SSP Boardreader documentation for DAQ review

Dr Martin Haigh

November 2016

1 Introduction

This document will describe the functionality and readiness of the SSP (SiPM Signal Processor) Boardreader program for PD, in the context of the DAQ design review for November 2016. The SSP itself will be discussed where relevant. The Boardreader is an artdaq process and part of the PD DAQ system, so its interaction with the rest of the DAQ will also be discussed.

The SSP hardware is responsible for recording, and making available to downstream components, waveforms from 36 SiPM photodetectors, ganged with 3 SiPMs per SSP channel; these comprise the readout hardware for the PDS. The SSP is capable of triggering internally on photon signals, on an external trigger source, and at fixed time intervals, and can also generate trigger signals for other detector components. Using an onboard FPGA, it can calculate metadata related to the observed pulses (heights,widths etc.), which may also be sent in lieu of the actual waveform data in order to reduce data volumes.

In turn, the Boardreader program is responsible for receiving data from an SSP unit, packaging it into fragments in time corresponding to those from the other detector systems, and sending it on to the Event Builder to produce full events. It also deals with configuration of the SSP hardware, by reading said configuration from a FHiCL file and communicating it to the hardware via the concrete setting of register values. Finally, the Boardreader deals with state transitions from the Run Control system by changing both its internal state and that of the SSP.

An SSP Boardreader program was implemented as a part of lbne-artdaq during preparations for the 35t run, and was used successfully for taking of 35t data. It is anticipated that only relatively minor changes will be required for PD, to account for changes in the SSP data structure, timing and triggering, and compatibility with newer versions of artdaq.

2 SSP hardware and data format

2.1 Description of the SSP hardware

The SSP is described in detail in [1]. Relevant information from the introduction to this document is reproduced here for convenience.

The Silicon Photomultiplier (SiPM) Signal Processor, or SSP, is an instrumentation module designed by the Electronics Group of the High Energy Physics division at Argonne National Laboratory to support the development of the photon detectors for the Deep Underground Neutrino Experiment (DUNE).

An SSP consists of 12 readout channels packaged in a self-contained 1U module. Each channel contains a fully differential voltage amplifier and a 14-bit, 150 MSPS analog-to-digital converter (ADC) that digitizes the waveforms received from the SiPMs. The digitized data is then processed by a Xilinx Artix-7 Field-Programmable Gate Array (FPGA). The FPGA implements an independent

Data Processor (DP) for each channel. The processing incorporates a leading edge discriminator for detecting events and a constant fraction discriminator (CFD) for sub clock timing resolution.

In the simplest mode of operation, the module can perform waveform capture, using either an internal trigger or an external trigger. Up to 2046 waveform samples may be read out for each event. When waveform readouts overlap the device can be configured to offset, truncate or completely suppress the overlapping waveform. Pile-up events can also be suppressed.

As an alternative to reading full waveforms, the Data Processors can be configured to perform a wide variety of data processing algorithms, including several techniques for measuring amplitude, and also timing of the event with respect to a reference clock. All timing and amplitude values are reported in a compact event record.

The SSP can be configured to trigger readout in several ways, including self-triggered, use of an external trigger, or use an external gate to readout all events within a time window. The digitization clock and timestamp across multiple SSP can be synchronized using an external clock source. A Xilinx Zynq FPGA, onboard the MicroZed system-on-module, handles the slow control and event data transfer. The SSP has two parallel communication interfaces; USB 2.0 and 10/100/1000 Ethernet. The 1 Gbps Ethernet supports full TCP/IP. The module includes a separate 12-bit high-voltage DAC for each channel to provide up to 30 V of bias to each SiPM. The module also features charge injection for performing diagnostics and linearity monitoring, and also voltage monitoring.

2.2 Data format

Data is read out from the SSP in individual triggers – since each channel can trigger independently, an "event" from the SSP contains data from a single set of 3 SiPMs. Hereafter, these units of data will be referred to as "packets", to avoid confusion with triggers and events from the global system.

An SSP packet comprises a header consisting of metadata and pulse properties as calculated by the onboard Data Processor, optionally followed by a digitised version of the input pulse, in a configurable number of 14-bit ADC values. The packet header consists of 12 32-bit words, of which the first word is always set to `0xAAAAAAAA`, with the next 16 bits giving the length of the packet. The header also contains the external timestamp, which is used to associate data from an SSP with corresponding data in the other SSPs and detector systems, and is maintained by the hardware through synchronization with the global timing system. The other information in the header is not used by the Boardreader, so it will not be discussed in detail here.

The packet header is immediately followed by waveform data. Each 14-bit sample is packed into 16 bits, with two samples in each 32-bit word. The spare bits in each 16-bit block are used to indicate the points at which the leading edge and constant fraction discriminators triggered, if relevant. The next packet follows directly in the datastream from the end of the waveform data, with another `0xAAAAAAAA` word indicating its start.

3 Boardreader interface with SSP

The Boardreader communicates with the SSP via TCP/IP over Ethernet – the SSP is set internally to a fixed IP, which the software uses to identify it over the network. One port on the SSP is used for slow control, and a separate port for data transfer. The SSP also possesses a USB interface for communication and data transfer; the Boardreader supports communication with the SSP via this interface but it is not anticipated that it will be used during data taking.

The SSP interface allows for register values to be set and read out using control packets sent over the network. All configuration of the SSP, as well as starting/stopping data taking, is handled by this mechanism. The Boardreader contains a map of the SSP register names to their literal values, allowing a user to describe the desired configuration using human-readable names – these names are the same as those listed in the SSP register map document [2].

The other port, used for event data, is not written to by the Boardreader – data are received unidirectionally from the queue.

4 Boardreader interface with DAQ

The SSP Boardreader program is a standard artdaq Boardreader, using a derived class of the artdaq `CommandableFragmentGenerator` specialized for the SSP. The direct interface with other layers of the DAQ is implemented entirely within this class.

The required interface is rather simple – at construction, the Fragment Generator receives a FHiCL parameter set containing the necessary configuration information for the DAQ software and hardware. It configures an object representing the interface with the hardware using the former, and this object is then asked to set the registers corresponding to the hardware configuration in the FHiCL. The Fragment Generator also implements start and stop methods to enable run start/stop commands from the DAQ to be processed, and the `getNext_` method, which the DAQ calls in order to get event data from the Fragment Generator. On an invocation of `getNext_`, the Fragment Generator asks the hardware interface object for data, which it stores in artdaq fragment objects and returns. It will return multiple fragments, if they are available.

5 Implementation

A simple schematic of the Boardreader internals and interfaces with the SSP and DAQ is shown in Figure 1.

The SSP Boardreader uses the `boost::asio` library internally to manage communication over Ethernet. Objects of class `boost::asio::ip::tcp::socket` are set up for the control and data ports on the SSP, and these are used for all communication with the hardware. For reading and writing registers, the SSP uses control packets with a simple structure. For data, a `boost::asio::buffer` object is used to fill a C++ container.

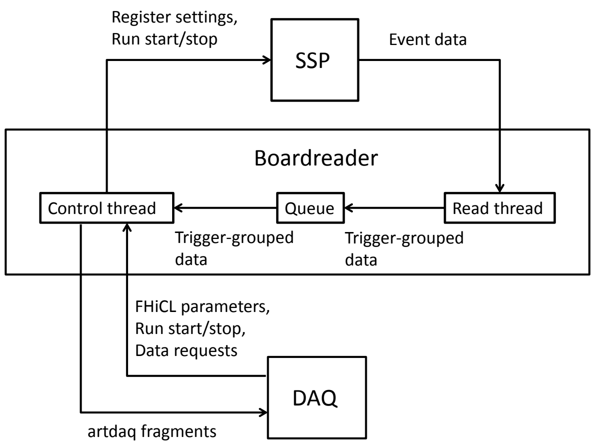

Figure 1: Block diagram showing interfaces of Boardreader to DAQ and SSP, and inter-thread communication within the Boardreader.

Upon initializing the Boardreader, the SSP is set to a stopped state by setting the relevant registers on the hardware. Starting the run can be done in one of two ways – either an immediate start is issued to the hardware, or the hardware is put in a state where a sync signal from the global timing unit will cause the collection of event data to start. When starting on timing synchronization, the synchronization timestamp can be obtained retrospectively from an SSP register, allowing the Boardreader to find out the run start time.

At the start of a run, a separate thread is started (the read thread), which is then responsible for periodically polling the Ethernet port for data. This thread first looks at individual data words to locate a packet header, and will then proceed to find out the packet size and read out all the data. An exception is raised if the full packet data do not appear within some timeout window after a header is observed. If the first word seen is not the expected front word of the header (`0xAAAAAAAA`), the thread will continue to look through subsequent words until it finds a valid header, but a warning will be issued in this instance.

The read thread is also responsible for forming the data into the larger units which will represent event fragments in the DAQ. For the 35t Boardreader, these units were fixed periods of time called millislices; however for PD data will be grouped based on belonging to a single global trigger received by the SSP, which will be passed to the Boardreader as part of the SSP packets. When all data corresponding to a trigger has been seen, it is put into the proper format with a short header, and added to a thread-safe queue. This is the structure that is popped from when the `getNext_` method of the Fragment Generator is called.

Between packets, the read thread monitors whether a stop command has been set by the main thread, and when this happens the thread will terminate. Once this happens, the control thread will send the necessary register settings to the SSP to put it into a stopped state.

As a further note, the library used for communication with the SSP allows for running without a read thread – in this case, the user must manually ask for events from the device interface. The library also supports connecting to the SSP without connecting to the data port and without setting any registers. These features can be used to run the SSP in "scope mode" or for slow control respectively.

References

[1] P. DeLurgio et al., *SiPM Signal Processor User Manual*, DUNE DocDB 1571.

[2] Z. Djurcic et al., *SSP and LCM Register Map*, DUNE DocDB 910.