BEAM SYNCHRONOUS DATA ACQUISITION FOR SWISSFEL TEST INJECTOR

B. Kalantari, T. Korhonen, Paul Scherrer Institute, Villigen, Switzerland

Abstract

A 250 MeV injector facility at PSI has been constructed to study the scientific and technological challenges of the SwissFEL [1] project. Since in such pulsed machines in principle every beam can have different characteristics, due to varying machine parameters and/or conditions, it is very crucial to be able to acquire and distinguish control system data from one pulse to the next. In this paper we describe the technique we have developed to perform beam synchronous data acquisition at 100 Hz rate. This has been particularly challenging since it had to provide us with a reliable and real-time data acquisition method in a non real-time control system. We describe how this can be achieved by employing a powerful and flexible timing system with well defined interfaces to the control system.

INTRODUCTION

The 250-MeV Linear accelerator [2] at PSI has been constructed as a test bed facility to help design, development of concepts and proof of principles toward realization of the SwissFEL project. This injector test facility will be also used as the injector of the SwissFEL.

In general in operation of pulsed machines, e.g. FELs, it is very crucial to be able to study and diagnose the machine behaviour on pulse-to-pulse bases. Therefore it is required that, the data generated by the machine components is measured and collected at each pulse. Furthermore it is also required that the collected data at each pulse is clearly distinguished from those of another pulse. The reliable, pulse-to-pulse data acquisition requires a tight cooperation and well defined interface of timing system with the control and measurement systems. Considering the repetition rate of such machines, which is normally high (from several tens to few hundred Hz), the data acquisition and collection, will not be anymore a trivial task to do. Our injector currently produces beam at 10 Hz repetition rate. However, since there are several subsystems, e.g. Laser systems, which require 100 Hz operation at the same time we have to be able to cope with data acquisition at 100 Hz.

The pulse-to-pulse data acquisition across several locally separated control/measurement nodes is also known as Beam-Synchronous Data Acquisition to which we refer by BS-DAQ in the rest of the paper. In the following we first describe our BS-DAQ concept and then briefly describe the timing system features and control system interface which helped us to implement this concept.

BS-DAQ CONCEPT

The problem of BS-DAQ is in fact that of real-time data acquisition. Hence the first and the most important issue to address is, to acquire and collect data in a specified time interval. The time interval is specified by the machine repetition rate, said the other way, time interval between two consecutive machine pulses. So it is crucial that a measurement device participating in BS-DAQ demonstrates a deterministic behaviour. Hence, in our mechanism we assume that the measurement device has this important property. Closely related to this, is the issue of presentation of the measurement in the control system. In this regard we also assume that the measured data is transformed to a control system object well within the deadline determined by the repetition rate.

Local Buffering Concept

The heart of the BS-DAQ concept is the local buffering of the generated data at the Input/Output Controller (IOC) node. A similar concept has been also employed in LCLS [3]. The major reason for this approach is that in our control system the communication between IOC’s is via an Ethernet-based protocol (EPICS Channel Access) which provides no guarantees in determinism of the data transmission delay (no upper bound on the delay). However since the IOC’s demonstrate a real-time behaviour at the local level, we collect the acquired data locally at the beam or machine rate and then after finishing the acquisition the data collected in the buffers can be transferred via the communication network to the remote client application for offline analysis i.e. correlation studies.

Role of the Timing System

To make the local buffering work, we still need to provide a minimal but reliable, real-time communication across all locally separated IOC’s which participate in a BS-DAQ run. This is because the measured data belonging to the same pulse has to be synchronously buffered at each IOC and it has to be possible to distinguish each buffered element by a *pulse ID* (pulse identification number). We use global event system of Micro Research Finland (MRF) [4] for our timing system. In the event system, an event generator in the central master timing IOC generates timing information and transmits them to all event receivers around the facility via optical links. The pulse ID or pulse marker is a unique, monotonically increasing number which is assigned to each machine pulse. It is generated in the central, master timing IOC and distributed via the timing system to all IOC’s connected to the timing system (via optical links). The pulse ID is received at each IOC via the event receiver and is transformed to a control system object where it can be used as an index of the acquired data at each pulse.

Role of the Control System

The control system at each IOC provides the means for control logic of the buffering mechanism. In our case all the logic, mechanisms and required software objects has been provided by the control system (EPICS) such that no special object type or low level driver had to be developed. The timing critical command/status data as mentioned is provided by the timing systems but finally presented as control system objects.

IMPLEMENTATION

In the implementation of the BS-DAQ mechanism, we have chosen right from the beginning to develop a generic control system software package so that the installation and usage for the control system engineers is a plug and play task which does not require any internal knowledge of the mechanism. In this regard we should emphasize that the EPICS toolkit had almost all the features that we required for implementation of the BS-DAQ.

Overview

The master timing IOC which houses the event generator is the central coordinator and synchronizer of the BS-DAQ. An EPICS program which is driven by the event systems interrupts determines, according to user specified parameters, at what pulses each IOC has to collect an EPICS channel. Then at each appropriate pulse, a command is sent synchronously to all IOC's via the timing system. When the receivers in each IOC receive the BS-DAQ commands, they find out if they have to copy value of the specified EPICS channel into the specified local buffer. The local buffers are realized by using a standard EPICS record type called "compress". Along with each data buffer there is also a pulse ID buffer to serve as the data index. A BS-DAQ run is initiated on-demand by the user for specified number of pulses. Typically the buffer length is selected such that at least 20 seconds interval can be covered. For example at 100 Hz repetition rate a buffer with 2000 element should be sufficient.

Control and Tweaking Knobs

A user (machine expert usually) who runs the BS-DAQ has some knobs to control. These are as the following:

- Number of acquisitions: this determines the total number of samples which will be collected in the local buffers at each IOC.
- Spacing: this parameter specifies the spacing between beam pulses which are sampled. For example zero spacing causes the data collection at each generated beam. A spacing of one, means samples at every second beam pulse will be collected.
- Defer cycles: this specifies how long to wait after each beam before buffering of each sample.
- Start/stop/abort/resume: these are the major control knobs to start, stop, resume or abort a BS-DAQ run.

Multi-user Operation

The BS-DAQ application is implemented such that it allows several users to simultaneously run their own independent BS-DAQ instance. Each user *leases* an acquisition *slot* and takes control over that until the acquisition is finished where the slot becomes *free* again (by the user). Currently six slots are supported and it can easily be expanded to more slots if required. In terms of logic, control and timing resources there is practically no limitation on increasing the number of acquisition slots. On the other hand number of machine experts that would use the application at the same time will not be a big number.

Dynamic Configuration

Most of the configuration parameters of the BS-DAQ can be specified at runtime. In particular the user can decide what data (EPICS channel) goes to what available buffer on an IOC. This makes the application very flexible since it is not necessary to specify measured data and the buffer which is going to be used in BS-DAQ at IOC initialization time, hence there is no need to restart the system. Furthermore, it allows better utilization of the IOC resources such as memory. It also leads to smaller and easier maintenance of the software package. Dynamic configuration, on the other hand, makes setting up an acquisition slot a rather complex task which involves several steps such as finding a free slot, slot leasing, finding free buffer, buffer assignments, etc. Currently the high level user application that retrieves and analyzes the collected data does this setup task. The high level software that our machine experts use for this purpose is Matlab.

First-Level Verifications

To provide a simple, first level cross-check in order to see if the BS-DAQ has done a successful run, we provided two verification methods. The first is to check if the data (EPICS channel) to be taken is not in the alarm state. The second is a check to see if there is any irregularities in the collected pulse ID's along with each collected data.

Support for Waveforms

The BS-DAQ is also able to collect the vector data (that has more than one element) at each sample in the same way as it does for scalar data. This was needed to collect the camera images or phase and amplitude waveforms in the LLRF systems. The BS-DAQ mechanism handles vectors data exactly in the same way as scalars. This was achieved by introducing a parameter to BS-DAQ logic, applied to every data, as *sample size* which is 1 for scalar data and N for a vector with size N.

CONCLUSION

In this paper we have presented our approach to perform Beam-Synchronous Data Acquisition (BS-DAQ) in the SwissFEL test injector facility. The concept of local buffering was presented and our implementation was discussed in details.

REFERENCES

[1] http://www.psi.ch/swissfel/swissfel

[2] T. Schietinger, M.Aiba, B. Beutner, M. Dach, A. Falone, R. Ganter, R. Ischebeck, F. Le Pimpec, N. Milas, P. Narang, G.L. Orlandi, M. Pedrozzi, S. Reiche, C. Vicario, " FIRST COMMISSIONING EXPERIENCE AT THE SwissFEL INJECTOR TEST FACILITY", Proceedings of Linear Accelerator Conference LINAC2010, Tsukuba, Japan Paul Scherrer Institut, CH-5232 Villigen PSI, Switzerland

[3] J. Dusatko, S. Allison, M. Browne, P. Krejcik, "THE LCLS TIMNG EVENT SYSTEM", Proceedings of BIW10, Santa Fe, New Mexico, US

[4] http://www.mrf.fi/