10.1109/ULTSYM.2012.0115

# Reconfigurable Ultrasonic System-on-Chip Hardware (RUSH) Platform for Real-Time Ultrasonic Imaging Applications

Pramod Govindan, Spenser Gilliland, Thomas Gonnot and Jafar Saniie
Department of Electrical and Computer Engineering
Illinois Institute of Technology,
Chicago, Illinois 60616

***Abstract* - Ultrasonic systems are widely used in industrial and medical imaging applications for diagnosis, nondestructive evaluation (NDE), defect recognition and classification. These applications require large amounts of data to be processed in real-time using computationally-intensive signal processing algorithms. In this study, we developed a Reconfigurable Ultrasonic System-on-Chip Hardware (RUSH) platform. This platform provides real-time signal processing for NDE and imaging applications using ultrasonic transducers ranging from 20 KHz to 20 MHz operational frequencies.**

## I. Introduction

System-on-Chip (SoC) solutions using FPGA are the best choice for the reconfigurable computational platform. In this study, a reconfigurable SoC is designed and developed on Xilinx Virtex-5 (XC5VLX110T) FPGA for ultrasonic NDE and imaging applications. This platform includes an embedded Microblaze processor to enable hardware-software (HW/SW) co-design. Computationally-intensive signal processing algorithms are implemented in the FPGA hardware which is reconfigurable for various algorithms. This platform has a 12-bit 250 MSPS ADC (Maxim MAX1215N 2012) with a sampling rate up to 250 MHz to capture ultrasonic signals and a DAC to deliver processed signals for real-time display. The platform also has PC interface for remote access and ultrasonic information display/storage. NDE of materials requires the design of transducers to be tightly coupled to data collection at high sampling rates and real-time computation. For high resolution imaging, ultrasonic measurements involve transducers operating in MHz frequency range. This results in a significant computational load for embedded systems. In this platform, access to the ultrasound data and custom IP cores is available through a gigabit Ethernet connection. The ability of the processor to execute platform-independent C code allows the reuse of a substantial portion of available software, thus reducing design and development time. This system also promotes reuse of hardware resources, as IP Cores can be easily interfaced to the embedded processor. Besides, the bus architecture has been integrated into an off-the-shelf operating system so that device drivers can be easily reused. The setup for NDE and imaging applications using the Reconfigurable Ultrasonic System-on-Chip Hardware (RUSH) platform is shown in Fig.1 and Fig.2.

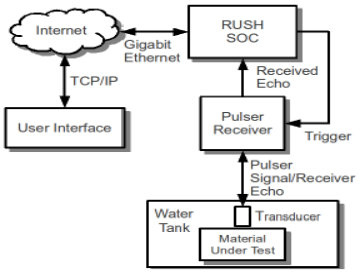


Fig. 1. Ultrasound Testing Setup using the RUSH Platform

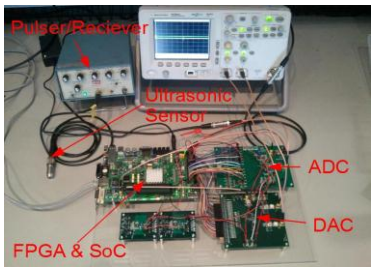


Fig. 2. RUSH system including Virtex-5 FPGA, ADC, DAC, pulser/receiver and ultrasonic transducers

## II. RUSH System Features

The RUSH system provides a common platform for both software and hardware developers using existing standards. The system takes advantage of the IP Cores in the Xilinx Embedded Development Kit (EDK). The EDK system implements a common bus based on the IBM Processor Local Bus (PLB) architecture. This allows reusable IP Cores to directly integrate with the PLB in order to provide processor, memory controller and I/O capabilities. The ability to run Linux allows the reuse of a substantial portion of available software and drivers. This reduces development time and provides a common industry standard POSIX API to software developers. As the bulk of the system is implemented in FPGA, the system is able to guarantee strict real-time demands in hardware; however, software interrupt and scheduling latencies should be contained to avoid buffer overruns when copying data from the ADC to the memory or network.

The RUSH was built by integrating several components including a Xilinx XUPV5 FPGA development platform (Xilinx 2010), a Maxim MAX1215N ADC Evaluation Kit (Maxim1215N 2012), a Maxim MAX5874 DAC Evaluation Kit (Maxim5874 2012) and two Maxim MAX1536 Power Supply Evaluation Kits. The system can be miniaturized by

creating a custom printed circuit board which includes chips from each of the components discussed above. The interconnections of these components are shown in Fig. 3.

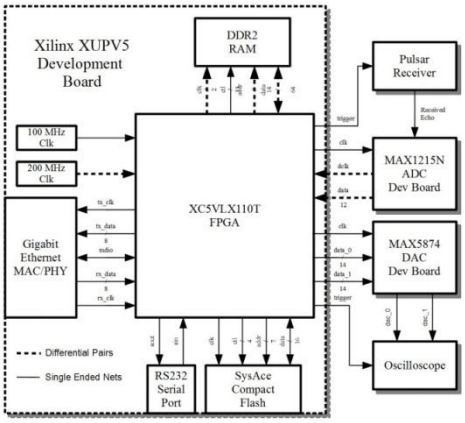


Fig. 3. RUSH block diagram showing hardware components and interconnections.

The FPGA System-on-Chip (SoC) is based around the Processor Local Bus. The SoC includes the following cores from Xilinx: A multi-port memory controller (MPMC), universal asynchronous receiver transmitter (UART), Gigabit Ethernet, System Advanced Configuration Environment (Sys ACE) compact flash controller, timer, clock generators and a Microblaze processor. The Microblaze processor is a 32 bit RISC CPU which can be customized to include an FPU (Floating-Point Unit) and streaming interface to either fast simplex link or AXI (Advanced eXtensible Interface) stream interfaces. This allows the core to have additional pseudo instructions added to the instruction set. In this design, 12 KB data and 4KB instruction caches were chosen in order to strike a balance between resource usage and performance. The cache is backed by the MPMC which interfaces to the DDR2 (Double Data Rate) memory.

In the RUSH design, the FPGA is used for signal processing. DSP48 blocks inside the FPGA provide constant multiply and accumulate (MAC), two-operand multiply and division capabilities. The system is capable of storing up to 256 MB of data in the DDR2 memory for future analysis. The DDR2-400 used in RUSH has a maximum theoretical bandwidth of 3200 MB/s and interfaces to the RUSH through the MPMC (Multi-Port Memory Controller) provided by Xilinx (Xilinx 2011). The gigabit Ethernet controller is connected to the processor using the PLB interface and an SDMA (Soft Direct Memory Access) PIM (Personality Interface Modules) of the MPMC. The SDMA PIM provides high speed scatter-gather DMA (Direct Memory Access) transfers to the Ethernet MAC so that the processor is free to process even when data is being copied into or out of main memory. The RUSH system communicates with the user using TCP/IP on top of gigabit Ethernet. The RUSH system is theoretically capable of providing wired gigabit Ethernet communication at 125 MB/s. Given a packet of 1500 bytes, the total overhead for IP and Ethernet is approximately 4%. Given another 1% of overhead for TCP, leaves a total of 118.75 MB/s for data transmission.

The ADC (MAX1215N) operates for a broad frequency spectrum of ultrasound from 20 KHz to 20 MHz. For a 250 MHz clock, the total throughput of the ADC is 375 MB/s. The ADC uses LVDS (low voltage differential signaling) when interfacing to the FPGA. The maximum operating frequency is governed by the delay, which depends on the length of wire and delay of the ADC. Because the edge rate of the ADC is significant with respect to the length of wire, the line must be treated as a transmission line. The transmission line has a delay of 170 ps/inch according to the measurement obtained in this study which compares an 18 inch vs. 12 inch coaxial wire. For this arrangement, the total skew is approximately 6.23 ns and limits the connection to a maximum sampling rate of 160.50 MHz. Because the connections between the external inputs and the FPGA are located across multiple FPGA IO banks, it was necessary to use a Digital Locked Loop (DLL) for the incoming signal as shown in Fig. 4, instead of using the BUFR (regional clock buffer) as recommended by Xilinx. In this scenario, the skew present at the first DFF (D flip-flop) is eliminated by using the DLL and the interface operates as expected.

## III. Evaluation of RUSH and Xilinx XUPV5 FPGA

Various ultrasonic signal processing applications have been examined on the proposed RUSH platform. Real-time ultrasonic flaw detection with a high repetition rate using split-spectrum processing was implemented as a HW/SW co-design having parallel & pipelined multiplications and additions for high throughput. This platform was also used for the real-time embedded implementation of the chirplet signal decomposition algorithm. Currently, we are evaluating the performance and implementation of 3D ultrasonic data compression using discrete wavelet transform incorporating different kernels. Finally, to improve the signal-to-noise ratio, real-time coherent averaging was implemented at a sampling rate of 200 MHz.

### *A. Split Spectrum Processing*

Split Spectrum processing (SSP) is an algorithm used for discerning frequency diverse echoes. The algorithm splits a broadband signal into multiple narrowband signals [2], [6], [7], [8], [9]. These signals are then post processed using

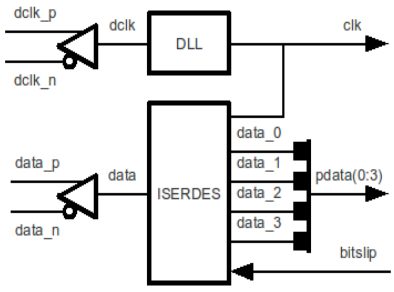


Fig. 4. Interconnection between ADC and FPGA

algorithms such as absolute minimization, median filtering, or more complex algorithms based on fuzzy set logic, ordered statistics and neural networks. The general goal is to remove the Rayleigh scattering in a system thereby increasing the target-to-clutter ratio. Clutter is a combination of the many echoes that are produced in the microstructures of a material (i.e., grains). When an ultrasound wave scatters from grains, it produces a stationary scatter which represents these grain scatterers. When the aggregate of all of the scatterer echoes are combined, they produce a static noise that hides the target echo. Therefore, these scatterings must be filtered from the signal in order to increase the SNR for the classification or detection system and enable accurate detection of the target. When a measurement is taken using a number of different narrowband pulses with different frequencies, the clutter echoes become randomly distributed across the pulse frequencies and can be filtered using statistical processing. In SSP realization the first component is data acquisition unit. The second processing component, Fast Fourier Transform (FFT), gives the frequency spectrum of the received echo signal. Third step, several filters split the spectrum into different frequency bands. Next step, inverse FFT gives the time domain signal of each individual frequency band. The signals from each individual frequency band are first normalized and then passed into a post processing block for detection. This detection processor can employ different techniques such as averaging, minimization, order statistic filters (e.g. minimization), Bayesian classifiers or neural networks [2], [6], [7], [8], [9].

An implementation of the SSP algorithm has been instantiated inside the FPGA fabric. The SSP hardware uses: FFT followed by a basic one-zero windowing algorithm to split spectrum; inverse FFT (iFFT) to obtain frequency-diverse signals; and pass-through absolute minimum processing to improve the visibility of target echo in presence of clutter. As the SSP algorithm uses a significant number of FFTs, it is important that the FFT implementation must be highly area efficient. The most efficient IP Core available for implementing an FFT/iFFT is the Xilinx Radix 2-Lite FFT IP Core (FFT 2012). At a transform size of 4096 points, this IP Core has a latency of 57,539 cycles; however, it uses only two DSP48 components, seven BRAMs, and approximately 250 slices. This low resource usage allows up to 13 FFTs to be implemented in tandem with the other components of the RUSH. The SSP implementation has synthesis parameters which can be altered in EDK for changing the number of channels, and the maximum number of samples. In the current configuration, the SSP algorithm uses twelve channels and 4096 samples which is the maximum number possible given the DSP48 resources available on the FPGA. For the SSP processing, the number of FFTs required is equal to the number of channels plus one. In the current configuration, the implementation has thirteen FFTs instantiated in the FPGA.

### *B. Chirplet Signal Decomposition*

Ultrasonic signals are often composed of many interfering echoes. Each echo is similar to a chirplet, and chirplets can be described with six parameters using chirplet signal decomposition (CSD) [3]. Therefore, CSD representation results in a major data reduction (compression) when compared to the raw data and it can be used for compressed data storage. Furthermore, the estimated chirplets can be used for material characterization and system identification. The chirplet signal can be defined as,

$$f_{\Theta}(t) = A\exp[-\alpha(t-\tau)^2/2 + j2\pi f_c(t-\tau) + j\beta(t-\tau)^2/2 + \phi]$$

where $\Theta = [\alpha, \beta, A, f_c, \tau, \phi]$. The term $\tau$ is the time-of-arrival, $f_c$ is the center frequency, A is the amplitude, $\beta$ is the chirp rate, $\phi$ is the phase, and $\alpha$ is the bandwidth factor of the echo. This representation accurately models the echoes from ultrasound experiments. In the current study, the CSD algorithm [3] is implemented in software using C language due to the high complexity. The algorithm has been optimized by using pre-computation and estimation methods. Additionally, the algorithm has been altered to provide deterministic behavior. The goal is to reduce the execution time as much as possible while maintaining an adequate SNR for echo estimation. The main performance impediment for CSD is the correlation operations which require regenerating a chirplet with different parameters thousands of times during each parameter estimation step in the algorithm. By pre-computing a table of these values and manipulating this table to perform the operation, the computation time was drastically reduced. This was accomplished by extracting expressions which have shared parts from the inner loops. Furthermore, by using lookup tables for cosine, sine and tangent, the total execution time was substantially reduced with only a slight reduction in accuracy. In Fig. 5, the echoes from the CSD algorithm are regenerated and compared to that in the original signal.

### *C. 3D DWT Compression*

Discrete Wavelet Transform (DWT) has been used for ultrasonic compression applications due to its high energy compaction properties. DWT with proper thresholding provides efficient compression without significantly degrading the reconstructed signal fidelity. When proper thresholding is applied, the transform coefficients in

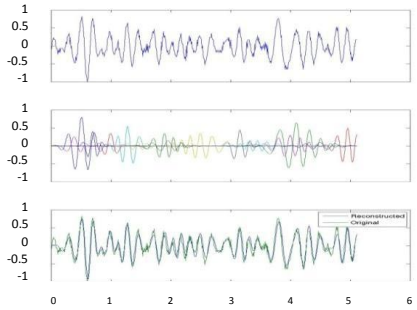


Fig. 5. CSD echo estimation. Top Trace: Original signal, Middle Trace: Estimated echoes, and Bottom Trace: Reconstructed signal superimposed on Original signal

frequency bands which contain a significant portion of signal energy are retained while others are discarded, thereby resulting in lesser number of coefficients. 3D DWT is implemented by applying 1D wavelet transform for each coordinates. A high-order wavelet Daubechies 10 (db10) is used for x-direction to improve data compaction. Db10 wavelets are orthogonal, compactly supported and continuous. It provides better frequency localization. A simple kernel (Haar) is used for both the spatial (y and z) directions to reduce the computation time and hardware requirements. Haar wavelet is very compact, orthogonal and symmetric. It has the highest time resolution.

The wavelet packet decomposition technique is used for x-direction, where both the low pass (L) and high pass (H) components are decomposed according to the energy distribution to achieve better compression. Through extensive evaluation of ultrasonic A-Scans, we have found that H, LH and LLHL coefficients have almost zero energy and, therefore, can be ignored resulting in a total compression of 79% in the x-direction. In spatial directions, Haar wavelet is used since it is simple and highly efficient to implement. Evaluation of Harr wavelet in y and z directions reveals that H & LH does not have large energies and can be discarded, resulting in 75% additional reduction of the compressed results obtained in x-direction. The original A-scan and reconstructed A-scan from the 95% compressed 3D data are shown in Fig. 6, which are extremely similar. For achieving this level of compression, the 3D compression algorithms implemented in Xilinx Virtex-5 FPGA system takes 210 ms to compress ~33 Mbytes of data into ~1.0 Mbytes [4].

### *D. Coherent Averaging*

Coherent averaging is a method for improving the signal-to-noise ratio of a pulsed signal and this improvement is proportional to the number of averaging signals. Coherent averaging heavily relies on precise synchronization between the pulse and capture logic in order to provide accurate results. Thus, this implementation takes advantage of the preprocessing block in the ADC as shown in Fig. 7 to create a highly accurate synchronization between the data acquisition capture clock and the excitation trigger pulse applied to the ultrasonic transducer. Since the preprocessing block is clocked by the *dclk* of the ADC, the timing is guaranteed to be accurate to within the frequency of the clock used to drive the ADC. The coherent averaging block is parallelized by a factor of four to meet timing

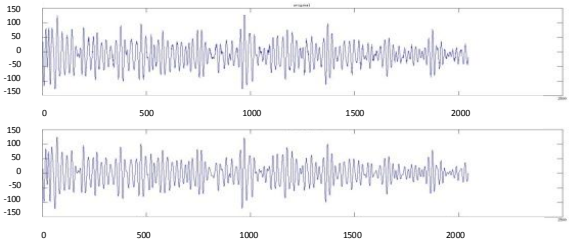


Fig. 6. Original (top trace) and 3D reconstructed (bottom trace) A-scan

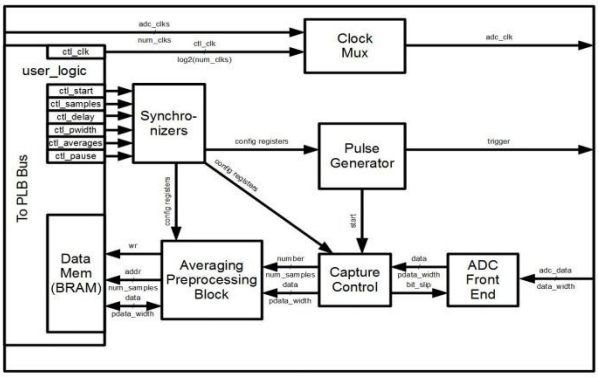


Fig. 7. ADC system implementation including coherent averaging block

requirements for real-time signal averaging at 200 MHz data rate.

## IV. Conclusion

An embedded reconfigurable real-time signal processing system is proposed to enable ultrasonic researchers to experiment with different algorithms using HW/SW co-designs. RUSH is an adaptable FPGA based platform for implementing real-time ultrasonic signal processing applications. The possibility of HW/SW co-design allows flexibility in developing highly efficient systems for different ultrasonic NDE and imaging applications, particularly of importance to ultrasonic handheld devices.

## References


[1] J. Saniie and E. Oruklu, "Introduction to Special Issue on Novel Embedded Systems for Ultrasonic Imaging and Signal Processing", *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 59, no. 7, pp. 1329-1332, July 2012 (invited paper).

[2] J. Saniie, E. Oruklu and S. Yoon, "System-on-Chip Design for Ultrasonic Target Detection Using Split-Spectrum Processing and Neural Networks", *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 59, no. 7, pp. 1354-1369, July 2012.

[3] Y. Lu, E. Oruklu and J. Saniie, "Fast Chirplet Transform With FPGA-Based Implementation", *IEEE Signal Processing Letters*, 15, pp.577- 580, 2008.

[4] C. Desmouliers, E. Oruklu and J. Saniie, "Adaptive 3D Ultrasonic Data Compression using Distributed Processing Engines", *IEEE Ultrasonics Symposium* , pp. 694 – 697, September 2009.

[5] Xilinx, 2012, Platform Specification Format Reference Manual v13.4, http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_4/psf_rm.pdf

[6] J. Saniie and D. T. Nagle, "Analysis of Order-Statistic CFAR Threshold Estimators for Improved Ultrasonic Flaw Detection", *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 39(5), pp.618-630, 1992.

[7] J. Saniie, K. D. Donohue, D. T. Nagle and N. M. Bilgutay, "Frequency Diversity Ultrasonic Flaw Detection using Order Statistic Filters", *Proceedings of IEEE Ultrasonics Symposium*, pp. 879-884 vol.2, 1988.

[8] J. Saniie, D. T. Nagle and K. D. Donohue, "Analysis of Order Statistic Filters Applied to Ultrasonic Flaw Detection using Split-Spectrum Processing", *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 38(2), pp.133-140, 1991.

[9] J. Saniie, T. Wang and X. Jin, "Performance Evaluation of Frequency Diverse Bayesian Ultrasonic Flaw Detection", *The Journal of the Acoustical Society of America*, 91, pp.2034-2041, 1992.