International Journal on Magnetic Particle Imaging
Vol 8, No 1, Suppl 1, Article ID 2203063, 4 Pages

Infinite Science Publishing

Proceedings Article

A Flexible High-Performance Signal Generation and Digitization Plattform based on Low-Cost Hardware

Niklas Hackelberg[1,2,*] · Jonas Schumacher[3,4] · Matthias Graeser[1,2,3,4] · Tobias Knopp[1,2]

[1]Section for Biomedical Imaging, University Medical Center Hamburg-Eppendorf, Hamburg, Germany
[2]Institute for Biomedical Imaging, Hamburg University of Technology, Hamburg, Germany
[3]Fraunhofer Research Institute for Individualized and Cell-based Medicine, Lübeck, Germany
[4]Institute for Medical Engineering, University of Lübeck, Lübeck, Germany
[*]Corresponding author, email: n.hackelberg@uke.de

Abstract

Modern imaging modalities, such as magnetic particle imaging (MPI), are based on complex sequences which require synchronous multi-channel signal generation and reception. The component of an MPI scanner responsible for this functionality is the data acquisition (DAQ) system. Different scanner topologies and the nature of (digital) signal processing impose varying requirements on such a system. In this work, we introduce the RedPitayaDAQServer project, which implements a flexible and scalable DAQ system. It is based on the low-cost hardware RedPitaya STEMlab 125-14 and is able to meet the requirements of most MPI scanner concepts that have been proposed to date.

I. Introduction

In magnetic particle imaging (MPI) the spatial distribution of superparamagnetic nanoparticles is determined based on their non-linear response to external magnetic fields [1]. A related method is magnetic particle spectroscopy (MPS), which can be seen as MPI without spatial encoding. One important component of an MPS/MPI device is a data acquisition (DAQ) system, which is capable of generating and measuring analog signals synchronously. Depending on the considered scanner topology, the number of synchronized in- and output signals and their waveform can differ. During an MPI measurement various magnetic fields are applied for a certain time interval and the particle response is measured simultaneously. Therefore, an MPI DAQ system needs to be able to acquire signals for the entire period length of one field cycle, which can vary between 21.5 ms and 0.5 s [1, 2] or even longer.

Systems for (digital) signal processing often use specialised hardware, which can result in high-performance, but with expensive and inflexible systems. Field-programmable gate arrays (FPGAs) are an option that can leverage the benefits of specialised hardware while remaining flexible. They are used in the crowd-funded RedPitaya project that aimed to provide DAQ instrumentation at a low-price. The project is a combination of proprietary RedPitaya STEMlab circuit boards and an open-source software ecosystem. Together they can be used as a variety of different measurement or control instruments. Nowadays, RedPitaya offers various circuit boards, but we focus on their first board, the STEMlab 125-14. We refer to this as the RedPitaya board throughout the remaining text. This board combines an FPGA with two 125 MHz analogue-to-digital (ADC) and digital-to-analogue (DAC) converters, as well as a general purpose processor and an ethernet connection.

While the RedPitaya hardware is suitable to meet the requirements of an MPI DAQ system, the stock software is not. In particular, the ring-buffer storing the samples during acquisition only holds 16.000 samples. That is

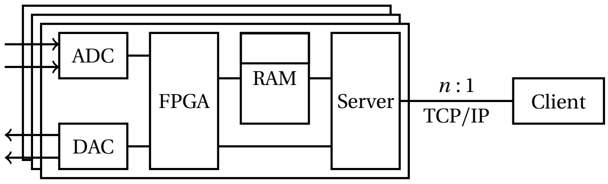

Figure 1: Hard- and software components of a RedPitayaDAQServer DAQ setup.

insufficient to accommodate the acquisition times of an MPI measurement. The aforementioned measurement interval of 0.5 s sampled at a rate of 1.953 MHz would result in the buffer being overwritten 61 times.

In this work, we introduce the open-source RedPitayaDAQServer project, which aims to remove the limitations of the stock RedPitaya software. It provides software components implementing a scalable and flexible data acquisition system. Furthermore, we show how the hard- and software can be used to build different MPI scanners.

II. Methods and Materials

The RedPitayaDAQServer project is realized as a distributed system in which one client connects to a cluster of RedPitaya boards, each running a server and a custom FPGA image. All components of the project can be seen in Figure 1. Communication between client and server happens via SCPI commands over a TCP/IP connection and the server is mostly used as an intermediary to the FPGA. The FPGA image is able to generate and acquire synchronized signals across multiple RedPitayas. A client controlling the cluster can be implemented using the provided reference Julia client library.

Signal acquisition within a cluster is based on a shared clock and trigger signal distributed via cables between the RedPitayas. Once triggered, all FPGAs continuously write the samples from their ADC channel to the sample ring-buffer with each clock tick. The sampling rate can be adjusted by setting a decimation parameter. To allow for longer acquisition times, the ring-buffer is moved from the FPGA to a reserved region of 128 MB size within the main memory of the RedPitaya board. The highest supported sampling rate is 15.625 MHz. At this rate a sample exists within the buffer for around two seconds before being overwritten. To retrieve samples from a RedPitaya, the client instructs the server to transmit portions of the reserved memory region. This portion is based on the number of samples requested and the sample to start with, counted from the beginning of the acquisition. The server tracks potential data loss and buffer state for each transmission. This can be used to identify and improve bottlenecks of a client implementation and to retry measurements.

For signal generation, each FPGA image can produce the following signals on the two DAC channel:

$$S_i(t) = a_{i,1}\, w(t, f_{i,1}, \varphi_{i,1}) + o_i + \tilde{o}_i(t) + \sum_{j=2}^{4} a_{i,j} \sin(2\pi f_{i,j}\, t + \varphi_{i,j}).$$

Each channel i in a cluster can generate a composite waveform consisting of four components, a fixed and a time-varying offset o_i and $\tilde{o}_i(t)$. Of the four components only the first component has a variable waveform w, which can be a sine, square, triangle or sawtooth wave. For each component, the amplitude $a_{i,j}$, frequency $f_{i,j}$ and phase shift $\varphi_{i,j}$ can be set by the client. The time-varying offsets are read from a ring-buffer managed by the server, which are transmitted by the client before a measurement is started. The access to this buffer and thus the time-varying offset can be consistently updated at a frequency of about 12 kHz. Using this approach each channel can generate a selection field, an offset field, a (multi-tone) drive-field excitation or a combination of these.

Further documentation with implementation details, small sample clients, the client library interface and development notes for the FPGA image, the server and own client applications can be found in the project's repository[1].

III. Results

Together with the other open-source projects MPIMeasurements.jl[2], MPIReco.jl [3] and MPIFiles.jl [4], the RedPitayaDAQServer forms the software stack of various MPS/MPI scanners [2, 5, 6]. Figure 2 shows how the RedPitaya boards were used in their respective setups.

The 1D MPS (a) requires a single RedPitaya producing a sinusoidal signal for particle excitation and receiving the particle signal as well as a reference signal that is used to control the applied field.

The pulsed MPS (b) [6] allows for different excitation waveforms and requires a second output channel for generating an offset field. In the example shown in the figure, the board generates a rectangular signal on the first channel overlapped with a step-wise time-varying offset. The second signal is a step function that changes values with each new period of the first signal.

The scanner for human brain-applications (c) [2] requires more than two in- and output signals and therefore uses a RedPitaya cluster of size two. The first RedPitaya is responsible for generating the dynamic two-channel selection field (SF) whereas the second RedPitaya generates the 1D drive field (DF). Both RedPitayas are used for particle signal and control signal acquisition.

[1] https://github.com/tknopp/RedPitayaDAQServer
[2] https://github.com/MagneticParticleImaging/MPIMeasurements.jl

10.18416/ijmpi.2022.2203063

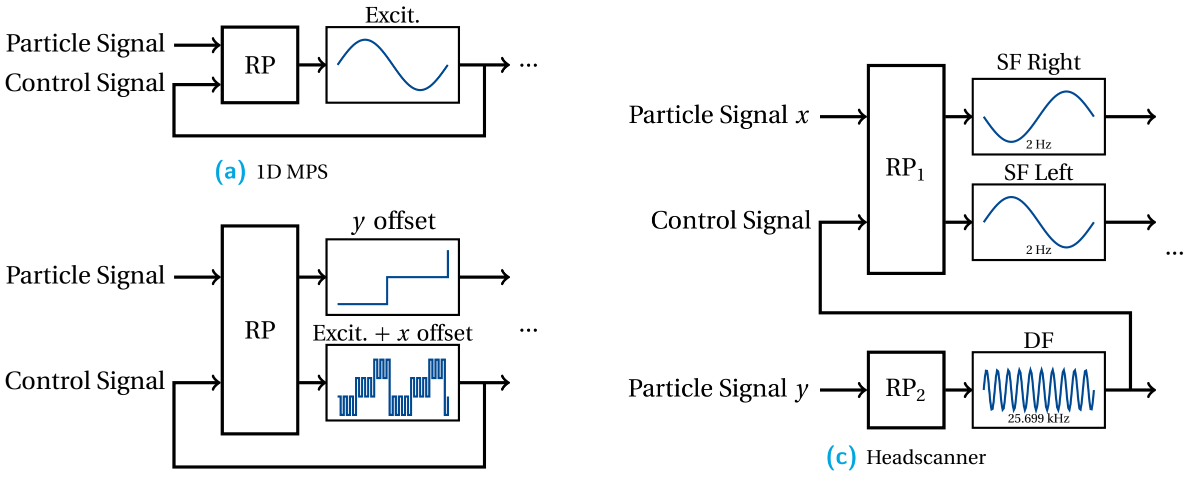

Figure 2: Block diagrams showing how the RedPitayaDAQServer is used in different MPI scanners and the type of signals they produce and receive. The RedPitayas (RPs) were setup to generate signals for drive and selection fields (DF, SF) as well as particle excitation and offset fields.

The performance of client-side sample retrieval depends on the queries of the client, its processing of the samples and the network bandwidth. At the highest sampling rate a single RedPitaya produces data at a rate of 500 Mbit/s and can transmit at rates slightly above that. With the performance tracking we could write and optimize a client that managed to acquire signals continuously for one hour without any data loss at sampling rates of 15.625 MHz for a single and 7.1825 MHz for cluster of two and three RedPitayas (four and six channels) using 1 Gbit/s ethernet. An example of performance tracking can be seen in Figure 3.

IV. Conclusion and Discussion

In this work we have shown how the RedPitaya hardware together with the RedPitayaDAQServer software can be used to realize a flexible and low-cost DAQ system that is able to meet the requirements of MPI systems. The DAQ system can be scaled to different numbers of in- and output signals by introducing new STEMlab boards to a cluster. The scalability is limited in regards to client-side (continuous) signal acquisition by the available network bandwidth and how a client application requests, receives and processes samples.

With the implementation of a client in the open-source Julia software stack for different MPI scanners, we have shown that the system can achieve high-performing data acquisitions without data loss during measurements. The data loss and buffer state tracking provided by the server proved to be an important tool in optimizing clients and testing performance of a system.

Because of the reprogrammable digital hardware and the language agnostic communication between client and server, the DAQ system can be adapted for different hard- and software setups and requirements.

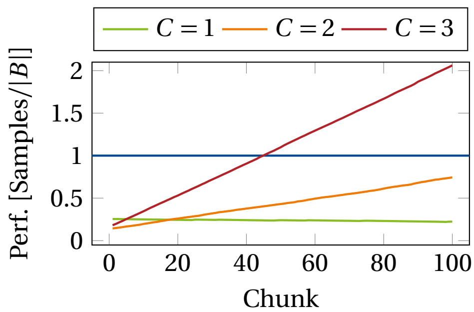

Figure 3: Averaged performance data of a cluster with size C based on the distance between the requested and the latest sample normalized by buffer size $|B|$. For each cluster three full buffers were read continuously at the highest sampling rate (15.625 MHz). The samples are retrieved in 100 equally sized chunks. A value larger than one means data was lost.

Author's statement

We would like to acknowledge the RedPitaya projects of Koheron[3] and Pavel Demin[4], which the RedPitayaDAQServer is partially based on. Research funding: The authors thankfully acknowledge the financial support by the German Research Foundation (DFG, grant number KN 1108/7-1 and GR 5287/2-1). Conflict of interest: Authors state no conflict of interest.

[3] https://www.koheron.com/blog/2016/11/29/red-pitaya-cluster
[4] https://github.com/pavel-demin/red-pitaya-notes

References

[1] B. Gleich and J. Weizenecker. Tomographic imaging using the non-linear response of magnetic particles. *Nature*, 435(7046):1214–1217, 2005, doi:10.1038/nature03808.

[2] M. Gräser, F. Thieben, P. Szwargulski, F. Werner, N. Gdaniec, M. Boberg, F. Griese, M. Möddel, P. Ludewig, D. van de Ven, *et al.* Human-sized magnetic particle imaging for brain applications. *Nature communications*, 10(1):1–9, 2019, doi:10.1038/s41467-019-09704-x.

[3] T. Knopp, P. Szwargulski, F. Griese, M. Grosser, M. Boberg, and M. Möddel. MPIReco.jl: Julia package for image reconstruction in MPI. *International Journal on Magnetic Particle Imaging*, 5(1-2), 2019, doi:10.18416/IJMPI.2019.1907001.

[4] T. Knopp, M. Möddel, F. Griese, F. Werner, P. Szwargulski, N. Gdaniec, and M. Boberg. MPIFiles.jl: A julia package for magnetic particle imaging files. *Journal of Open Source Software*, 4(38):1331, 2019, doi:10.21105/joss.01331.

[5] F. Griese, T. Knopp, C. Gruettner, F. Thieben, K. Müller, S. Loges, P. Ludewig, and N. Gdaniec. Simultaneous magnetic particle imaging and navigation of large superparamagnetic nanoparticles in bifurcation flow experiments. *Journal of Magnetism and Magnetic Materials*, 498:166206, 2020, doi:10.1016/j.jmmm.2019.166206.

[6] F. Mohn, T. Knopp, M. Boberg, F. Thieben, P. Szwargulski, and M. Graeser, System matrix based reconstruction for pulsed sequences in magnetic particle imaging, 2021. arXiv: `2108.10073 [physics.med-ph]`.

