

Recursive Filters for Subband Decomposition Algorithms in Ultrasonic Detection Applications

Erdal Oruklu, Joshua Weber and Jafar Saniie
Department of Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, Illinois, 60616

Abstract—In this study, we present a low-cost, low-power, and compact hardware platform for real-time ultrasonic detection applications. In order to cope with the computational complexity of the inherent signal processing algorithms, recursive IIR filter architectures are proposed for realizing the frequency diverse transforms; discrete Fourier transform (DFT) and discrete cosine transform (DCT), which provide robust detection performance. Both Goertzel (for DFT implementation) and folding kernel for DCT offer significant hardware reductions compared to standard transform implementations. In order to sustain the high throughput required for real-time signal processing, a sparse transform method is proposed where only the required transform coefficients are computed. FPGA implementation results confirm savings in hardware usage and viable performance for a portable ultrasonic imaging device.

I. INTRODUCTION

Embedded processing architectures are rapidly advancing and making a large impact in many fields. Specifically, Field Programmable Gate Arrays (FPGA) devices facilitate fast development time and adaptable architectures for signal processing applications in many domains [1]. In [2], a hardware platform has been presented to demonstrate the feasibility of FPGA based real-time ultrasonic detection for non-destructive evaluation (NDE) applications. In NDE detection applications, high scattering noise (clutter) echoes must be suppressed by signal processing methods such as frequency-diverse transforms and subband decomposition. Computational complexity of these algorithms and real-time requirements place tight constraints on the execution time and bandwidth requirements; requiring expansive and power inefficient hardware units.

In this study, we propose recursive filter architectures for realizing ultrasonic target detection algorithms, specifically targeting low-cost, low-power and compact devices. Recursive filters are utilized for hardware efficient implementation of Fourier transform (DFT) and discrete cosine transform (DCT). Both DFT and DCT can be used for subband decomposition and frequency analysis of ultrasonic signals. In particular, split-spectrum processing (SSP) algorithm [3] has been shown to work with DFT and DCT transforms with robust detection results.

For recursive DFT realization, the Goertzel algorithm [4], [5] is used; for recursive DCT implementation, folded DCT kernel [6], [7] is used. Both recursive implementations use simple IIR filters and significantly reduce hardware requirements compared to the conventional transform implementa-

tions, at the expense of overall system throughput. However, in order to speed up the transform operations, a *sparse* transform operation can be performed by exploiting the unique characteristic of the SSP algorithm: SSP algorithm does not act on the whole set of frequency data; only a subset of the whole frequency response is used, the rest is discarded during the subband filtering stage. Recursive filter implementations generate each transform coefficient output independent from other coefficients, therefore, significant computation can be avoided by selectively generating transform coefficients.

In Section II, we discuss the ultrasonic detection algorithms that utilize frequency-diverse transforms. Section III shows the derivation of recursive transform implementations. Proposed FPGA system architecture and the performance results are presented in Section IV and Section V, respectively.

II. ULTRASONIC DETECTION ALGORITHMS

Ultrasonic target detection is made difficult by the presence of high scattering micro structure noise. The use of frequency diverse testing provides ultrasonic echo data containing high amounts of statistical variation in scattering noise. This scattering noise is the result of a large number of small randomly distributed scatters arising from the micro structure of the material. However, the clutter noise are sensitive to frequency shifts whereas flaw/target echoes are less susceptible to frequency shifts. The SSP algorithm uses this fact to achieve decorrelation between the target echo and clutter noise. The algorithm, as shown in Figure 1, works by decomposing the wide band input signal into a series of overlapping narrow subbands using orthogonal transforms such as DFT and DCT. Due to the fact that clutter noise is highly frequency sensitive, clutter noise will exhibit high degree of randomness while the target information will remain relatively constant across subbands. These subbands are then combined back together in a post-processing unit utilizing order statistics [3]. Within order statistics, it has been shown that an absolute minimizer is able to achieve substantial performance improvements.

For SSP, the most important performance metric is the target-to-clutter (TCR) ratio and it is used to judge the overall performance of the algorithm. Example SSP implementation results (using DFT and DCT transforms) with ultrasonic experimental data are shown in Figure 2. The results (typically >10dB improvement) demonstrate the ability of the SSP

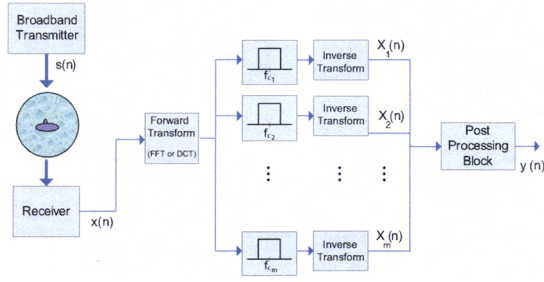


Fig. 1. Split spectrum processing

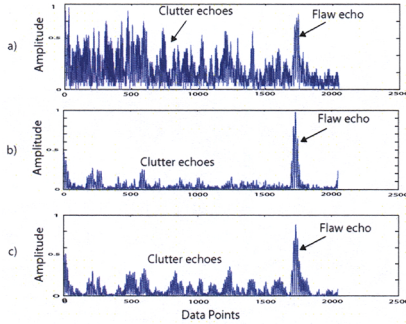


Fig. 2. FCR improvement using SSP algorithm a) Experimental signal b) FFT detection result c) DCT detection result

algorithm to perform target detection robustly even when the input TCR is very poor.

III. RECURSIVE FILTERS FOR SUBBAND DECOMPOSITION

For Fourier transform, the recursive filter architecture is derived from the Goertzel algorithm, yielding a first-order IIR filter. Similarly, DCT can be realized as a recursive structure if kernel folding is applied to DCT. The resulting kernel is a second-order IIR filter. For calculating even and odd DCT coefficients, two filter structures can be used in parallel for doubling the computation rate.

A. Goertzel based Fourier Transform

Fourier transform kernel can be written as:

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

$$W_N = e^{-j(2\pi/N)} \quad (2)$$

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

$$W_N^{-kN} = e^{j(2\pi/N)Nk} = e^{j2\pi k} = 1 \quad (3)$$

Due to (3), we can multiply the right side of (1) by W_N^{-kn} without affecting the result.

$$X[k] = W_N^{-kN} \sum_{r=0}^{N-1} x[r] W_N^{kr} = \sum_{r=0}^{N-1} x[r] W_N^{-k(N-r)} \quad (4)$$

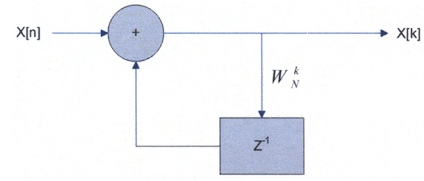


Fig. 3. Goertzel DFT Block Diagram

Define the sequence

$$P_k[n] = \sum_{r=-\infty}^{\infty} x[r] W_N^{-k(n-r)} u[n-r] \quad (5)$$

From (4) and (5) and the fact that $x[n] = 0$ for $n < 0$ and $n \geq N$, it follows that

$$X[k] = P_k[n] \Big|_{n=N} \quad (6)$$

Examining (5) and (6), indicate that $X[k]$ can be obtained after N iteration of a filter with the following system function:

$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}} \quad (7)$$

The Goertzel IIR filter structure is shown in Figure 3.

B. Recursive filter architectures for DCT

Discrete Cosine Transform kernel can be written as:

$$y[k] = \sqrt{\frac{2}{N}} E_k \sum_{n=0}^{N-1} x[n] \cos \frac{(2n+1)k\pi}{2N} \quad k = 1, 2, \dots, N-1 \quad (8)$$

$$E_k = \sqrt{\frac{1}{2}} \text{ for } k = 0 \text{ and } E_k = 1 \text{ for } k \neq 0 \quad (9)$$

DCT can be implemented with a recursive IIR filter structure based on Clenshaw's recurrence formula [6]. Although hardware requirements are very basic for the recursive structure, computationally, it requires N^2 clock cycles for N data points. In [7], 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 inputs can be processed separately with additional IIR filter blocks. A single folding operation can be described as:

$$w_k[n] = x[n] + (-1)^k x[N-1-n] \quad (10)$$

and

$$y[k] = \sqrt{\frac{2}{N}} E_k \sum_{n=0}^{\frac{N}{2}-1} w_k[n] \cos \frac{(2n+1)k\pi}{2N} \quad (11)$$

For $k = \text{even samples}$, define:

$$\text{Let } g_j(k) = \sum_{n=0}^j w_k[j-n] \cos(n + \frac{1}{2})\theta_k \quad (12)$$

$$y[k] = \sqrt{\frac{2}{N}} E_k (-1)^{\frac{k}{2}} g_{\frac{N}{2}-1}[k] \quad (13)$$

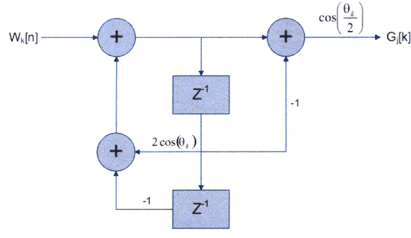


Fig. 4. Recursive DCT for even coefficients

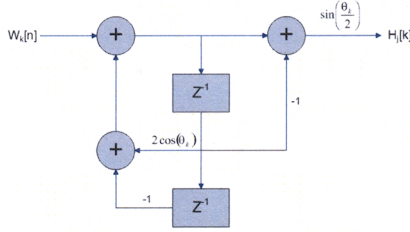


Fig. 5. Recursive DCT for odd coefficients

Using the Chebyshev polynomials on (12), a recursion is introduced:

$$g_j[k] = \cos\left(\frac{\theta_k}{2}\right)\{w_k[j] - w_k[j-1]\} + 2 \cos(\theta_k)g_{j-1}[k] - g_{j-2}[k] \quad (14)$$

Equation (14) can be represented as a filter with the following system function:

$$\frac{g_k(z)}{w(z)} = \frac{\cos(\frac{\theta_k}{2})(1 - z^{-1})}{1 - 2 \cos \theta_k z^{-1} + z^{-2}} \quad (15)$$

With a similar derivation, for $k = \text{odd}$ values, define:

$$h_j[k] = \sum_{n=0}^j w_k[j-n] \sin[(n + \frac{1}{2})\theta_k] \quad (16)$$

$$y[k] = \sqrt{\frac{2}{N}} E_k (-1)^{\frac{k-1}{2}} h_{\frac{N}{2}-1}[k] \quad (17)$$

Using the Chebyshev polynomials on (16), a recursion is introduced:

$$h_j[k] = \sin\left(\frac{\theta_k}{2}\right)\{w_k[j] - w_k[j-1]\} + 2 \cos(\theta_k)h_{j-1}[k] - h_{j-2}[k] \quad (18)$$

Equation (18) can be represented as a filter with the following system function:

$$\frac{h_k(z)}{w(z)} = \frac{\sin(\frac{\theta_k}{2})(1 + z^{-1})}{1 - 2 \cos \theta_k z^{-1} + z^{-2}} \quad (19)$$

Figure 4 and Figure 5 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 $\frac{N}{2}$ cycles.

In the following section, an FPGA implementation of the SSP algorithm using recursive filters is presented.

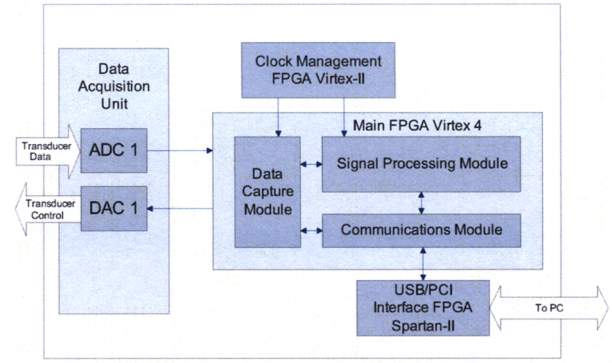


Fig. 6. Hardware platform overview

IV. SYSTEM IMPLEMENTATION

An existing architecture (presented in [2]) is used as a reference platform. A block diagram of the platform is shown in Figure 6. This architecture is then implemented on a Virtex 4 FPGA using a Nallatech XtremeDSP development kit. Since the system has been designed with a high degree of modularity, and the individual modules are independent, it is straight forward to replace the transform implementations. The original hardware platform uses a conventional Fourier transform implementation which is either based on radix-2 or radix-4 FFT algorithm. The SSP algorithm is executed in parallel, with separate computational units for the forward transform and each of the inverse transform units. For comparison purpose, the reference platform is modified by swapping the transform modules. FFT and inverse FFT modules are replaced with Goertzel based units or with recursive DCT implementations. No other component is replaced. This allows for an objective comparison between the performance of the conventional transform implementations with the recursive methods.

A. Goertzel FFT Implementation

The Goertzel filter structure shown in Figure 3 can calculate a single frequency component output after N iterations. It can be further noted that this implementation requires very few resources; a single complex adder and a single complex multiplier. A complex adder is easily constructed from two parallel standard adders. The complex multiplier is implemented as a Xilinx IP core. The IP core utilizes the built-in DSP48 elements (providing hardware 18×18 multiplier and accumulator units). In order to reduce computational effort, W_N^k is precomputed for all k and a fixed $N = 1024$ and placed in a ROM for quick access.

The primary advantage of the Goertzel DFT implementation is its ability to selectively calculate the frequency components and perform a sparse transformation. For 1024-point input data samples, it has been demonstrated that subband bandwidth of interest is limited to approximately 30 frequency components. When a traditional DFT method is used, a large percentage of the calculated components are discarded.

Inverse DFT operations can also be reduced. Although all 1024 points need to be calculated, we can use the knowledge of the incoming signal to reduce computational complexity. Since the incoming signal to the inverse DFT stage is filtered by a windowing operation, it will consist of a string of '0' coefficients leading and following the actual subband data. To reduce computation time, all the multiplication factors can be combined together into a single value. This enables computation of each resulting data point (in time domain) in only 30 clock cycles. These savings in forward and inverse transform operations provide improved efficiency to the design.

B. Recursive DCT Implementation

The recursive DCT structure performs in a similar manner to the Goertzel algorithm, using an IIR filter structure to calculate frequency components individually. Implementing two IIR filters in parallel consumes a total of 7 total adders and 4 multipliers. Note the additional adder required to perform the input folding as demonstrated in (10). Unlike the Fourier transform, the DCT is a real transform and hence all adders and multipliers are also real. Once again, all trigonometric factors are precomputed and placed in ROM memory to be accessed by the multipliers. Similar to Goertzel DFT, recursive DCT kernel calculates only the necessary transform coefficients, providing a major improvement to system throughput.

V. PERFORMANCE AND RESOURCE USAGE

Total execution times for the various SSP stages are shown in Table I. Execution times are calculated based on a clock rate of 115 MHz for all architectures. FFT algorithm is able to perform with higher throughput compared to the recursive implementations. This is expected since they need to calculate every transform coefficient with $O(N)$ iterations. Nevertheless, through sparse transform operations, the total execution time is still comparable to FFT and meets the typical timing requirements of real-time ultrasonic imaging. The difference between the Goertzel and recursive DCT is due to the initial folding stage, and simultaneous generation of two frequency components (even and odd) every $\frac{N}{2}$ cycles in DCT realization.

The Goertzel and recursive DCT implementations have several advantages over the FFT. The execution time becomes an easily controllable parameter of the design. The recursive algorithms produce each frequency component independently. This introduces a higher level of data parallelism into the design. It is trivial to reduce transform time by instantiating additional Goertzel or DCT processing units. Through the introduction of one more recursive filter, both the transform and inverse transform execution time can be halved.

Another important metric for measuring performance is resource consumption. Specifically, in a Virtex-4 FPGA, there are three major hardware resources; slices of configurable logic, embedded DSP48 multiply and accumulators, and embedded block RAM. The total resource consumption for the complete system utilizing each of the implementation algorithms is shown in Table II. Since Goertzel and DCT

TABLE I
FPGA PROCESSING TIME OF THE SSP ALGORITHM

Algorithm Stage	Hardware (Radix-2 FFT) (Cycles)	Hardware (Goertzel) (Cycles)	Hardware (DCT) (Cycles)
Frequency Transform	5,190	30,720	7,680
Window Filtering	1,024	1024	1,024
Inverse Transform	5,190	30,720	7,680
Post Processing	1,024	1024	1,024
Total cycles	12,428	63,488	17,408
Total time	108 μ s	552 μ s	151 μ s

TABLE II
FPGA RESOURCE USAGE

	Radix-2	Goertzel DFT	Recursive DCT
Slices	8,378	3,434	3,928
DSP48	30	16	20
RAM16	28	48	50

implementations utilize very simple IIR filters, they lack the complexity of FFT implementation. Correspondingly, this imposes a very large reduction in overall resource consumption except the RAM usage. Cooley-Tukey algorithm in FFT implementations makes very efficient use of memory storage, utilizing in-place memory calculations. On the other hand, Goertzel and recursive DCT use separate input and output memory units and additional ROMs to store all precomputed coefficients.

VI. CONCLUSION

This paper presents two recursive architectures for frequency-diverse transforms in ultrasonic detection applications. Transform coefficients can be calculated independently from each other; hence, only a subset of the overall frequency components are computed in the SSP algorithm. Close to 50% reduction in resource consumption can be achieved with only a moderate degradation in execution time, compared with FFT. They introduce more data level parallelism, allowing for configuration of multiple processing units for a trade-off between resource consumption and overall execution time.

REFERENCES

- [1] J. J. Rodriguez-Andina, M. J. Moure, and M. D. Valdes, "Features, design, tools, and application domains of FPGAs," *IEEE Transactions on Industrial Electronics*, vol. 54, pp. 1810–1823, August 2007.
- [2] J. Weber, E. Oruklu, and J. Saniie, "Configurable hardware design for frequency-diverse target detection," in *IEEE International Midwest Symposium on Circuits and Systems*, August 2008, pp. 890–893.
- [3] J. Saniie, D. Nagle, and K. Donohue, "Analysis of order statistic filters applied to ultrasonic flaw detection using split spectrum processing," *IEEE Transactions on Ferroelectrics and Frequency Control*, vol. 38, no. 2, pp. 133–140, March 1991.
- [4] T. Dulik, "An FPGA implementation of Goertzel algorithm," in *Lecture Notes in Computer Science*, vol. 1673, September 1999, pp. 339–346.
- [5] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing Principles, Algorithms, and Applications*, 4th ed. Prentice Hall, April 2006.
- [6] M. F. Aburdene, J. Zheng, and R. J. Kozick, "Computation of discrete cosine transform using Clenshaw's recurrence formula," in *IEEE Signal Processing Letters*, vol. 1, no. 7, August 1995, pp. 101–102.
- [7] C. Chen, B. Liu, J. Yang, and J. Wang, "Efficient recursive structure for forward and inverse discrete cosine," in *IEEE Transactions on Signal Processing*, vol. 52, no. 9, September 2004.