FPGA-based discrete wavelet transforms design using MatLab/Simulink

Gavrincea Ciprian, Tisan Alin, Oniga Stefan, Buchman Attila

Electrotehnic Department
North University of Baia Mare
Baia Mare, Romania
gcg@ubm.ro

Abstract

This paper presents theoretical and practical aspects in conjunction with hardware implementation of wavelet transform. In the last decade, discrete wavelet transforms have shown to be an effective tool for signal and image processing. The paper analyzes different solution for hardware implementation of wavelet transform in FPGAs using MatLab-Simulink environment.

1. INTRODUCTION

During the last several years the wavelet transform (WT) has emerged as an important signal processing research tool. Wavelet transform theory as it relates to filter banks was refined by several researchers in the late 1980s and early 1990s, including Mallat (1989), Daubeehies (1992), Vetterli and Herley (1992), and Vaidyanathan (1993). Though work on wavelet theory has been extensive, research into wavelet transform applications is still in its infancy. Of particular interest is the use of the wavelet transform for the analysis of transient signals, since it is seen as an improvement over the traditionally used short-time Fourier transform (STFT). This is based on the fact that by using different scales of the analyzing wavelet, the wavelet transform is able to locate in.

There are several types of wavelet transforms and, depending on the application, one may be preferred to the other. For a continuous input signal, the time and scale parameters can be continuous leading to the Continuous Wavelet Transform (CWT). The wavelet transform can be defined for discrete time signals leading to the Discrete Wavelet Transform (DWT). In the latter case it uses multirate signal processing technique and is related to subband coding schemes used in speech and image compression.

2. WAVELET TRANSFORM

The basic idea underlying wavelet analysis consists of expressing a signal as a linear combination of a particular set of functions obtained by shifting and dilating one single function called a mother wavelet. Several different mother wavelets have been studied in Daubechies (1988) and Meyer (1989). The decomposition of the signal into the basis of wavelet functions implies the computation of the inner products between the signal and the basis functions, leading to a set of coefficients called wavelet coefficients. The signal can consequently be reconstructed as a linear combination of the basis functions weighted by the wavelet coefficients. In order to obtain an accurate reconstruction of the signal, a sufficient number of coefficients have to be computed.

A fundamental property of the wavelet transform (WT) is that the time resolution and frequency resolution vary in the time- frequency plane. The continuous wavelet transform allows a variable coverage of the time-frequency plane. The transform is defined as:

$$CWT_x(\tau,a)=\frac{1}{\sqrt{a}}\int x(t)\Psi^*\left(\frac{t-\tau}{a}\right)dt \qquad (1)$$

where a is called the scaling factor, τ is the translation parameter, and Ψ is the window function or wavelet.[2]

2.1. Discrete wavelet transform

Though the CWT is useful for the mathematical derivation of wavelet transform theorems and properties, in computational applications (where signals and filters are discrete), it is the discrete wavelet transform (DTWT) that is used.

ISBN 978-973-713-188-1

Discrete wavelet transform can be implemented as a set of filter banks comprising a high-pass and a low-pass filters, each followed by downsampling by two. The low-pass filtered and decimated output is recursively passed through similar filter banks to add the dimension of varying resolution at every stage. This is mathematically expressed:

$$DWDT_{x(n)} = \begin{cases} d_{j,k} = \sum x(n) h_j^*\left(n - 2^j k\right) \\ a_{j,k} = \sum x(n) g_j^*\left(n - 2^j k\right) \end{cases} \quad (2)$$

The coefficients aj,k and dj,k refer to the approximation and detail components in the signal, respectively. The functions h(n) and g(n) in this equation represent the coefficients of the high-pass and the low-pass filters.[2]

2.2. Wavelet analysis

The wavelet transform processes a signal by decomposing it into successive approximation and detail signals. The approximation signal is re-sampled at each stage, and the detailed coefficients are kept. For decomposition into J scales, the transform coefficients consist of J scales of detailed coefficients and the Jth scale approximation coefficient. The process of signal decomposition using wavelet transform is called wavelet analysis.

Figure 1 illustrates one stage from a wavelet analysis circuit. On each stage, input data is passed through a low pass filter and a high pass filter. The output of the low pass filter provides approximation signal, while the output of the high pass filter provides detailed signal. Because output of each filter has the same number of sample as the original signal, detailed and approximation signals are decimated by 2.

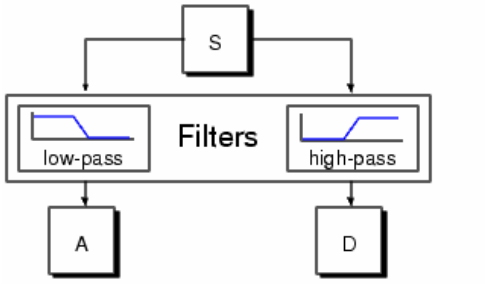

Fig. 1. Wavelet analysis concept.

2.3. Wavelet synthesis

The process of reconstruction of the original signal as a linear combination of the detailed and approximation coefficients is called wavelet synthesis. Figure 2 illustrates the concept of a full wavelet transform operation. To synchronize the signals obtained from analysis, each detailed signal from each scale has to be delayed.

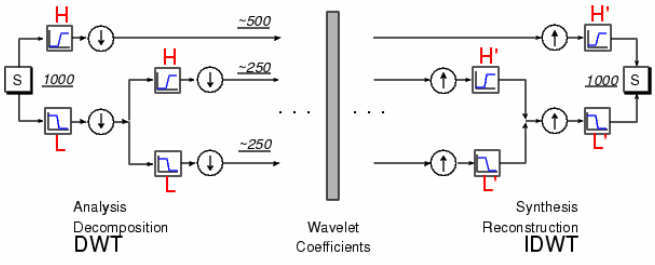

Fig. 2. Wavelet analysis-synthesis concept.

3. HARDWARE IMPLEMENTATION

Perfect reconstruction filters are used to perform analysis and synthesis of the signal. Filter coefficients can be obtained from Matlab environment. For this study we use a four level decomposition using Daubechies 4 wavelet. FIR filters are implemented using a dual ram memory for storing filter coefficients and data. The solution for the hardware implementation of the FIR filter is illustrated in figure 5.

The main blocks for implementation of FIR filter consist in a dual ram memory for storing filter coefficients and data, a multiplier, an accumulator and a control block. Input data are stored in ram memory on address range 0 to n, where n represents filter length. Filter coefficients are stored on address range n to 2n-1. Addresses for dual ram block are generated by the control block. Beside addresses, control block is used to generate write enable signal for memory block, reset signal for accumulator and enable signal for output register.

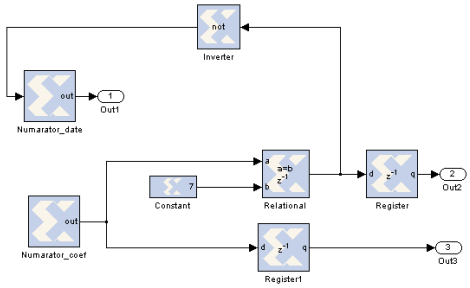

Fig. 4. Control logic block

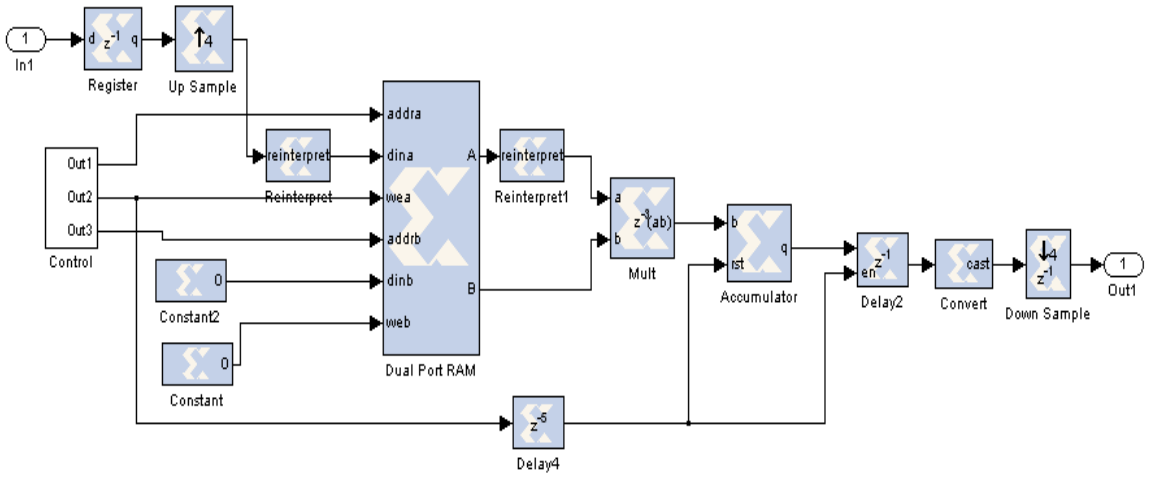

Fig. 5. Hardware implementation of FIR filter

Table 1 presents estimated resources needed to implement a 4 scale wavelet filter, on XILINX Spartan3E XC3S500E. In order to save hardware resources is best to use hardware multiplier. Because FPGA chips have a small number of hardware multipliers, a careful management of this type of resources is required.

Resources	Spartan3 HW multipliers	%	Spartan3	%
Slice	2372	51	4340	93,2
FFs	3616	38,8	7008	75,2
Block RAM	16		16	
LUT	2104	22,6	6640	71,3
Multiplier	16/20	80	0	0

Tab. 1. Hardware resources.

Another solution for the implementation of wavelet decomposition is presented in figure 6. Both filters from the same stage of decomposition are implemented using a single multiplier. This can be accomplish by storing two sets of filter coefficients in the same memory block (one set for low pass filter and one for high pass filter).

On wavelet decomposition stage, same input data are passed through a low pass and a high pass filter. One multiplier can be used for each stage if both filter coefficients are store in the same memory and input data is being processed sequentially. Two accumulators are used to separate the result for each filter. The control signals are generated using a more complex logic circuit.

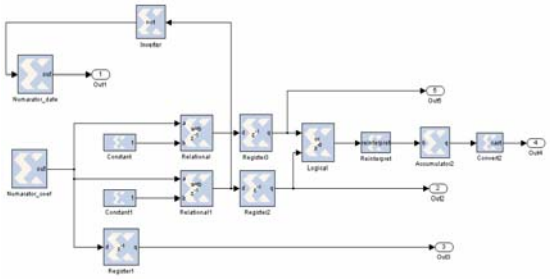

Fig. 5. Control logic block

Table 2 presents estimated resources needed to implement a 4 scale wavelet filter, using the method presented above, on XILINX Spartan3E XC3S500E. In case of implementation using built-in hardware multiplier the number of multipliers and ram blocks needed to implement the design is being reduced by 25%. Also the rest of the logic is being reduced by roughly 8%. If the design is being implemented without built-in hardware multiplier, the hardware resources are being reduced by 20%.

Resources	Spartan3 HW multipliers	%	Spartan3	%
Slice	1980	42,5	3440	73,8
FFs	2884	30,9	5436	58,3
Block RAM	12		12	
LUT	1952	20,9	5104	54,8
Multiplier	12/20	80	0	0

Tab. 2. Hardware resources.

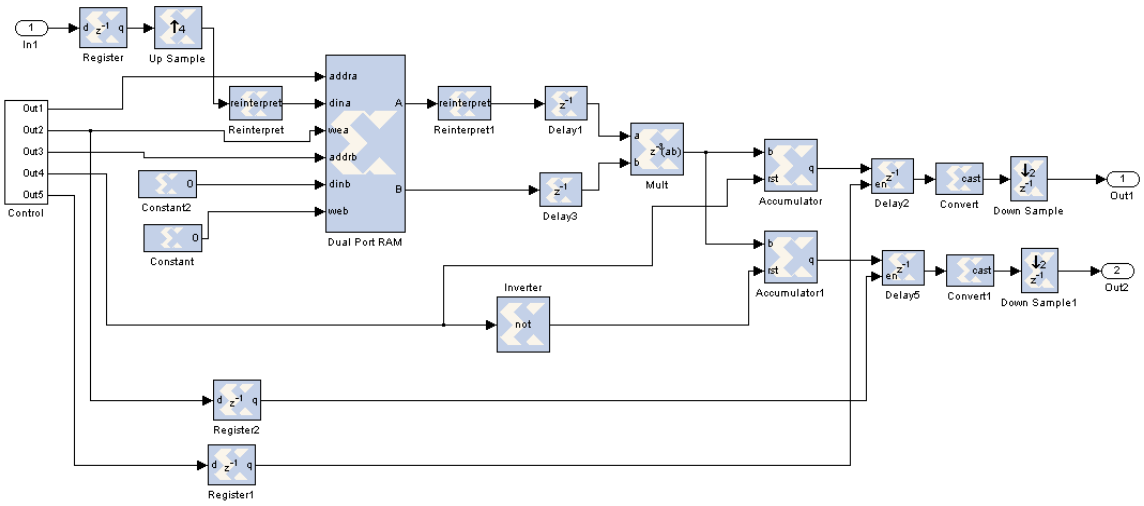

Fig.6. Wavelet decomposition with 1 multiplier

3. CONCLUSION

Wavelets are used to analyze signals in much the same way as complex exponentials (sine and cosine functions) used in Fourier analysis of signals. The compactness and finite energy characteristic of wavelet functions differentiate wavelet decompositions from other Fourier like analysis in their applicability to different circumstances. Wavelet functions not only can be used to analyze stationary signals but also it can be used to decompose nonstationary, time-varying or transient signals.

Design of current DSP applications using state-of-the art multi-million gates devices requires a broad foundation of the engineering skills ranging from knowledge of hardware-efficient DSP algorithms to CAD design tools. The requirement of short time-to-market, however, requires replacing the traditional HDL based designs by a MatLab/Simulink based design flow. This not only allows MatLab users to design FPGAs but also to by-pass the hardware design engineer leading to a significant reduction in development time.

Simulink design flow for FPGAs is an interesting alternative both for a University lab as well as for the professional developers in industry. The design flow allows a software developer to quickly explore FPGA design options in terms of size and speed and to check if the resulting design fulfills the design constrains.

REFERENCES

[1] D. Moshou, I Hostens, G Papaioannou, H Ramon, "Wavelets and self-organising maps in electromyogram analysis", *ESIT 2000*, 14-15 September 2000, Aachen, Germany

[2] C. S. Burrus, R. A. Gopinath, *Introduction to Wavelets and Wavelet Transforms*, Prentice Hall Inc.,Upper Saddle River, New Jersey, 1998.

[3] Yousef M. Hawwar_, Ali M. Reza_, Robert D. Turney, „Filtering (denoising) in the wavlet transform domain",

[4] S.G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation", IEEE Trans.Pattern Anal. Machine Intell., vol. 2, pp. 674-693, 1989.

[5] S. Masud and J.V. McCanny, "Rapid design of biorthogonal wavelet transforms", IEEE Proceedings of Circuits, Devices and Systems, Volume: 147 Issue: 5, 2000, pp: 293 -296