VLSI DESIGN AND IMPLEMENTATION OF 2-D INVERSE DISCRETE WAVELET TRANSFORM

Paul McCanny[1], *John McCanny*[1,2], *Shahid Masud*[2]

[1]DSiP Laboratories, Asbhy Building, Stranmillis Road, Queens University of Belfast,
Belfast BT9 5AH, Northern Ireland
[2]Amphion Semiconductor Ltd, 50 Malone Road, Belfast BT9 5BS, Northern Ireland

e-mail : {p.mccanny, j.mccanny}@ee.qub.ac.uk
smasud@amphion.com

ABSTRACT

This paper proposes a JPEG-2000 compliant architecture capable of computing the 2-D Inverse Discrete Wavelet Transform. The proposed architecture uses a single processor and a row-based schedule to minimize control and routing complexity and to ensure that processor utilization is kept at 100%. The design incorporates the handling of borders through the use of symmetric extension. The architecture has been implemented on the Xilinx Virtex2 FPGA.

1. INTRODUCTION

The Inverse Discrete Wavelet Transform (IDWT) is an important part in the design of a wavelet based imaging system yet there have been very few architectures developed to compute the IDWT. This may be based on the assumption that the forward DWT (FDWT) architectures is able to compute the IDWT simply by reversing the data flow. This method, albeit simple, ignores the up-sampling and boundary techniques needed for perfect reconstruction. Hence, resulting designs are inefficient.

Typically a multi-processor scheme is favoured for IDWT computation. In this, several 1-D DWT processors are used and configured to be either row or column processors. This results in complex routing and control needed to synchronize the processors. The larger multi-processor cores have limited application when a high performance is required. Also, the advent of System On a Chip (SOC) design has meant that smaller competent cores and ease of integration are desirable characteristics.

In this paper, a single processor architecture is proposed that uses a simple scheduling algorithm to reduce the complexity of the circuit and achieve 100% processor utilization. The size and efficiency of this design is superior to existing solutions.

2. BACKGROUND

The two dimensional FDWT of an image produces four sub-bands, the LL, LH, HL and HH sub-bands. although the FDWT of the LL sub-band may be computed again to produce the (LL)LL, (LL)LH, (LL)HL and (LL)HH sub-bands. These sub-bands may be reconstructed to produce the original image, e.g. the $(LL)^J$, $(LL)^{J-1}LH$, $(LL)^{J-1}HL$ and $(LL)^{J-1}HH$ sub-band can be recombined to produce the $(LL)^{J-1}$ sub-band.

There are two methods of implementing the IDWT, a level-first scheme in which an entire LL sub-band is synthesized before the reconstruction of the next finer sub-band. If this is the case then a $N^2/4$ (where NxN is the size of the image) buffer memory is required is to store the LL data between resolutions. The direct approach can be used in this way to compute the IDWT. This computes the IDWT along the columns, storing the coefficients in memory before computing the IDWT along the rows. The advantage of this approach is that processor utilization is kept at 100% throughout the computation. The disadvantage is the large latency and storage space ($N^2/2$) required.

The alternative depth-first scheme does not wait until a sub-band is fully reconstructed before reconstructing the next. This results in a significant reduction in latency, but design complexity is increased.

Vishnawath and Owens proposed a common architecture for the DWT and IDWT [1]. In this paper parallels are drawn between the scheduling required for a depth-first computation of both the forward and inverse DWT, namely the Recursive Pyramid Algorithm (RPA) and the Inverse Recursive Pyramid Algorithm (IPRA).

Chakrabarti and Mumford proposed an architecture using the IRPA that is able to compute the wavelet

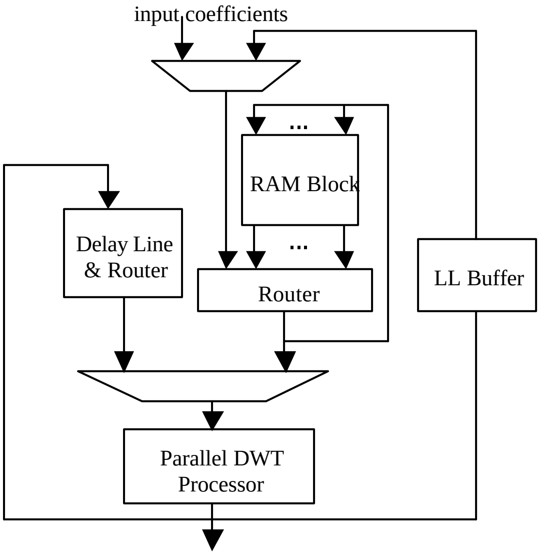

Figure 1: Proposed Architecture

transform in O(N^2) cycles [2]. This transform computes the IDWT in a row-major fashion and uses four parallel filters. Two of its filters are used for computing the DWT along the rows and two for computing the column DWT. It also requires JN/2 storage (where J is the number of octaves to be reconstructured). Two scheduling schemes are proposed in this paper which reduce latency or storage size. However the processors used in this design are, at best, active 66% of the time. This architecture also does not take into account the storage required for the column processors to store the input LL, LH, HL and HH coefficients.

A similar depth-first approach was also adopted by Yu and Chen [3]. In this architecture a scheme similar to RPA is adopted. However, this seeks to implement RPA in two dimensions simultaneously. The architecture requires just NL (where L is the maximum filter length) storage space but the complexity of the controller and routing in this design is quite high. This architecture also computes the IDWT in O(N^2) cycles.

3. PROPOSED ARCHITECTURE

The architecture proposed in this paper improves the efficiency of previous designs by using a row-based scheduling method to achieve a higher processor utilization. The results are better than what is possible using the RPA scheduling [4]. This has the effect of completely eliminating all the spare cycles in the RPA, therefore, completing the IDWT of the image in fewer cycles.

The proposed architecture is shown in Figure 1. It consists of a main memory unit, an LL buffer unit, a delay line and a single processing unit. The coefficients are input in a raster scan manner along the rows. These are input to a router and stored in the main memory unit. The coefficients are input in a specific order. In the first row the LL and HL sub-bands are input. These are interlaced, e.g. coefficient LL(x, y) is input first, immediately followed by coefficient HL(x, y), followed by coefficient LL(x +1, y) and so on. The HL and HH coefficients are input in a similar manner.

Cycle	Set			
	0	1	2	3
1	LL/LH	HL/HH	LL/LH	HL/HH
2	L	H	L	H

Table 1: Input schedule to parallel processor

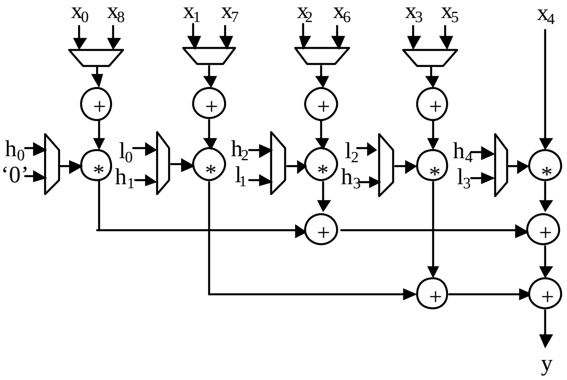

Figure 2: Parallel Processor for the 9,7 DWT

These coefficients are then input to the parallel processor so that the column IDWT can be computed. This produces the L and H coefficients, which are stored in the delay line. The schedule for the computation of the inverse DWT is shown in Table 1.

3.1. Processor Design

Two alternatives, lifting-based and filter-based, can be used when designing a DWT processor. The lifting scheme requires considerably less multipliers than required by the convolutional filter, [6]. However, many extra pipeline registers must be inserted to a lifting-based design to maintain the correct order of computation. This means that, whilst the lifting-based design is more efficient in terms of multiplications required for a serial transform, it does not translate well to a parallel implementation as the extra registers needed can result in a 50% increase in the amount of memory required. A convolutional filter translates easily to a parallel architecture with a minimum of memory but with a 50% increase in the number of

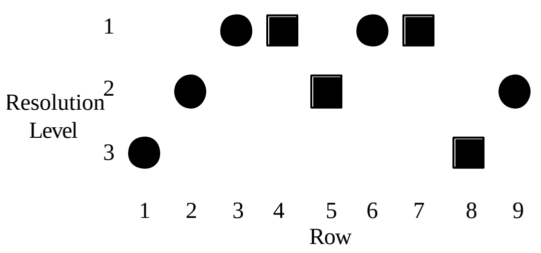

Figure 3: Row-based Input Schedule.

(A row of interlaced LL and HL coefficients are indicated by a circle, LH and HH coefficients are indicated by a square.)

multiplications required compared to the lifting-based design.

The processor is shown in Figure 2. This processor exploits the up-sampling of the IDWT by merging both the analysis coefficients of the low-pass and high-pass filter into the same structure. This enables the computation of the low and high pass outputs simultaneously. The inputs are also folded so as to exploit the symmetrical nature of the biorthogonal coefficients and therefore reduce the number of multiplications needed [7].

3.2. Multi-Resolution Operation

If more than one resolution is required then a buffer is used to store the reconstructed LL coefficients from the previous octave(s). This buffer need only consist of N/2 storage elements and can be implemented using either RAM or a register file.

The operation of the multi-resolution architecture is similar to that of the first resolution circuit. Initially the coarsest resolution (the Jth octave) is input, i.e. the LL^{J}/(LL)$^{J-1}$HL sub-bands are input for the reconstruction of the LL^{J-1} sub-band. When an LL^{J-1} coefficient is generated it is stored in the buffer until a complete row of LL^{J-1} coefficients have been output. The reconstruction of the LL^{J-2} sub-band is achieved by inputting the LL^{J-1}/(LL)$^{J-2}$HL sub-bands. Again, the LL^{J-2} coefficients are stored in the buffer until an entire row is completed.

This process continues reconstructing the next LL sub-band until the LL^{1}/HL^{1} sub-bands have been input to the parallel processor to reconstruct one full line of the image. Now the LH^{1}/HH^{1} coefficients are used to reconstruct a second line of the original image. When this is completed the (LL)^{1}LH/(LL)^{1}HH coefficients are used to generate a second row of LL^{1} coefficients. Figure 3 illustrates the scheduling method used for the row-based 2-D IDWT.

This row-based approach uses significantly less interconnections than the IRPA alternative [1] thus decreasing the complexity of the routing required. The overall processor utilization of a row-based circuit is also fixed at 100%, regardless of number of resolution levels used.

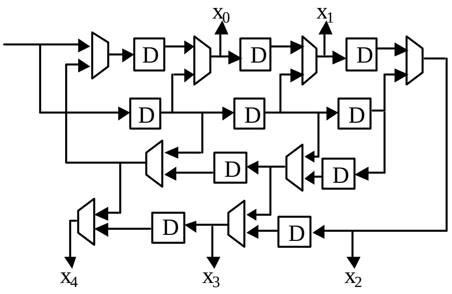

Figure 4: Serial delay line for the 5,3 DWT

Symmetric extension is used in JPEG-2000 to ensure perfect reconstruction [5]. It has been shown that processor utilization when using the RPA schedule significantly drops at the boundaries [8]. The row-based schedule used in this design performs significantly better than the RPA schedule at the image boundaries. Since one row of a single resolution is processed at a time only one row controller is required. Symmetric extension of the columns is handled by a separate controller.

3.3. Delay Line Structure

The structure of the delay line is shown in Figure 4. The delay line also incorporates a symmetric extension router necessary for compatibility with the JPEG-2000 standard [5]. The delay line consists of L output registers, as well as L auxiliary registers, for symmetric extension. This structure consists of four lines of registers, lines 1 -4. The inputs to the registers on lines 1 and 4 are input to the parallel processor. Line 2 is used to continuously receive input coefficients, thereby ensuring that there is no requirement to stall the input whilst the end of a row is being symmetrically extended. Line 3 is used as a row of buffer registers to shift data backwards so as to symmetrically extend the end of the signal. The signal is extended symmetrically at the start by routing the inputs from the coefficients stored in line 2 to the inputs of the registers in lines 1 and 4. The layout of this structure mimics the natural fold in the centre of the biorthogonal wavelet transform thereby keeping the signals close together when the inputs are applied to the biorthogonal parallel filter.

The advantages of using this structure, instead of a separate delay line and router, is that a constant data flow can be achieved. Therefore, when a new row is input the structure can store the new row while simultaneously symmetrically extending the end of the old row. This keeps the processor active 100% of the time. Also, the routing required is shorter and more regular than that required by a standard router.

Filter Pair	Res Level	Dimensions	Slices	Max Freq (MHz)	Fps
9,7	1	512x512	625	29.9	57.1
9,7	4	512x512	813	27.2	51.9
5,3	4	512x512	514	30.8	58.8
9,7	3	352x288	741	29.6	145.9

Table 2: Sample Implementation Results on Xilinx Virtex-2 XC2V250 -5 FPGA

4. IMPLEMENTATION

The circuit has been implemented on the Xilinx Virtex 2 FPGA. This device features embedded RAM and multiplier blocks, both of which were used in the design. The processor uses L/2 embedded multipliers and (L–1)*N RAMBlocks are used for the main storage unit. Another RAMBlock is used as the LL Buffer.

Symmetric Extension routers, as discussed in Section 3, are used to ensure compatibility with JPEG-2000. Sample implementation results are shown in Table 2.

5. CONCLUSIONS

An efficient architecture for computing the Inverse DWT is proposed. This architecture retains efficiency regardless of the number of resolution levels required. It is fully parameterisable in terms of resolution level, DWT coefficients, image size, data word-length and coefficient word-length. The architecture uses a row-based scheduling method to maximise processor utilization. Although the memory used in this design is comparable to that required by other designs, the scheduling used also means that routing and control is considerably less complex than that required by the RPA schedule.

The scheduling used in this architecture is flexible so that to achieve even higher performance two dedicated high and low pass processors (with constant coefficient multipliers) can be used to reduce, by half, the time taken from $O(2N^2)$ to $O(N^2)$ for reconstruction of a single octave and from $O(3N^2)$ to $O(1.5N^2)$ for a multi-resolution design. The utilization of the two processors is also kept at 100%.

6. ACKNOWLEDGEMENTS

This research has been supported by Amphion Semiconductor Ltd. and an EU funded Research Studentship.

7. REFERENCES

[1] M. Vishwanath, R.M. Owens "A Common Architecture for the DWT and IDWT", Proceedings of International Conference on Application Specific Systems, Architectures and Processors, pp 193-198, 1996

[2] C. Chakrabarti, C. Mumford, "Efficient Realizations of Encoders and Decoders Based on the 2-D Discrete Wavelet Transform" IEEE Transactions On VLSI Systems, Vol. 7, No. 3, pp 289-298, Sept. 1999

[3] C. Yu, S.J. Chen, "Efficient VLSI Architecture for 2-D Inverse Discrete Wavelet Transforms", Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 524-527, 1999

[4] P. McCanny, S. Masud, J.V. McCanny, "An Efficient Architecture for the 2-D biorthogonal Wavelet Transform Architecture", International Conference on Image Processing, Vol 3, pp 314-317, 2001

[5] M. Boliek, C. Christopoulos, E. Majani, "JPEG 2000 Part I Final Committee Draft Version 1.0", ISO/IEC JTC 1/SC 29/WG 1 N1646, March 2000

[6] W. Sweldens, "The Lifting Scheme: A custom-design construction of biorthogonal wavelets", Applied & Computational Harmonic Analysis, vol.3, no.2, pp.186-200, April 1996

[7] S. Masud, J.V. McCanny, "Design of silicon IP cores for biorthogonal wavelet transforms" Journal of VLSI Signal Processing, vol.29, no.3, pp.179-96, Nov. 2001

[8] M. Feretti, D. Rizzo, "Handling Borders in Systolic Architectures for the 1-D Discrete Wavelet Transform for Perfect Reconstruction" IEEE Transactions on Signal Processing, Vol. 48, No. 5, pp 1365- 1378, May 2000