ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA088

TIMING SYSTEM AT ESS

J. Cereijo García*, T. Korhonen, J. H. Lee, ESS, Lund, Sweden

Abstract

The European Spallation Source (ESS) timing system is based on the hardware developed by Micro-Research Finland (MRF). The main purposes of the timing system are: generation and distribution of synchronous clock signals and trigger events to the facility, providing a time base so that data from different systems can be time-correlated and synchronous transmission of beam-related data for for different subsystems of the facility. The timing system has a tree topology: one Event Generator (EVG) sends the events, clocks and data to an array of Event Receivers (EVRs) through an optical distribution layer (fan-out modules).

The event clock frequency for ESS will be 88.0525 MHz, divided down from the bunch frequency of 352.21 MHz. An integer number of ticks of this clock will define the beam macro pulse full length, around 2.86 ms, with a repetition rate of 14 Hz. An active delay compensation mechanism will provide stability against long-term drifts. A novelty of ESS compared to other facilities is the use of the features provided by EVRs in µTCA form factor, such as trigger and clock distribution over the backplane. These EVRs are already being deployed in some systems and test stands.

INTRODUCTION

ESS [1] is a collaboration of several European countries to build the leading facility in Europe in neutron science. It is being built in Lund (Sweden) and will start initial beam operations in 2019 and complete construction by 2025. It will produce neutrons by the spallation process, that basically consists of shooting a high energy proton beam to a tungsten target. The neutrons generated are used by a number of neutron instruments to do science in a broad number of disciplines. The proton pulses are up to 2.86 ms long and generated with a frequency of 14 Hz. The Integrated Control System Division (ICS) provides control systems for the whole facility, including the global timing system used to synchronise the operation of ESS subsystems to make it operate as a single facility [2]. The timing system has the following functionality that performs with deterministic latency: distribution of trigger events and clock signals across the facility, timestamping of the events, actions and data, and broadcasting of beam-related parameters.

TIMING SYSTEM ARCHITECTURE

The ESS timing system consists of an EVG, EVRs and fan-out (FOUT) modules. Its main functionalities are:

- Trigger event distribution.
- Fast beam data distribution.
- Timestamping.
- Distribution of clock signals.
- Delay compensation for stability against long term thermal drifts.

A central top-level EVG generates a bitstream containing timing events, data and synchronous clocks, which is sent to the EVRs through several levels of FOUTs over an optical link. The EVG sends events from sequencers stored in memory, from multiplexed counters that can also be used to trigger the sequencers, from software or from external inputs. When no other event is scheduled, the EVG sends a null event, that is periodically replaced by timestamping-related special events and a heartbeat. The time used for timestamping is derived from GPS time. It is incremented internally but periodically re-synchronised with GPS to avoid drifting too far away from the GPS-defined time. In each event clock cycle the EVG also can send either data bytes from a memory buffer or a distributed bus (DBUS) with 8 simultaneous clock signals sampled at half the event clock frequency(each of them is sent every other cycle). Figure 1 shows the structure of the event clock cycles in the bitstream.

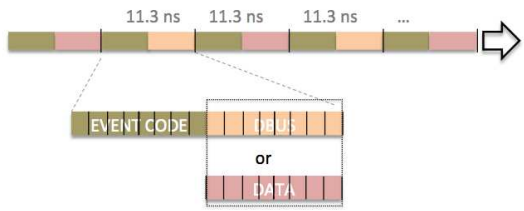

Figure 1: Frame structure of the timing system bit stream.

The EVG creates the event clock by dividing the frequency of the RF master oscillator of 352.21 MHz by 4, thus being 88.0525 MHz. The proton beam pulse frequency of 14 Hz is created by counting 6,289,464 cycles of the event clock, of 11.357 ns each. Figure 2 shows a rough sketch of the machine timeline, where an event at 14 Hz defines the beam cycle, and other important events are sent in relation to it.

The EVRs receive the bitstream, decode it and perform the necessary actions, mainly generating output triggers with parameters, such as delay, width or polarity, controlled by software. Software triggers and interrupts are also generated by EVRs, as well as sharing the beam parameters. The EVRs also perform the timestamping with the "wall-clock" time distributed by the EVG and which is kept internally in between pulses. Figure 3 shows the timing distribution system.

Beam Parameter Data

In addition to the timing events, the timing system has the capability to broadcast important parameters of the proton beam to the entire facility in a fast and reliable manner. The data is written to a 2 kilobyte data buffer of the EVG and then sent to the EVRs one byte every clock cycle when the

* javier.cereijogarcia@esss.se

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA088

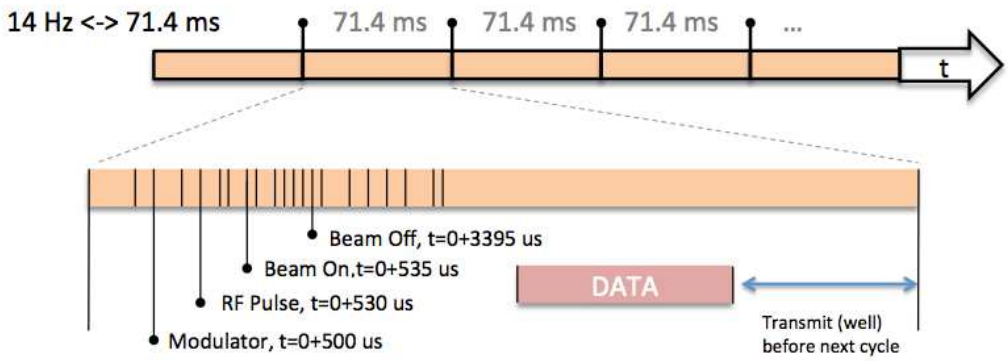

Figure 2: Machine timeline example.

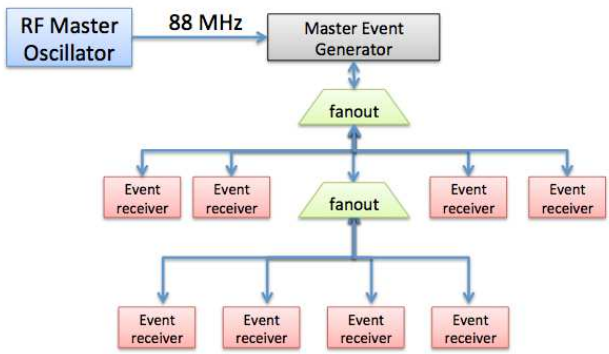

Figure 3: Timing distribution structure.

Table 1: Preliminary List of Beam Parameters Distributed by the Timing System. Data items marked with * are further explained in the following subsections.

Data item	Format
Beam repetition rate	Unsigned integer (Hz, 1-14)
Beam present	Boolean
Accelerator mode*	Enumerated integer
Beam envelope mode*	Enumerated integer
Pulse length	Float (ms)
Proton energy	Float (MeV)
Raster pattern*	Enumerated integer
Target segment	Unsigned integer

distribution is triggered. The data buffer is segmented to allow partial updates of the data, with a segment length of 16 bytes. The transmission starts with a segment number that defines the starting address in the data buffer. After an EVR writes the received segments into its data buffer it raises a completion flag, but only for the starting segment, even if the transmission includes several consecutive segments. There are also checksum error flags and segment overflow flags. The last segment of the data buffer is reserved for delay compensation data.

The data buffer is updated during one accelerator cycle with information about the following cycle, so that it is available in advance of the start of the cycle and the controlled subsystems can get ready. Table 1 shows a preliminary list of the data to be included in the data buffer.

Accelerator Mode One of the parameters sent in the data transmission is the accelerator mode, which defines the activities allowed in the accelerator. The accelerator modes are shown in Table 2. The transitions between modes are slow and driven by operator intervention. Only some transitions between the different accelerator modes are allowed, and they require authorization of safety and protection systems.

Beam Envelope Mode The beam envelope mode describes an "envelope" (limits) of accepted characteristics of the proton beam, assuming a peak current of 62.5 mA, but lower values are accepted and specified in other beam parameters. Transitions need permit from the protection systems. Some possible examples of the beam envelope mode are: no beam, probe beam (0 to 5 µs, 0 to 1 Hz), fast tuning (0 to 5 µs, 0 to 14 Hz), slow tuning (0 to 50 µs, 0 to 1 Hz) long pulse verification (2.86 ms, 1/30 Hz), shielding verification (low power, ~30 kW, production (2.86 ms, 1-14 Hz, 5 MW).

Raster Pattern The raster pattern is a way of implementing the beam repetition rate in an almost arbitrary pattern. It has 14 slots where each slot may or may not have beam. There are a number of predefined patterns.

EPICS INTEGRATION

The Experimental Physics and Industrial Control System (EPICS) [3] is a well known software toolbox for creating distributed soft real-time control systems for scientific facilities such as a particle accelerators, telescopes and other large scientific experiments. The communication between EPICS and the real world is realised by Input/Output Controllers (IOCs).

The EPICS integration of the timing system is realised by the *mrfioc2* driver [4], developed and widely used by the EPICS community, and which provides access to the whole functionality of the EVG and EVR modules. ESS

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA088

Table 2: List of Accelerator Modes. LoS stands for Line-of-Sight.

Name	Description
Shutdown	Ion source, most components off, except non-hazardous systems. Tunnel access possible.
RF Power	RF systems could be on, but proton production off. Intended for RF test.
Source	Source plasma on, no beam extraction. Access to tunnel (but not source cage) possible.
Studies Dump	P-beam to Tuning dump; tunnel closed. Access to LoS instruments only if primary shutters closed.
Studies Target	P-beam to target. Target areas and tunnel "searched and locked"; no access to LoS instruments.
StartUp Target Test	Same as Studies Target.
StartUp Low Power	Same as Studies Target
StartUp Ramping	Same as Studies Target.
Production	Same as Studies Target. Neutron production.

has selected EPICS and *mrfioc2* for configuring the timing system components, for example the actions of EVRs to event codes or how they handle the data distribution.

HARDWARE

ESS uses the MTCA-EVR-300 and PCIe-EVR-300DC EVRs provided by Micro-Research Finland (MRF) [5], which are already being deployed in some systems and test stands. The PCIe-EVR-300DC, due to its small form factor, only has a SFP and a micro-SCSI type connector that interfaces the IFB-300, a board with 8 slots for double outputs/inputs mezzanine modules available in different types (TTL, TTL 5V, LVPECL, etc). The MTCA-EVR-300 comes in two flavours, one with the same micro-SCSI type connector as the PCIe-EVR-300DC, and another with two slots for the universal modules. Both flavours also have a SFP, 2 TLL inputs, 4 TTL outputs, 32 RTM connectors and 10 backplane connectors. These backplane connectors are of special importance at ESS as they will allow for a reliable and clean installation with less cables, since all the fast acquisition and processing will be done in µTCA AMCs and RTMs [6, 7]. The backplane connectors run the 8 bussed trigger lines shared between all the AMCs, which are used for triggers, and the 2 clock lines for providing low jitter signals that can be used for clock generation or arbitrary pattern generation (following certain rules); the main use will be distributing the event clock and other high precision clocks. The MTCA-EVR-300 with micro-SCSI type connector is shown in Figure 4.

ESS will use an EVG and FOUT integrated into one module, called Event Master (EVM) in µTCA form factor. This

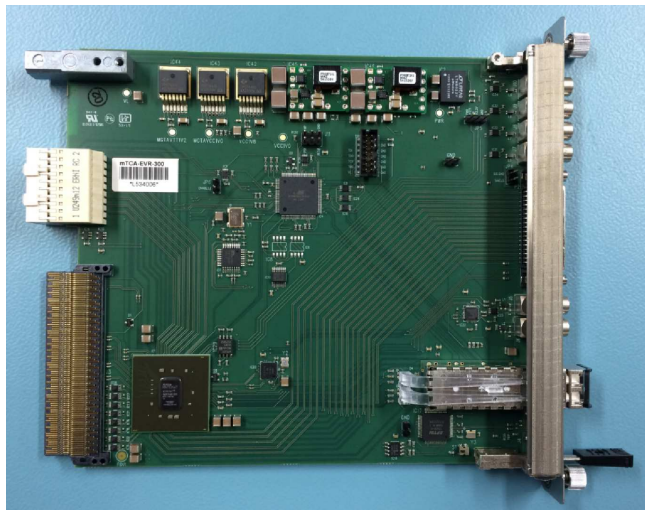

Figure 4: MTCA-EVR-300.

EVM is not yet available, so the initial installation will consist of VME-EVM-300 in VME form factor acting as FOUTs and a MTCA-EVR-300 with a specific firmware (FW) acting as EVG, since it meets all the hardware requirements.

STANDALONE MODE

In the current phase of ESS a lot of systems are still being designed and tested, many of them requiring timing for running. These systems are located in many different locations, not just in Lund where ESS is being built, but also in different European countries where ESS in-kind partners are designing and buildings parts of ESS. Because of that and the cost of providing every system with its own copy of the timing system, including at least one EVG and one EVR, a different approach for providing timing has been implemented. The latest versions of the MTCA-EVR-300 and PCIe-EVR-300DC EVRs FW have a sequencer implemented, which is identical to the sequencers found in any EVG. The sequencer is usually triggered by prescalers that divide down the frequency generated by a fractional synthesizer that is present in the EVR, but can also be triggered from other sources such as pulsers, the distributed bus, or manually with a software trigger. The most common use of the sequencer is just to provide a 14 Hz event that mimics the beam cycle, which is used to trigger different outputs. Harmonics of this frequency are usually also implemented.

UPLINK STREAM

There is an uplink stream from the EVRs to the EVG that works in parallel to the normal, downlink bitstream. It is used as a fast communication mechanism to inform the EVG mainly about hardware or other type of failures and trigger the *post mortem* functionality. The protocol used is the same as the downlink stream: there are 256 events available to be sent at a frequency of 88.0525 MHz. The events are triggered from the EVRs' TTL inputs, specially by the machine protection system and interlock system. When

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA088

the uplink events reach the EVG, they are replicated and sent over the downlink stream to all the EVRs. The EVRs then stop a circular buffer present in the µTCA crates that stores the last actions and their timestamps. This circular buffer is consumed by the *post mortem* system to gain information on the moments previous to the failures.

MINI-IOC

An in-kind partner of ESS, the Tallinn University of Technology, is developing an embedded EVR called miniIOC based on Zynq [8], a System-on-Chip that integrates an ARM [9] processor and FPGA. Its purpose is to replace the µTCA crate that ICS uses [6, 7] in remote locations with limited space and accessibility, where the µTCA crate is not feasible or cost efficient. The miniIOC is a light and compact standalone embedded computer system consisting of a central CPU and programmable logic (together integrated in the Zynq SoC). It is capable of running Linux with limited functionality and IO interfaces but precise timing and synchronization, with EVR functionality integrated in the FPGA that also provides interfacing logic for FMC extension IO cards. It has an optical transceiver for the timing bitstream, DIN-rail mountable form factor, and an Ethernet interface for communicating with the control system, remote management (reboot/restart, power cycling, software updates, etc) and booting from network, with no local storage of SW images (Uboot, linux image, FPGA image, etc). Special attention is turned to high reliability and remote manageability aspects, low power consumption with passive cooling only and no moving parts such as cooling fans. Figure 5 shows the system architecture sketch of the miniIOC.

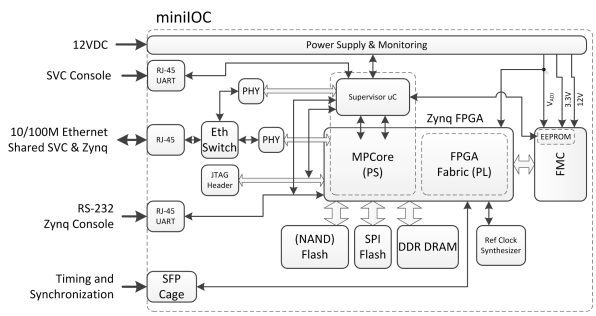

Figure 5: System architecture sketch of the miniIOC.

CONCLUSION

This paper presents the timing system used at ESS. After a general overview and description of its integration in the ESS control system, a series of characteristics is described in more detail: the beam parameter data broadcast mechanism, the specific timing hardware used at ESS, the new standalone mode intended for testing and the uplink stream. Finally the development of the miniIOC by an in-kind partner is introduced. As ESS is still in design and construction phase some of the details presented in this paper might change.

REFERENCES

[1] European Spallation Source ERIC website, http://europeanspallationsource.se

[2] T. Korhonen *et al.*, "Status of the European Spallation Source Control System", in *Proc. 15th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'15)*, Melbourne, Australia, Oct. 2015, paper FRB3O02, pp. 1177–1181, http://jacow.org/icalepcs2015/papers/frb3o02.pdf, doi:10.18429/JACoW-ICALEPCS2015-FRB3O02, 2015.

[3] Experimental Physcics and Industrial Control System website, http://www.aps.anl.gov/epics/

[4] mrfioc2 repository in github website, https://github.com/epics-modules/mrfioc2

[5] Micro-Research Finland website, http://www.mrf.fi

[6] J.P.S. Martins, S. Farina, J. H. Lee, D. Piso, "MicroTCA.4 integration at ESS: from the front-end electronics to the EPICS OPI", presented at *16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17)*, Barcelona, Spain, Oct. 2017, paper THPHA133, this conference.

[7] S. Farina, J. H. Lee, J. P. S. Martins, D. Piso, "MicroTCA Generic Data Acquisition Systems at ESS", presented at *16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17)*, Barcelona, Spain, Oct. 2017, paper TUAPL01, this conference.

[8] Xilinx's All Programmable SoC website, https://www.xilinx.com/products/silicon-devices/soc.html

[9] ARM processors website, https://www.arm.com/products/processors