REAL-TIME DATA ACQUISITION WITH CompactPCI Serial PLATFORM AT PSI

R. Rybaniec*, O. Bründler, B. Stef, Paul Scherrer Institut, Villigen, Switzerland

Abstract

Data acquisition (DAQ) is an ubiquitous feature in modern particle accelerator measurement and control systems. At the Paul Scherrer Institut (PSI), a next generation of electronic devices is being designed to meet the demands of upcoming renewal of facilities. The new developments utilize the CompactPCI Serial (CPCI-S.0) platform, and will cover a diverse set of applications, including Low Level Radio Frequency (LLRF), Longitudinal Beam Loss Monitoring (LBLM), and Filling Pattern Monitoring (FPM) systems. Careful design considerations and selection of an optimal architecture are crucial to fulfill a variety of DAQ requirements such as maximum frequency of acquisition, size of the data and different modes of triggering. In this contribution, we focus on the real-time DAQ implementations utilizing a multiprocessor system on chip (MPSoC) technology. We review the IP components developed in-house at PSI that provide the DAQ functionality. We demonstrate, that by reusing the IP components development, prototyping and testing of applications requiring the DAQ are accelerated.

INTRODUCTION

Real-time Data Acquisition (DAQ) is a crucial feature in modern particle accelerators responsible for high-speed, low-latency data recording and processing tasks, maintaining precise timing synchronization. When implemented with a multiprocessor system on chip (MPSoC), this involves utilization of a programmable logic (PL), digital signal processing (DSP) units, memory resources, data interfaces and specialized hardware modules such as analog-to-digital converters (ADCs) to efficiently capture, filter, and transmit incoming data streams.

The next generation of the PSI facilities like Swiss Light Source 2.0 (SLS 2.0) use a new development hardware based on the CompactPCI Serial (CPCI-S.0) [1] standard. The hardware upgrades will affect a diverse set of applications, including Low Level Radio Frequency (LLRF) [2], Longitudinal Beam Loss Monitoring (LBLM) [3], and Filling Pattern Monitoring (FPM) [4] systems. These systems have different characteristic which put constrains on their DAQ designs.

Table 1 summarizes a selected set of DAQ requirements for some systems in preparation for SLS 2.0. More complex applications, like LLRF, require many DAQ channels with different acquisition frequencies, lengths of the data buffers and triggering mechanisms. A precision time-stamping of the data is another important feature, which is required by all considered applications.

In parallel to the new hardware designs, a library of reusable PL entities [5] is undergoing continuous expansion. The library portfolio encompasses components designed to facilitate prevalent design concepts such as clock-domain crossing and memory generation, as well as the implementation of DSP operations utilizing fixed-point arithmetic. Build on top of this, there is a collection of the reusable intellectual property cores (IP cores) which implement high level functions, including complete DAQ operations.

In this paper we focus on two IP blocks currently used at PSI which are DataRecorder (DataRec) and Single Stream DAQ (SIS_DAQ). We illustrate their usage within the LLRF application.

DAQ IMPLEMENTATIONS ON MODERN MPSoCs

Modern MPSoCs are highly sophisticated devices integrating many logic components, providing flexibility in choosing an architecture to implement functionalities such as DAQ. The pivotal aspect in this selection process is the choice of memory resources.

Commonly used at PSI, Zynq UltraScale+ MPSoCs, provide distributed memory (LUTRAM, memory is implemented using the programmable logic resources), block RAM resources (BRAM, dedicated components acting as a memory), ultra RAM blocks (similar to BRAM blocks, with some limitations), and memory controllers for interfacing with external RAM chips.

From the designer perspective, LUTRAMs and BRAMs are most straightforward to use. However, those resources have too limited capacity to store all data required by some applications.

DataRec COMPONENT

The DataRec IP core (Fig. 1) implements a straightforward, general-purpose data recorder, leveraging the internal memory resources available in the PL. It interfaces with the processing system via an AXI4 Lite bus.

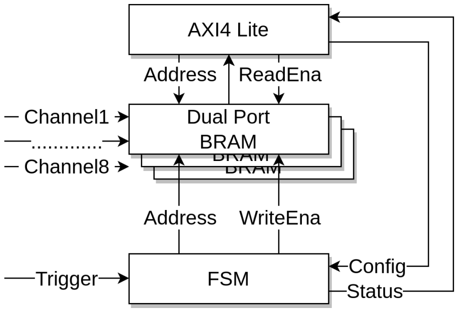

Figure 1: DataRec architecture diagram.

* radoslaw.rybaniec@psi.ch

Table 1: DAQ requirements for a selected set of applications for SLS 2.0

Application	Source	Channels	Samples	Data rate	Pre-trigger	IP core
LLRF	Main	34	31228	100 ksps	No	SIS_DAQ
LLRF	Post-mortem	34	499648	7.8125 Msps	Yes	SIS_DAQ
LLRF	Turn	8	512	250 Msps	No	DataRec
LLRF	Debug	2	131072	250 Msps	No	DataRec
LBLM	Main	2	16384	250 Msps	No	DataRec
FPM	Main	1	92160	300 Msps	No	SIS_DAQ

The main features of the DataRec IP core are:

- Pre- and post-trigger recording, so it is possible to record a desired number of the data samples before and after the trigger condition. Number of samples recorded is configurable in the runtime.
- A variety of trigger modes: *free-running* where recording starts immediately; *external-trigger*, where the component waits for external trigger signal, and *self-trigger* mode, where an internal trigger is generated based on the levels of the input signals.
- Up to 8 channels recorded in the same time, with automatic deinterleaving of the data, where each channel is stored in a continuous manner in the address space.

The Finite State Machine (FSM) controlling the component has three major states: (1) waiting for the start condition from the software and a trigger signal, (2) recording the data, (3) stopping the acquisition and sending interrupt to a supervising software, so that it can read out the data.

The component has two major limitations, which makes it unsuitable for some applications: limited data length and non-continuous operation. The former constraint stems from its reliance on internal memory resources in the PL, which enables a relatively straightforward architecture, but at the same time put limit on the maximum size of recorded data. The latter limitation arises from the FSM behaviour: in states (1) and (3) samples are not recorded and are consequently lost. This issue could be potentially mitigated by deploying two IP cores operating in parallel, this however, imposes even stricter constraints on the maximum data size.

SIS_DAQ COMPONENT

The SIS_DAQ IP core overcomes the limitations of the DataRec block. The architecture of the SIS_DAQ (Fig. 2) is simplified by processing only a single interleaved data stream. Complexity is reduced even further by reuse of the existing AXI4 infrastructure available in the MPSoC and off-loading some operations such as data deinterleaving to a real-time software.

Architecture

The component uses a pair of first-in-first-out memories (FIFO) to capture the data and program a simple FSM, which has two states only:

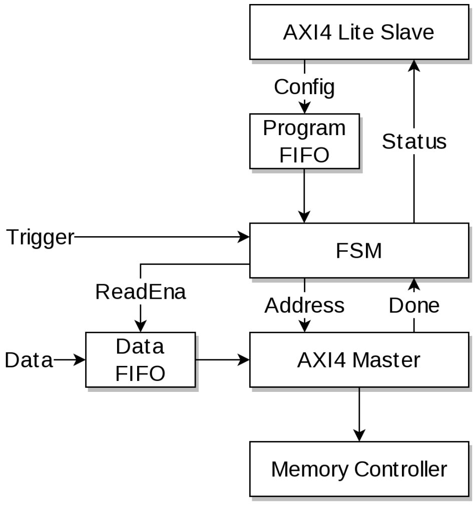

Figure 2: SIS_DAQ architecture diagram.

- In the **IDLE** state the core is waiting for a programming command from the supervising software and a trigger signal. It starts the transmission and transitions into the RECORDING state.
- During the **RECORDING** state, the component transfers data to an external memory via the AXI4 master interface. It monitors the number of data sent, initiates transactions on the AXI4 bus, and compares the total number of elements transmitted with the requested quantity. The component reverts to the IDLE state upon completion of the transmission.

This architecture, coupled with the software component, offers considerable flexibility, encompassing one-shot, continuous, and post-mortem DAQ, along with pre-triggering capabilities. A module for the EPICS control system is provided, that facilitates seamless integration of the IP core via a single command line within the startup script of the high level software. This enables rapid re-usage of the SIS_DAQ component in different applications.

Continuous Acquisition

The component enables continuous data recording, employing a triple buffering approach. The software consistently configures three data windows via the FIFO, with only two programmed simultaneously. Under normal conditions, the software reads data and automatically configures a new window. However, integrity of the recorded data remains guaranteed, even if the software operates with sub-optimal speed and is not able to process the data on time.

Post-mortem Acquisition

Post-mortem data acquisition is facilitated through the inclusion of an additional controller component within the PL. This component generates an event signal upon activation of one of its input sources, providing both the source responsible for the event and its corresponding timestamp. Data are continuously streamed into a memory across three buffers. Upon detection, the acquisition is stopped in a manner such that the sample captured alongside the event is stored in the central buffer, extended by two buffers containing samples recorded before and after the central buffer. Timestamps for each sample within the buffers are computed, enabling the assembly of an output buffer with user-defined pre-trigger and post-trigger sample counts. Typically a subset of the samples is displayed on the user interface panel, with the option to archive all available data.

DAQ SCHEME IN THE SLS 2.0 LLRF SYSTEM

The described IP cores has been employed in the LLRF application for SLS 2.0. A set of FMC Carrier/ADC mezzanine boards contains a MPSoC for real-time signal processing and communication purposes, as well as 8-channel 16bit 250 Msps ADC for 50 MHz RF input signals.

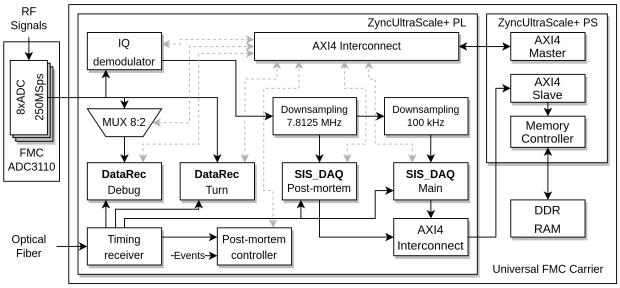

Figure 3: Block diagram of the real-time DAQ part of the SLS 2.0 LLRF application.

The signal processing (Fig. 3) consists of IQ demodulators, down-sampling and Cartesian to polar converters, and four DAQ streams. The main continuous acquisition channel and post-mortem channel are implemented using two SIS_DAQ IP cores, while a channel covering the duration of one revolution in the storage ring (Turn) and two debug channels with the ADC sampling rate are provided by the DataRec IP blocks.

A timing receiver decodes the input from the optical fiber and delivers precision timestamps for the DAQ components and a post-mortem controller.

The presented scheme was successfully implemented and tested on the target hardware, with the resource utilization given in Tab. 2. It illustrates that the BRAM occupancy for the DataRec increases proportionally with the volume of data, which is not the case for the SIS_DAQ.

Table 2: Resource utilization for the DAQ IP core instances in the SLS 2.0 LLRF application implemented on the Zynq UltraScale+ xczu11eg MPSoC

Source	Data size	BRAM	LUT
Main	2 MiB	1.5 [0.3%]	1194 [0.4%]
Post-mortem	32.5 MiB	2 [0.3%]	1196 [0.4%]
Turn	8 KiB	4 [0.7%]	856 [0.3%]
Debug	512 KiB	120 [20%]	905 [0.3%]

CONCLUSION

We have described two methodologies for implementing real-time DAQ systems on contemporary MPSoC devices, using the LLRF system for SLS 2.0 as a case study for various scenarios. These approaches are readily portable and have already been effectively deployed in other SLS 2.0 systems, such as beam diagnostics. By balancing the tasks between the PL and processors, it is possible to optimally use available storage resources. No modifications beyond top-level parameter adjustments were necessary for the IP cores, significantly reducing the development time. The methodology presented herein will also be applied to other projects, including the forthcoming High Intensity Proton Accelerator Facility upgrade.

ACKNOWLEDGEMENTS

We extend our sincere gratitude to all colleagues from the PSI AEK department and LLRF group who contributed to the development of the CPCI-S.0 ecosystem. Special thanks are due to D. Felici for providing invaluable insights regarding the FPM system and to R. Kalt and T. Schilcher for their corrections.

REFERENCES

[1] I. Johnson *et al.*, “Application development on CPCI-S.0 Hardware at PSI”, in *Proc. ICALEPCS’23*, Cape Town, South Africa, 2024, paper THPDP071, pp. 1508–1511. doi:10.18429/JACoW-ICALEPCS2023-THPDP071

[2] R. Rybaniec, B. Stef, R. Kalt, and A. Dietrich, “Signal processing architecture of the next generation LLRF systems at PSI”, presented at LLRF Workshop’23, Gyeongju, Rep. of Korea, 2023, unpublished.

[3] C. Ozkan-Loch, R. Ischebeck, and A. Stampfli, “CMOS Based Beam Loss Monitor at the SLS”, in *Proc. IBIC’21*, Pohang, Rep. of Korea, 2021, paper TUOB02, pp. 186–188. doi:10.18429/JACoW-IBIC2021-TUOB02

[4] B. Kalantari, T. Korhonen, and V. Schlott, “Bunch Pattern Control in Top-up Mode at the SLS”, in *Proc. EPAC’04*, Lucerne, Switzerland, 2004, paper THPLT186, pp. 2885–2887.

[5] B. Stef, J. Purtschert, and R. Rybaniec, “PSI’s open-source FPGA DSP libraries”, presented at IPAC’24, Nashville, USA, 2024, paper TUPR83, this conference.