Modeling and FPGA-based implementation of an efficient and simple envelope detector using a Hilbert Transform FIR filter for ultrasound imaging applications

Amauri Amorin Assef1,2*, Breno Mendes Ferreira2, Joaquim Miguel Maia1,3, Eduardo Tavares Costa4

1 Graduate Program in Electrical and Computer Engineering, Electrical Engineering Department, Federal University of Technology - Parana, Curitiba, PR, Brazil.
2 Graduate Program in Energy Systems, Electrical Engineering Department, Federal University of Technology - Parana, Curitiba, PR, Brazil.
3 Graduate Program in Biomedical Engineering, Electronic Engineering Department, Federal University of Technology - Parana, Curitiba, PR, Brazil.
4 Biomedical Engineering Department, School of Electrical and Computing Engineering, State University of Campinas, Campinas, SP, Brazil.

Abstract

Introduction: Although the envelope detection is a widely used method in medical ultrasound (US) imaging to demodulate the amplitude of the received echo signal before any back-end processing, novel hardware-based approaches have been proposed for reducing its computational cost and complexity. In this paper, we present the modeling and FPGA implementation of an efficient envelope detector based on a Hilbert Transform (HT) approximation for US imaging applications. Method: The proposed model exploits both the symmetry and the alternating zero-valued coefficients of a HT finite impulse response (FIR) filter to generate the in-phase and quadrature components that are necessary for the envelope computation. The hardware design was synthesized for a Stratix IV FPGA, by using the Simulink and the integrated DSP Builder toolbox, and implemented on a Terasic DE4-230 board. The accuracy of our algorithm was evaluated by the normalized root mean square error (NRMSE) cost function in comparison with the conventional method based on the absolute value of the discrete-time analytic signal via FFT. Results: An excellent agreement was achieved between the theoretical simulations with the experimental result. The NRMSE was 0.42% and the overall FPGA utilization was less than 1.5%. Additionally, the proposed envelope detector is capable of generating envelope data at every FPGA clock cycle after 19 (0.48 µs) cycles of latency. Conclusion: The presented results corroborate the simplicity, flexibility and efficiency of our model for generating US envelope data in real-time, while reducing the hardware cost by up to 75%.

Keywords Ultrasound, Envelope detection, Hilbert transform, FPGA, Simulink.

Introduction

In medical ultrasound (US) imaging, the envelope detection is a commonly used digital signal processing (DSP) technique to extract the magnitude of the oscillating broadband radio-frequency (RF) signals from the high-frequency echo carrier before any back-end processing (Chang et al., 2007). Typically, there are three conventional methods that can be used to extract the low frequency envelope of the received echo signals: Hilbert Transform (HT) based demodulation, squaring and filtering based demodulation, and quadrature demodulation (Zhou and Zheng, 2015). As discussed by Levesque and Sawan (2009), although HT is more accurate and efficient than other mixing methods, filtering and quadrature algorithms are generally preferred, because of their lower computational requirements.

In order to reduce the complexity of the HT based demodulation, several hardware-based algorithms, typically implemented in programmable logic devices, such as Field Programmable Gate Array (FPGA), have been proposed by the research community (Chang et al., 2007; Hassan and Kadah, 2013; Levesque and Sawan, 2009; Qiu et al., 2012). Such demodulation algorithms involve extracting the analytic signal of the received RF signal using the HT. As the analytic signal is a complex signal, where the real part (I-in-phase) is the...
original signal and the imaginary part (Q-quadrature) is the HT of the original signal with a 90-degree phase shift in the operation band, its magnitude is calculated as the square root of the sum of the squares of the I and Q components (Schlaikjer et al., 2003). For example, Chang et al. (2007) proposed an envelope detector using the look-up table (LUT) method to compute the magnitude of the I/Q signals extracted from the quadrature demodulation. On the other hand, Levesque and Sawan (2009) presented a fully hardware-based quadrature demodulation processor that combines two finite impulse response (FIR) filters and a piecewise linear function to complete a square root unit. Qiu et al. (2012) used the same approach, however, the authors included a Cordic algorithm to calculate the modulus of I/Q data. In common, all these studies reported significant storage and/or arithmetic requirements, which can be a problem in terms of power consumption, delay and FPGA chip area occupation.

In this study, we present the modeling, validation and implementation of a fully FPGA-based digital envelope detector based on an optimized HT FIR filter for US imaging applications using the Matlab/Simulink (MathWorks, USA) software. In addition to the inherent nature of the symmetric coefficients, the proposed envelope detector exploits the alternating zero-valued coefficients in the HT FIR filter impulse response to achieve a cost-efficient hardware implementation with low complexity and latency. The discrete system is built by using the DSP Builder development tool (Intel Corp., USA) and synthesized for an Intel Stratix IV FPGA. The accuracy of the design is evaluated by the normalized root mean square error (NRMSE), and its operating time is estimated.

**Methods**

HT based demodulation algorithms can be built using different methods and techniques (Levesque and Sawan, 2009; Qiu et al., 2012). However, due to its inherent stability, linear phase and easiness for realization (DeBrunner and Wang, 2006; Soderstrand et al., 2000), a digital FIR filter is used in this model implementation.

The conventional digital FIR filter with constant coefficients of order \( N \) can be expressed in the following discrete convolution sum form:

\[
y(n) = a_0 x(n) + a_1 x(n-1) + \ldots + a_N x(n-N) = \sum_{i=0}^{N} a_i x(n-i) = \sum_{i=0}^{N} a_i x(n-i)
\]  

(1)

where \( n \) is the sample index (\( n = 0, 1, 2, \ldots, N \)), \( x(n) \) is the input signal, \( y(n) \) is the output signal and \( a_i \) are the length-\( N \) FIR filter coefficients. By taking advantage of the odd anti-symmetry impulse response (i.e., \( a_0 = -a_N, a_1 = -a_{N-1}, \ldots \)) that we are considering in this work, Equation 1 can be written as

\[
y(n) = a_0 \left[ x(n) - \sum_{i=0}^{N/2} a_i x(n-N+i) \right] + a_N x(n-N) + \ldots + a_N x(n-N) + a_1 x(n-1) + \ldots + a_0 x(0)
\]

(2)

As discussed by Zhou and Zheng (2015), the conventional FIR convolution algorithm can also be applied to a HT approximation, which is characterized by an impulse response with interleaved zeros coefficients (i.e., \( a_0 = a_2 = \ldots = a_N = 0 \)). Therefore, assuming that the even \( \text{N}\)-order \((N+1)\) taps filter should be such that the zero-valued coefficients form the first and last entry of the impulse response, where \( N \) must obey \( N = 4 + 4n \), it can be shown that

\[
y(n) = a_0 \left[ x(n) - \sum_{i=0}^{N/4} a_{2i} x(n-2i) - \sum_{i=0}^{N/4} a_{2i-1} x(n-2i+1) \right] + a_N x(n-N) + \ldots + a_1 x(n-1) + \ldots + a_0 x(0)
\]

(3)

From (3), the proposed cost-efficient hardware-based HT FIR filter architecture is shown Figure 1. The output signal \( Q(n) \) is the quadrature component of the signal produced by the HT FIR filter and \( I(n) \) is the in-phase component, which corresponds to the input signal \( x(n) \) delayed by an appropriate amount of cycles to compensate the phase delay of the FIR process employed for generating the \( Q(n) \) output. As described before, this scheme exploits the HT FIR filter coefficients properties to reduce the required number of multiplication operations and adders to \( N/4 \), and shift registers to \( N/2-1 \) in the convolution sum algorithm.

For simplicity and according to Levesque and Sawan (2009) and Qiu et al. (2012), which investigated a satisfactory trade-off between delay and filter order for

![Figure 1. Proposed HT FIR filter architecture to produce an efficient hardware realization.](image-url)
envelope detectors, we choose to evaluate a 32\textsuperscript{th}-order HT FIR filter in this study. The filter coefficient values were calculated by using the Matlab FDATool with the equiripple FIR filter design method (McClellan et al., 1973). The HT FIR filter impulse response with normalized pass-band of 0.05 to 0.95 is presented in Figure 2.

The FPGA-based envelope detector was modeled in Simulink by using the integrated DSP Builder toolbox, allowing fast and automatic generation of hardware description language (HDL) code. Figure 3 shows the top-level design of the proposed hardware model. Initially, the “Input” block casts double precision RF data loaded from the Matlab workspace into 16-bit signed fixed-point representation for hardware efficiency. Then, the HT FIR filter structure computes the I and Q signals of the input \textit{RF\_signal} to minimize the amount of computation needed. As a result, the implemented DSP modeling function was reduced to: (1) 9 “Parallel Adder Subtractor” blocks, where the + and - operators determine whether each input is added to or subtracted from the total; (2) 15 “Delay” blocks with two pipeline stages; (3) and 8 shared “Multiplier” blocks based on predefined signed fractional coefficients, which are imported from Matlab workspace and stored as constants. Following the 8-input “Parallel Adder” block, the output bit width was limited to 16 bits to save hardware resources in the subsequent signal processing chain. After the HT FIR filter, the sum of the squares of the I and Q signals was achieved by the “Multiply Add” block with

![Figure 2](image-url) Impulse response of the 33-tap HT FIR filter with negative odd symmetry and interleaved zero-valued coefficients.

![Figure 3](image-url) Architecture implementation of the FPGA-based envelope detector using the DSP Builder Blockset.
two multipliers. Before the next operation, an “AltBus” module was used to optimize the bit width of the output signal to 32 bits. Finally, the “Square Root” block returns the square root of the received argument with 16-bit resolution through the “Output” block envelope. The generated output signals I, Q and envelope data, labeled as $I_n$, $Q_n$ and $Envelope$, respectively, were exported to the Matlab workspace for subsequent off-line analysis. To complete the hardware implementation, a “ROM” block that maps data to an embedded RAM in the FPGA was included to store the echo signal.

To evaluate the performance of the proposed model, we used real US data (2000 samples) captured by a 3.2 MHz central frequency AT3C52B (BROADsound Corp., Taiwan) convex array transducer connected to an US research system that has been developed in our University (Assef et al., 2012). This RF signal was acquired from a tissue-cyst mimicking phantom (84-317, Nuclear Associates) and sampled at 40 MHz with 12-bit resolution. The complete acquisition setup is described by Assef et al. (2016). Once verified and validated in both Simulink and ModelSim (Intel Corp., USA) softwares, the generated VHDL (Very High Speed Integrated Circuits HDL) output files were synthesized and compiled with the Quartus II (Intel Corp., USA) software. For the experimental implementation, we used a DE4-230 FPGA development board (Terasici Tech., Taiwan), containing a Stratix IV EP4SGX230KF40C2 FPGA, and the SignalTap II Logic Analyzer, available in Quartus II, for data acquisition.

The accuracy of the envelope detector model was evaluated graphically, as well as quantified by the NRMSE cost function in comparison with the ideal HT (Chang et al., 2007), calculated from the absolute value of the discrete-time analytic signal (DTAS) via Fast-Fourier Transform (FFT) (Marple, 1999) in Matlab.

**Results**

Figures 4a and b show the graphical comparison between the Matlab simulation and experimental results, in which Figure 4b is the enlargement of Figure 4a from 11 to 14 µs, for better comparison. Here, the original input data (RF signal), the simulated envelope obtained with the reference method (DTAS-FFT response) and the resultant envelope information acquired by the FPGA (FPGA HT FIR response) are presented. As it can be seen, an excellent agreement was achieved between the reference simulation and our method, which was also confirmed by the calculated NRMSE of 0.42% and by the respective frequency spectrum responses (Figure 5).

In terms of FPGA clock cycles, the latency for the I and Q computation is 16 clock cycles and for the square and square root operations is 3 clock cycles. As a result, the proposed 33-tap HT FIR filter model is capable of generating real-time envelope data at every FPGA clock cycle after 19 clock cycles $\lfloor N/2+3 \rfloor$ of latency. Therefore, 2019 clock cycles of 40 MHz are needed to complete the envelope detection process, so that the fastest triggering frequency is about 19.81 kHz.

Additionally, other filters with order 8, 16, 64 and 128 were also tested using this method. As expected, the latency values were 0.18, 0.28, 0.88 and 1.68 µs, respectively. However, comparing to the ideal HT, the filters with order 8 and 16 were not efficient enough to compute the envelope data, resulting in a weak NRMSE of 47.45% and 11.45%, respectively. On the other hand, although the filters with higher order have produced an excellent NRMSE (<1%), no significant difference
FPGA-based envelope detector

The fully hardware-based digital envelope detector model presented here offers an excellent alternative when compared with other demodulation methods that require, for example, the use of complex FFT algorithms (Hans, 2005) or mixing sine and cosine functions (Qiu et al., 2012) to compute the analytic signal of the RF echo data in real-time. The proposed model is more simple, easy to implement and computationally efficient in terms of hardware requirements, while yielding similar results. Additionally, as the DSP Builder automatically translates the Simulink design into VHDL code, biomedical students and researchers do not need previous knowledge of HDL programming to implement, simulate and synthesize the algorithm in FPGA. Consequently, this methodology accelerates the investigation of new DSP algorithms and shortens the development cycle considerably.

In conclusion, we have successfully modeled and evaluated an efficient FPGA-based envelope detector for US imaging applications. The HT FIR filter algorithm has been realized easily and quickly by combining the Matlab/Simulink and DSP Builder tool, and proved to be able to produce accurate results with less computational cost.

Acknowledgements

The authors would like to thank CNPq and CAPES for their financial support. We would also like to thank the Intel Corporation for the donation of the DE4-230 FPGA board, which was used in this work.
References


