Performance Analysis of System-on-Chip Architectures for Ultrasonic Data Compression

Boyang Wang*, Pramod Govindan** and Jafar Saniie*

*Embedded Computing and Signal Processing Laboratory
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, Illinois

**School of Engineering
University of North Florida
Jacksonville, Florida

Abstract—Ultrasonic NDE and imaging applications utilize a large amount of information. Most of these applications demand real-time data processing with low power consumption. Compression of the collected ultrasonic data helps to reduce the storage, as well as rapid data transmission to remote locations for expert analysis. The objective of this study is to develop embedded system-on-chip architectures for ultrasonic data compression, and analyze the performance of different design methods according to the application requirements. The system is implemented using Xilinx Zynq System-on-Chip (SoC), which combines both ARM processor and field-programmable gate array (FPGA) on the same chip. The major parameters analyzed in this study are signal fidelity, hardware resource utilization and computational processing speed. The hardware and software co-design implementation is about five times faster compared to software only implementation using Zynq SoC.

Keywords - Ultrasonics; Data Compression; FPGA; Zynq SoC

I. INTRODUCTION

The ultrasonic testing platform demands high sampling rates and computational speed in order to realize real-time processing. This paper explores the implementation of ultrasonic signal acquisition and processing on Xilinx Zynq System-on-Chip (SoC) platform [1]. In our previous work [2], the system architecture of a reconfigurable ultrasonic testing system is presented, where the hardware setup of this system is introduced. Figure 1 shows the block diagram of the ultrasonic testing system. The system is capable of generating and capturing high frequency ultrasonic echoes. With a high speed ADC interfaced to the system through FPGA, the system will sample the high frequency ultrasonic echoes at 250MSPS, where each sample consists of 16 bits. The sampled data will be buffered and processed on the Zynq SoC. In this study, a low cost Zynq based development board called Zedboard is used as the main controller and processor of the system.

Section II discusses the efficiency of various ultrasonic data compression algorithms. In this study, ultrasonic data compression is implemented by using three processing methods: discrete wavelet transform (DWT), Frequency domain windowing, and signal decimation in time domain.

II. INTRODUCTION TO COMPRESSION ALGORITHMS

This section details three different compression algorithms which have been explored in this study.
A. DWT based Compression

Certain signal transforms are suitable for data compression because of their data decorrelation and energy compaction properties. DWT converts the signal from the time domain to the wavelet domain. The signal is represented by a series of wavelet signals with different coefficients. By expressing the signal in the wavelet domain, the signal will be represented in a more compact format. The DWT algorithm used in this study is shown in Figure 2, which follows the Mallet algorithm [3]. The original signal is decomposed into low-pass (LP) and high-pass (HP) frequency components after each level of DWT filter pair. Each DWT filter pair contains one LP FIR filter and one HP FIR filter. The coefficients of the FIR filters are decided by the mother wavelet. After a careful study and comparison of different wavelets [4], Daubechies 10 wavelet is selected as the mother wavelet suitable for ultrasonic application requirements. According to the result of the experiments, we found that the majority of the energy is concentrated in LLL and LLHH frequency bands. Therefore, other frequency components are discarded to compress the signal.

![Diagram of DWT Decomposition Filter Set](image)

Figure 2. DWT Decomposition Filter Set

B. Spectral based Compression

Another compression method examined in this study is based on examining the frequency spectrum of the original signal using FFT [5]. After the Fourier transform operation, the original signal is represented in the frequency domain. Most of the energy of the Fourier transform of the ultrasonic signal is concentrated in certain frequency bands. By omitting those frequency bands of the transformed signal with very low energy content, the signal is expressed in a more compact format. The decompression methodology is to pad the compressed spectrum with zeros and perform Inverse FFT (IFFT).

As shown in Figure 3, the energy of the transformed signal is concentrated in the low frequency band marked by a red rectangle. During the compression, only the content in the marked region which contains 256 samples will be preserved as the compressed spectral result. When performing the reconstruction, the compressed signal is padded with zeros and then IFFT is applied to recover the original time domain signal.

![Diagram of Power Spectrum of Original Ultrasonic Signal](image)

Figure 3. Power Spectrum of Original Ultrasonic Signal

C. Time Domain Decimation based Compression

A practical method of compression is to perform time-domain decimation of the original signal. The reconstruction of the original signal can be achieved by using the time-shift property of Fourier transform [6]. The Fourier transform has the property that a delay in time domain maps to a phase shift in the frequency domain. As it is shown in the following equation, X[k] is the Fourier transform of the discrete signal x[n], if x[n] is delayed by D samples. Then, each value in the FFT of the signal is modulated by $e^{-j2\pi nkD/N}$.

$$x[n] \leftrightarrow X[k]$$

$$x[n-D] \leftrightarrow e^{-j2\pi nkD/N}X[k]$$

The spectral of the decimated signal is modulated in the frequency domain multiple times to recover the decimated samples of the original signal. Figure 4 demonstrates the procedure of compression by decimation and reconstruction by time-shift interpolation. Figure 4 (a) shows the first 40 samples of the original signal. After the decimation, only 10 samples are preserved as a compressed signal. Figure 4 (b) shows the compressed signal that is decimated by 4. By implementing the time-shift property, the decimated signal will be time shifted to recover the samples that are omitted in the process of decimation. Figure 4 (c) shows the reconstructed original signal.

![Diagram of Signal Compression by Decimation and Reconstruction](image)

Figure 4. Signal Compression by Decimation and Reconstruction by Utilizing Time-shift Property of Fourier Transform
III. SIGNAL PROCESSING ON ZYNQ SoC

This section provides the implementation details of the signal processing algorithm on Zynq SoC. The ARM processor and FPGA available on Zynq SoC allows hardware and software co-design. Hardware acceleration modules available on the FPGA can be used to boost the processing speed of the signal processing algorithms.

The embedded ARM processor enables many complex operations. By using C or C++ libraries developed for the embedded ARM processor, users can easily program the Zynq SoC to perform different operations such as UART communication (aka asynchronous serial communication), Ethernet communication, and signal processing algorithms. The communication between the ARM processor and FPGA is achieved via the AXI4 bus interface. A DMA logic block is available in the system to transfer the data between FPGA and RAM, which will offload the CPU.

Figure 5 presents the block diagram of the FFT/IFFT acceleration block implemented on the Zynq SoC. The PS in the figure indicates the Processing System which mainly consists of the ARM processors; PL indicates the Programmable Logic which is the FPGA. AXI-Lite is a low speed communication interface which is used to configure the DMA, and AXI-General Purpose I/O (GPIO) blocks. AXI-High Performance (AXI-HP) interface is used to transfer the data between signal processing acceleration block within the FPGA and the RAM.

As shown in Figure 5, the acquired ultrasonic signal is stored in the RAM (DDR Memory). The AXI-GPIO block is used to convert AXI-Lite interface signals to GPIO control signals. The ARM processor will configure the FFT/IFFT acceleration block through the AXI-GPIO block. Then the DMA is triggered by the ARM processor to start data transfer from the RAM to the FFT/IFFT acceleration block to complete the ultrasonic signal processing algorithm. After the processing is performed, the ARM processor will again command the DMA to transfer the processed data back into the RAM.

IV. EXPERIMENTAL RESULTS

In this section, the performance of software-only implementation and hardware-software co-design implementation are analyzed and compared. The fidelity of different algorithms is evaluated by calculating the correlation between the reconstructed signal and the original signal [6]. Moreover, the compression ratio is considered as an important performance factor of the compression algorithm. Table 1 shows the fidelity and compression ratio of three algorithms mentioned in Section II.

<table>
<thead>
<tr>
<th>Compression Method</th>
<th>Compression Ratio</th>
<th>Fidelity (Correlation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWT based</td>
<td>81.25%</td>
<td>98%</td>
</tr>
<tr>
<td>Spectral based</td>
<td>87.5%</td>
<td>99%</td>
</tr>
<tr>
<td>Decimation based</td>
<td>75%</td>
<td>97%</td>
</tr>
</tbody>
</table>

In this study, the core signal processing functions such as FFT and FIR are implemented by using FPGA acceleration. Different compression algorithms are generally a linear combination of these two operations. Additionally, the C implementation of these algorithms are executed on the Zynq SoC using the embedded ARM processor. The ARM is running at 33.33 MHz, and the FPGA is running at the frequency of 100 MHz. Table 2 shows the execution time of FFT and FIR algorithms. These execution times include the data transfer between the RAM and Zynq SoC before and after the execution of the algorithms.

<table>
<thead>
<tr>
<th>Execution Method</th>
<th>Time(us)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT in C</td>
<td>254.66</td>
</tr>
<tr>
<td>FFT in FPGA</td>
<td>47.31</td>
</tr>
<tr>
<td>FIR in C</td>
<td>1029.33</td>
</tr>
<tr>
<td>FIR in FPGA</td>
<td>246.98</td>
</tr>
</tbody>
</table>

By knowing this information, a reasonable estimation of different compression algorithms can be made. The DWT based compression/reconstruction algorithm is a combination of multiple FIR filters. Our preliminary results show that the FPGA implementation (see Table 2) can be 4.16 times faster than the C implementation for both DWT compression and reconstruction. Note that the spectral based compression/reconstruction algorithm and the time-domain based decimation/reconstruction are calculated using multiple FFT operations. As shown in Table 2, the FPGA implementation of these two algorithms is 5.38 times faster than the C implementation using the embedded ARM processor.
V. CONCLUSION

In this research, three different compression algorithms are discussed. The performance of these algorithms is compared by listing the fidelity of the reconstructed signal and the compression ratios. Furthermore, a hardware-software co-design structure is used to accelerate such algorithms on the Zynq SoC. Software implementation is also run on the ARM processor embedded on the Zynq SoC and the performance is compared against the hardware-software co-design performance using FPGA fabric. Results show that hardware-software co-design is faster than the software design. Furthermore, by utilizing the parallel processing capabilities of the FPGA, the computational speed can be improved further. This paper presents a high speed ultrasonic signal processing system on Xilinx Zynq SoC, suitable for real-time ultrasonic imaging applications.

REFERENCES