Dynamically Reconfigurable Architecture Design for Ultrasonic Imaging

Erdal Oruklu, *Senior Member, IEEE*, and Jafar Saniie, *Senior Member, IEEE*

Abstract—Ultrasonic imaging algorithms, including detection and compression, are computationally complex and difficult to implement in hardware for real-time applications. In this paper, we present an ultrasonic reconfigurable subband decomposition processor (RSDP) that can employ wavelet filters for frequency diverse signal processing. This architecture enables parallel implementation of a lifting-based discrete wavelet transform. The configurability of the architecture applies to the selection of wavelet kernels and scales for subband decomposition, thresholding operation for compression, and the postprocessing detection algorithm. The underlying hardware design makes use of the fact that both compression and detection applications share the same algorithm fundamentals. A unified architecture has been designed that implements signal decomposition and reconstruction with forward and inverse discrete wavelet transforms. After the forward transform step, a windowing operation is applied to discriminate frequency bands for target detection. Using the same architecture, a thresholding operation is applied to wavelet coefficients for data compression. The flexibility and the modular design make this reconfigurable architecture an effective and practical solution for real-time ultrasonic imaging applications. The resulting architecture is adaptable, fast, and suitable for a system-on-a-chip implementation that requires minimal logic resources.

Index Terms—Compression, detectors, field-programmable gate array (FPGA), reconfigurable architectures, wavelet transforms.

I. INTRODUCTION

ULTRASONIC target detection and classification in the presence of high scattering noise (clutter) is a significant and challenging problem. Another challenge for a real-time ultrasonic imaging application is the large amount of data that must be processed and compressed for image formation and/or image transmission for remote analysis by experts through wireless or wired communication channels or computer networks. In this paper, we present a reconfigurable architecture for ultrasonic signal compression and target detection. This design is based on the development of a run-time configurable architecture, which provides increased flexibility and adaptability. In addition, this architecture manages the high computational load of real-time applications while minimizing area and power consumption. Target detection algorithms are based on the premise that clutter echoes exhibit randomness and are more sensitive to frequency shifts than target echoes [1]. Therefore, frequency diverse signal decomposition methods such as discrete wavelet transform (DWT) can be used for differentiating the target information from the clutter echoes. In an earlier work [2], a comparative study of DWT kernels for target detection has been performed. The results indicate a target-to-clutter ratio (TCR) enhancement of 6–13 dB when the measured TCR is 0 dB or less. Wavelet filters can also be beneficial to ultrasonic signal compression due to their energy compaction properties. It has been shown [3], [4] that the DWT can achieve with high fidelity signal compression ratios up to 90% when applied to ultrasonic broadband echoes.

For hardware realization of ultrasonic detection and compression applications, we have designed a reconfigurable subband decomposition processor (RSDP) that can implement various wavelet kernels for subband decomposition of the ultrasonic data. An earlier version of the reconfigurable architecture was proposed in our earlier work [5]. In this paper, an improved architecture is presented with a case study using a field-programmable gate array (FPGA) implementation. Furthermore, detailed discussions of the wavelet-based detection algorithm and the RSDP hardware architecture are given. The advantage of the RSDP architecture is the freedom to fine-tune the target detection and compression algorithms for different environments. The DWT component of this architecture is based on the lifting scheme [6]. The lifting scheme requires two to four times fewer arithmetic operations than the conventional filter convolution architecture for the DWT [7].

Section II describes the algorithms that are used for a frequency-diverse ultrasonic target detection, which involves subband decomposition and frequency band selection for TCR enhancement, as well as DWT-based compression methods. A novel, reconfigurable, and pipelined architecture that is capable of implementing detection and compression is presented in Section III. A case study is also given in Section III-E. Section IV concludes this paper.

II. BACKGROUND

A. Subband Decomposition and Target Detection

In the ultrasonic imaging of materials, an effective method of obtaining frequency diverse information is through split spectrum processing (SSP) of the broadband echoes [1], [8]. SSP can be implemented via subband signal decomposition. In Rayleigh scattering, where the signal wavelength is significantly larger than the microstructure of materials that consists of randomly distributed reflectors and grains, the detected
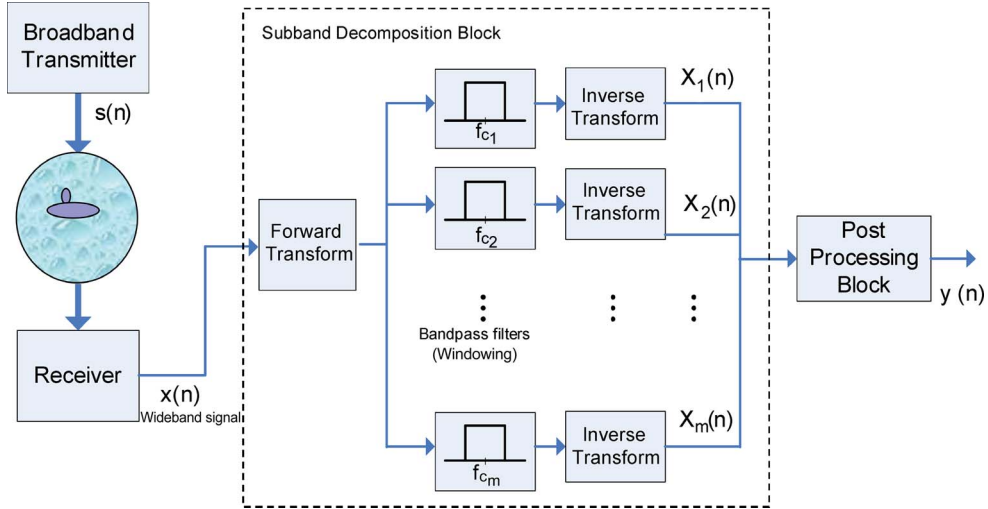
Fig. 1. Ultrasonic target detection system using subband decomposition.

echoes exhibit randomness in amplitude and are sensitive to shifts in the transmitted frequency. In contrast, targets are often larger in size and less vulnerable to variation in the transmitted frequency. In general, target echoes exhibit different distributions as a function of frequency when compared with microstructure scattering. Therefore, at any given time, the outputs of bandpass filters can be represented as a random feature vector that contains information that is related to target and grain echoes.

The SSP procedure has several steps (see Fig. 1). The first step is data acquisition. The experimental setup for data acquisition utilizes a pulse generator to produce the electrical impulses to drive the ultrasonic transducer. The pulse receiver is used to receive the backscattered ultrasonic echoes. The received signal is then digitized and passed through several bandpass filters using forward and inverse fast Fourier transforms (FFTs) to split the spectrum into different subbands (i.e., observation channels). The output signals from the subbands are then passed into a postdetection processor for target detection [1], [8]. This processor can employ different techniques such as signal averaging, Bayesian classifiers, and order statistic filters, including minimization. For the minimization algorithm, minimum amplitudes of the observation channels are obtained for each particular time, i.e.,

$$y(n) = \min\left[|x_1(n)|, |x_2(n)|, \ldots, |x_m(n)|\right] \qquad (1)$$

where $x_i(n)$ is the observation channel output corresponding to a certain frequency band.

In [2], the DWT has been shown to offer similar subband decomposition performance with efficient hardware implementations and, therefore, is a viable alternative to the FFT in detection algorithms.

B. DWT for SSP

Since the DWT does not have a prefixed kernel as in the FFT, a wavelet kernel can be chosen based on the detection or compression performance of the wavelet filters depending on the ultrasonic application. In particular, the compactness properties of the DWT allow a region of interest to be determined

in a time–frequency representation, which is essential for target detection [2]. TCR enhancement is governed by the degree of the compactness of the target echo. The clear benefit of using the DWT is the capability of fine-tuning the wavelet kernel for compacting the target echo information while spreading the clutter energy over a 2-D plane.

The DWT decomposes the digitized ultrasonic signal into subbands and provides a time–frequency representation. The task of the target detection algorithm is to select a number of windows to discriminate the target echoes from the clutter echoes. Here, a *window* represents a group of wavelet scales that function as a bandpass filter similar to bandpass filtering in SSP. Inverse DWT is applied to each window operation, and the resulting time-domain signals are then fed into the postprocessing block. The postprocessor in the final stage is a decision block that reconstructs the time-domain signal from the incoming channels according to order statistics rules. In the reconstructed signal, the target echo is made more visible due to the vulnerability of clutter echoes to the change in wavelet scales.

For ultrasonic experimental results, we have used 2048 data points, which correspond to a maximum of 11 wavelet scales in the wavelet transform domain. An important question is determining the frequency bands (scales) to be used for postprocessing. Since the clutter echo spectrum is shifted toward higher frequencies [8], the target echo is expected to be the dominant information in lower frequencies. Wavelet domain scales in Fig. 2 confirm that the lowest scales (high frequencies) are mostly clutter information, whereas the higher scales represent the low-pass approximation of the ultrasonic data. Inspection of Fig. 2 also shows that the intermediate scales 3–6 contain dominating target information. Therefore, intermediate scales are a desirable choice for postprocessing. Increasing the number of scales to be used in the algorithm increases the chance of integrating scales that do not carry target information into the reconstruction step.

The following steps are critical for target detection applications that incorporate DWT decomposition.

1) Choose an appropriate wavelet kernel to maximize the target echo compactness.

(a)

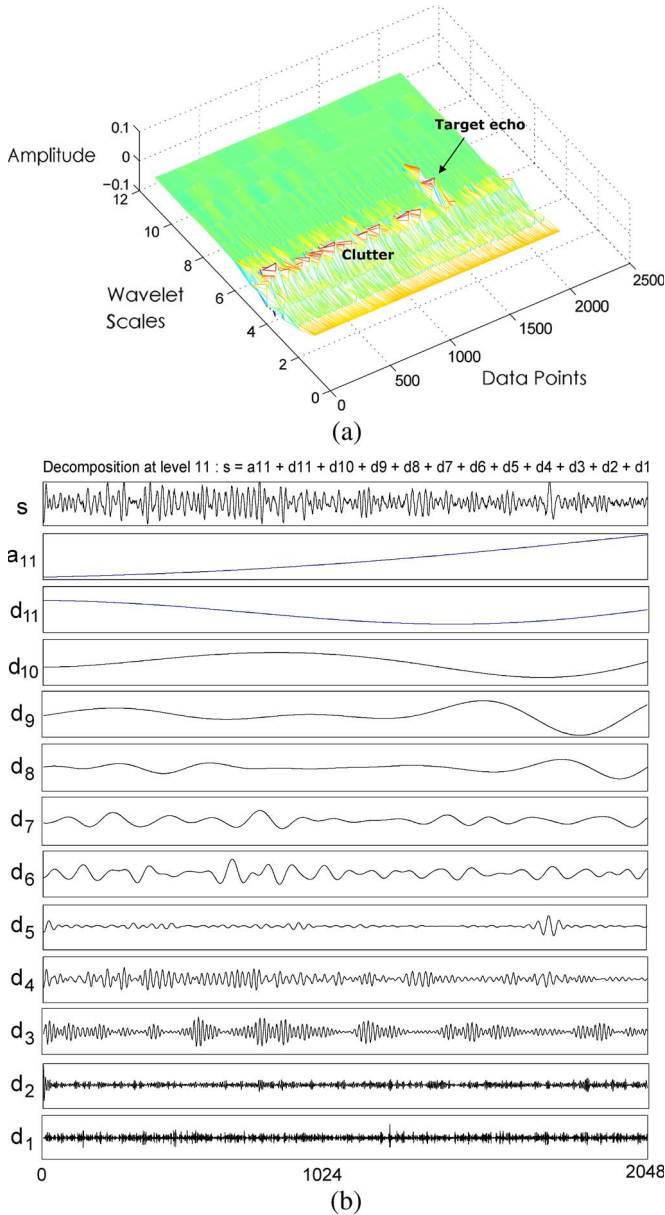Decomposition at level 11 : s = a11 + d11 + d10 + d9 + d8 + d7 + d6 + d5 + d4 + d3 + d2 + d1

(b)

Fig. 2. Wavelet scales in (a) 3-D and (b) 2-D representation. (a) Time–frequency representation of experimental ultrasonic data. (b) Wavelet scales (d1–d11 are detail levels, where d1 represents the highest frequency band, and a11 is the approximation level).

2) Identify the wavelet scales that carry target echo spectrum information.
3) Determine how many windows are to be utilized for signal reconstruction.
4) Find the number of scales that has to be integrated in each window.

It is important to evaluate the performance of the wavelet kernels for target detection since they differ in their compactness properties. Here, compactness means that the signal can be represented by fewer wavelet coefficients and scales. To understand the compactness of the kernel, we examined the similarity between the signal echo and the wavelet kernel. If the wavelet kernel is similar to the ultrasonic target echo, then the target echo becomes dominant in a particular scale. A

TABLE I
DWT WAVELET KERNEL PERFORMANCE EVALUATION

Transform Method	Ascan1	Ascan2	Ascan3	Ascan4	Ascan5	Ascan6	Average TCR
EXPERIMENTAL DATA							
Input Signal TCR	-0.14dB	-0.75dB	-2.07dB	-3.52dB	-3.74dB	-4.19dB	-2.40dB
FFT	10.75	7.21	14.82	10.17	12.33	12.78	11.54
D4	7.41	6.1	8.35	5.9	9.48	10.8	8.00
D10	5.63	2.19	10.86	11.2	9.43	7.58	7.81
Vaidyanathan	9.18	6.46	8.71	7.95	11.83	11.85	9.33
Coiflet	6.92	7.04	8.78	10.27	13.47	12.5	9.83
Symmlet8	5.24	2.75	7.62	11.19	10.62	6.85	7.37
Symmlet10	6.52	6.32	9.24	10.46	13.22	11.54	9.55
Battle1	6.87	7.48	11.81	8.47	10.85	12.72	9.70
Battle5	10.02	6.76	7.89	8.69	12.33	11.88	9.59
Target-to-Clutter Ratio (TCR) Enhancement (dB)							

desirable property of wavelet kernels is retaining target echo information in as many frequency scales (subbands) as possible. This is an important property for the minimization detection algorithm.

C. Performance Results

For performance analysis and testing, the experimental A-Scan data from a steel block (type 1018, a grain size of 50 μm) are acquired and analyzed. A Panametric (type 5052) pulser/receiver is used to drive the ultrasonic transducers, and it is used to receive the ultrasonic echoes. The received echo signals are then converted to digital data for SSP. The A-scan measurements were conducted using a broadband unfocused ultrasonic transducer of 0.5-in diameter with a 5-MHz center frequency. Data were acquired with a 100-MHz sampling rate, and each sample is 8 bits. The steel block has several holes (1.5 mm in diameter) at known separate locations. All the A-scan measurements probe the hole positions within the steel block. For performance analysis, the TCR is calculated by finding the maximum target echo (reflection from hole location) amplitude in the reconstructed signal, i.e.,

$$\text{TCR} = 20 * \log_{10}(T/C) \tag{2}$$

where T is the maximum target echo amplitude, and C is the maximum clutter echo amplitude. This value is compared with the largest amplitude of clutter echoes.

Table I and Fig. 3 show the performance of Daubechies (D4, D10), Symmlet (8,10), Coiflet (1,5), Battle-Lemarie (1,5), and Vaidyanathan wavelets and the FFT on different sets of experimental ultrasonic data consisting of a target echo that is highly masked by a clutter (i.e., microstructure scattering) for detection analysis. The number of filters in the filter bank for the FFT is eight. For the DWT, three windows are used for signal reconstruction. For postprocessing, absolute minimization is applied to the filter-bank outputs. It can be seen that the detection performance of wavelet transforms closely matches that of the FFT-based SSP method, and, for any given experimental data, up to 13-dB TCR enhancement is possible using wavelet decomposition.

In general, the algorithm performs well for multiple targets as long as the target echoes are not within each other's proximity. If multiple target echoes are close enough to interfere
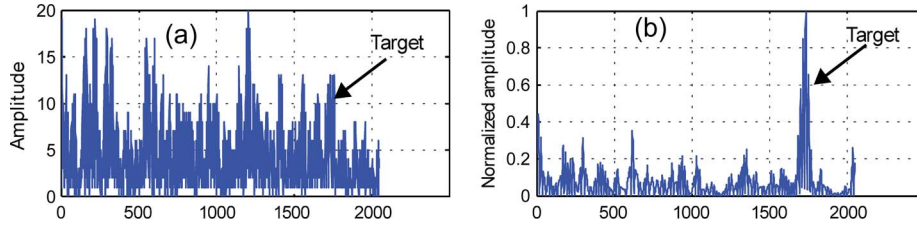
Fig. 3.   TCR improvement result using the Coiflet wavelet. (a) Experimental data. (b) After SSP operation.
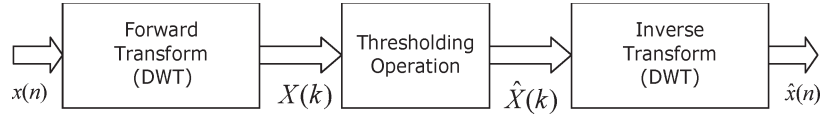

Fig. 4.   Thresholding operation for ultrasonic data compression.

with each other, then the performance of the algorithm may deteriorate. The minimization postprocessing algorithm detects the presence of the targets but may not differentiate individual targets. Nevertheless, other postprocessing techniques such as neural networks can be employed to resolve the existence of multiple targets [9].

D. Ultrasonic Signal Compression

In ultrasonic imaging applications, it is desirable to use data compression techniques to reduce the data size while maintaining the signal integrity. Fig. 4 shows the ultrasonic data compression algorithm. The data compression of a given signal $x(n)$ is successful when the redundant and noise components of $x(n)$ are reduced or removed. The signal $\hat{x}(n)$ is the compressed representation of $x(n)$.

Thresholding can be applied to transform the coefficients of the original ultrasonic signal for data compression. In the hard thresholding method, all coefficients that are smaller than τ are set to zero, and all coefficients that are greater than τ are kept the same, i.e.,

$$\hat{X}(k) = \begin{cases} 0, & X(k) < \tau \\ X(k), & X(k) \geq \tau \end{cases}. \qquad (3)$$

For a signal that is corrupted by white Gaussian noise with variance σ^2, it has been shown [10] that the optimal threshold for N number of sample points is given by τ, i.e.,

$$\tau = \frac{\sigma}{\sqrt{N}} \sqrt{2 \ln(N-1)}. \qquad (4)$$

This threshold value can be used as the initial threshold value. Based on the compression ratios achieved, this value can dynamically be adjusted. Data compression performance of the DWT depends on the wavelet kernel and its compactness properties. The data compression performance of six different wavelet kernels [3] is shown in Fig. 5.

This figure shows how much energy is concentrated in the five most dominant coefficients of the DWT as a function of the bandwidth of the ultrasonic signal. These results indicate that the Daubechies (Daub20) wavelet kernel has the best data compression performance, whereas the Haar wavelet kernel has the worst data compression performance. Furthermore, a robust compression architecture must be reconfigurable and must sup-
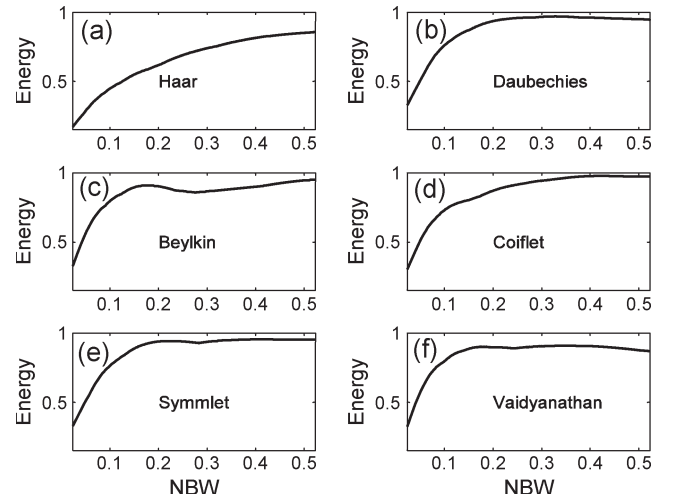

Fig. 5.   Energy accumulated among the five most dominant coefficients of the DWT using the following kernels: (a) Haar, (b) Daubechies (Daub20), (c) Beylkin, (d) Coiflet, (e) Symmlet, and (f) Vaidyanathan. Energy given as a function of ultrasonic echo bandwidth (NBW: normalized bandwidth) [3], [4].

port multiple wavelet kernels to achieve optimal performance for different ultrasonic testing environments.

III. RSDP ARCHITECTURE

Multipurpose design of the ultrasonic processor demands a reconfigurable architecture that is capable of realizing both target detection and data compression algorithms. In addition, this architecture requires adaptable wavelet-based subband decomposition for optimal performance. Therefore, the RSDP architecture has been designed to accommodate these objectives.

The RSDP design is based on the development of a dynamically configurable architecture, which provides increased flexibility and adaptability to ultrasonic imaging applications. The configurability of the architecture applies to the application type (i.e., compression or detection), selection of the subband transform method, algorithm parameters, thresholding operations, and the postprocessing algorithm.

Fig. 6 shows the system components of the ultrasonic processor architecture. The input memory holds the ultrasonic data. The ultrasonic data are fed into the forward DWT block using processing elements (PEs). The intermediate results are stored
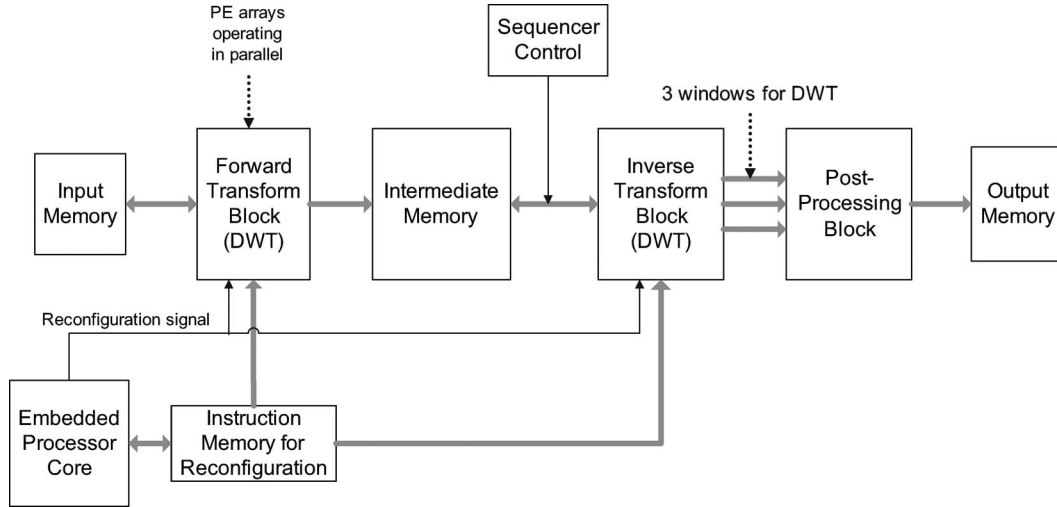
Fig. 6. RSDP architecture.

in a buffer. If the selected operation mode is data compression, then a hard threshold is applied to the amplitudes of the coefficients by a *sequencer* block. Since a major portion of the transform coefficients are below the threshold value, this results in a significant reduction of the data size. If the operation mode is target detection, the intermediate results are processed by the *sequencer*, which selects certain wavelet scales based on the windowing method. This windowing operation discriminates those subbands where target information is dominant, and they are selected for signal reconstruction (up to three windows for the DWT). Therefore, the same sequencer block is utilized for both compression and detection. The inverse transform block uses the same hardware resources as the forward transform block, and they can be reconfigured for multiple inverse DWT operations. The postprocessing block applies order statistics methods such as minimization. The outcome is stored in the output memory to be transmitted or displayed.

The embedded processor core in Fig. 6 acts as a control logic unit and is designed to perform the following tasks:

1) determine the type of the operation that can either be data compression or target detection;
2) select the filter coefficients (i.e., the wavelet kernel) for the DWT;
3) apply amplitude thresholding and wavelet scale selection parameters;
4) reallocate hardware resources for inverse transform channels for target detection.

The framework of the wavelet kernel implementation in the RSDP is based on the lifting scheme [11], [12]. The lifting scheme offers several advantages compared with conventional filter convolution architectures. It requires fewer arithmetic operations, and in-place calculation and integer-based operations are possible [7], [12].

A. Lifting Scheme Implementation of the DWT

The lifting scheme is computationally less complex, and any wavelet filter can be decomposed into a finite sequence of simple filtering steps, which are called lifting steps [13].

This decomposition is a factorization of the polyphase matrix of the wavelet filter into prediction and update steps, which are implemented as alternating upper and lower triangular matrices and a constant diagonal matrix, i.e.,

$$P(z) = \prod_{i=1}^{m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \quad (5)$$

where $s_i(z)$ and $t_i(z)$ are the Laurent polynomials, K is the scaling factor, and m is determined by the wavelet kernel factorization. For example, the Cohen–Daubechies–Feauveau 9/7 tap wavelet filter (CDF 9/7) can be factored into lifting steps as

$$P(z) = \begin{bmatrix} 1 & \alpha(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{bmatrix}$$
$$\times \begin{bmatrix} 1 & \gamma(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \delta(1+z) & 1 \end{bmatrix} \begin{bmatrix} \zeta & 0 \\ 0 & 1/\zeta \end{bmatrix} \quad (6)$$

where $\alpha = -1.58613$, $\beta = -0.05268$, $\gamma = 0.88291$, $\delta = 0.44351$, and $\zeta = 1.14960$.

B. PE Design for the Lifting Scheme

The RSDP architecture consists of PE arrays that operate as multifunction data-path elements. Each PE is designed to be capable of realizing a single lifting step computation. Therefore, in theory, even a single PE is sufficient to complete a single wavelet decomposition level by sequential operations. However, the throughput can effectively be increased by introducing arrays of PEs into the system, where arrays simultaneously execute multiple decomposition levels.

For a universal PE design, it is important to analyze different types of update and predict filters $s_i(z)$ and $t_i(z)$, respectively, in the lifting factorization. These filters may have different implementations and data-path requirements depending on the wavelet kernel [14]–[16], as shown in Fig. 7. To realize the circuits in Fig. 7, the PE architecture is designed to include two adder units, two registers, four multiplexers, and a multiplier unit as a single data-path element (see Fig. 8). Control signals for multiplexers enable the desired type of lifting circuits. The
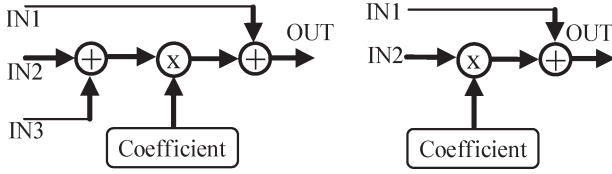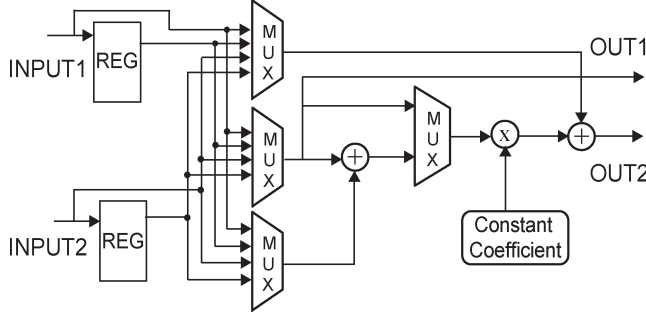
Fig. 7. DWT lifting step circuits.

Fig. 8. PE architecture for the RSDP.

registers are used to provide the filter taps. If there are more than two taps in either $s_i(z)$ or $t_i(z)$, these filters require a combination of more than one PE for a single lifting step.

C. System Data-Path Design

The RSDP architecture supports lifting factorizations with a variable number of lifting steps through the dynamic reconfiguration of the PE interconnection network. Although a crossbar network, the Clos network, or tree structures [17] can be implemented for PE interconnection, these network types are costly in terms of bus size and logical units. Consequently, a simpler network communication is designed for the RSDP architecture. In this network, arrays of PEs are formed by feeding the output result either to the next neighboring PE or to an output buffer memory. Each PE generates two outputs—s (approximation or a low-pass result) and d (detail or a high-pass result). For the last PE in an array, the d output is stored in the wavelet coefficient memory, and the s output is forwarded to the beginning of the next PE array (see Fig. 9). To achieve this I/O operation, multiplexers are used for controlling the input and output destinations.

In summary, subband decomposition using the RSDP has the following steps.

1) The wavelet filter is decomposed into lifting steps, and lifting factorization is obtained.
2) PEs can be configured for any wavelet filter implementation. A context memory holds the required configuration states and coefficient values. A finite-state machine (FSM) in the control block routes the desired signals to the PEs during the initialization phase.
3) PEs are cascaded to build the lifting architecture. Registers (D-blocks in Fig. 9) are used for pipelining the lifting steps. The number of lifting steps from the factorization in step 1 determines the number of PEs that is required in one array. For example, CDF 9/7 or Daubechies-4 (D4) can be implemented in four lifting steps, whereas CDF 5/3 and CDF 2/6 require two lifting steps [18]–[20].

4) For a faster implementation, more PE arrays can be constructed, which can execute different wavelet decomposition stages in parallel. The low-pass output signal from each array is fed into the next array input. Each array expects two samples from the previous array, and the sampling rate is half of the previous stage. Therefore, execution at each stage can simultaneously be finished with a minimum amount of latency.
5) A splitter block at the beginning of each PE array (see Fig. 9) is required to split the input signal into even and odd samples. For an inverse DWT, a merger block is used to combine the coefficients from the wavelet memory and the output of the previous stage.
6) For each PE array, a buffer memory is used to store the high-pass signal outputs of each stage (see Fig. 9).
7) Windowing operation for detection is accomplished by zero-padding the coefficients outside the window using the sequencer block in Fig. 10. The width and the location of windows are also configurable based on the processing profiles that are stored in the memory.
8) For compression applications, the sequencer can be utilized to perform thresholding on the amplitudes, as shown in Fig. 10. By retaining only the coefficients above the threshold and adjusting this threshold value, desired compression ratios can be achieved.
9) For an inverse transform, the ultrasonic target detection algorithm requires up to three inverse DWT channels. Therefore, the available PE resources should be partitioned for each channel for concurrent operation. In the parallel-mode operation, the output of each channel can immediately be fetched to the postprocessing block, and the decision results can be obtained without waiting for all subsequent inverse transform channels to finish.

D. Control Logic

Each PE unit includes FSM logic for dynamic reconfiguration by an external control unit or a CPU. FSM logic is responsible for loading the constant lifting coefficient from the coefficient memory, selecting the correct lifting circuit type, and choosing the source input and the destination output for the desired PE array size. The FSM machine is connected to the embedded processor unit through a control bus, and the instructions that are issued by the processor are received by all PE elements via a shared bus. Fig. 11 shows the state diagram for the PE FSM logic. In the *idle* state, the PE waits for the assertion of either an execute signal for starting the execution or a configure signal for reconfiguring the PE execution parameters.

In the idle state, output latches are disabled, and dynamic power consumption is minimum in the data-path elements. If the execute signal is asserted, the PE changes its state to *execute* and starts processing the data. If the configure signal is asserted, the PE is now in the *reconfigure* mode. Initially, the PE checks the control bus and reads the control instruction word. The format of this instruction is shown in Fig. 12. The most significant m bits are used to address PEs (2^m PEs are supported). If this address does not match the PE's inherent address, the PE
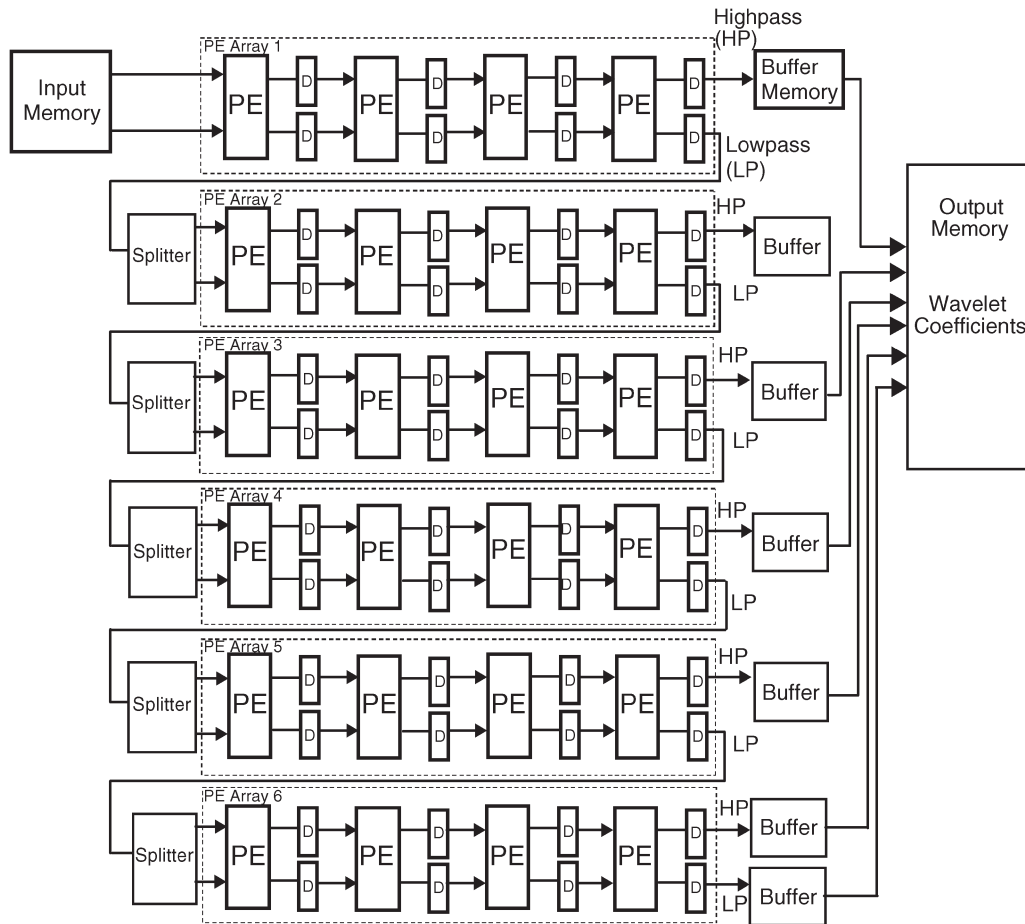
Fig. 9.   Implementations of concurrent wavelet decompositions using PE arrays.
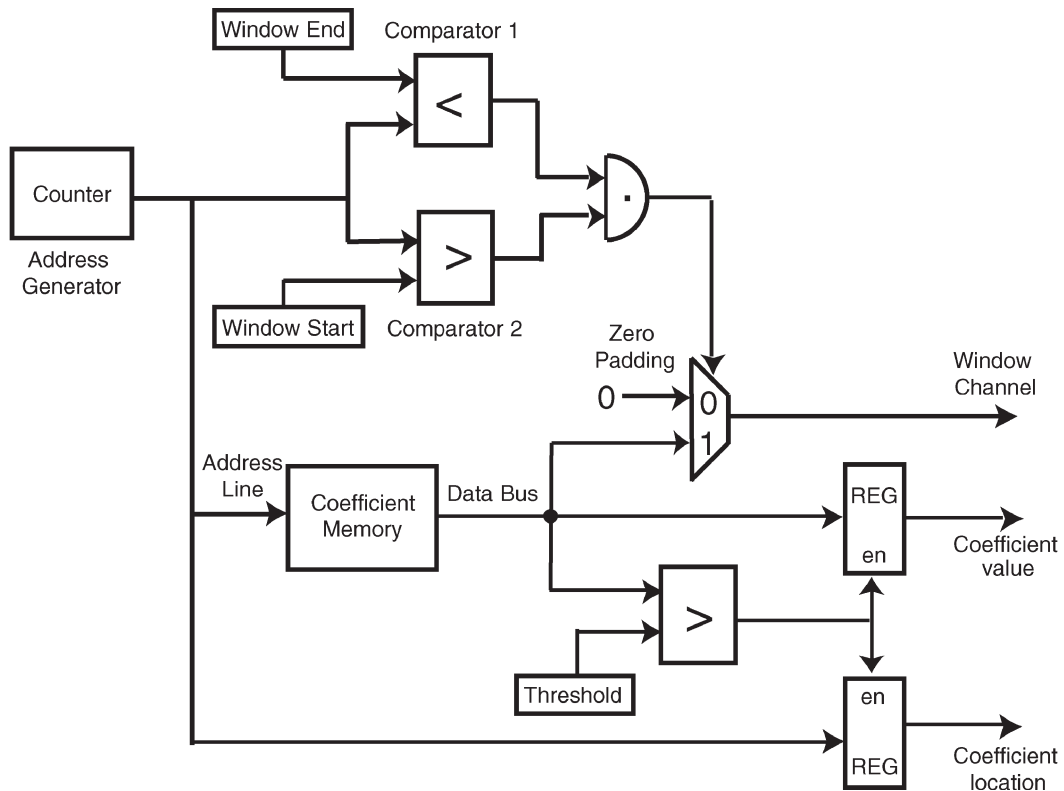

Fig. 10.   Sequencer-thresholding and windowing operations for compression and detection applications.
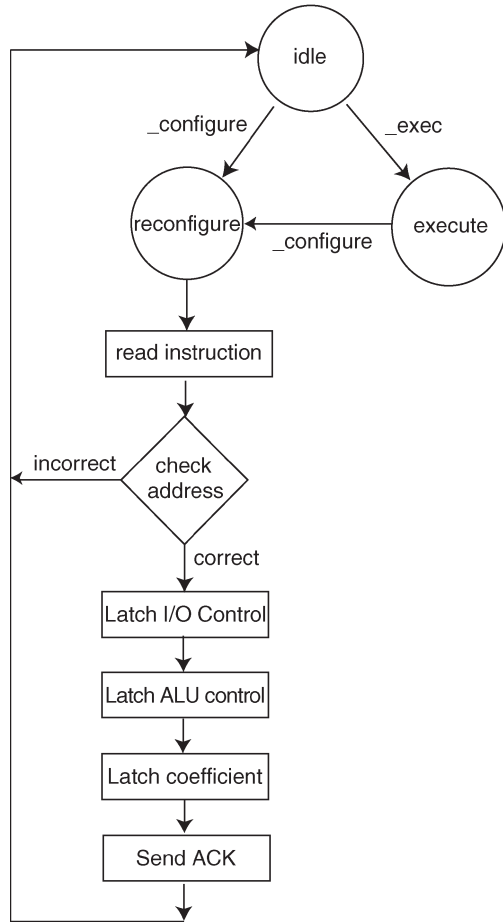
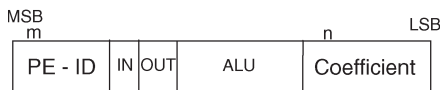Fig. 11.   State machine for the PE control.

Fig. 12.   Instruction word format for PE configuration.

returns to the idle state. If the address is correct, the PE latches the configuration data. The next control bits IN and OUT are input and output selection bits for the multiplexers: $IN = 0$ selects the previous PE as the input source. $IN = 1$ selects the memory buffer as the input source. $OUT = 0$ selects the next PE for output destination. $OUT = 1$ selects the memory buffer. The next bits are arithmetic-logic unit configuration bits, which control the multiplexers in Fig. 12 for selecting the lifting step circuits. The least significant n bits contain the fixed lifting coefficient. After this coefficient is stored to a register, the PE sends an acknowledgement signal to the control unit and returns to the idle state. The control unit sends configuration instructions to all the required PEs and updates the resource table of PEs with the incoming acknowledgement signals. After all the PEs are configured, it issues the execute signal. In this setup, the reconfiguration overhead is directly proportional to the number of PEs that is required by the algorithm. If the configuration instructions are stored in a context memory, fetching and issuing instructions can be done very quickly. Therefore, run-time reconfiguration of PEs is feasible and efficient in the RSDP architecture.
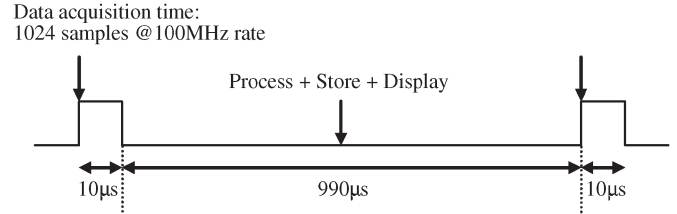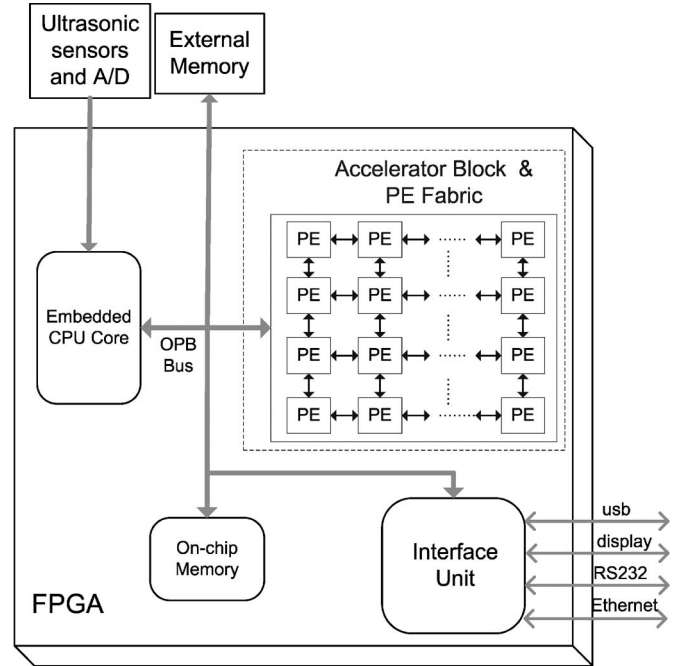

Fig. 13.   Typical timing requirements.

Fig. 14.   System architecture.

E. Case Study

The repetition rate in ultrasonic imaging systems dictates the processing time for real-time target detection or compression. For real-time systems, a typical value for a repetition rate is 1000 Hz, resulting in 1-ms time intervals for processing the acquired data. Fig. 13 shows the timing requirements for a typical application. Data acquisition takes 10 μs (considering 1-K samples acquired at a 100-MHz sampling rate). Consequently, the RSDP architecture has to process the data and store or transmit the results in 990 μs.

A hardware/software codesign scheme is utilized to obtain the proposed adaptable ultrasonic imaging architecture. The target platform is an FPGA chip with an embedded PowerPC CPU core. Fig. 14 shows the Xilinx Virtex-II Pro FPGA [21] device that is configured for RSDP operation. In this system, the embedded core processor is for preprocessing and synchronizing the streaming input data that are acquired through an ultrasonic sensor and an A/D block. The on-chip peripheral bus [22] allows communication between the processor, the on-chip memory, and the accelerator block. The accelerator block implements the required DWT data-path functions and the thresholding operations for compression via PEs. PEs are configured to establish systolic arrays for the desired DWT lifting scheme.

For this case study, the accelerator block incorporates 24 PEs to implement the ultrasonic imaging system (see Fig. 9). The

TABLE II
FPGA SYNTHESIS RESULTS

Resources	Accelerator Block	Percentile of Utilized Blocks in FPGA
Number of Slices	5592	40%
Number of Registers	5784	21%
Number of LUTs	9384	34%
Clock Frequency	141 MHz	-

CDF 9/7 biorthogonal wavelet filter [23] is used for subband decomposition. This filter is known to work best with compression applications (i.e., JPEG2000 [24], [25]), and many efficient hardware realizations of the 2-D CDF 9/7 transform have been proposed particularly for image compression algorithms [26]. Ultrasonic data (1024 points) that are sampled at 100 MHz are processed by an array of PEs, which carry out the data-path computations that are required for each lifting step. Each PE has two 16-bit adder units and a 16-bit multiplier. PEs are pipelined to achieve higher throughput. These elements are configured during run-time initialization for different wavelet kernels. Depending on the wavelet kernel and the corresponding lifting factorization, a number of PEs are cascaded to form an array of PEs. For example, 9/7 filter factorization requires four lifting steps [13], which are implemented by four PEs. For 1024 points, up to ten wavelet decomposition stages have to be completed for all the wavelet coefficients. If more than one PE array is available, the throughput can significantly be increased by concurrent operations. The lifting algorithm is sequential in nature: Second-stage calculations can immediately start after two low-pass results are generated from the first stage.

Once the pipeline is full, all the PE arrays start concurrent execution, and multiple wavelet decomposition levels are simultaneously computed. Since 24 PEs are available, six PE arrays can be used for 100% PE utilization (see Fig. 9). If the number of lifting steps is not equal to four, it is still possible to use more than one PE array by readjusting the size of the PE arrays. The number of inverse transforms that is required for signal reconstruction determines the number of windows that are used in the 2-D windowing stage. This number also dictates the throughput of the RSDP. In this case study, three windows are used; therefore, three inverse transforms are required. The same 24 PEs that are used in the forward DWT are reconfigured for inverse transform operations. In this case, two PE arrays are allocated for each inverse transform block. They operate in parallel, and all channel outputs are passed into a minimization postprocessing block.

F. Implementation Results

The PE accelerator block for the RSDP case study has been synthesized with Xilinx Virtex-II Pro (XC2VP30-7) FPGA using the Xilinx ISE 8.2 software. Table II shows the implementation results. The logic required for the PE fabric consumes up to 40% of the available FPGA resources. This indicates that more complex postprocessing methods, such as neural networks, can also be realized within the RSDP FPGA architecture [9]. Furthermore, the RSDP data-path design achieves

a clock rate of 141 MHz, which meets the real-time operation requirements of ultrasonic imaging, including data acquisition, data processing, and data storage. The proposed architecture can execute the SSP algorithm (one forward DWT, three inverse DWTs, thresholding, and postprocessing) in less than 100 μs. As a comparison, a software-based implementation of the SSP algorithm is implemented (using FFTs for frequency decomposition) on an embedded Microblaze microprocessor running at 100 MHz in a Virtex-II Pro FPGA. This software implementation requires 37.72 ms to execute the SSP algorithm, which is clearly not sufficient for a repetition rate of 1 ms (see Fig. 13).

G. Throughput Analysis of RSDP Wavelet Implementation

For forward DWT implementation, the required clock cycles are computed according to the following equations for direct or parallel implementation using the RSDP (see Fig. 9). This relies on the assumption that each PE operation can be executed in a single clock cycle. The computational parameters are:

- number of data points N;
- number of wavelet decomposition stages $= NS$ (maximum number of stages $= \log_2 N$);
- number of PEs $= NPE$;
- number of lifting steps for a single wavelet decomposition level $= NLS$;
- number of PE arrays $= NA = \lfloor NPE/NLS \rfloor$;
- latency for single-level DWT decomposition $= T_d = NLS$;
- number of passes $= NPass = \lceil NS/NA \rceil$.

Direct implementation without any parallelism (i.e., one array of four PEs) results in the following equation:

$$
\begin{aligned}
&\text{Total\_ClockCycles} \\
&= \sum_{i=0}^{NS-1} 2^{NS-1-i} + \text{Total\_Latency} \\
&= 2^{NS-1} + 2^{NS-2} + 2^{NS-3} + \cdots + 1 + T_d * NS. \quad (7)
\end{aligned}
$$

Hence, for 1024-point and 10-level wavelet decomposition in the case study, 1064 cycles are needed, i.e.,

$$
\begin{aligned}
\text{Total\_ClockCycles} &= 2^9 + 2^8 + 2^7 + \cdots + 2 + 1 + 4 * 10 \\
&= 1023 + 40 = 1063. \quad (8)
\end{aligned}
$$

Parallel reconfigurable implementation (i.e., NA concurrent decomposition levels) results in the following equation:

$$
\begin{aligned}
&\text{Total\_ClockCycles} \\
&= \sum_{i=1}^{NPass} 2^{NS-1-(i-1)*NA} + \text{Total\_Latency} \\
&= 2^{(NS-1)} + 2^{(NS-1)-NA} + 2^{(NS-1)-2*NA} + \cdots \\
&\quad + 2^{(NS-1)-(NPass-1)*NA} + (NA * T_d) * Npass. \quad (9)
\end{aligned}
$$

Using six arrays of four PEs, 568 clock cycles are required for 1024-point and 10-level wavelet decomposition (see Fig. 9), i.e.,

$$
NPass = \left\lceil \frac{10}{6} \right\rceil = 2 \quad (10)
$$

$$
\text{Total\_ClockCycles} = 2^9 + 2^3 + (6 * 4 * 2) = 568. \quad (11)
$$

Further improvement can be obtained by increasing the PE arrays; however, the throughput gain will be less significant. The throughput of the forward DWT could be the basis for evaluating the total performance for the RSDP architecture for a complex system.

IV. CONCLUSION

In summary, the RSDP realization offers several benefits to ultrasonic imaging applications. The detection or compression algorithms can be time-multiplexed to perform forward transform, bandpass filtering, thresholding, inverse transform, and postprocessing. Data-path elements have dynamically been reallocated and reconfigured for each task. The type of the DWT kernel is an important factor that affects the performance of both ultrasonic data detection and compression algorithms. With the reconfigurable data-path logic, several wavelet kernels can be supported and implemented during the algorithm run time. The window size and scale selections can adaptively be adjusted for better target detection performance. These concepts are crucial for a high-performance and adaptable ultrasonic imaging system. Therefore, the RSDP FPGA prototype that has been presented in this paper offers significant adaptability and area improvements compared with conventional logic implementations, and it is feasible to use it in portable ultrasonic devices for real-time applications. An application-specific integrated circuit implementation with integration of ultrasonic sensors can result in a very compact system-in-package solution for industrial and medical ultrasonic applications.

ACKNOWLEDGMENT

REFERENCES

[1] J. Saniie, D. T. Nagle, and K. D. Donohue, "Analysis of order statistic filters applied to ultrasonic target detection using split spectrum processing," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 38, no. 2, pp. 133–140, Mar. 1991.

[2] E. Oruklu and J. Saniie, "Ultrasonic target detection using discrete wavelet transform for NDE applications," in *Proc. IEEE Symp. Ultrason.*, Aug. 2004, vol. 2, pp. 1054–1057.

[3] G. Cardoso and J. Saniie, "Performance evaluation of DWT, DCT and WHT for compression of ultrasonic signals," in *Proc. IEEE Ultrason., Ferroelectr., Freq. Control Symp.*, Aug. 2004, vol. 3, pp. 2314–2317.

[4] G. Cardoso, "Compression, estimation, and analysis of ultrasonic signals," Ph.D. dissertation, Illinois Inst. Technol., Chicago, IL, 2005.

[5] E. Oruklu, G. Cardoso, and J. Saniie, "Reconfigurable architecture for ultrasonic signal compression and target detection," in *Proc. IEEE ICASSP*, Mar. 2005, vol. 5, pp. 129–132.

[6] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, Apr. 1996.

[7] J. Reichel, "On the arithmetic and bandwidth complexity of the lifting scheme," in *Proc. Int. Conf. Image Process.*, Oct. 2001, vol. 3, pp. 198–201.

[8] J. Saniie and D. T. Nagle, "Analysis of order statistic CFAR threshold estimators for improved ultrasonic target detection," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 39, no. 5, pp. 618–630, Sep. 1992.

[9] S. Yoon, E. Oruklu, and J. Saniie, "Dynamically reconfigurable neural network architectures for ultrasonic flaw detection," in *Proc. IEEE Ultrason. Symp.*, Oct. 2006, pp. 1377–1380.

[10] D. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard, "Density estimation by wavelet thresholding," Dept. Statist., Stanford Univ., Stanford, CA, 1992. Tech. rep.

[11] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Anal.*, vol. 29, no. 2, pp. 511–546, 1997.

[12] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Proc. SPIE, Wavelet Appl. Signal Image Process. III*, 1995, vol. 2569, pp. 68–79.

[13] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," Bell Laboratories, Murray Hill, NJ, 1996. Tech. rep.

[14] C. Huang, P. Tseng, and L. Chen, "Efficient VLSI architectures of lifting based discrete wavelet transform by systematic design method," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2002, vol. 5, pp. 565–569.

[15] P. Tseng, C. Huang, and L. Chen, "Reconfigurable discrete wavelet transform architecture for advanced multimedia systems," in *Proc. IEEE Workshop Signal Process. Syst.*, Aug. 2003, pp. 137–141.

[16] H. Liao, M. K. Mandal, and B. F. Cockburn, "Efficient architectures for 1-D and 2-D lifting based wavelet transforms," *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp. 1315–1326, May 2004.

[17] W. Wolf, *FPGA Based System Design*, ser. Prentice Hall Modern Semiconductor Design Series. Upper Saddle River, NJ: Prentice-Hall, 2004.

[18] K. Andra, C. Chakrabarti, and T. Acharya, "A VLSI architecture for lifting-based forward and inverse wavelet transform," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 966–977, Apr. 2002.

[19] T. Acharya and C. Chakrabarti, "A survey of lifting-based discrete wavelet transform architectures," *J. VLSI Signal Process.*, vol. 42, no. 3, pp. 321–339, Mar. 2006.

[20] W. Jiang and A. Ortega, "A lifting factorization-based discrete wavelet transform architecture design," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 651–657, May 2000.

[21] Xilinx, Inc., *Virtex-II Pro and Virtex-II Pro X FPGA User Guide, UG012(v4.1)*, Mar. 2007. [Online]. Available: http://direct.xilinx.com/bvdocs/userguides/ug012.pdf

[22] Xilinx, Inc., *OPB IPIF Product Specification DS414*, Dec. 2005. [Online]. Available: http://www.xilinx.com/bvdocs/ipcenter/data_sheet/opb_ipif.pdf

[23] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 45, no. 5, pp. 485–560, Jun. 1992.

[24] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. Consum. Electron.*, vol. 46, no. 4, pp. 1103–1127, Nov. 2000.

[25] K. Andra, C. Chakrabarti, and T. Acharya, "A high-performance JPEG2000 architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 3, pp. 209–218, Mar. 2003.

[26] S. L. Bishop, S. Rai, B. Gunturk, J. L. Trahan, and R. Vaidyanathan, "Reconfigurable implementation of wavelet integer lifting transforms for image compression," in *Proc. IEEE Int. Conf. Reconfigurable Comput. FPGA's*, Sep. 2006, pp. 1–9.

Erdal Oruklu (S'01–M'05–SM'09) was born in Istanbul, Turkey, in 1973. He received the B.S. degree in electronics and communication engineering from the Technical University of Istanbul, Istanbul, in 1995, the M.S. degree in electrical engineering from Bogazici University, Istanbul, in 1999, and the Ph.D. degree in computer engineering from Illinois Institute of Technology, Chicago, in 2005.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, where he is also the Director of the very large scale integration (VLSI) and system-on-a-chip (SoC) research laboratory. His research interests include reconfigurable computing, advanced computer architectures, hardware/software codesign, and embedded systems. The main focus of his studies is the research and development of SoC frameworks for FPGA and VLSI implementations of real-time signal processing applications, including ultrasonic target detection, data compression, and time–frequency transforms.

Dr. Oruklu is a member of Eta Kappa Nu.

Jafar Saniie (S'80–M'81–SM'91) was born in Iran on March 21, 1952. He received the B.S. degree in electrical engineering from the University of Maryland, College Park, in 1974, the M.S. degree in biomedical engineering from Case Western Reserve University, Cleveland, OH, in 1977, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1981.

In 1981, he was with the Department of Applied Physics, University of Helsinki, Helsinki, Finland, where he conducted research in photothermal and photoacoustic imaging. Since 1983, he has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, where he is a Filmer Professor, the Associate Chair, and the Director of the Embedded Computing and Signal Processing Research Laboratory. His research interests and activities include ultrasonic signal and image processing, statistical pattern recognition, estimation and detection, embedded digital systems, digital signal processing with FPGAs, and ultrasonic nondestructive testing and imaging. In particular, he has performed extensive work in the areas of frequency diverse ultrasonic flaw enhancement techniques, embedded signal processing architectures for ultrasonic imaging, ultrasonic data compression, nonlinear signal processing in target detection, ultrasonic imaging of reverberant multilayer structures, morphological processing and pattern recognition in ultrasonic imaging, time–frequency analysis of ultrasonic signals, and applications of neural networks for detecting flaw echoes and classifying microstructural scattering.

Dr. Saniie is a member of Sigma Xi, Tau Beta Pi, and Eta Kappa Nu and had been the IEEE Branch Counselor (1983–1990). He has been a Technical Committee Member of the IEEE Ultrasonics Symposium since 1987 (currently, he is the Chair of Sensors, NDE, and Industrial Applications), was the Program Coordinator and the Local Chair of the Conference on Properties and Applications of Magnetic Materials from 1985 to 2006, was an Editorial Advisory Member of the *Nondestructive Testing and Evaluation Journal* from 1986 to 1996, and was the Associate Editor of the IEEE TRANSACTIONS ON ULTRASONICS, FERROELECTRICS, AND FREQUENCY CONTROL since 1994. He was a recipient of the 1986 Outstanding IEEE Student Counselor Award and of the 2005 and 2006 University Outstanding Faculty Awards in recognition of his excellence in teaching.