A Real-time Streaming DAQ for Ultrasonix Research Scanner

Mateusz Walczak, Marcin Lewandowski, Norbert Żołek
Department of Ultrasound
Institute of Fundamental Technological Research PAS
Warsaw, Poland
Email: mlew@ippt.pan.pl

Abstract — A new 128 channel parallel acquisition module for the Ultrasonix SonixTouch ultrasound scanner was developed. Although the module provides similar functionality to the original SonixDAQ system, it presents new possibilities of real-time data streaming and processing on GPU, thanks to a fast PCIe communication interface. Direct access and processing of pre-beamformed channel data with fully programmable transmit schemes enables the research implementation of new advanced imaging modalities (eg. plane wave imaging, vector Doppler, shear wave elastography). The presented RX-DAQ system is enriched with a GPU software framework, for Python and Matlab, enabling the integration of user processing functions.

Keywords — ultrasonic imaging; pre-beamformed data; real-time processing; GPU; FPGA

I. INTRODUCTION

The development of new methods and algorithms of signal processing and visualization of ultrasound signals requires access to raw pre-beamformed data. Ultrasound imaging is a real-time modality, thus the implementation and validation of new methods have to be performed in this time regime. Unfortunately, most ultrasound devices used in research laboratories support very limited access to raw data and mostly for offline processing only.

There are a few clinical/research ultrasound systems capable of acquiring channel data – one of them is the Ultrasonix SonixTouch (Analogic, USA) with the SonixDAQ 128 parallel acquisition module [1]. The SonixDAQ is connected directly to the ultrasound probe, in parallel with the ultrasound scanner. Internal routing inside the SonixTouch scanner connects the active probe connector to the aux connector where the SonixDAQ is sitting. The Ultrasonix research package lets you control the ultrasound transmission/acquisition process (Texo API). The SonixDAQ can acquire signals from all 128 probe’s elements and store them on an internal 16GB memory buffer. Later, the data are transferred to a PC via USB 2.0 for further processing [2]. A USB data transfer, limited to ~30MB/s, prevents the implementation of any real-time imaging using raw data.

We developed a 64-channel parallel acquisition module (RX64) as a part of the complete Versatile Research Ultrasound Platform based on a PCIe communication architecture and optimized for software GPU processing. We decided to spin off a product that provides us and other research labs with real-time streaming and processing capabilities for already owned SonixTouch scanners (fig. 1).

The RX-DAQ is the first available ultrasound acquisition system built for a new paradigm of GPU processing. The implemented software framework and example kernels for advanced image reconstruction methods give a fast start into the world of software oriented ultrasound signal processing.

Figure 1. The RX-DAQ module (without housing) connected to Ultrasonix SonixTouch in place of SonixDAQ.
II. HARDWARE

The RX-DAQ was built on two RX64 [3] acquisition cards connected via a custom backplane. Each RX64 card contains an FPGA Altera Stratix IV, two 2-4 GB 64-bit DDR3 memory modules, and two 32-channel ultrasound front-end modules CSM9132 (Cephasonics, USA), which provide analog signal conditioning of ultrasound echoes and A/D conversion with 12-bit resolution at the speed of 65 MSPS. Serialized 64-channels data are transferred to the FPGA using LVDS links and after deserialization are stored in the local DDR3 memory modules. Next, the data are transferred through the PCIe 2.0 8 lane interface to the CPU RAM memory. Real time data streaming from the module is performed by the DMA (Direct Memory Access), resulting in high throughput and low CPU load.

The custom backplane card integrates two RX64 cards and connects their analog channels via a probe connector directly to the Ultrasonix scanner, in place of SonixDAQ (fig. 2). An embedded PCIe switch (PLX8632) aggregates two RX64’s PCIe 8 lane interfaces into one 16 lane interface and connects the RX-DAQ to the PC via a PCIe cable.

The total bandwidth capacity of the interface is about 6 GB/s, which allows for real-time transfer of over 6000 RF frames (128 channel data with 4096 samples) per second.

External clock and trigger input signals allow for synchronization of the RX-DAQ with the SonixTouch master clocks, as well as the transmit/receive timing, which can be easily used in photoacoustic applications.

![Figure 2. A block diagram of the RX-DAQ system.](image)

III. SOFTWARE

The software package consists of (fig. 3):

- A low-level system driver for the RX-DAQ
- A user API for the RX-DAQ (C library)
- A high level software framework for streaming and processing
- A Python wrapper for the Ultrasonix Texo API

A low level C API provides full control over RX-DAQ acquisition and data streaming. Additionally, two higher level software processing frameworks: one for Python with Nvidia CUDA integration and the other for Matlab® with OpenCL integration were developed.

![Figure 3. An architecture of the RX-DAQ’s software.](image)

Both frameworks support integration of user supplied GPU kernels. A Python scripting environment provides free and easy to use (fig. 4) interface for building data flow and processing. Developed or user provided GPU CUDA kernels [4] can be integrated into a flowchart defining consecutive processing steps. The implemented blocks for data acquisition, envelop detection, logarithmic compression and display allow to build a complete ultrasound visualization system. The display functions utilize OpenGL.

The other software package is developed as toolbox for Matlab® and uses OpenCL [5] framework for efficient processing and ultrasound data reconstruction.
Figure 4. A screenshot of Python/CUDA user interface for acquiring and processing ultrasound data from our RX-DAQ module connected to the Ultrasonix SonixTouch.

An implementation of the complete ultrasound method on the system requires:

1. a TX schema to be programmed into the SonixTouch scanner and synchronized with the acquisition pattern of the RX-DAQ,
2. an acquisition pattern programmed into the RX-DAQ,
3. a software setup of the data communication parameters optimized for a given acquisition pattern,
4. a processing flow (set of steps/algorithms) for data processing - image reconstruction.

The SonixTouch is programmed with TX patterns using a Python wrapper for the Ultrasonix’s Texo API. The clock and trigger signals from the scanner are used to synchronize acquisition process in the RX-DAQ. For evaluation purposes several different ultrasound methods were implemented:

- standard STA and MSTA [8],
- cumulative STA with virtual receiving subaperture [9],
- plane-wave imaging [10],
- short-lag spatial coherence (SLSC) [11],
- motion analysis: speckle tracking [12], and Doppler flow analysis [13].

All processing of the RF data (data filtering, image reconstruction) was optimized for GPUs [4,5,6] and can provide real-time visualization on a modern PC platform. A processing speed at the level of 50 frames per second for Python/CUDA based software and about 10 frames per second using Matlab®/OpenCL software, when run on a PC equipped with NVIDIA K5000 Quadro GPU, was attained. Slower processing using Matlab® is due its lack of optimized functions for GPU data exchange and the display of GPU data buffers.

Depending on the algorithm complexity, the amount of raw data per image and target image resolution one may need to hand optimize GPU kernels and/or use multi-GPU solutions to achieve satisfactory frame rates.

Time consuming and complex mathematical functions (eg. sin, cos) of the data processing pipeline, used for apodization in STA or PWI algorithms, were approximated using polynomials [14]. Only native relational functions were used in case of branching the code and most of the conditional constructions. The local memory of multiprocessors in GPU was extensively used to minimize data reading and writing delays especially for the manipulating and filtering of the received data.

IV. CONCLUSIONS

We developed a new 128-channel RF parallel acquisition system, compatible with the Ultrasonix SonixTouch Research scanner, enabling real-time streaming and processing of pre-beamformed data. The reconstruction algorithms were run on the GPU (CUDA/OpenCL), showing the full potential of the system. The RX-DAQ greatly extends the research capabilities of the Ultrasonix Research scanner, especially for emerging applications, such as: synthetic aperture imaging, vector Doppler or elastography.

ACKNOWLEDGMENT

Project POIG.01.03.01-14-012/08-00 co-financed by the European Regional Development Fund under the Innovative Economy Operational Programme.

REFERENCES


