System-on-Chip Subband Decomposition Architectures for Ultrasonic Detection Applications

Erdal Oruklu · Joshua Weber · Jafar Saniie

Received: 22 April 2011 / Revised: 26 July 2011 / Accepted: 29 August 2011 / Published online: 22 September 2011 © Springer Science+Business Media, LLC 2011

Abstract This paper presents a hardware efficient systemon-chip (SoC) sensor architecture for ultrasonic imaging applications that uses the split-spectrum processing (SSP) algorithm. The SSP design is realized using recursive subband decomposition techniques for achieving minimal hardware and power consumption. Recursive implementations of discrete Fourier transform (DFT) and discrete cosine transform (DCT) are presented for subband decomposition which result in sparse transform operations and significantly reduced hardware and power requirements. A comparative study and performance results present the advantages of the recursive hardware architecture compared to the conventional implementation of the SSP algorithm using IP cores for FFT.

Keywords Recursive DCT \cdot Ultrasound \cdot Subband decomposition \cdot Goertzel

1 Introduction

Ultrasonic imaging has applications in many industries, ranging from manufacturing, medical imaging, to nondestructive evaluation for quality control and safety. In particular, ultrasonic testing has been widely used for flaw detection and monitoring the integrity of the civil structures such as bridges [1, 2], detection of cracks in welds [3, 4] and concretes [5].

Ultrasonic sensors can identify the problems often hidden within these structures, becoming an effective procedure for the safety, maintenance, repair and life extension of critical structures. However, the nature of measured ultrasonic echoes is complex; measured scattered echoes often consist of many multiple interfering echoes. Detection and unraveling these echoes for evaluating and characterizing source of echo scattering require advanced signal processing methods such as frequency-diverse signal decompositions [6]. These algorithms present a great computational challenge due to real-time sensor data processing and scalability needs. A conventional hardware design based on microcontrollers and digital signal processors falls far short of meeting the combined demands of high speed, compact area, low power and adaptability requirements. On the other hand, Field Programmable Gate Array (FPGA) devices facilitate fast development time and adaptable architectures for signal processing applications in many domains [7–9], including ultrasonic testing and measurements [10]. Within the field of ultrasonics, FPGA use is becoming more prevalent, with studies demonstrating their use in multi-mode techniques for reducing scanning time [11] and in ultrasonic pulsed Doppler flow measurements [10]. Due to the unparalleled adaptability and scalability of FPGAs, they can be re-programmed, designs can be modified, changed and improved continuously with no extra cost overhead.

In this paper, we propose FPGA based smart sensor nodes for ultrasonic imaging algorithms, specifically targeting low-power and reduced hardware implementation suitable for distributed sensor networks. In order to meet these objectives, optimizations are required at both algorithm level and architectural level. The architectures explored in this paper target a well-established ultrasonic flaw detection algorithm, split-spectrum processing (SSP) [6]. The SSP algorithm is based on subband decomposition techniques and it has high computational load and memory

E. Oruklu (⊠) · J. Weber · J. Saniie Illinois Institute of Technology, Chicago, IL, USA e-mail: erdal@ece.iit.edu

requirements. In this study, in order to meet the high performance and low power design goals, recursive filter structures is introduced and realized on FPGA logic. These recursive filter structures are able to perform the signal decomposition with significantly reduced logic resources and power consumption. In addition, the filter structures are able to produce each signal component independently, allowing for fine grained control of parallelism. This enables a sparse transform resulting in a large reduction in computation complexity while maintaining the same overall performance.

In Section 2, SSP algorithm for flaw detection is discussed. Section 3 presents the FPGA based platform designed for exploring multiple hardware/software (HW/ SW) co-design approaches. Recursive implementations and their analysis are presented in Section 4. The results and comparisons of the different architectures are discussed in Section 5.

2 Ultrasonic Detection Algorithms

Ultrasonic target detection is made difficult by the presence of high scattering microstructure noise. This scattering noise is the result of a large number of small randomly distributed scatterers arising from the microstructure of the materials. When the transmitted wavelength of the ultrasonic signal is larger than the microstructure of the material under test, the echoes exhibit Rayleigh scattering [12]. These clutter echoes exhibit a large degree of randomness in amplitude and sensitivity to frequency shifts. Specifically, microstructure scattering results in an upward shift in the expected frequency of broadband ultrasonic scattering echoes or clutter. On the other hand, targets (flaws) are most often much larger than the transmitted wavelength and behave like geometrical reflectors. Consequently, target echoes often display a downward shift in their expected frequency which is caused by the overall effect of attenuation governed by the physical properties of the propagation path. This downward frequency shift of the target echoes is a productive attribute since it enables the exploration of the frequency content of ultrasonic signals and subbands in order to improve the target visibility.

The SSP algorithm uses this fact to achieve decorrelation between the target (flaw) echo and clutter noise. The algorithm, as shown in Fig. 1, works by decomposing the wideband input signal into a series of overlapping narrow subbands using transform techniques such as FFT or DCT. The subbands are placed as a series of overlapping bandpass filters which contain information about the flaw echoes and a subset of the scattering noise. These subband channel outputs are then combined back together in a postprocessing unit utilizing Bayesian [12] or order statistics [6]. Within order statistics, it is shown that an absolute minimizer is able to achieve substantial performance improvements.

Figure 2 shows the output of eight subband channels for an experimental ultrasonic signal using a 5 MHz transducer and testing a steel block with an embedded flaw. It can be seen that clutter echoes are much more susceptible to frequency variation compared to the flaw echo. For SSP, the most important performance metric is the Flaw-to-Clutter (FCR) ratio and it is used to judge the overall performance of the algorithm. It relates the strength of the



Figure 1 Split-spectrum processing target detection algorithm.



Figure 2 Subband channels showing random variation for clutter echoes.

flaw echo to the surrounding clutter noise. This ratio is a direct representation of how a flaw echo can be detected in the surrounding noise. The equation to calculate the FCR is given in (1).

$$FCR = 20*\log_{10}(F/C) \tag{1}$$

where F is the maximum target echo amplitude and C is the maximum clutter echo amplitude.

An example SSP implementation result of ultrasonic experimental data using minimum detector is shown in Fig. 3. The results (typically more than 10 dB improvement) demonstrate the ability of the SSP algorithm to perform flaw detection robustly even when the input FCR is very poor.

3 FPGA Platform and Architectural Investigation

A highly modular and flexible FPGA system, based on Xilinx Virtex 4 FPGA and A/D chips, has been designed and demonstrated in [13]. We leverage the modularity of this platform to compare multiple architectural and implementation variations for efficiency. The modularity provides the reconfigurability of the design to support the many parameter changes necessitated by the SSP algorithm. To accomplish this task, the system tasks are broken into three modules, signal capture, signal processing, and communications as shown in Fig. 4.

Signal capture module is composed of ADC chips which capture the incoming data from the ultrasonic transducer. The ADC converter allows for continuous data acquisition at 14-bits of precision and 105 MHz sample rate. Furthermore, it has a dynamic range of ± 1 V, giving the ability to resolve very small signal changes. The ADC unit is controlled from and provides data to the main FPGA (Virtex-4 FPGA by Xilinx). The incoming signal data is captured using a single ADC chip. The sampled data is the echo response from the pulsed firing of an ultrasonic transducer. The system controls the firing of the transducer and capturing of all echo data coming in from the dedicated ADC chips. It also provides pre-processing, by implementing a configurable amplifier to the incoming data. The sampling of the incoming data is performed independent of the signal processing; allowing for a configurable clock rate and a selectable sample rate for the incoming data.





The communication module provides a channel that allows for command and control from the host PC. It can monitor the status of the FPGA operations through 16 DMA channels providing a high bandwidth direction connection to internal Block RAMs (BRAMs).

The signal processing module is different for each architectural variation. In particular, radix-2, radix-4 FFT IP-cores and recursive implementations of DFT and DCT transforms are investigated. The implementation of interfaces to other modules and the implementation of the order statistic post processor (minimization) remain the same in all designs compared.

As a base reference system, a 4-channel Radix-2 FFT with absolute minimization post processing signal processing module is created. Using higher number of subbands (i.e., 8-channels instead of 4) typically increases the performance of the SSP algorithm by exploiting the frequency sensitivity of the clutter echoes; however, the system hardware requirements are also increased significantly. Both FCR detection performance and resource usage results will be presented in Section 4.



Figure 4 FPGA based ultrasonic imaging platform.

3.1 FFT Architectural Variations

The signal processing base system was modified to utilize three different DFT based implementation techniques including Radix-2 and Radix-4 FFT designs using Xilinx IP cores [14] and a hardware efficient design based on Goertzel's algorithm [15]. The radix-2 and radix-4 FFT designs are based on the well-known Cooley-Tukey technique. These IP cores are highly optimized to utilize the FPGA resources, specifically for efficient use of DSP48 units and embedded memory elements within the Virtex-4 FPGA.

3.2 Goertzel Architecture

Upon inspection of the SSP algorithm, it is clear that not all transform coefficients are used for post-processing. Much of the frequency information (transform coefficients) is unnecessary, and filtered out during the subband decomposition. Specifically, only around 10% of the total frequency information is preserved in the subband decompositions (see Fig. 5 for the frequency region of interest). This is due to the downward frequency shift of the flaw echoes and upward frequency shift of the clutter echoes in the Rayleigh scattering region as explained in Section 2. If we want to detect flaw echoes, low-frequency region is the primary region of interest.

Consequently, we reason that much of the computational load could be reduced by generating only the necessary coefficients. Hence, the use of a sparse transform can be more computationally efficient. In order to take advantage of this fact, we have implemented a Goertzel based Fourier transform [16, 17]. The Goertzel algorithm has the advantage that it can compute each frequency component individually. While slower on average to compute a single frequency component, we leverage the fact that we can now compute only a subset of the total frequency components to produce a more efficient design.

$$X[[k]] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad k = 0, 1, \dots, N-1$$
(2)

where
$$W_N = e^{-j\left(\frac{2\pi}{N}\right)}$$
 (3)

Goertzel takes advantage of the periodicity of W_N^{-kn} in order to reduce computation. We observe that:

$$W_N^{-kN} = e^{j(\frac{2\pi}{N})Nk} = e^{j2\pi k} = 1$$
(4)

Due to (4), right side of (2) can be multiplied by W_N^{-kN} without affecting the result:

$$X[k] = W_N^{-kN} \sum_{r=0}^{N-1} x[r] W_N^{kr} = \sum_{r=0}^{N-1} x[r] W_N^{-k(N-r)}$$
(5)

Define the sequence:

$$y_k[n] = \sum_{r=-\infty}^{\infty} x[r] W_N^{-k(n-r)} u[n-r]$$
(6)

From (5) and (6) and the fact that x[n] = 0 for n < 0 and $n \ge N$, it follows that

$$X[k] = y_k[n]|_{n=N} \tag{7}$$

Hence, X[k] can be obtained after N iteration of a filter with the following system transfer function:

$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}} \tag{8}$$

The Goertzel IIR filter structure is shown in Fig. 6. This filter structure is very straightforward to implement. It uses very few hardware resources and takes advantage of the built-in DSP48 elements in Virtex-4 FPGA [18] to perform fast single cycle multiplications. The small resource







consumption of this implementation is advantageous in many ways. As each frequency component is generated independently, multiple instances of this filter structure can be implemented to calculate in parallel. This allows for flexible control over the performance of the design against the cost in resource consumption. SSP implementation in Fig. 7 illustrates this. If five Goertzel filter kernels are available in the system; they can be used for both forward and inverse Fourier transform operations. For the forward transform, all five kernels operate in parallel, calculating different frequency coefficients since they are independent. Hence, execution time of forward Fourier transform will be five times faster compared to single Goertzel kernel use. For the inverse transform, four subband channels are necessary; each Goertzel kernel executes one inverse DFT. The throughput of the system can be easily doubled by instantiating five more Goertzel filters. In Section 5, we present results utilizing both five and ten Goertzel filter kernels.

In Fig. 7, an absolute minimizer (an order statistics filter) is used for post-processing and the SSP result.

$$y_{\min}(n) = \min[|x_i(n)|, \quad i = 1, 2, \dots, k]$$
 (9)



Figure 7 SSP implementation using 5 Goertzel filter kernels.



where y is the SSP output, x_i is the SSP channel *i*, and *k* is the total number of the SSP channels. This post processor step uses the partially uncorrelated observations and makes use of statistical differences in the channels corresponding to random processes inherent to microstructure and flaw echoes for improved flaw detection. The statistical differences of microstructure and target echoes can be exploited for improving the FCR.

3.3 DCT Architectures

Discrete cosine transform can be used in split-spectrum processing for subband decomposition requiring no complex number operations. In this work, we have created a hardware implementation of the DCT and integrated it into the system platform, allowing comparisons between the performance of DCT and FFT transforms.

As SSP works with very large data sets, most prior work into DCT implementations is of limited applicability. But similar to the Goertzel algorithm, we utilize a technique allowing for a sparse transform. Decomposing the DCT down into a recursive type structure allows for calculation of values in a similar manner to the Goertzel based FFT. DCT can be implemented with a recursive IIR filter structure using Clenshaw's recurrence formula [19]. Although hardware requirements are very basic for the recursive structure, computationally, it requires N² clock cycles for N data points. In [20], faster recursive structures have been presented to improve the computation time. These structures employ a folding operation by exploiting the symmetry properties of the cosine terms. Furthermore, even and odd



Table 1 Flaw-to-clutter ratio improvement.

	Matlab FFT (dB)	HW/SW Co-design (dB)	Hardware 4-channel (dB)	Hardware 8-channel (dB)	Hardware Goertzel DFT (dB)	Matlab DCT (dB)	Recursive DCT (dB)
Average Improvement	10.89	10.08	7.43	10.69	10.79	12.35	10.32
Standard Deviation	3.29	3.14	3.49	1.99	3.23	3.13	3.11

inputs can be processed separately with additional IIR filter blocks.

DCT kernel can be written as:

$$Y[k] = \sqrt{\frac{2}{N}} E_k \sum_{n=0}^{N-1} x[n] \cos \frac{(2n+1)k\pi}{2N} \quad k = 1, 2, \dots, N-1$$
(10)

$$E_k = \sqrt{\frac{1}{2}} \text{ for } k = 0 \text{ and } E_k = 1 \text{ for } k \neq 0$$
(11)

A single folding operation applied to DCT results in:

$$Y[k] = \sqrt{\frac{2}{N}} E_k \sum_{n=0}^{\frac{N}{2}-1} w_k[n] \cos\frac{(2n+1)k\pi}{2N}$$
(12)

where

$$w_k[n] = x[n] + (-1)^k x[N-1-n]$$
(13)

For k = even samples, define:

$$Y[k] = \sqrt{\frac{2}{N}} E_k (-1)^{\frac{k}{2}} g_k \left[\frac{N}{2} - 1\right]$$
(14)

where

$$g_k[j] = \sum_{n=0}^{j} w_k[j-n] \cos\left(n + \frac{1}{2}\right) \theta_k \tag{15}$$

Table 2 FPGA processing time of the SSP algorithm.

Z-transform of the convolution given in Eq. 15 can be represented as a filter with the following transfer function:

$$\frac{G_k[z]}{W_k[z]} = \frac{\cos(\frac{\theta_k}{2})(1-z^{-1})}{1-2\cos\theta_k z^{-1}+z^{-2}}$$
(16)

Similarly for k = odd samples, define:

$$Y[k] = \sqrt{\frac{2}{N}} E_k(-1)^{\frac{k-1}{2}} h_k \left[\frac{N}{2} - 1\right]$$
(17)

where

$$h_k[j] = \sum_{n=0}^{j} w_k[j-n] \sin\left(n + \frac{1}{2}\right) \theta_k$$
(18)

The Eq. 18 can be represented as a filter with the following transfer function:

$$\frac{H_k[z]}{W_k[z]} = \frac{\sin(\frac{\theta_k}{2})(1+z^{-1})}{1-2\cos\theta_k z^{-1}+z^{-2}}$$
(19)

Figures 8 and 9 show second order IIR filter structures for even and odd k, respectively. By implementing these two structures in parallel, two frequency components can be computed every N/2 cycles. Like the Goertzel based Fourier transform, it is straightforward to implement the DCT IIR filters. By using DSP48 elements, we can enable and instantiate additional units to increase performance results.

Algorithm stage	HW/SW codesign (with FFT accelerator)	Hardware radix-4 FFT	Hardware radix-2 FFT	Hardware Goertzel DFT (5 filter kernels)	Hardware Goertzel DFT (10 filter kernels)	Hardware recursive DCT (5 filter kernels)	Hardware recursive DCT (10 filter kernels)
Forward Transform	3,456	1,322	5,190	6,144	3,072	1,536	768
Window Filtering	91,456	1,024	1,024	1,024	1,024	1,024	1,024
Inverse Transform	13,824	1,322	5,190	30,720	15,360	7,680	3,840
Post Processing	196,661	1,024	1,024	1,024	1,024	1,024	1,024
Total Cycles	305,397	4,692	12,428	38,912	20,480	11,264	6,656
Clock Frequency	100 MHz	115 MHz	115 MHz	115 MHz	115 MHz	115 MHz	115 MHz
Total Time	3,050 µs	41 µs	108 µs	338 µs	178 µs	97 μs	57 μs

Table 3FPGA resource usage.

	HW/SW codesign (with FFT accelerator)	Hardware radix-4 FFT	Hardware radix-2 FFT	Hardware Goertzel DFT (5 filter kernels)	Hardware Goertzel DFT (10 filter kernels)	Hardware recursive DCT (5 filter kernels)	Hardware recursive DCT (10 filter kernels)
Slice	6,949	17,365	8,388	4,035	4,797	4,228	5,795
LUTs	8,768	18,302	9,415	5,495	7,089	5,530	7,576
DSP48s	25	90	30	20	40	12	24
RAM16	68	64	34	37	67	34	61

4 Experimental Results

4.1 Flaw-to-Clutter Ratio Enhancements

A set of ultrasonic A-scan data measurements (data size is 1024 samples per A-scan) are processed by the SSP architectures and benchmarked according to their FCR performance. Table 1 presents the average FCR improvement results for all the implementations investigated in this research. It is also important to point out that parameter changes such as location, overlap amount and the number of subbands can induce significant impact on FCR performance. For fair comparison, these parameters were identical in each implementation.

The FFT and the DCT results are obtained from Matlab implementations which use floating-point representation and serve as reference point and benchmark for FPGAbased hardware designs. A hardware/software (HW/SW) codesign implementation [13] which uses a C-program running on the soft-core Microblaze processor with a dedicated FFT accelerator IP-core is also shown in Table 1. Variations between architectures occur due to the impact of finite word-length precision (16-bit internal datapath used in all cases). In general, the recursive structures based on Goertzel and DCT are able to achieve nearly identical performance to Matlab implementations, outperforming FFT IP-core designs. Accumulator registers have been tailored to prevent overflow while maintaining the highest level of precision.

4.2 Execution Time

The main objective of the proposed ultrasonic smart sensor architectures is to achieve real-time processing of Amplitude Scan (A-Scan) data with minimal resource usage and power dissipation. Typically, a processing rate exceeding 1 KHz can be considered real-time for ultrasonic imaging. This gives only a 1 ms time window to perform capture and processing of A-Scan. The execution time results for all the architectures are shown in Table 2. Execution times for each processing step in the SSP algorithm are also presented in Table 2.

All of the hardware architectures are able to achieve the necessary repetition rate. It is important to note that the proposed recursive DCT structures are able to perform faster than radix-2 FFT implementations due to sparse transform operations. In addition, the recursive structures are much more adaptable to performance requirements. Since the recursive structures are able to produce each frequency component separately, multiple components could be produced in parallel by instantiating additional filter structures. Hence, doubling the number of filter kernels from 5 to 10 reduces the computation time almost by half for both Goertzel and DCT implementations (see Table 2). This allows for direct control over the trade-off between needed performance and resource consumption. The fastest implementation is based on radix-4 FFT IP-core; however, it is also the most expensive implementation with respect to area and power.



Resource consumption for all the designs is presented in Table 3. For the target platform Virtex-4 FPGA, there are four major resources, slices of configurable logic, look-up tables (LUTs), embedded DSP48 multiply and accumulators, and embedded block RAM. Radix-4 FFT IP core implementation requires the largest resource consumption with minimal performance gain against recursive techniques. Goertzel and DCT implementations use similar amount of resources with the exception of DSP48 component. Significant savings (i.e., 75% and 50% less logic slices; 75% and 33% less DSP48s compared to radix-4 and radix-2 techniques, respectively) are observed against conventional techniques while using five filter kernels in parallel. Table 3 also indicates that scaling is very efficient. Using ten filter kernels increase the hardware resources marginally while almost doubling the performance.

Power consumption results are shown in Fig. 10. Total dynamic power results follow the trend observed in resource usage given in Table 3. Recursive architectures, in particular DCT implementation, dissipate less dynamic power (i.e., 25% compared to radix-2 implementation). Due to fixed properties of the FPGA fabric, static power consumption is almost same and the margin of difference for dynamic power is not as high as expected among different architectures. However, for an ASIC implementation, power savings would be much more pronounced. If the timing requirements are relaxed, the HW/SW codesign can be used for least power consumption. It offers 50% less power consumption against recursive DCT.

5 Conclusion

In this paper, a hardware efficient implementation of ultrasonic detection algorithms is studied using an FPGA based smart sensor platform. Recursive filters and sparse transform operations are proposed for reducing area and power while achieving real-time operation. The synthesis results show that all design corners are improved when compared against the traditional FFT implementations and HW/SW codesign plat-forms. The recursive architectures presented here are scalable, power efficient and especially well-suited for distributed sensor applications such as structural health monitoring where smart and sustainable ultrasonic sensors are required.

References

 Sinha, S. K., Schokker, A. J., & Iyer, S. R. (2003). Non-contact ultrasonic imaging of post tensioned bridges to investigate corrosion and void status. *IEEE Proceedings of Sensors*, 1, 487–492.

- Mori, H., Oshima, T., Mikami, S., Honma, M., & Funatsu, M. (1994). "Effect of individual decision of bridge expert on total evaluation of bridge integrity". *Journal of Constructional Steel*, 2.
- Chassignole, B., Villard, D., Dubuget, M., Baboux, J. C., & El Guerjouma, R. (2000). Characterization of austenitic stainless steel welds for ultrasonic NDT. *Review of Progress in QNDE*, 20, 1325–1332.
- Halkjaer, S., Sorensen, M. P., & Kristensen, W. D. (2000). The propagation of ultrasound in a austenitic weld. *Ultrasonics*, 38, 256–261.
- 5. Carino, N. J., Sansalone, M., & Hsu, N. H. (1986). "Flaw detection in concrete by frequency spectrum analysis of impactecho waveforms". *International Advances in Nondestructive Testing*, 12.
- Saniie, J., Nagle, D., & Donohue, K. (1991). Analysis of order statistic filters applied to ultrasonic flaw detection using splitspectrum processing. *IEEE Transactions on Ferroelectrics and Frequency Control, 38*(2), 133–140.
- Lu, Y., Oruklu, E., & Saniie, J. (2008). Fast chirplet transform with FPGA-based implementation. *IEEE Signal Processing Letters*, 15(1), 577–580.
- Jung, S., & Kim, S. S. (2007). Hardware implementation of a realtime neural network controller with a DSP and an FPGA for nonlinear systems. *IEEE Transactions on Industrial Electronics*, 54(1), 265–271.
- Rodriguez-Andina, J. J., Moure, M. J., & Valdes, M. D. (2007). Features, design, tools, and application domains of FPGAs. *IEEE Transactions on Industrial Electronics*, 54, 1810–1823.
- Hong Hu, C., Zhou, Q., & Shung, K. (2008). Design and implementation of high frequency ultrasound pulsed-wave Doppler using FPGA. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 55(9), 2109–2111.
- Hernandez, A., Urena, J., Hernanz, D., Garcia, J. J., Mazo, M., Derutin, J. P., et al. (2003). Real-time implementation of an efficient Golay correlator (EGC) applied to ultrasonic sensorial systems. *Microprocessors and Microsystems*, 27(8), 397–406.
- Saniie, J., & Nagle, D. (1992). Analysis of order statistic CFAR threshold estimators for improved ultrasonic flaw detection. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 35(5), 618–630.
- Weber, J., Oruklu, E., & Saniie, J. (2011). FPGA-based configurable frequency diverse ultrasonic target detection system. *IEEE Transactions on Industrial Electronics*, 58(3), 871–879.
- Xilinx LogiCORE IP Fast Fourier Transform (FFT), Product Specification, DS260, June 2009. Available at: http://www.xilinx. com/support/documentation/ip documentation/xfft ds260.pdf.
- Beck, R., Dempster, A. G., & Kale, I. (2001). Finite-precision Goertzel filters used for signal tone detection. *IEEE Transactions* on Circuits and Systems-II: Analog and Digital Signal Processing, 48(7), 691–700.
- Hwang, J.-K., & Li, Y.-P. (2010). Efficient recursive IDFT scheme for complex-valued signals in tap-selective maximum-likelihood channel estimation. *Journal of Signal Processing Systems*, 60(1), 71–80.
- Oruklu, E., Weber, J., & Saniie, J. (2008). "Recursive filters for subband decomposition algorithms in ultrasonic detection applications", *IEEE Ultrasonics Symposium*, pp. 1881–1884, November 2008.
- Xilinx XtremeDSP for Virtex-4 FPGAs User Guide, UG073, May 15, 2008. Available at http://www.xilinx.com/support/documentation/ user guides/ug073.pdf.
- Aburdene, M. F., Zheng, J., & Kozick, R. J. (1995). Computation of discrete cosine transform using Clenshaw's recurrence formula. *IEEE Signal Processing Letters*, 1(7), 101–102.
- Chen, C., Liu, B., Yang, J., & Wang, J. (2004). Efficient recursive structures for forward and inverse discrete cosine. *IEEE Transactions on Signal Processing*, 52(9), 2665–2669.



Erdal Oruklu received his B.S. degree in Electronics and Communications Engineering from Technical University of Istanbul in 1995 and M.S. degree in Electrical Engineering from Bogazici University, Istanbul, Turkey in 1999. He received his Ph.D. degree in Computer Engineering from Illinois Institute of Technology, Chicago, Illinois in 2005. He is currently an Assistant Professor at Department of Electrical and Computer Engineering in Illinois Institute of Technology where he is also the Director of VLSI and SoC Research Laboratory. Dr. Oruklu's research interests are signal processing hardware, reconfigurable computing, advanced computer architectures, hardware/software co-design and embedded systems.



Joshua Weber received his B.S.E degree in Computer Systems Engineering from Arizona State University in 2005 and M.S degree in Computer Engineering from Illinois Institute of Technology in 2008. He is currently pursuing his Ph.D. at Illinois Institute of Technology. He is a member of the VLSI and SOC Research Laboratory. Mr. Weber's research interests are in embedded systems, reconfigurable computing, hardware/software co-design and digital logic design with FPGAs.



Jafar Saniie received his B.S. degree in electrical engineering from the University of Maryland in 1974. He received his M.S. degree in biomedical engineering in 1977 from Case Western Reserve University, Cleveland, OH, and his Ph.D. degree in electrical engineering in 1981 from Purdue University, West Lafayette, IN. In 1981 Dr. Saniie joined the Department of Applied Physics, University of Helsinki, Finland, to conduct research in photothermal and photoacoustic imaging. Since 1983 he has been with the Department of Electrical and Computer Engineering at Illinois Institute of Technology where he is a Filmer Professor, Associate Chair and Director of the Embedded Computing and Signal Processing (ECASP) Research Laboratory. Dr. Saniie's research interests and activities are in ultrasonic signal and image processing, statistical pattern recognition, estimation and detection, embedded digital systems, digital signal processing with field programmable gate arrays, and ultrasonic nondestructive testing and imaging. Dr. Saniie is an IEEE Fellow for his contributions to "Ultrasonic Signal Processing for Detection, Estimation and Imaging".