Signal Denoising by Wavelet Packet Transform on FPGA Technology

Mohamed I. Mahmoud, Moawad I. M. Dessouky, Salah Deyab, and Fatma H. Elfouly

***Abstract*—** A denoising method based on wavelet packet shrinkage was developed in this research. The principle of wavelet packet shrinkage for denoising and the selection of thresholds and threshold functions were analyzed. The design of a low-cost, field programmable gate array (FPGA) based digital hardware platform that implements wavelet packet transform algorithms for real-time signal de-noising is presented.

***Keywords*—** wavelet packet transform, denoising, FPGA.

I. Introduction

ALL signals obtained as instrumental response of analytical apparatus are affected by noise. The noise degrades the accuracy and precision of an analysis, and it also reduces the detection limit of the instrumental technique. Signal denoising is therefore highly desirable in analytical response optimization.

For the applications of interest, noise is primarily high frequency, while the signal of interest is primarily low frequency. Because the wavelet transform decomposes the signal neatly into approximation (low frequency) and detail (high frequency) coefficients, the detail coefficients will contain much of the noise. This suggests a method for de-noising the signal: simply reduce the size of the detail coefficients before using them to reconstruct the signal. This approach is called *thresholding or shrinkage* the detail coefficients. Of course, we cannot throw away the detail coefficients entirely; they still contain some important features of the original signal. Various kinds of thresholding have been proposed, and which kind of thresholding is best depends on the application. The two different approaches which are usually applied to denoise: hard thresholding or soft thresholding. The hard thresholding method consists in setting all the wavelet coefficients below a given threshold value equal to zero, while in soft thresholding the wavelet coefficients are reduced by a quantity equal to the threshold value [5]. A generalization of the discrete wavelet transform is the discrete wavelet packet transform (DWPT) which keeps splitting both lowpass and highpass subbands at all scales in the filter bank implementation, thus Wavelet Packet obtains a flexible and a detail analysis transform. So we used the Wavelet Packet transform for de-noising.

Signal de-noising using wavelet packet transform consists of the following three steps:

The main steps of signal denoising are:

1. Wavelet packet transform of observed signal.
2. Shrinkage of the empirical wavelet coefficients.
3. Inverse wavelet packet transform of the modified coefficients.

The denoising procedure requires the estimation of the noise level. In this work Stein's Unbiased Estimate of Risk (SURE) [6] has been chosen as a principle for selecting a threshold to be used for denoising.

Previous research on signal de-noising using wavelet is off-line in nature; the signal is sampled in real-time, but then captured in memory or on hard disk, and de-noised after the fact on a traditional personal computer or workstation using a software tool such as Matlab. However, many applications require *real-time processing*, in which the signal must be processed as it is received. These real-time applications require that the signal be processed at the same rate that it is produced; in other words, the throughput in samples per second of data coming out of the de-noising system must be equal to the throughput of data going into the system. A small amount of latency, or lag from input to output, is acceptable (and necessary, since computations can not be done instantaneously). The goal of this research is to demonstrate that signal de-noising can be done in real-time efficiently and inexpensively by using a field programmable gate array as the computational platform.

The rest of the paper is organized as follows. Section II describes the wavelet packet algorithm; Section III explains why FPGA are an appealing choice for implementation of the de-noising part of the system. Section IV describes the denoising principle. Section V details our FPGA implementation of signal denoising. Section VI gives the simulation results. Section VII gives the synthesis results. And Section VIII draws conclusions.

II. Wavelet Packet Algorithm

M. I. Mahmoud, M. I. M. Dessouky, S. Deyab are with Faculty of Electronic Engineering, Menouf, Egypt.

F. H. Elfouly is with HIE, Alshorouk academy, Cairo, Egypt.

Wavelet Packet Transform (WPT) is now becoming an efficient tool for signal analysis. Compare with the normal wavelet analysis, it has special abilities to achieve higher discrimination by analyzing the higher frequency domains of a signal. The frequency domains divided by the wavelet packet can be easily selected and classified according to the characteristics of the analyzed signal. So the wavelet packet is more suitable than wavelet in signal analysis and has much wider applications such as signal and image compression, denoising and speech coding [7].

Wavelet packet transform uses a pair of low pass and high pass filters to split a space corresponds to splitting the frequency content of a signal into roughly a low-frequency and a high-frequency component. In wavelet decomposition we leave the high-frequency part alone and keep splitting the low-frequency part. In wavelet packet decomposition, we can choose to split the high-frequency part also into a low-frequency part and a high-frequency part. So in general, wavelet packet decomposition divides the frequency space into various parts and allows better frequency localization of signals [7].

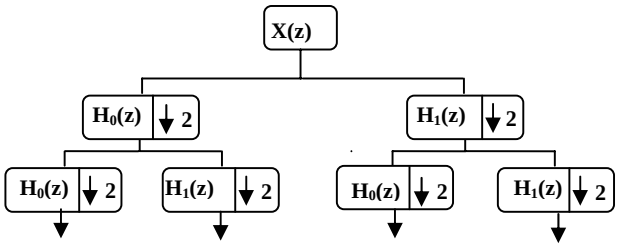

Fig. 1 Wavelet packet tree

As shown in Fig. 1, the wavelet packet transform can be viewed as a tree. The root of the tree is the original data set. The next level of the tree is the result of one step of the wavelet transform. Subsequent levels in the tree are constructed by recursively applying the wavelet transform step to the low and high pass filter results from the previous wavelet transform step [7]. Similarly the inverse wavelet packet can reconstruct the original signal from the wavelet packet decomposition spectrum. The inverse wavelet packet is done starting from the coarsest decomposition level where the WPT coefficients are upsampled before passing through a pair of reconstruction filters. Note that, the wavelet that is used as a base for decomposition cannot be changed if we want to reconstruct the original signal.

Daubechies 18-tap wavelet has been chosen for this implementation. The filters coefficients corresponding to this wavelet type are shown in Table 1.

III. Advantages of FPGA-based implementation.

Several computer hardware platforms can be considered for processing of signals from optical imaging systems; traditional choices for implementing such a system are a microprocessor, a digital signal processor, or an application-specific integrated circuit (ASIC). Microprocessors and digital signal processors offer the advantage of being inexpensive, off-the-shelf devices, easily programmed to perform a variety of tasks. On the other hand, an ASIC, while expensive to design and fabricate and inherently inflexible once the design is complete, offers an advantage in terms of processing speed [8].

Recent advances in FPGA technology have made FPGA extremely attractive for implementation of all types of computational systems. FPGA represent a new middle ground between microprocessors and ASICs in terms of computational performance and cost. Like microprocessors, FPGA are inexpensive, off-the-shelf, and easily reprogrammed for new applications [8]. Like ASICs, FPGA offer a high degree of control over the underlying computer hardware, and therefore allow the system designer to specify hardware architecture tailored to the application at hand, thus providing additional processing speed. Once relegated to small "glue logic" applications, FPGA are now capable of implementing complex computational systems. In the last few years, systems have been built or proposed for a variety of applications dominated by mathematical computations, including a cross-correlator for radio astronomy, a sonar beam former, one- and two-dimensional convolvers [8], a decimation filter, and a fast Fourier transform. This prior research shows that FPGA -based implementations are typically at least one order of magnitude faster than processor-based implementations, without incurring the high cost of fabrication and development required for application specific integrated circuits.

IV. Denoising Principle

A. Model of Noise-containing Signals and Principles of Denoising Based on Wavelet Packet Shrinkage

In engineering, a one-dimensional model of signals with additive noises can be shown as follows:

$$y(n) = x(n) + \sigma\, e(n), n = 1,2,..., N \qquad (1)$$

Where, y(n) denotes noise-containing signals, x(n) denotes real signals, e(n) is white Gaussian noises with a normal distribution, and N (0,1) denotes the deviation of noise signals. In engineering, the useful real signals usually behave in the form of low-frequency signals or certain relatively stable signals, while noise signals are usually in the form of high-frequency signals. Signal x(n) can be depicted by wavelet packet coefficients decomposed from wavelet packet, with larger wavelet packet coefficients carrying more signal energy and smaller carrying less [8,9]. The basic idea of denoising with wavelet packet shrinkage is (according to the characteristic that wavelet packet coefficients of noises and signals) behaves differently in different scales (namely, different bands). To eliminate

wavelet components of different scales produced by noises, especially components of noise-dominated scales, and the preserved wavelet packet coefficients are the very wavelet packet coefficients of original signals, then the original signals are reconstructed via the wavelet packet transform reconstruction algorithm. Therefore, we know the key to denoising based on wavelet packet shrinkage is how to filter out wavelet packet decomposition coefficients produced by noises. Appropriate thresholds are chosen in engineering to quantify wavelet packet decomposition coefficients, wavelet packet coefficients lower than or equal to the threshold are treated as zero, and only data above the threshold are used to reconstruct signals x(n). In this way, most of noises are eliminated, while the singularity points and characteristics of the original signals are preserved [9,10]. Obviously, the choice of threshold directly influences the effectiveness of the denoising algorithm. Too high a threshold would result in too many wavelet packet decomposition coefficients being reset as zero, and thus destroying too many details of the signal, while with too low a threshold the expected denoising effect could not be achieved.

The process of denoising based on wavelet packet shrinkage is divided into three steps:

$$y = W(s) \tag{2}$$

$$z = D(y,t) \tag{3}$$

$$\hat{s} = W^{-1}(z) \tag{4}$$

Where, W(•) and W-1(•) denotes the decomposition and reconstruction algorithm of wavelet packet respectively, D(y,t) denotes the shrinkage of wavelet packet coefficients with the given threshold t , s denotes noise-containing signals, y denotes the wavelet packet decomposition coefficient of s, z denotes the wavelet packet coefficient after shrinkage, and ŝ denotes denoised signals. The whole denoising process of wavelet packet shrinkage is illustrated as in Fig. 2.

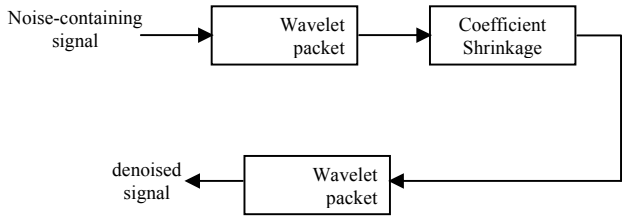

Fig. 2 Flow diagram of denoising based on wavelet packet shrinkage.

In the denoising process of noise-containing signals, the most important question is how to choose a threshold and a threshold function.

B. Threshold estimation

This analysis illustrates the use of Stein's Unbiased Estimate of Risk (SURE) as a principle for selecting a threshold to be used for de-noising. Stein Unbiased Risk Estimate (SURE) is an adaptive threshold selection rule. It is data driven. The aim of estimate is to minimize the risk. Because the coefficients of true signal are unknown, the true risk is also not unknown. We derive the unbiased estimate of true risk for generalized threshold functions; then SURE threshold value minimizes the unbiased risk estimate [6]. This technique calls for setting the threshold T to

$$T = \sqrt{2\log_e(n\log_2(n))} \tag{5}$$

Where n is the length of the signal.

C. Selection of Threshold Function

For any threshold, two kinds of threshold function can be used: hard-threshold function, soft-threshold function. Their mathematical expressions are as follows [9]:

Hard-threshold function:

$$D^H(y,t) = \begin{cases} y & |y| \geq t \\ 0 & |y| < t \end{cases} \tag{6}$$

$$D^s(y,t) = \begin{cases} sign(y)(|y| - t) & |y| \geq t \\ 0 & |y| < t \end{cases} \tag{7}$$

In formula (6) ~ (7), denotes the wavelet packet decomposition coefficient, t denotes the threshold, and D(y,t) denotes the estimated value of wavelet packet decomposition coefficient of denoised signals.

V. Signal Denoising on FPGA

The signal de-noising process is implemented on a field programmable gate array (FPGA) using a six-level Daubechies wavelet with soft. The wavelet packet transform consists of the analysis and synthesis banks. The analysis bank does the six level Daubechies wavelet transform, separating out the noisy signal into approximation coefficients and six levels of detail coefficients. The analysis bank is made up of low and high pass filters and downsampling blocks. The synthesis bank reconstructs the signal by recombining the approximation and detail coefficients, and is made up of upsampling blocks and filters. For a signal de-noising application, a thresholding block is placed between the analysis and synthesis banks. We now describe the architecture used to implement the signal de-noising system on a field programmable gate array.

A. The analysis bank

The analysis bank consists of an FIR filter followed by a down-sampling operator [11]. Down-sampling an input sequence x[n] by an integer value of 2, consists of generating an output sequence y[n] according to the relation y[n] = x[2n]. Accordingly, the sequence y[n] has a

sampling rate equal to half of that of x[n]. We implemented the decimator as shown in Fig. 3.

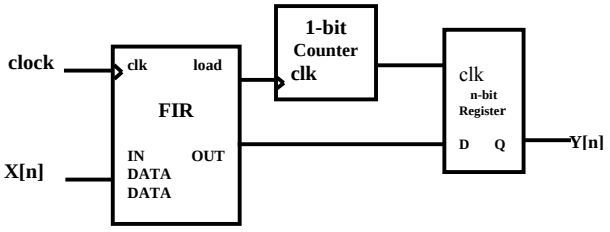

Fig. 3 implementation of the basic blocks of the Analysis bank

An active-high output control pin, labeled load, has been implemented in FIR filter structure and connected directly to the CLK input of a 1-bit counter. The input port of the FIR filter is connected to the input samples source, the input port of the FIR filter is connected to the input samples source, whereas the output port is connected to a parallel-load register. The register loads its input bits in parallel upon receiving a high signal on its load input from the 1-bit counter, and blocks its input otherwise. Assuming unsigned 8-bit input samples, the decimator operates as follows. When the load signal is activated, every time the FIR completes a filter operation, it triggers the counter to advance to the next state. If the new state is 1, the parallel-load register is activated, and it stores the data received at its input from the FIR filter. If the new state is 0, the register is disabled, and consequently the FIR output is blocked from entering the register, and ultimately discarded. The above procedure repeats, so that when the state machine has 1 on its output, the FIR data is stored, and when it has a 0 on its output, the FIR data is discarded.

B. The synthesis bank

The synthesis bank consists of an FIR filter proceeded by an up-sampling operator [11]. The up-sampler inserts an equidistant zero-valued sample between every two consecutive samples on the input sequence x[n] to develop an output sequence y[n] such that y[n] = x[n/2] for even indices of n, and 0 otherwise. The sampling rate of the output sequence y[n] is thus twice as large as the sampling rate of the original sequence x[n]. We implemented the interpolation filter as shown in Fig. 4.

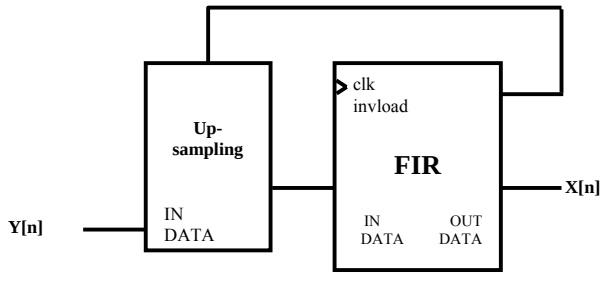

Fig. 4 implementation of the basic blocks of the Synthesis bank

The input port of the FIR filter is connected to the output port of the up-sampling block; whereas the input port of the up-sampling block which is described by a state machine is connected directly to the input samples source. The operation of the state machine depends on the load signal received from FIR filter; it triggers the state machine to advance to the next state. If the load signal is 1, the input sample will appear at the output port of the state machine. Otherwise the output will be zero.

C. The thresholder

As implemented, the system uses soft thresholding because the soft threshold provides smoother results in comparison with the hard threshold. The thresholder is the simplest block in the system. As the detail coefficients exit the synthesis bank, the thresholder uses a comparator to see whether a given coefficient's magnitude is grater than or equal the threshold. If it is, a subtractor is used to subtract threshold from that coefficient and a multiplexer is used to replace that coefficient with the output of the subtractor in the coefficient stream. Else, a multiplexer is used to replace that coefficient with a zero in the coefficient stream.

VI. SIMULATION OF DWT ON FPGA

Once the design entry phase is terminated by a successful compilation of the complete hierarchical design. The next step is the simulation of the design to illustrate how it works. For this purpose a test bench facility is available in the EDA tool which is the most suitable method to run a complete simulation for the design. It describes with the VHDL code. The test bench provides access to text file which contains the data of the encoded noisy signal file generated by matlab program. Fig. 5 illustrates the VHDL code of the test bench only.

```
ARCHITECTURE gfewq OF reconyt_tester IS
file infile : text is in "D:\data\spnoise1.txt";
BEGIN
process (clk2 )
variable inline : line;
variable dataread : Bit_vector (7 downto 0);
variable adc_out : std_logic_vector (7 downto 0);
BEGIN
xin <= "00000000";
    IF (clk2'EVENT AND clk2 ='1') THEN
if (NOT endfile(infile)) then
readline (infile , inline);
read(inline , dataread);
adc_out := to_stdlogicvector( dataread);
   end if;
 end if;
xin <= adc_out;
end process;
```

Fig. 5 VHDL code of the test bench.

The designed test bench has been run and the noisy signal was applied as the input of the denoising system with clock period equal to 1800 ns as shown in Fig. 6. The test bench result for the input signal is presented in Fig. 7.

VII. Synthesis of DWT on FPGA

We have implemented the design using Altera FPGA device, EP1C6Q240. This device contains 5980 logic elements.

VIII. Conclusion

Based on wavelet packet analysis, its denoising effect is better than wavelet transform. In this paper, we have studied signal denoising by wavelet packet shrinkage. Stein's Unbiased Estimate of Risk (SURE) has been chosen as a principle for selecting a threshold to be used for denoising. The field programmable gate arrays are an inexpensive and viable computational platform for processing of signals denoising. The suggested design is tested. The simulation and synthesis result of the suggested design is presented.

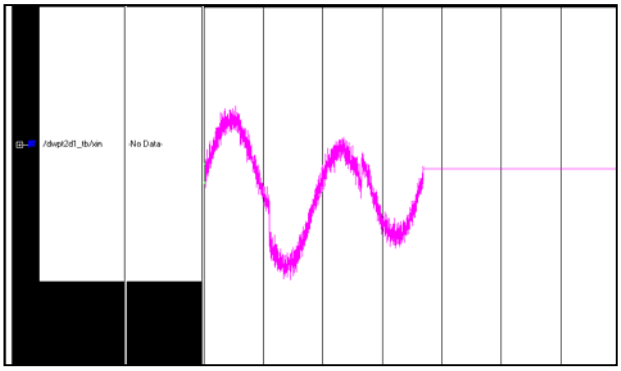

Fig. 6. Noisy signal with SNR = 7 db.

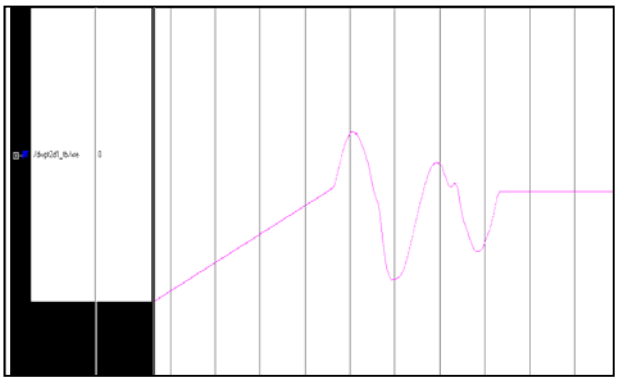

Fig. 7. Denoised signal

References

[1] C.K. Chen, Introduction to Wavelets, Academic Press, Boston, 1991.

[2] C.R. Mittermayer, S.G. Nikolov, H. Hutter, M. Grasserbauer,Wavelet denoising of Gaussian peaks: a comparative study,Chemom. Intell. Lab. Syst. 34 1996 187–202.

[3] B. Walczak, B. van den Bogaert, D.L. Massart, Application of wavelet packet transform in pattern recognition of near-IR data, Anal. Chem. 68 1996 1742–1747.

[4] D.L. Donoho, I.M. Johnstone, Ideal spatial adaptation via wavelet shrinkage, Biometrika 81 1994 425–455.

[5] D.L. Donoho, Denoising by soft-thresholding, IEEE Trans. Information Theory 41 3 1995 613–627.

[6] Xiao-Ping Zhang,M.Desai. Adaptive denoising based on sure risk, IEEE signal processing letters, vol. 5, no. 10, 1998.

[7] http://documents.wolfram.com/applications/wavelet/Fundamentalsof Wavelets/1.5.6.html.

[8] J. Carletta, G. Giakos N. Patnekar, L. Fraiwan and F. Krach, Design of a field programmable gate array-based platform for real-time de-noising of optical imaging signals using wavelet transforms , Elsevier, vol.36, 2004 289-296.

[9] http://documents.wolfram.com/applications/wavelet/Fundamentalsof Wavelets/1.5.6.html.

[10] Fgfh Peng Jiang, Qingbo Huang, Yaguang Kong1, and Li Chai, Research on a Denoising Method Based on Wavelet Packet Shrinkage for Pulp Thickness Signals. The First International IEEE conference of Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06).2006.

[11] D. L. Donoho, "De-noising by soft-thresholding," IEEE Trans on information theory, vol. 41, pp. 612-627, 1995.

[12] Nick, K. and Julian, M. 1997. Wavelet Transform in Image Processing. Signal Processing and Prediction I, Eurasip, ICT press, 23-34.