

Rapid Design of Discrete Orthonormal Wavelet Transforms using Silicon IP Components

Shahid Masud *Student Member IEEE* and John V. McCanny *Senior Member IEEE*

DSiP™ Laboratories, School of Electrical Engineering and Computer Science,
The Queen's University of Belfast, Stranmillis Road, Belfast BT9 5AH, Northern Ireland
Email: Shahid.Masud@ee.qub.ac.uk

Abstract

A rapid design methodology for orthonormal wavelet transform cores has been developed. This methodology is based on a generic, scalable architecture utilising time-interleaved coefficients for the wavelet transform filters. The architecture has been captured in VHDL and parameterised in terms of wavelet family, wavelet type, data word length and coefficient word length. The control circuit is embedded within the cores and allows them to be cascaded without any interface glue logic for any desired level of decomposition. Case studies for stand alone and cascaded silicon cores for single and multi-stage wavelet analysis respectively are reported. The design time to produce silicon layout of a wavelet based system has been reduced to typically less than a day. The cores are comparable in area and performance to hand-crafted designs. The designs are portable across a range of foundries and are also applicable to FPGA and PLD implementations.

1. Introduction

The design for 'reuse' methodology is being considered as the only way forward for designing complex dsp hardware. This methodology reduces the time-to-market through the use of silicon cores or mega-cells in the VLSI design of a digital signal processing chip. The value of a silicon core is based on its reusability which in turn is based on the degree of parameterisation envisaged in the design process. Thus for all new applications, a component does not need to be designed from scratch leading to a reduction in design time and development expenditure and an improvement in design quality [1]. A considerable skill in the development of silicon cores is required so as to provide an efficient architecture along with comprehensive reusability. Due to this aspect, the silicon cores are also widely cited as *Intellectual Property Components*.

The description of complex circuits through VHDL has created a paradigm shift in the ASIC design process. Not only does it allow various architectural templates of dsp functional components to be described at different levels of abstraction, the components can be parameterised in a variety of dimensions and functionally tested within the VHDL environment. Also, the adoption of IEEE standard for VHDL has enhanced its portability across a range of commercial synthesis and simulation tools [2].

This paper presents the application of VHDL based 'reuse' methodology for the implementation of discrete wavelet transforms. A brief overview of wavelet transforms is presented

in section 2. This is followed by a discussion on their implementation. A rapid design strategy for wavelet transforms implementation is presented in section 3. Several case studies are reported in section 4 followed by the summary and conclusions.

2.0 Discrete Wavelet Transforms

The use of wavelet transforms is becoming increasingly popular in speech and image processing applications. The main advantages in using a wavelet transform is the non-block based signal decomposition along with the fact that the transformed coefficients are localized in both space and frequency thus providing a multi-resolution analysis.

A discrete wavelet transform performs a multi-stage signal decomposition using a filter bank structure shown in figure 1. The filter bank comprises a lowpass and a highpass filter, each followed by decimation by two. The decimated lowpass output from the preceding stage acts as the filter-bank input for the succeeding stage and so on. Accordingly, any number of stages can be cascaded to produce a wavelet based decomposition. From a practical perspective, a three stage decomposition is considered adequate in most applications.

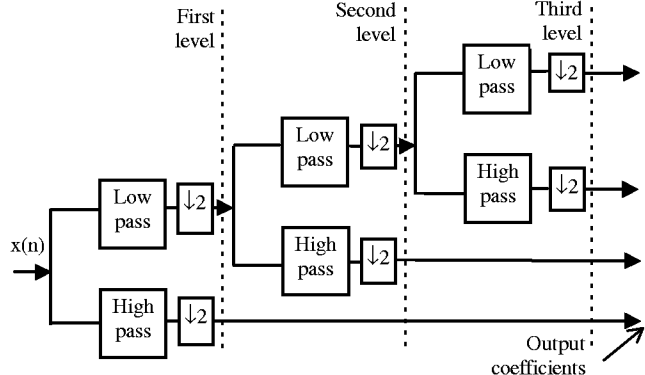


Fig 1: A three level wavelet decomposition system

The hardware implementation of discrete wavelet transforms is quite demanding in terms of circuit area requirements [3]. Also, the varying sample rate at every stage adds to the complexity of the control circuit. However, from a VLSI perspective, the interesting aspect is the similarity in architecture of the filter banks required. An efficient architecture and design of a single level decomposition allows 're-use' of the same in subsequent stages, albeit at different sample rates and different word lengths.

To date, most of the implementation schemes have been based on biorthogonal type of wavelet functions [4] mainly because symmetry in filter coefficients can be exploited to reduce the number of multipliers. Moreover, some biorthogonal wavelet filter afford integer coefficients which makes the system design even simpler [5]. Orthonormal wavelets, which are the subject of this paper, offer important signal processing properties such as regularity and higher vanishing moments which are desirable in many applications. The orthonormal filters have un-symmetric, real number coefficients which can make their implementation more complex. The purpose of this paper is to describe architectural synthesis and methods for the rapid silicon design [6, 7] of a very broad range of orthonormal wavelet transforms. A key aspect has been the development of a canonic architecture which is scaleable and can be applied to any general orthonormal wavelet transform. Previous attempts at simplifying the implementation of orthonormal wavelets have mostly been directed towards specific wavelets such as ‘Daubechies’ wavelet functions [8, 9]. Such architectures are not universal and can only be used in very specific applications.

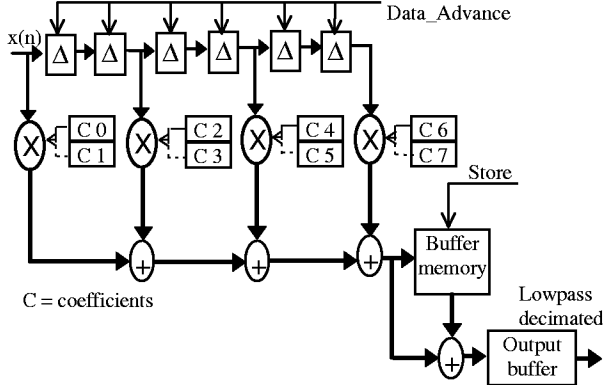


Fig 2: A time inter-leaved implementation of a 8 taps wavelet filter

References [10, 11, 12] contain a comprehensive overview of the architectural approaches for wavelet transforms. The generic architectural block developed for the illustration of our rapid design methodology is shown in figure 2. It is a canonic implementation of decimated filters and offers very high efficiency in terms of computation and area [13]. This scheme is derived from bi-phase decomposition of wavelet analysis filters through which, the even and odd samples are separately processed and the accumulated result is produced after processing two consecutive samples.

We have focused our effort on high throughput applications and consequently a bit-parallel, word-serial filter implementation has been assumed (although the architecture can also be simply extended to other word formats, such as bit-serial or digit serial data).

3. Rapid Design of Wavelet Cores

A detailed analysis of orthonormal wavelet families suggests that all of these wavelets can be suitably implemented using a filter bank structure. These wavelets differ in the choice of filter coefficients and the number of taps of the filter. A flexible and area efficient scheme must be able to scale the architecture on the fly based on generic specification of required wavelet family.

Moreover, due to the nature of wavelet transforms and the fact that new wavelet transforms are continually being reported in literature, we should be able to add a new wavelet family whenever required.

The architecture presented in figure 2 is broken into three parts as shown in figures 3 to 5. The *input* component in figure 3 is the first part of the filter. It comprises a multiplier and a delay. The delay is incorporated to synchronise the data read-in and hence it facilitates the seamless interfacing of multiple wavelet core blocks.

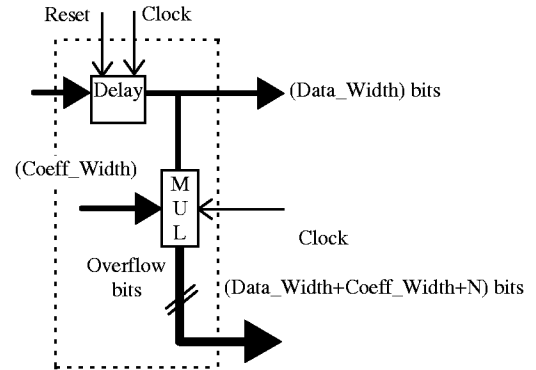


Fig 3: The ‘input’ component in Wavelets Package

The second component represents the *repeating taps* of the filter. The presence of two delays signify the difference between an ordinary FIR filter and the decimated approach that we have adopted. The other elements in this block are a multiplier/adder (MAC). This component directly interfaces to the *input*, *output* or an identical block.

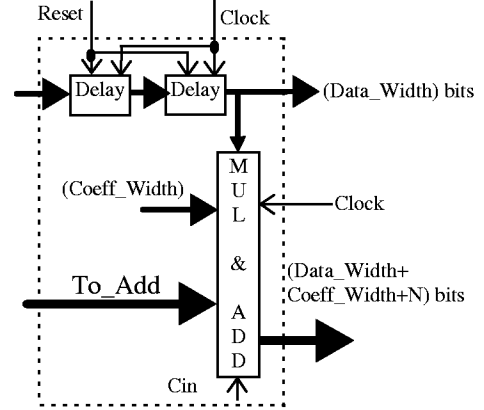


Fig 4: The ‘repeating taps’ component in Wavelets Package

The *output* block represents the accumulator and decimator. The output is only available when two samples have been processed.

The control circuit and coefficient scheduling mechanism is a part of the core block. The only external signals required for complete functioning are *Clock* and *Reset*. The *Clock* signal also directly determines the sample frequency.

The methodology allows us to rapidly explore the design space by incorporating appropriate multiplier / accumulator components from a synthesizable library. Means are provided whereby specific hardware instantiations captured

[illegible]

Each of these functional parts has been parameterised in terms of word length and overflow bits. These correspond to sign extension in 2's complement arithmetic. The blocks were developed using synthesizable components from VHDL libraries supplied by ISS Ltd [6]. Specifically, a Booth encoded, Wallace tree multiplier, a carry-save adder and asynchronous clear D-flip-flops were used to develop these core processing blocks. The architecture based on these synthesizable blocks can be extended during synthesis to cater for any wavelet family and type.

A MATLAB code was used to generate a text file containing the coefficients for any desired wavelet. Some of the commonly used wavelet coefficients are now available over the Internet [14]. These can be directly used in our scheme. The use of a numerical package offers un-limited possibilities regarding the choice and future development of wavelets. The text file containing real number coefficients is converted to 2's complement format at 16 bits resolution using VHDL code. The coefficients are then appropriately placed in a VHDL package. The required word length specified in the generic description is chosen at the time of instantiation. Each wavelet is identified by a separate number which is specified in generics to access those coefficients. Addition of a new wavelet is trivial and involves repeating these steps. Table 1 shows a few entries in the coefficients package.

Generic Number	Wavelet Type
1	Daubechies 8 taps
2	Symmlet 12 taps
3	Daubechies 4 taps
4	Morlet
5	Coiflet1
6	Coiflet2

Numerous design examples have been undertaken to illustrate the ease and speed with which wavelet transforms can be implemented. The methods presented allow a non-specialist DSP engineer to develop a silicon implementation of a full wavelet system in parallel with algorithm development. The following generic parameters are all that need to be specified at the time of instantiation of any wavelet block:

1. Data word length,
2. Coefficient word length,
3. Word length extension to prevent overflow (and to cater for cascading stages),
4. Wavelet family,
5. Values for taps (or type) of the filter required.

The block diagram illustrates the proposed architecture for wavelet transform. It consists of the following components and data flow:

- Input:** The input signal enters the system.
- MUX (Multiplexer):** Receives the input and directs it to either the Wavelet Transform Processor or the Memory block.
- Wavelet Transform Processor (Core Block):** The central processing unit that performs the wavelet transform. It outputs **Detail Coefficients** and sends data to the Memory and Switch blocks.
- Memory:** Stores data and provides it to the Switch block.
- Switch:** Routes data from the Memory and Wavelet Transform Processor to the **Approximation** output and the **Qr[]** block.
- Qr[]:** A register or queue that receives data from the Switch and feeds back into the MUX.
- Control and address generator:** Provides control signals to the MUX, Wavelet Transform Processor, Memory, and Switch.

4.1 Daubechies 4 taps wavelet analysis, three stages

4.2 Symmlet 12 taps wavelet analysis, two stages

In this case a two level wavelet decomposition based on a Symmlet 12 taps wavelet was synthesised. The coefficient resolution in 2's complement format was set at 10 bits and the data resolution was 9 bits. The internal word length accuracy was 25 bits and 32 bits for the first and second stage

respectively using a 9 bit truncation between the two stages. The silicon area required in this case is 2.19 mm^2 ($1.19 \text{ mm} \times 1.84 \text{ mm}$). The core requires 110 K transistors ($\sim 35 \text{ K}$ gates). Typical power dissipation for this core at 100 MHz data throughput is around 600 mW.

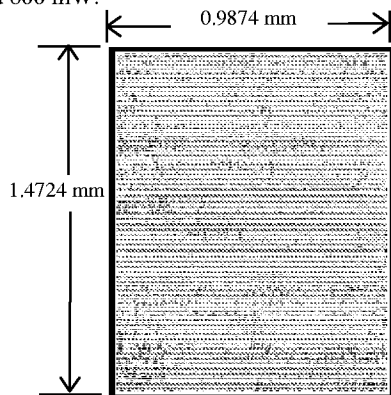


Fig 7 : The silicon layout for Daubechies 4 taps, 3 level wavelet analysis

4.3 Rapid Prototyping on FPGAs

The layout for a Xilinx 4000 series FPGA was produced using fpga_analyzerTM and Design ManagerTM toolsets. The total time to generate the layout was less than a day. The design comprises 353 Configurable Logic Blocks (CLB) and 50 Input/Output Blocks (IOB) for 8 bits data and 7 bits coefficient resolution for Daubechies 8 taps wavelet analysis. The clock rate targeting an XC4025E, speed grade-4 device was found to be of the order of 30 MHz. This is sufficient for carrying out a three level wavelet transform of a 256×256 pixels image at 25 frames/sec using the configuration shown in figure 6.

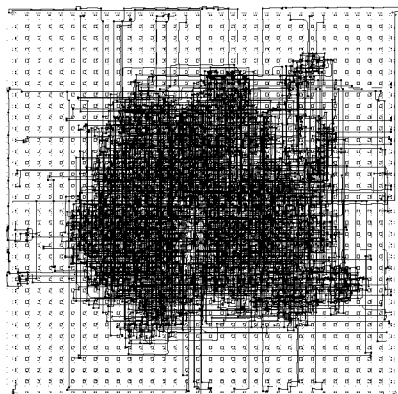


Fig 8: The XC4025-4 FPGA floorplan for Daubechies 8 taps wavelet transform processor

5. Conclusion

A method for the rapid design and implementation of discrete orthonormal wavelet transforms has been developed. The approach allows a non-specialist to implement a wide range of wavelet transform based systems directly in silicon. A new wavelet family can easily be added whenever required to give additional flexibility to the design space. The implementation is independent of silicon technology and can easily be ported to

FPGAs and PLDs. An efficient architecture for the decimated filters was chosen as the basis of the generator presented. This has been parameterised in terms of word lengths so that the same block can be 're-used' in a multi stage wavelet analysis system. The results obtained from three synthesis examples given in the paper demonstrate the effectiveness of this approach.

6. Acknowledgement

The support from the Commonwealth Scholarship Commission and the Northern Ireland Industrial Research and Technology Unit under the Technology Development Programme is gratefully acknowledged.

7. References

- [1] M.S.B. Romdhane, V.K. Madiseti, J.W. Hines, 'Quick-Turnaround ASIC Design in VHDL Core Based Behavioral Synthesis', Kluwer Academic Publishers, 1996
- [2] IEEE Design and Test of Computers, April 1986, Special Issue on VHDL
- [3] O. Rioul, P. Duhamel, 'Fast Algorithms for Discrete Wavelet Transforms', IEEE Transactions on Information Theory, March 1992, pp 569-586
- [4] 'Biorthogonal Wavelet Filter Megafunction', Altera Corp, Feb 1997, <http://www.altera.com>
- [5] G. Lafruit, B. Vanhoof, J. Bormans, M. Engels, I. Bolsens, 'Implementation Aspects of FIR Filtering in a wavelet Compression Scheme', Proceedings IWISP, Nov 1996, pp 521-524.
- [6] J.V. McCanny, D. Ridge, Y. Hu, J. Hunter, 'Hierarchical VHDL Libraries for DSP ASIC Design', Proc. IEEE ICASSP, Munich, 1997, pp 675-678
- [7] J.V. McCanny, D. Trainor, Y. Hu, T.J. Ding, 'Rapid Design of Complex DSP Cores', invited paper, Proc. IEEE European Solid State Circuits Conference, Southampton, UK, Sept 1997, pp 284-287
- [8] A. Klappenecker, V. Baumgarte, A. Nückel, T. Beth, 'Methods for Regular VLSI Implementation of Wavelet Filters', Proceedings SPIE, 1996, pp 443-454
- [9] A.S. Lewis, G. Knowles, 'VLSI Architecture for 2-D Daubechies Wavelet Transform without Multipliers', Electronics Letters, Jan 17, 1991, pp 171-173
- [10] M. Vishwanath, R.M. Owens, M.J. Irwin, 'VLSI Architectures for the Discrete Wavelet Transform', IEEE Transactions on Circuits and Systems II, Vol 42, 1995, pp 305-316
- [11] C. Chakrabarti, M. Vishwanath, R.M. Owens, 'Architectures for Wavelet Transforms : A Survey', Journal of VLSI Signal Processing, 1996, pp 171-192
- [12] K.K. Parhi, T. Nishitani, 'VLSI Architectures for Discrete Wavelet Transforms', IEEE Transactions on VLSI Systems, June 1993, pp 191-202
- [13] J. McCanny, J. McWhirter, E. Swartzlander Jr., Editors, 'Structural Optimization for Video Rate FIR Filters', Systolic Array Processors, Prentice Hall, 1989, pp 360-368.
- [14] Video coding group at Univ. of Bath, UK, <http://dmsun4.bath.ac.uk/wavelets/>