Fusion Engineering and Design 85 (2010) 286–291

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

Conceptual model of data acquisition hardware for long pulse fusion experiments: Event driven and time synchronization issues

Adriano Luchetta*, Gabriele Manduchi, Oliviero Barana, Anton Soppelsa, Cesare Taliercio, Antonio Barbalace

Consorzio RFX, Associazione Euratom/ENEA sulla Fusione, Corso Stati Uniti, 4, 35127 Padova, Italy

ARTICLE INFO

Article history:
Available online 21 May 2010

Keywords:
Data acquisition
Event-driven
Time synchronization
Time distribution

ABSTRACT

The paper addresses two important aspects: event-driven data acquisition and accurate timing reconstruction. Event-driven data acquisition allows reducing the requirements in data storage and transfer bandwidth retaining at the same time all the necessary information for a proper reconstruction of acquired signals. In many cases, in fact, the required bandwidth in data acquisition is not constant, being higher only in correspondence of particular events during the discharge. Event detection is performed in real time and the system has to guarantee a higher sampling speed in a time window centered on the event occurrence. Possible technological solutions are discussed and a new "soft trigger" approach is suggested, which allows using the same hardware for both low and high sampling speed acquisition, as well as using the general network for event transmission in place of dedicated links. Accurate timing reconstruction of acquired signals is necessary, regardless of the duration of the discharge. Care is then required in the implementation of timing systems in order to ensure both time stability and reliability over the whole plasma discharge, and possibly over the whole experiment lifetime. The typical architecture for such systems is based on a GPS clock generator that is connected to all data acquisition devices via a dedicated fiber-optics link carrying the synchronization clock. The paper introduces a different approach in timing reconstruction using IEEE 1588, a recent standard for accurate time synchronization, and discusses its potentialities and limits in data acquisition systems.

1. Introduction

ITER will address experimental sessions with plasma duration up to 3600 s, considerably extending the operation scenarios of current large tokamaks. In such long pulses, data collection is required to use techniques for continuous data acquisition.

Continuous data acquisition in long-duration fusion experiments has been addressed successfully in Tore-Supra and LHD [1,2]. Valuable development in the field has been also carried out in W7X and validated on the WEGA Experiment [3]. Recently also EAST has come into operation and SST-1 will do so soon [4,5].

In short-pulse experiments, data acquisition is typically based on the simple "batch model", in which data are stored temporarily in local memory of the ADC modules and uploaded into a central data repository after pulse [6]. In long pulse experiments the batch model is not adequate and, thus, continuous data streaming has been introduced [7]. One reason to stream data continuously is functional, due to the requirement of processing and displaying real time data during the pulse. A second reason is technological, due to the required dimension of the local memory storage that is proportional to channel number, sampling frequency and pulse duration.

Systems based on parallel busses, such as VMEbus, CompactPCI and PCI, were typically used for the implementation of continuous data acquisition systems, routing data in real time through the bus. In some cases, new commercial ADC modules were equipped on board with standard (Ethernet) or custom fast communication links to stream data during acquisition [8–10], avoiding the bus throughput bottleneck (in principle up to 80 MB/s in VME64 and 133 MB/s in 33 MHz CompactPCI). Data acquisition modules exploiting high bandwidth serial links, such as PCI Express, are currently being developed, typically in AdvacedTCA and PXIe form factors [11–13].

In addition to the ability of handling a continuous data stream, knowledge of time is of prime importance. A system-wide knowledge of time can be achieved by two approaches: distribution of a centralized clock signal to all system components that must be time-aware or use of local clocks kept synchronized by some mechanism.

This paper will concentrate on a conceptual proposal to implement effective data acquisition and time synchronization in

* Corresponding author. Tel.: +39 049 8295043; fax: +39 049 8700718.
E-mail address: adriano.luchetta@igi.cnr.it (A. Luchetta).

0920-3796/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.fusengdes.2010.04.051

hardware and will give a roadmap to implement the concepts presented. The application is triggered by the design of the ITER Neutral Beam Test Facility to be built in Padova, Italy, where beam pulse duration can reach up to 1 h and the harmonic content of many diagnostics signals to acquire can reach the value of several hundreds kHz up to MHz. The estimated amount of data to be acquired by continuous data acquisition would be very high and, so, to avoid over-designing the system, data reduction techniques have been investigated.

2. Event-driven data acquisition

Analogue signals are typically acquired through an ADC that cyclically repeats the process of holding the analogue signal, sampling it and making available the associated digital value in a buffer register. The clock driving the ADC may be a stand-alone local clock or a synchronized clock that is kept in step with a reference clock through an external synchronization mechanism. An important piece of information in this process is the point in time at which the analogue signal is held in the ADC and in general the result of the measurement process is the pair (value, time), referred to as a sample hereafter. In some cases, as for instance simple real time control applications, the time value may not be important as far as the control process proceeds cyclically (acquisition of input, logic computation, and emission of output). The time information allows sequencing events and data acquired and allows discriminating causes and effects improving, thus, the comprehension of the dynamics of physics phenomena.

The data acquisition can be a continuous process that is triggered by some sort of start event, proceeds cyclically at a constant clock frequency and is terminated by a stop event. For long-duration pulse and large harmonic content signals this technique can produce masses of data that require large amounts of memory space for data storage, high bandwidth networks for data transmission and massive human resources for data analysis. Data reduction algorithms can be used after pulse to filter data and retain only those with a high content of information and, in addition, smart techniques can be developed to mine massive databases and extract information efficiently [14]. This approach requires designing the data communication channels to the maximum data throughput. To avoid over-designing the communication networks, data compression can be applied in real time [15]. These are good strategies that, in the authors' opinion, may be complemented by real time data filtering in order to upload into the data storage system only pre-filtered data.

The information content in the data acquired may continuously vary in time and be lower in some phases and higher in some other. Accordingly to the information content level the sampling frequency could be modified in order to sample faster when the information content is high (and presumably the harmonic content) and to sample slower (down to null sampling frequency) when the information content is low. Moreover, a signal may require sampling only when its value is within a certain band of interest. In general more complex schemes may be required to handle effective data acquisition. A common situation may be the case of a signal that requires high sampling frequency in a time window centered on the occurrence of some sort of event, whereas outside the windows requires sampling at a lower baseline sampling frequency.

Hereafter we will define as event-driven data acquisition, an acquisition process that can change its parameters, for instance have its sampling frequency modified, depending on the occurrence of some events occurring somewhere. Events may be related directly to the signal under sampling (internal events) or related to any asynchronous occurrence (external event) that may happen somewhere in the experiment. Example of an internal event is the trespassing of some threshold of the sampled signal, whereas an example of an external event may be a status change somewhere in the experiment, for instance the occurrence of some instability such as a Vertical Displacement Event or an Edge Localized Mode.

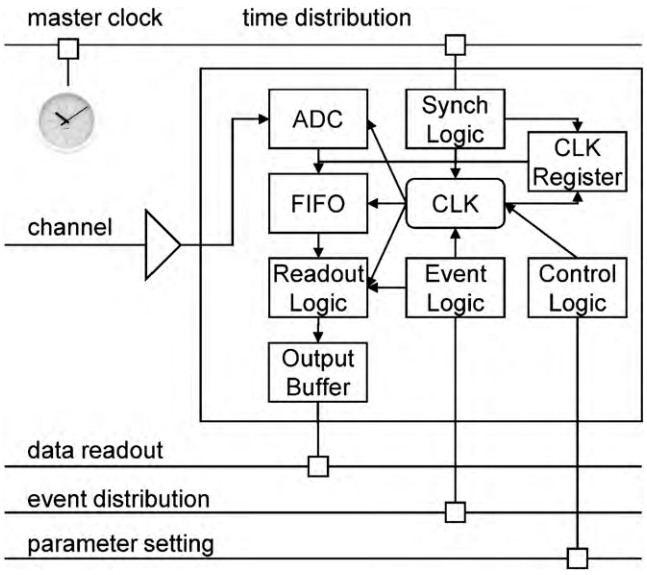

Fig. 1. Block diagram of an event-driven data acquisition module.

3. Simple model of event-driven ADC process

Fig. 1 proposes a block diagram for an event-driven data acquisition module and identifies a set of functions.

The line *channel* represents the analogue input signal that is sampled by the module. Only one channel is shown, but the concept explained below can be easily extended to multiple channels. The module is typically part of a data acquisition system, including other external entities that are hereafter referred to as the *environment*. The ADC conversion is executed by a standard ADC that is clocked at a constant working frequency f_s driven by the clock CLK. It is supposed that some external front-end electronics implements the low-pass filter to suppress harmonic components exceeding the Nyquist frequency $f_s/2$. The module needs the knowledge of time and, for this purpose, it is connected to the *time distribution* channel on which time information is routed. No assumption is necessary at this stage on the absolute or relative nature of the time distributed. The mechanism to handle time is not within the scope of this section, and thus is not described here, but will be dealt with later. The *Synch Logic* keeps the module clock *CLK* synchronized with a central *master clock* that provides a unique time base in the entire environment. A *clock register* maintains locally the value of time. At each clock pulse a sample can be produced by the module and enqueued into the *FIFO*. A sample is defined as a pair (value, timestamp), the value and the timestamp being the output of *ADC* and *clock register*, respectively. The *Readout Logic* is in charge of dequeuing the FIFO and moving the output samples into the *Output Buffer*, where they are available for data readout and can be reached from the environment through the *data readout* channel. The *Readout Logic* is programmed to apply various readout policies. It can simply route the last acquired sample into the *Output Buffer*, thus providing data sampling at the frequency f_s. The *Readout Logic* can perform more complex functions, such as re-sampling the signal at a lower sampling frequency $f_r = f_s/n$ where integer $n > 0$. This process is generally called decimation and n is referred to as the decimation factor. The process of selecting one