# FAIR TIMING SYSTEM DEVELOPMENTS BASED ON WHITE RABBIT

C. Prados, R. Bär, D. Beck, J. Hoffmann, N. Kurz, S. Rauch, W. Terpstra, M. Zweig (GSI, Darmstadt, Germany), M. Kreider (GSI, Darmstadt, Germany; Glyndwr University, U.K.)

## *Abstract*

A new timing system based on White Rabbit (WR) is being developed for the upcoming FAIR facility at GSI, in collaboration with CERN, other institutes and industry partners. The timing system is responsible for the synchronization of nodes with nanosecond accuracy and distribution of timing events, which allows for real-time control of the accelerator equipment. WR is a fully deterministic Ethernet-based network for general data transfer and synchronization, which is based on Synchronous Ethernet and PTP. The ongoing development at GSI aims for a miniature timing system, which is part of a control system of a proton source, that will be used at one of the accelerators at FAIR. Such a timing system consists of a Data Master generating timing messages, which are forwarded by a WR switch to a handful of timing receivers. The next step is an enhancement of the robustness, reliability and scalability of the system. These features will be integrated in the forthcoming CRYRING control system in GSI. CRYRING serves as a prototype and testing ground for the final control system for FAIR. The contribution presents the overall design and status of the timing system development.

## INTRODUCTION

The GMT triggers and synchronizes accelerator equipment accordingly to the accelerator cycles [1]. Cycle lengths range from 20 ms (present UNILAC), several seconds (synchrotrons SIS18 and SIS100/300) to several hours (storage rings) [2]. The beam production chain is an important concept in accelerator control systems. It describes the production of a beam from an ion source through the accelerators to a target. Properties of such a beam production chain include the ion type (from protons to uranium), energy, intensity, focus and emittance and other parameters at the final destination. This information is conveyed to the GMT, which has an integral view on the tightly synchronized accelerators and beam transfer sections. The GMT must take into account the execution of several beam production chains in the accelerator complex at the same time. For each part of the machine, switching between different beam production chains will be possible between cycles, which implies a high degree of true parallel operation.

The GMT is made of an interconnected network of Timing Receiver Nodes (TRN) and special nodes so-called Masters. The TRN are devices synchronized to the Timing Master, source of time and frequency for all the network, and they are also the receivers of timing messages from the Data Master. The interconnection is established using WR Switches, which are responsible of the propagation of the synchronization and timing messages to the TRN.

In about four years, first beam will be injected into the FAIR accelerator complex. By then, the General Machine Timing system must support about 2000 Front End Controllers (FEC) with integrated TRN. The main tasks of the GMT are distribution of timing messages, clocks and timestamps. Timing messages synchronize actions of the accelerator in hard real-time on a level of about 1 ns, even if components are a few kilometers apart. Clock and timestamps distribution, inherently provided by White Rabbit PTP, will be used for correlating data acquired at distinct places throughout the facility.

The GMT is linked to other systems. It must react with upper bound latency on external signals like interlock signals [3] by executing predefined alternatives in the schedule. All systems connected to the timing system depend on it's high availability. Distribution of timing messages, clocks and timestamps must be guaranteed for commissioning and testing even when the accelerator does not produce beam. As a failure in synchronizing equipment may led to loss of beam, the distribution of timing messages must be robust - at most one timing messages per year may be lost. Furthermore, critical components of the GMT must be implemented redundantly.

The GMT not only serves for the operation of the FAIR accelerator machines. Also Data AcQuisition (DAQ) systems of experiments will link to the GMT for correlating experimental data with accelerator actions. Moreover, a connection to the GMT allows for time stamping of data in case of globally triggered DAQ systems, fan-out of time stamps for free running DAQ systems and finally provides distribution of precise synchronous clock signals.

## DESIGN

GMT is a decentralized design, Fig. 1, based on a network where every kind of device is responsible of carrying out an specific tasks. The Clock Master is source of time and frequency for the network. The Clock Master creates, schedules and sends timing messages to control the accelerator and the WR Network (WRN) provides synchronization propagation and transports the data (e.g timing messages) to the TRN. The hardware and software design of this devices is being developed in GSI (e.g TRNs) and in collaboration with other institutes and companies (e.g WR Switch).

### *White Rabbit*

White Rabbit (WR [4]) is a protocol developed to synchronize nodes in a packet-based network with sub-ns accuracy. WR is based on existing standards: Ethernet (IEEE 802.3 [5]), Synchronous Ethernet (SyncE [6]) and PTP [7].

ISBN 978-3-95450-139-7

A WR network offers low-latency, deterministic packet delivery and network-wide, transparent, high-accuracy timing distribution.

The timing network at SIS/FAIR is a White Rabbit network interconnecting TRNs using WR Switches and fiber optic links.

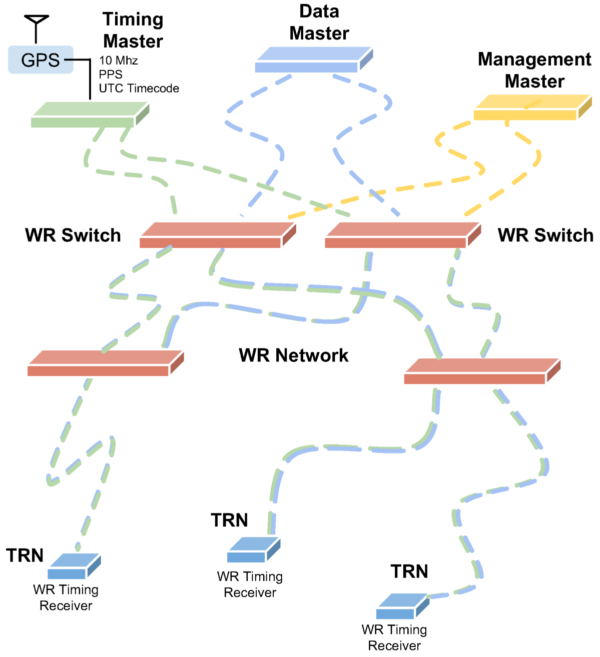

Figure 1: General Machine Time.

### Data Master

The Data Master (DM) for the FAIR accelerator deals with various different tasks. The first step is to take machine commands from the LHC Software Architecture (LSA) and convert these into sequence programs, depicting beam production chains [8]. The corresponding actions that are to be taken are preprogrammed into the timing endpoints. The DM will run a multiple of these sequence programs in parallel and each is generating timing messages containing action IDs and their execution times. These are scheduled and handed over to a dispatcher in order to be sent over the timing network. The sequence programs will communicate with each other to resolve dependencies in beam production chains and are able to take external signals and interlocks into account.

The hardware and functionality of the DM is subdivided into three parts:

*CPU - API Block*

Its CPU is fed by the LHC Software Architecture (LSA) with machine parameters derived from physical requirements for beam production chains. These parameters are converted into sequence programs, compiled and uploaded to the FPGA.

*FPGA - SoftCPU Cluster*

These programs are run in parallel on SoftCPU macros residing in the FPGA. They deal with sending out timing messages paired with an execution time to WR nodes, reacting to interlocks and mutual synchronisation.

*FPGA - Timing Message Concentrator*

Its primary functions are aggregation of messages into Ethernet Frames and to schedule transmission of these timning messages over the timing network so they arrive on time at the respective nodes.

The first version of the DM is using hardware cores for specific tasks. This version will still use fixed sequence programs with parameters rather than code auto-generated by the LSA system. It is currently under development and shortly awaiting final testing. It is meant to control the CRYRING accelerator. The second generation of the DM should scale up to 32 soft CPUs and will use sequence programs automatically generated by the LSA core. After being tested on the CRYRING, it will be used to control GSI's SIS18 synchrotron and the UNILAC linear accelerator.

### Clock Master

Timing is distributed in the WRN from a switch called Clock Master (CM) to all the other TRNs/WR switches in the network. All the devices in the WRN lock their frequency and adjust their local clocks to that of the CM. The CM is basically a WR Switch configured as Master Clock in the PTP protocol. It uses 10 MHZ and PPS signal from a GPS for synthonization and the UTC Time for the synchronization of the network. The Fig. 1 shows that the CM is on top of the network connected to the first layer of WR switches.

### Management Master

At startup, the WR networking devices need essential configuration parameters (e.g. ip address). During operation, the network has to be constantly monitored in order to prevent, identify and solve malfunctions or breakdowns of devices.

Currently the Management Master (MM) is serving ip addresses to the management port of switches and WR nodes using BOOTP [9]. Information about the status of the network (e.g. link up/down, synchronization status etc...) is gathered in the MM using a Distributed Information Management tool. In the future the MM will offer advance management and monitoring capabilities using the SNMP [10] and sFlow [11].

### Timing Receiver Nodes

FAIR requires multiple form factor variants of timing receiver nodes. In order to reduce the maintenance effort, all of our different FPGAs utilize a common system on chip (SoC) design. This design is centered around the Wishbone bus system, which combines the standard timing receiver functionality with the form factor specific bus interfaces. The standard functionality included in every timing receiver consists of White Rabbit, an Event-Condition-Action (ECA) scheduler, and a timestamp latch unit (TLU). The form factor specific interfaces fall into three categories.

First, master interfaces for controlling the timing receiver, such as PCIe, VME, USB, and Ethernet. Second, slave interfaces for controlling off-chip resources, such as DDR3, SRAM, flash, daughter boards, and displays. Third, raw IO interfaces suitable for capturing signals to timestamps using the TLU or generating high precision timing signals using the ECA. These last interfaces are generally LEMO, LVDS, or high density connectors.

The ECA unit is a gateware component for producing control signals at preprogrammed times. The idea is to split high-level timing command, namely events, which should be carried out by multiple devices at a given time, from the actions an individual timing receiver must take. To this end, after attaching a timing receiver to a controlled device, one programs the ECA's condition table. This rules in this table specify which events from the DM require action by the timing receiver. Furthermore, the rules may include a time offset to compensate for local delays due to the attached table length or delays inherent to the controlled device.

As a concrete example, one could program the ECA to respond to ramp events by outputting new set values to the magnet's power supply. When the DM broadcasts a timing message to apply in 200us, all the timing receivers process this request. Those timing receivers which control magnets on the ring recognize this event requires their action. They calculate when they must power their magnets to achieve the 200us target and schedule an action for that time. When the time arrives, the ECA executes the required action, accurate to 8ns. In the future, we intend to leverage Altera's PLL phase shifting technology to reach 100ps accuracy.

Similar to the ECA unit, every timing receiver includes a TLU. For each input connected to the TLU, there is a timestamp queue. The rising or falling edge of a signal on the input causes an absolute timestamp to be recorded in the respective queue. Currently these timestamps are only accurate to 8ns, though we intend to improve this to better than 1ns.

Although different form factors provide different physical control interfaces, we have unified all of them to a single C library interface. A user of this library does not need to care how the timing receiver is attached to his computer. Using this library, all devices in the timing receiver can be automatically discovered using the self-describing bus standard. Thus, one does not need to know the particular SoC address layout for a given timing receiver. Instead, one simply locates the component to control, complains if it is missing, and then proceeds to access it via the C interface to Wishbone.

Although the master interfaces for each form factor differ, they all include a network connection in order to run White Rabbit. It is possible to control the Wishbone bus of a timing receiver over the network using the Etherbone protocol. therbone is simply a serial version of the Wishbone bus protocol.

**SCU** Most timing receivers will be built in the Scalable Control Unit (SCU) form factor [12]. It is planned to run around 1200 units in FAIR. The SCU is a combination of a carrier board with an Arria II FPGA and a COMExpress board running Linux. The communication between FPGA and COMExpress board is done via PCIe. An SCU controls up to 12 slave cards via the SCU bus and can connect to an optional daughter board for additional IOs. The main use cases in FAIR for the SCU is the control of the ACU (Adaptive Control Unit) for ramped power suplies, control of radio frequency devices and to serve as a gateway to existing control system components connected via a bus derived from a MIL-STD-1553.

**PEXARIA** The pexaria5 is a 4-lane PCIe card intended to be used in a standard PC. It is based on the newer Arria V FPGA platform. It sports up to four SFP cages, a 26-pin trigger bus interface, a USB port. In the future it will also host a daughter board with external IO ports.

**EXPLODER** The exploder is a portable timing receiver with many IO ports. It is a two-planed, enclosed, hand-sized device hosting an Arria II FPGA and four SFPs. The top plane contains the application-specific IOs, such as LCD display, LEMOs, LVDS, NIM, trigger bus, knobs, etc. The only additional master interface it provides is USB.

**VETAR** The vetar is a VMEbus IEEE-1014 card intended to be hosted in VME crates. It also based on the Arria II FPGA and is endowed with one SFP cage, SRAM, LCD display, EEPROM memory and USB interface. LVDS and lemo I/Os are available on the main board or on the extension board with additional high density connector.

### *Timing Network*

The Timing Network interconnects the above described components of the Timing System using WR switches [13]. A stable and continuous synchronization of all the TRNs with an appropriate accuracy and reliable and deterministic distribution of timing messages are the key requirements. Both timing and data resilience against network failures is achieved using redundant connections. Lower layer protocols establish a spanning tree topology setting ports connected to cyclic paths to block/passive state. Upper-bound delivery latency of timing messages from the DM to TRNs is guaranteed using a QoS tagging [5] of the traffic and Cut-through [14] switching.

## DEVELOPMENT AND IMPLEMENTATION PLAN

The new challenge for building the FAIR timing system is twofold: combination of already consolidate technologies like WR and Etherbone, and to create a reliable system capable of scaling to more than 2000 nodes.

The iterative development methodology is based on the creation of a relative simple GMT with minimal features and then increase the complexity and functionality, going through iterations that will result in a running system.

### *Proton Linac Source*

A complete control system will be delivered for testing the source of the proton linac, that will later serve as an

ISBN 978-3-95450-139-7

additional injector into the existing synchrotron SIS18 at GSI/FAIR. In summer 2013 a functional prototype of this control system has been set-up and passed its acceptance test. From the point of the timing system, this small control system is a simple pulse generator for a handful of TRNs. However, this is the first productive timing system that uses prototypes of the FAIR timing system, such as a timing master, a timing network and timing receivers.

### *CRYRING*

The next step will be a timing system for the CRYRING, a small synchrotron and storage ring, which has been reallocated from the Manne Siegbahn Laboratory in Stockholm [15] to GSI in 2013. It is presently being set up next to the Experimental Storage Ring (ESR). One of the motivations behind the CRYRING project is its explicit usage for FAIR development tests. Although the operation of the ring will commence in stand-alone mode, it covers nearly all relevant aspects of an accelerator facility. Thus, it presents an ideal test ground for new sub-systems of the accelerator control system such as the timing system. Here, new development and features for FAIR can be tested without the overhead required for routine operation. It is estimated that between 20 and 50 TRNs have to be connected to a first timing system.

### *New Timing System for SIS18 and ESR*

The timing systems for the proton linac source or CRYRING have been either very small in scale our operated under experimental conditions. Integrating the existing timing system at the synchrotron SIS18 is a important milestone, since SIS18 serves as an injector into the new FAIR machines. Operation at SIS18 requires the timing system to work reliably in routine operation 24/7. The old timing system, which is based on an extension of the MIL-STD-1553 bus, needs to be replaced by the new GMT.

### *Integration of UNILAC*

Next to the new proton linac, the existing heavy ion linear accelerator UNILAC is the second injector into the FAIR accelerator complex. Here, the existing MIL-based timing system will most likely not be replaced for the reasons of cost and effort. Moreover, the UNILAC is special with respect to timing. First, it is operated at 50Hz and phase locked to the mains voltage delivered to GSI. Second, the ion sources feeding the UNILAC require fixed repetition rates for reasons such as thermal stability. However, injecting the UNILAC beam into the SIS18 requires linking the UNILAC timing to the schedule of the FAIR accelerator complex.

### *FAIR Timing System*

All of the previous instances of the new GMT had different aspects: Proof of principle at the proton linac source, prototyping and development of final solutions at the CRYRING and reliability of routine operation at the SIS18 and the ESR. One important step towards the final timing system is the combination of all these aspects. This is eased, since the timing master and the first two layers of switches will be physically located in an existing building close to the SIS18. This allows to set up first components of the timing master for CRYRING, SIS18 and ESR already. By this, the timing master, its infrastructure and its interfaces will be continuously developed, improved and tested at its final location over a few years. Another important step is to address the issue of scaling the new GMT to the size required for the final FAIR facility. As it is planned to purchase the equipment not just before FAIR machine commissioning but over a period of several years, scalability can be addressed and tested in several steps. One option would be to use test areas in buildings of the existing facility.

## CONCLUSION

A new general machine timing system for FAIR is presently being developed in iteration cycles, where iteration provides a functional timing system focusing in certain aspect of the final timing system. The next iterations will implement timing system for the source of the FAIR proton linac, followed by the CRYRING at GSI. The replacement of the existing timing system at the SIS18 and the ESR combined with addressing scalability will pave the way to the timing system for the FAIR facility.

## REFERENCES

[1] T. Fleck et al., FAIR Accelerator Control System Baseline Technical Report, General Machine Timing System, 2012.

[2] D. Beck et al., F-DS-C-05e, FAIR Detailed Specification General Machine Timing System, 2012.

[3] J. Fitzek et al., F-DS-C-08e, FAIR Detailed Specification Interlock System.

[4] White Rabbit `http://www.ohwr.org/projects/white-rabbit`

[5] IEEE Std. 802.3-2008, 2008.

[6] Timing characteristics of a synchronous Ethernet equipment slave clock (EEC), ITU-T Std. G.8262, 2007.

[7] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IEEE Std. 1588-2008, 2008.

[8] The Timing Master for the FAIR Accelerator Facility, Bär, T. Fleck, M. Kreider, S. Mauro. ICALEPCS 11.

[9] Bootstrap Protocol (bootp), RF 951, B. Croft J. Gilmore.

[10] Simple Network Management Protocol, RFC 3411-3418.

[11] sFlow, `http://wwww.sflow.org`

[12] Performance of Standar FAIR Equipment Controller Prototype, Stefan Rauch.

[13] White Rabbit Switch `http://www.ohwr.org/projects/white-rabbit/wiki/Switch`

[14] The All-New Switch Book, Rich Seifert, James Edwards, Wiley Publsing 2010.

[15] M. Lestinsky et al, CRYRING@ESR:A study group report, `http://www.gsi.de Darmstadt, July 26, 2012.`

ISBN 978-3-95450-139-7