International Journal of Research in Advent Technology, Vol.2, No.2, February 2014

E-ISSN: 2321-9637

Design of Area-Delay Efficient VLSI Architecture for Multilevel 2D DWT

Maya.s [1]

PG student, Applied Electronics, Loyola Institute of Technology and Science,Nagercoil, India[1]

mayaece35@gmail.com[1]

Abstract: This project presents a concept to solve the computational complexity for efficient realization of 2D DWT in VLSI application. For this purpose, the proposed design scheme introduced a Memory efficient architecture for multilevel 2D DWT using a generic structure 3(K-2) M/4 , where K is the order of the filter and M is the image height. The proposed operation is based on Daubechies as well as biorthogonal filters and the structure does not involve frame-buffer. It involves line-buffers which is independent of throughput rate. The main advantage of this work is reduced Area Delay Product(ADP) and Energy Per Image(EPI).The concept can also highlight the independency of throughput which makes it convenient for higher application. The proposed structure can provide regular data flow, small cycle period(Tm) and 100% hardware utilization efficiency.

Index terms—2-D discrete wavelet transform (DWT), DWT, lifting, systolic array, very large scale integration (VLSI).

1. INTRODUCTION

The discrete wavelet transform (DWT) has become one of the most used techniques for signal analysis and image processing applications. The discrete wavelet transform (DWT) performs a multiresolution signal analysis which has adjustable locality in both time and frequency domains. Two dimensional (2-D) discrete wavelet transform (DWT) is widely used in image and video compression. The input image is required to be decomposed into multilevel DWT to achieve higher compression ratio. At present, many VLSI architectures for the 2-D DWT have been proposed to meet the temporal requirements of real-time processing. For real time image compression, DWT has to process massive amounts of data at high speeds. The use of software implementation of DWT image compression provides flexibility for manipulation but it may not meet timing constraints in certain applications. Lifting and convolution are the two computing approaches to achieve the discrete wavelet transform.

While conventional lifting based architectures require fewer arithmetic operations compared to the convolution-based approach for DWT, they sometimes have long critical paths.

1.1. Discrete wavelet transform

DWT can decompose the input samples in multiresolution. The implementation of the discrete wavelet transform is based on the filter banks. After each filtering, the number of the output samples is decimated by a factor of 2. The samples generated by the high pass filters are completely decomposed; meanwhile, the other samples generated by the low pass filters are applied to the next-level computation or further decomposition.

1.2. 2D Discrete Wavelet Transform

The basic idea of 2-D architecture is similar to 1D architecture. A 2D DWT can be seen as a 1D wavelet scheme which transform along the rows and then a 1D wavelet transform along the column. The 2D DWT operates in a straightforward manner by inserting array transposition between the two 1D DWT.Multilevel 2D DWT can be implemented by recursive pyramid algorithm (RPA) [1].But the hardware utilization efficiency is always less than 100% and it requires complex control circuits.

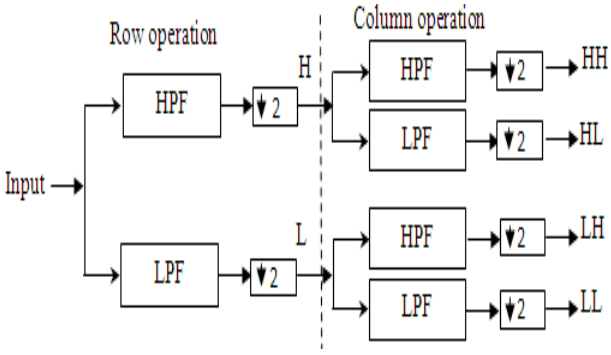

Fig 1. Block diagram of the 2-D DWT

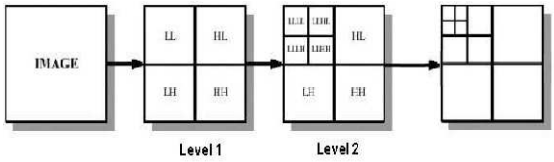

Fig 2. Multi Level Decomposition

1.3. Design Strategy

Convolution-based scheme along with appropriate scheduling of multilevel decomposition could have lower complexity than the lifting-based design[8].The parallel data acess scheme of [4] helps to reduce on-chip memory but at the same time it increases complexity of the frame buffer. The block based design [6] involves large on-chip memory and introduces significant overhead.The memory savings offered by the convolution-based scheme is significantly higher than the saving of arithmetic components. Here propose a scheme to derive a memory-efficient computing structure for multilevel 2-D DWT.

1) DWT levels are computed concurrently to avoid FB.
2) Convolution scheme is used for orthogonal as well as biorthogonal wavelet filters to derive maximum advantage of parallel data-access scheme.
3) Parallel data access is applied in each DWT level to reduce memory complexity of the overall structure.
4) Due to down-sampled filter computation, appropriate block size (*P*) need to be selected for the first level such that parallel data-access scheme could be applied to maximum possible DWT levels and 100% HUE could be achieved.

Using the proposed scheme, we have derived a parallel and pipeline architecture for the computation of three-level 2-D DWT.

Table 1.Minimum Input Block Size for Different DWT Levels

DWT Levels(J)	Input Block Size(P)
1	1
2	4
3	16
4	64

2.PROPOSED ARCHITECTURE

Because of down-sampled filtering, the computational complexity after each level of decomposition steadily decreases by a factor of four. Maximum (100%) HUE may be achieved by introducing four times more parallelism in two-level DWT computation. Similarly, for three-level DWT, we need to have 16 times more parallelism which could be too high for many applications.

This paper derived a pipeline structure for three-level 2-D DWT. The proposed structure is shown in fig.3. It consists of three processing units (PUs), where PU-1, PU-2, and PU-3, respectively, perform computations of first-level, second-level and third-level.

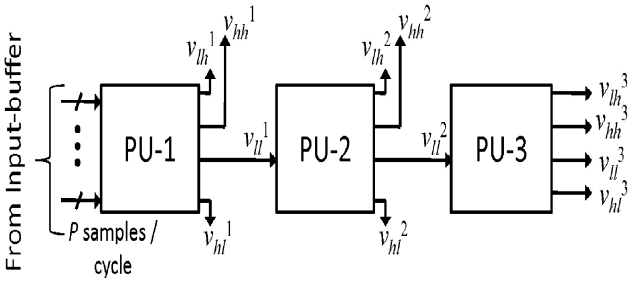

Fig. 3. Proposed structure for computation of three-level 2-D DWT

According to Table1, the minimum input block size for PU-1 is P =16. Structure of PU-1 is shown in fig. 5. It consists of eight processing elements (PEs). Each input block is extended by (K−2) samples such that adjacent input blocks of a row are overlapped by (K − 2) samples, where K is the filter order. From the input matrix (**X**), extended input blocks (**I**(m, i)) are fed to PU-1 block-by-block in every cycle. The input block **I**(m, i) corresponding to the mth row of (**X**) contains the samples {x(m, 16i), x(m, 16i + 1), . . . , x(m, 16i +K + 12), x(m, 16i + K + 13)}, for $0 \le m \le M - 1$ and $0 \le i \le (N/16) - 1$. In the first cycle, the first input block of the first row of **X** is fed then during the second cycle, the first input block of the second row is fed to PU- 1, such that the first input blocks of all the M rows of **X** are fed in M cycles and in the next set of M cycles, second input blocks of all the M rows are fed to the structure. The entire MN/16 input blocks of **X** are fed to the structure in MN/16 cycles. The input buffer is an interleaved memory, which consists of P banks (shown in Fig. 4) to read a block of P samples simultaneously.

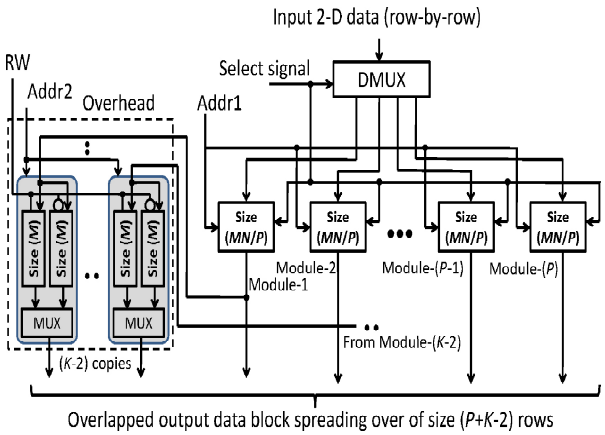

Fig.4 Structure of the input buffer using interleaved-memory

A set of eight data vectors (**B***q*+1, for $0 \le q \le 7$) of size K are derived from each input block **I**(m, i) where the adjacent data vectors are overlapped by ($K - 2$) samples. These data vectors are fed in parallel to the PEs. Structure of PE is shown in Fig. 6. It consists of a pair of identical subcells (subcell-1 and subcell-2) and one delay-unit (DU-1).

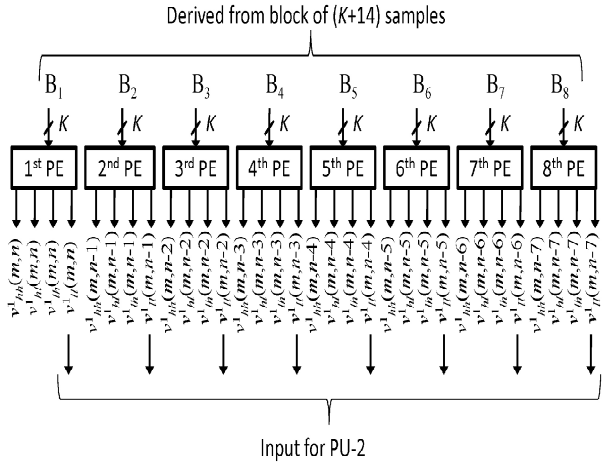

Fig.5 Structure of PU-1 for block size $P = 16$

Subcell-1 performs the necessary DWT computation along the row-direction and it generates a pair of low-pass (u_l) and highpass (u_h) intermediate components in each cycle. Successive outputs of subcell-1 belong to the same column. Subcell-2 does the processing of the intermediate matrices [U_l] and [U_h] columnwise in time-multiplexed form. The delay unit (DU-1) provides the necessary column delay to the intermediate results and feeds them into subcell-2 in time multiplexed form to perform down-sampled DWT computation. Internal structure of DU-1 is shown in Fig. 6(b).

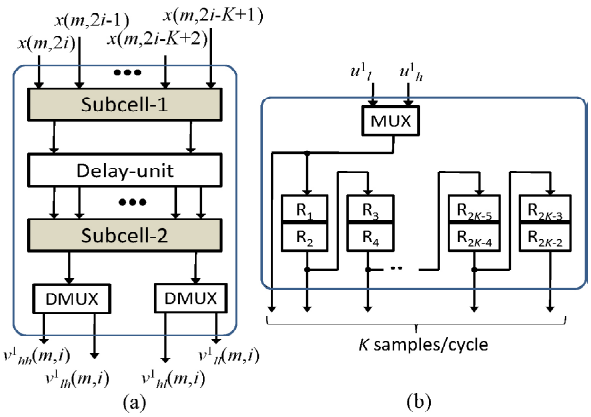

Fig.6 (a) Structure of the processing element (PE). (b) Structure of delay unit (DU-1).

Subcell-2 receives components of u_l during every even-numbered cycles and those of u_h during every-odd numbered cycles. One component of a pair subbands [v^1_{ll}, v^1_{lh}] or [v^1_{hl}, v^1_{hh}] is obtained from each PE in every cycle. First-level DWT of the entire input matrix is obtained in MN/16 cycles.

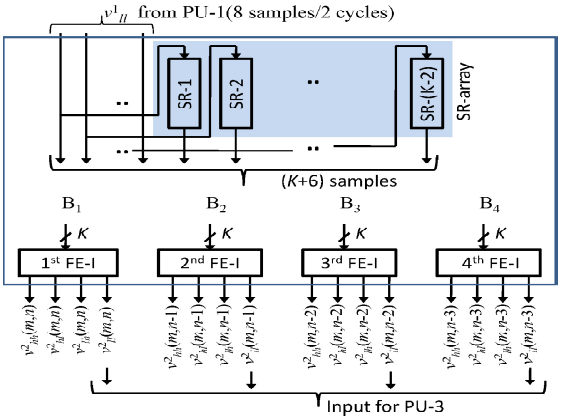

Fig. 7 Structure of PU-2

Components of low-low subband v^1_{ll} are processed by PU-2 to compute DWT components of second-level. PU-2 receives a block eight components of a particular row of v^1_{ll} from PU-1 in every alternate cycle. The structure of PU-2 is shown in Fig. 7. It consists of one SR-array and four functional elements (FEs) of type FE-I. SR-array introduces embedded down-sampling along the rows for second-level DWT computation. Each input block is extended by ($K - 2$) samples and the adjacent input block of a row is overlapped by ($K - 2$) samples. Input blocks of v^1_{ll} arrives at PU-2 columnwise in alternate cycles such that one column of the input blocks of v^1_{ll} arrives at the PU-2 in M cycles, and input blocks of the entire subband v^1_{ll} in MN/16 cycles.

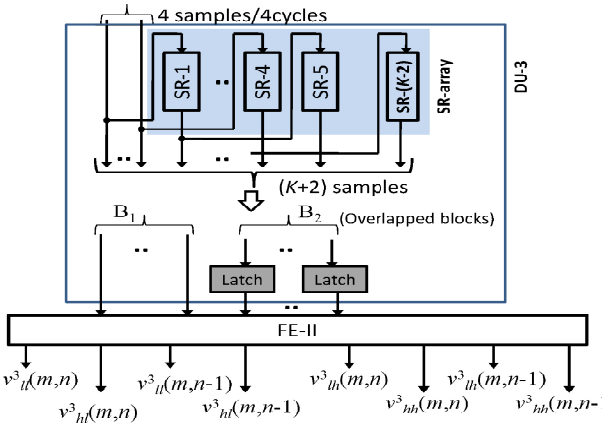

Fig. 8 Structure of PU-3

Components of subband v^2_{ll} are sent to PU-3 to compute third-level DWT. PU-3 receives a block of four components corresponding to four consecutive columns of v^2_{ll} from PU-2 in every fourth cycle. The structure of PU-3 is shown in Fig. 8. It consists of one delay-unit (DU-3) and one FE-II. An input block of v^2_{ll} arrives at PU-3 columnwise. A pair of DWT components corresponding to two adjacent columns of two subbands (v^3_{ll}, v^3_{lh}) is obtained from FE-II during even-numbered period of eight cycles. During the odd numbered sets of eight cycles, a pair of components of other two subbands (v^3_{hl}, v^3_{hh}) of the same two columns is produced. Two columns of each of four subbands are obtained in *M* cycles and subband components of the third-level DWT are obtained in *NM*/16 cycles.

Compared with the parallel structure of [6] the proposed structure requires ($PK_l/8K_0$) times less multipliers and adders and less on-chip storage.

3. EXPERIMENTAL RESULTS

The simulation results and synthesis of proposed method shows the proposed method has less delay, less area and high throughput compared to existing method.

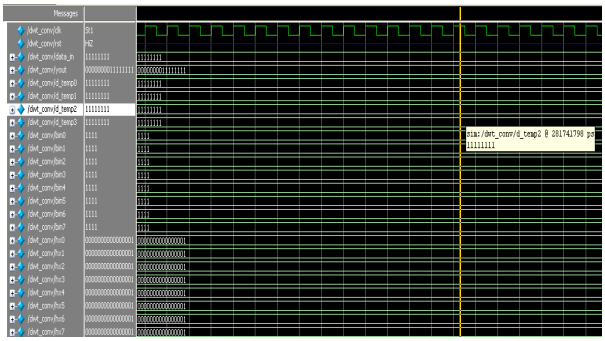

Figure 9: Simulation Results of Proposed Method

3.1. Synthesis Report

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	356	1,920	18%
Number of 4 input LUTs	892	1,920	46%
Logic Distribution			
Number of occupied Slices	542	960	56%
Number of Slices containing only related logic	542	542	100%
Number of Slices containing unrelated logic	0	542	0%
Total Number of 4 input LUTs	950	1,920	49%
Number used as logic	892		
Number used as a route-thru	58		
Number of bonded IOBs	44	66	66%
IOB Flip Flops	16		
Number of GCLKs	1	24	4%

3.2. Timing Summary

```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: 9.806ns (Maximum Frequency: 101.974MHz)
   Minimum input arrival time before clock: 6.911ns
   Maximum output required time after clock: 5.318ns
   Maximum combinational path delay: No path found
```

3.3. Map Report

```
Number of errors:     0
Number of warnings:   0
Logic Utilization:
  Number of Slice Flip Flops:        356 out of   1,920   18%
  Number of 4 input LUTs:            892 out of   1,920   46%
Logic Distribution:
  Number of occupied Slices:                       542 out of     960   56%
    Number of Slices containing only related logic:     542 out of     542  100%
    Number of Slices containing unrelated logic:          0 out of     542    0%
      *See NOTES below for an explanation of the effects of unrelated logic
Total Number of 4 input LUTs:           950 out of   1,920   49%
  Number used as logic:                892
  Number used as a route-thru:          58
  Number of bonded IOBs:               44 out of      66   66%
    IOB Flip Flops:                    16
  Number of GCLKs:                     1 out of      24    4%
```

4.CONCLUSION

The most important issue for the efficient realization of 2-D DWT is the memory complexity. This paper suggested a memory efficient design strategy for the

computation of three level 2D DWT in VLSI applications. The proposed system derived a convolution based generic structure based on Daubechies as well as biorthogonal wavelet filters. The proposed structure does not involve FB and involves line-buffers of size $3(K - 2)M/4$ which is independent of throughput rate. This is used as a major advantage when the structure is implemented for higher throughput rate. For high performance image processing applications, the proposed structure can be used as a area delay efficient and energy efficient implementation of multilevel 2D DWT.

Acknowledgments

First the Author thanks the God and also wishes to thanks the Management of Loyola Institute of Technology and Science for providing excellent computing facility and encouragement.

References

[1] M. Vishwanath, "The recursive pyramid algorithm for the discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 42, no. 3, pp. 673–676, Mar. 1994.

[2] P.-C. Wu and L.-G. Chen, "An efficient architecture for 2-D discrete wavelet transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11,
no. 4, pp. 536–545, Apr. 2001.

[3] C.-T. Huang, P.-C. Tseng, and L.-G. Chen, "Generic RAM-based architectures for 2D discrete wavelet transform with line-based method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 910–920, Jul. 2005.

[4] P. K. Meher, B. K. Mohanty, and J. C. Patra, "Hardware-efficient systolic-like modular design for 2-D discrete wavelet transform," *IEEE Trans Circuits Syst. II, Exp. Briefs*, vol. 55, no. 2, pp. 151–154, Feb. 2008.

[5] C. Cheng and K. K. Parhi, "High-speed VLSI implementation of 2-D discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 393–403, Jan. 2008.

[6] B. K. Mohanty and P. K. Meher, "Memory-efficient modular VLSI architecture for high-throughput and low-latency implementation of multilevel lifting 2-D DWT," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2072–2084, May 2011.

[7] X. Tian, L. Wu, Y.-H. Tan, and J.-W. Tian, " Efficient multi-input/ multi-output VLSI architecture for 2-D lifting-based discrete wavelet transform," *IEEE Trans. Comput.*, vol. 60, no. 8, pp. 1207–1211, Aug. 2011.

[8] C.-H. Hsia, J.-M. Guo, and J.-S. Chiang, "Improved low-complexity algorithm for 2-D integer lifting-based discrete wavelet transform using symmectric mask-based scheme," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 19, no. 8, pp. 1202–1208, Aug. 2009.

[9] C. Zhang, C. Wang, and M. O. Ahmed, "A pipeline VLSI architecture for fast computation of the 2-D discrete wavelet transform," *IEEE*

Maya.s received the B.E degree in Electronics and Communication Engineering from Marthandam College of Engineering and Technology,Marthandam,TamilNadu in 2012 and currently doing the M.E degree in Applied Electronics from Loyola Institute of Technology and Science.Nagercoil,Tamil Nadu.