorial and research survey", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 3, pp. 464 -474, Sept. 1998

[10] K. K. Parhi, *VLSI signal processing systems*, John Wiley & Sons, 1999

[11] D. C. Wong, G. DeMicheli, and M. J. Flynn, "Designing high-performance digital circuits using wave pipelining: Algorithms and Practical Experiences", *IEEE Transactions on computer Aided Design of Integrated Circuits and Systems*, vol.12, no.1, pages 25-46, January 1993

[12] D. Ghosh and S. K. Nandy, "Design and realization of high-performance wave-pipelined 8×8b multiplier in CMOS technology", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, no. 1 , pp. 36 -48, March 1995

[13] E.I. Boemo, S. Lopez-Buedo and J. M. Meneses, "Wave pipelines via look-up tables", *IEEE International Symposium on Circuits and Systems ISCAS '96*, vol.4 , pp. 185 -188, 1996.

[14] H.Guild, "Fully Iterative Fast Array for Binary Multiplication and Addition", *Electronic Letters*, vol. 5, no.12, pp. 263, June 1969

[15] G. Lakshminarayanan, B. Venkataramani, "Optimization techniques for FPGA based wave-pipelined DSP blocks", to be published in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 7, July 2005

[16] G. Lakshminarayanan, B. Venkataramani, M. Yousuff Shariff, T. Rajavelu and M. Ramesh, "Self tuning circuit for FPGA based wavepipelined multipliers", Proceedings of VLSI Design & Test workshop VDAT04, pp. 93-101, August 2004, Mysore

[17] J.Senthil Kumar, G.Lakshminarayanan, B.Venkataramani, G.Sriram, M.S.Jambunathan, "Design and Implementation of FPGA based Fast Multipliers with Optimum Placement and Routing Using Structure Organizer", Proc. Of National Conference on VLSI Design and Testing, PSG Tech., Coimbatore, India. Feb 21-22, 2003.

[18] G.Seetharaman B.Venkataramani V.Amutha, Anurag Saundattikar, "System on chip implementation of 2D DWT using lifting scheme", presented in the international Asia and South Pacific Conference on Embedded SOCs (ASPICES 2005), July 5-8, 2005, Bangalore.

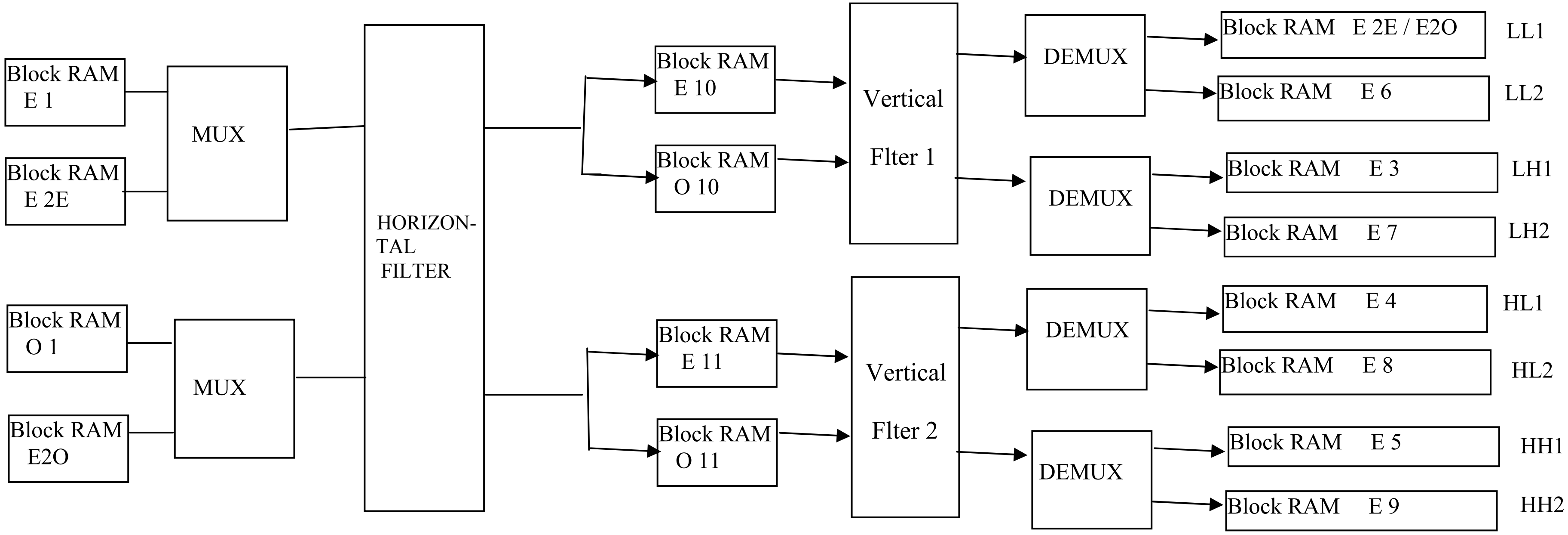

Fig. 7 Block diagram of two level DWT scheme

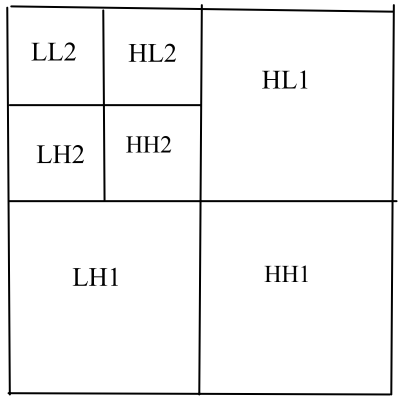

Fig.8 Sub bands of the two level 2D DWT