Configurable Hardware Design for Frequency-Diverse Target Detection

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

Abstract—In this paper, we present a reconfigurable hardware architecture and the accompanying application software to perform frequency-diverse target detection. This architecture can be utilized in diverse applications and environments such as real-time ultrasonic testing due to the adaptability of the design. The system design allows for operational level and design level configurability. In addition to the design flow, this paper presents the impact of parameter changes on the detection algorithm performance and FPGA implementation results. The experimental results show that the proposed modular hardware architecture meets real-time operation timing requirements and the algorithm performs robustly.

I. INTRODUCTION

Ultrasonic target (flaw) detection has applications in many industries ranging from manufacturing to maintenance of materials. To achieve target detection, high scattering noise (clutter) echoes must be suppressed by signal processing methods. In addition, real-time requirements place tight constraints on computation time and increase communication bandwidth needs for the capture, processing, analysis, and presentation of data. The system architecture presented in this paper takes advantage of well established flaw/target detection algorithms, specifically Split-Spectrum Processing (SSP) [1]. The SSP algorithm is based on the subband decomposition technique and contains many parameters (such as number of bandpass filters, filter bandwidths and locations) that strongly impact its performance with different data sets. Through the configurable implementation of the real time SSP processing unit, the flexibility of the design to the changes in SSP parameters can be maximized and all the requirements can be met.

FPGA devices facilitate fast development time and adaptable architectures for ultrasonic sensors and signal processing applications [2], [3]. In [4], a hardware platform was presented to demonstrate the feasibility of FPGA based SSP applications. In this work, the configurable architecture is further improved by introducing a greater flexibility in the design and detailed hardware & software implementation results are presented and compared. In the proposed configurable architecture, the system flexibility is present in two forms. The first is through highly modular design. With the emphasis on decomposition of the algorithm to independent processing blocks, a flexible system can be created. By increasing or decreasing the total number of processing units or by replacing individual units, the impact of each change on the overall system can be evaluated in a quick and robust manner. The second core flexibility is the creation of a control channel to the host computer. Through this communication channel, the host computer can configure many parameters of the design on the fly during runtime. This allows the system operator to tailor the parameters to the current conditions. These two forms of flexibility allow the user to evaluate the parameter impact on the results, both in performance of the algorithm and in resource utilization. Therefore, the design space can be optimized through these interactions to meet all requirements while at the same time minimizing resource usage.

II. SPLIT SPECTRUM PROCESSING

SSP is based on the fact that clutter echoes exhibit randomness in amplitude and are highly sensitive to transmitted frequency. Conversely, target echoes are less vulnerable to transmitted frequency. The SSP algorithm uses this fact to achieve decorrelation between the target echo and clutter [1]. The algorithm, as shown in Fig. 1, works by decomposing the wide band input signal into a series of overlapping narrow subbands. These subbands are then combined back together in a post-processing unit, most commonly an absolute minimizer. This technique allows for near optimum Flaw-to-Clutter (FCR) ratio improvements [1].

The choice of the narrow subbands is very important to the performance of the algorithm. The subbands are a series of overlapping bandpass filters. Their placement is controlled by three primary parameters, starting frequency offset, subband width, and overlap. N filters of width W are placed, each starting at

\[
\text{Offset} + (n - 1) \times \text{Overlap} \quad \text{for } n = 1 \cdots N
\]

To maximize the performance of this algorithm, it is important to optimize the filter placement. Each subband needs to contain the target echo and a subset of the overall clutter in order to perform optimally. Performance of the algorithm is also dependent on the number of subbands used, with a trend of higher FCR ratios for more subbands. For SSP, the most important performance metric is the FCR ratio and FCR is used to judge the overall performance of the algorithm. SSP implementation result of ultrasonic experimental data is shown in Fig. 2. The results (typically >10dB improvement) demonstrate the ability of the SSP algorithm to perform flaw detection robustly even when the input FCR is very poor.
III. SYSTEM IMPLEMENTATION

A. System Overview

An overview of the test system is shown in Fig. 3. The system is composed of three major components. The first is a host computer, providing control of the system to the user and presentation of all data and results. Running on this PC system is application software developed to enable communication with the hardware components. The next component is an XtremeDSP hardware development board produced by Nallatech [5]. This board combines a Xilinx Virtex 4 FPGA with dedicated ADC and DAC chips into a convenient package. The final component is the transducer and pulse generator. The pulse generator takes in a low voltage TTL level signal and generates the high voltage pulse to fire the transducer. In addition it provides pre-processing of the reflected echo signal.

B. Hardware Realization

1) Hardware Overview: The hardware design facilitates increased modularity. The modularity gives the ability to quickly modify the design to support the many parameter changes necessitated by the SSP algorithm. As the first step for achieving this goal, the system tasks are broken into three tasks, communications, signal processing, and signal capture.

The modules for the system have been designed in such a manner that they act independently of each other. They are all clocked from separate clock domains, allowing very fine grained control of clocking to each part of the design. In addition, they do not make assumptions about the other modules within the system. This allows the modules to be highly reusable. Any module can be redesigned or changed in the system without having to make structure changes to any other part of the design. This technique has proved to be invaluable aid in research, allowing quick reconfigurations of the system to test individual parameter impacts.

All three modules are packaged in a Xilinx Virtex 4 FPGA [6]. This FPGA is provided on the Nallatech ExtremeDSP development kit as the main user FPGA. As seen in Fig. 4, this board also provides other components used in our system. They include dedicated ADC and DAC chips, fixed and programmable oscillators, a dedicated Virtex-II FPGA for clock management, and a dedicated Spartan-II FPGA for PCI/USB communications control.

2) Data acquisition unit: At the front end of the system is the Data Acquisition Unit. This 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 ±1V, giving the ability to resolve very small signal changes.

3) Signal Capture Module: The signal capture module controls the firing of the transducer and capturing of all echo data coming in from the dedicated ADC chips. It also provides a level of pre-processing, by adding a configurable amplifier to the incoming data. It is important to note that the clock domain in this module is separate from the data processing element. The result of this is that the sampling rate of the device is independent of the clock rate of the processing unit. This enables optimization of the clock rate and sample rate independently for maximum performance.

4) Communication Module: The communication module provides an interface to the host PC and oversees all communications. It provides two primary services. The creation of a register file and the ability to access those registers through a memory mapped interface and a DMA interface that...
can be easily connected to internal Block RAMs (BRAMs). It accomplishes this through the aid of a separate dedicated FPGA component. This FPGA provides all the PCI or USB interfacing requirements, simplifying the design to a more manageable interface. The design of the register file is independent of the overall system. Registers can be added or removed through small code changes. In addition, registers can be made read or write only with minimal code impact.

In addition to direct memory mapped access is support for DMA requests. The system provides 16 separate channels for DMA access. These 16 channels can be connected to any internal BRAM within the FPGA. This is a very beneficial in debugging and verification of the system.

5) Signal Processing Module: This module implements the SSP algorithm. It performs all the transformations and computations to produce the final output data. It gathers its input data from the signal capture module, and obtains parameter values from the communication module. As data flow into the module, it is first transformed to the frequency domain through a FFT module. The FFT module used is a Radix-2 based IP core provided by Xilinx [7]. This module is highly optimized by Xilinx to take full advantage of the dedicated DSP blocks within the Virtex 4 device. As a design configuration parameter, FFT operation is implemented with an 8-bit input precision. The data is allowed to dynamically grow to a 19-bit output, which is then truncated to 16-bits to be passed on to the rest of the design. The output of the FFT module is then decomposed by narrowband filters. The filters are dynamic in their set points. Both the center frequency and bandwidth of the filters is controlled by user writeable registers in the communication module. This allows the user to change the narrowband filter parameters without reconfiguration of the FPGA device. The subband channels are then transformed through the use of IFFT modules. Since all subbands are independent, parallel IFFT modules are used to increase performance. The IFFT modules, like the FFT module, are highly optimized Xilinx IP cores. They take the 16-bit input and allow it to grow to 27-bits. The resulting data are then processed without further truncation. By eliminating truncation, a very high dynamic range is achieved. This helps the hardware design to perform very closely to the ideal software representation. As a final step the subbands are combined into the final output. The system currently uses an absolute minimum post processor. This technique is simple to implement and allows for near optimum PCR performance.

6) Design Advantages: The system architecture has been created with modularity and reconfigurability in mind. This configurability is provided at two levels:

i. Design Configurability
   Design configuration takes advantage of the modularity of the system. It enables subsystem modifications within the design to facilitate large changes to the algorithm, such as changes in number of channels or changes between types of modules themselves.
   - Number of subband channels
   - Bit lengths of processing
   - Choice of frequency transformation implementations, Radix-2/Radix-4 for FFT.
   - Choice of post processing techniques, e.g. minimization, maximization, averaging, neural networks.

ii. Operational Configurability
   Operational configuration provides run-time flexible parameters, algorithmic changes that can be controlled through the user interface, without requiring a reprogramming of the device.
   - Subband Window Parameters (Width, Overlap, and Offset)
   - Data sampling rate and acquisition window location
   - Signal pre-processing and gain

IV. APPLICATION SOFTWARE

The application software has been developed to work in conjunction with the hardware. It primarily acts as a user friendly Graphical User Interface (GUI). This allows for configuration and programming of the FPGA, control of all system functions, support for modification of all run-time configurable algorithm parameters, and support for display and analysis of results.

In order to maintain the modularity and reusability, the software package has been designed in a layered manner. At the most basic level is the PCI/USB communication drivers, controlled by the PC operating system. On top of that is the communication API developed by Nallatech. This provides support for basic communications of two types between the host PC and the hardware components. It provides memory map register access and 16 DMA channel access to the internal memory blocks.

To enable a simple interface to the hardware, a new MALTAB toolbox has been developed. This toolbox acts as a wrapper around Nallatech developed API. This provides a much more convenient and flexible design for the most common communications between the host PC and the hardware. Furthermore, the toolbox can be easily updated for accelerating the development of new applications.

V. PERFORMANCE AND RESOURCE UTILIZATION

In this section, performance results based on the FPGA implementation are presented. As shown in Table I, an average
Flaw-to-Clutter ratio (dB)

<table>
<thead>
<tr>
<th>Data</th>
<th>Starting Value</th>
<th>After SSP Processing</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.6905</td>
<td>6.7109</td>
<td>3.0204</td>
</tr>
<tr>
<td>2</td>
<td>-2.3301</td>
<td>9.9768</td>
<td>12.3069</td>
</tr>
<tr>
<td>3</td>
<td>2.2395</td>
<td>10.9734</td>
<td>8.7309</td>
</tr>
<tr>
<td>4</td>
<td>-3.7417</td>
<td>6.3667</td>
<td>10.1084</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>7.5143</td>
<td>7.7455</td>
</tr>
<tr>
<td>6</td>
<td>1.5368</td>
<td>8.9643</td>
<td>7.4275</td>
</tr>
<tr>
<td>7</td>
<td>2.694</td>
<td>11.1871</td>
<td>8.4931</td>
</tr>
<tr>
<td>8</td>
<td>4.1425</td>
<td>9.6398</td>
<td>5.4973</td>
</tr>
<tr>
<td>9</td>
<td>4.2522</td>
<td>5.1244</td>
<td>0.8722</td>
</tr>
<tr>
<td>10</td>
<td>-1.743</td>
<td>7.3577</td>
<td>10.4787</td>
</tr>
</tbody>
</table>

Starting 4692 cycles per frame.

Also of importance is the total resource usage consumed by the design which is critical for both cost and performance reasons. In this case, the flexibility of the design is demonstrated by implementing two separate designs, one utilizing a 4-channel algorithm and an additional one with an 8-channel implementation. Using an 8-channel implementation improves the FCR by 2dB on the average while increasing the resource usage as shown in Table III. The architecture takes advantage of many of the advanced features of the Virtex-4 device. It makes significant use of the built-in RAM blocks and the DSP48 processing elements [6]. When comparing resource usage it’s important to consider the trade-off between resource usage and the performance increase. Although software implementation uses significantly less resources, the throughput is not sufficient for real-time ultrasonic detection applications.

VI. Conclusion

In this paper we have presented a configurable hardware implementation of the SSP algorithm and its application to real-time ultrasonic signal processing. The SSP algorithm parameters have a large impact on its overall performance. With this in mind, a modular architecture has been designed and presented. The design keeps flexibility in configuration as its primary goal. The hardware implementation can improve the execution time of the software implementation by almost 1000 times. Furthermore, the modular architecture presented in this paper is a valuable research tool and it can be used to fully explore the design space of the SSP algorithm through parameter changes. It provides a hardware framework for experimenting with new subband decomposition algorithms.

REFERENCES