LAUNCHING THE FAIR TIMING SYSTEM WITH CRYRING

M. Kreider, J. Bai, R. Bär, D. Beck, A. Hahn, C. Prados, S. Rauch, W. W. Terpstra, M. Zweig
GSI Helmholtzzentrum für Schwerionenforschung, D-64291 Darmstadt, Germany

Abstract

During the past two years, significant progress has been made on the development of the General Machine Timing (GMT) system for the upcoming FAIR facility at GSI. The primary features are time-synchronization of 2000-3000 nodes using the White Rabbit Precision Time Protocol (WR-PTP), distribution of International Atomic Time (TAI) timestamps and synchronized command and control of FAIR control system equipment.

A White Rabbit network has been set up connecting parts of the existing facility. A next generation of the Timing Master has been developed. Timing Receiver nodes in the form factors Scalable Control Unit (standard front-end controller for FAIR), VME, PCIe and standalone have been developed. CRYRING is the first machine on the GSI/FAIR campus to be operated with this new timing system and serves as a test-ground for the complete control system. Installation of equipment starts in late autumn 2014 followed by commissioning of equipment in winter.

INTRODUCTION

The primary task of the General Machine Timing (GMT) system is the hard real-time control of the GSI and FAIR accelerator complex with sub-ns precision [1]. This is a two-step process. First, the *Settings Management* [2] distributes multiple settings to concerned Front-End Computers (FECs) via the normal network of the accelerator. Activities[1] are prepared by the Front-End Software FESA [3]. Second, the GMT generates on-time actions at the FECs. Such an action triggers a prepared activity at the FEC and provides an index referencing one of the preloaded settings.

The fundamental idea behind the GMT is the concept of time-based control. The distribution of information and timely execution of activities are decoupled. As a prerequisite, all nodes of the GMT share a common notion of time provided by WR-PTP [4] via the dedicated White Rabbit network (timing network). The central component of the GMT is the Data Master (DM) [5]. At some time the DM receives an (updated) schedule recipe for the operation of the facility from the settings managements. This recipe only contains indices to and time intervals between actions. Based on this recipe, the DM controls the facility in hard real-time. This is again a two-step process. First, the DM broadcasts timing messages including the indices, but with absolute execution time stamps, via the timing network to the Timing Receivers (TR) embedded in the FECs. The messages must be distributed with an upper bound latency. Second, messages are received by the TRs where they are filtered. Relevant messages remain pending until the specified execution time when the TR performs the action. Depending on the configuration of the TR, such an action could be digital signal generation, complex activity such as ramping a radio-frequency system, or signaling an event to the front-end software via an interrupt request (IRQ).

In 2014 the primary features have been implemented in such a way that the GMT and other components of the FAIR control system can be used coherently for the control of a real machine like CRYRING. This synchrotron has in the meantime moved from its original site in Stockholm to GSI and is presently installed in a refurbished cave behind the existing Experimental Storage Ring (ESR). While the infrastructure for the installation of the control system is presently being completed, the GMT components relevant for the recommissioning of CRYRING have been implemented and tested. This paper reports on the on-going work and summarizes the present situation.

ASTERISK

For the nodes of the GMT, the timing team at GSI supports a variety of hardware types and functionality. The interfaces of the GMT provide a common "look and feel" to the users, hiding the complexity and differences between the form factors to a large extent. Although some features (e.g. a display) may not exist on all form factors, common functionality must be presented identically at the interfaces.

To guarantee these requirements, the timing team builds releases. Such a release includes hardware, gateware (FPGA code), firmware (embedded CPU code) as well as software (host system code) in a consistent way. The first version, named *Asterisk*, has been released in July 2014. It includes all features of the GMT required for the next milestones of the whole FAIR control system.

Hardware

Asterisk includes four form factors. The Scalable Control Unit (SCU) is the standard FEC for the FAIR control system and has been developed by the hardware section of the control system department [6]. Three other form factors have been developed by the department of Experiment Electronics and are intended for usage by the department of Beam Instrumentation as well as Data AcQuisition (DAQ) systems of FAIR experiments. The most important one is the PCIe module PEXARIA5, since this module represents the reference implementation based on an ARRIA V FPGA for all form factors provided through the GSI timing team. The two remaining modules, the VME board VETAR and the standalone form factor EXPLODER are still based on ARRIA II FPGAs.

[1] Example: Ramping of a magnet.

ISBN 978-3-95450-146-5

Gateware

Gateware is synthesized Hardware Description Language (VHDL) code for Field Programmable Gate Arrays (FPGAs). For the GMT, the gateware is based on a Wishbone System-on-Chip architecture [7]. This way, the functionality can be clearly separated and implemented by dedicated Wishbone master or slave devices attached to a hierarchy of several Wishbone crossbars. Each Wishbone device is identified by vendor ID, device ID as well as major and minor revision required for the implementation of a Self-Describing Bus (SDB) record. The connection to host bus system or Ethernet are provided by Wishbone masters.

Typical components addressed by users would be: A Timestamp Latch Unit (TLU) allows timestamping of incoming digital signals with 1 ns precision. An Event-Condition-Action (ECA) unit filters incoming timing messages [6]: Relevant messages are transferred to so-called action channels, where they are sorted according their execution time [8]. On-time, the ECA spits out the message data with a granularity of 8 ns to Receiving Components (RC), that are also implemented in VHDL. Examples of RCs are a message queue required for IRQ handling to the host system, or General Purpose IO (GPIO) that allows output of digital signals with a 1 ns granularity. Of course, the gateware includes the WR-PTP core required for time synchronization.

An important development has been the implementation of an on-chip CPU cluster of Lattice Micro 32 (LM32) softcores [9]. The cluster is connected to the Wishbone bus and implements shared memory and Message Signaled Interrupts (MSI) for synchronization.

Software

The main software components within *Asterisk* include a kernel driver specific for each host bus system, a generic Wishbone kernel driver and the userland Etherbone [10] API. This is sufficient to provide transparent access to the on-chip Wishbone devices from userspace. The supported interfaces are PCIe, VME, USB and UDP/Ethernet. This is complemented by software tools and libraries for specific Wishbone devices like a flash controller, the WR-PTP core or the ECA.

Data Master

The Data Master is in charge of command and control of the FAIR accelerator complex in hard real-time. It is implemented as a hybrid system. A high-end industrial PC serves as an interface to the settings management and other components of the control system. A PEXARIA PCIe module serves as the main hardware component. Its key feature is a FPGA hosting a multi-core cluster of LM32 softcores. Distinct processes on these cores are each in charge of real-time generation of timing messages of a particular part of the accelerator complex. A dedicated tool chain allows configuration of the DM with a schedule recipe in XML format and control of the operational state. Only one instance of the Data Master will be used for CRYRING and later-on for FAIR. For more details see [5].

Timing Network

The timing network is composed of commercial White Rabbit switches. For the start of CRYRING, only top down timing messages from the Data Master are allowed next to WR-PTP. These messages are forwarded by the switches to the nodes via cut-through routing. As their is no other traffic, the latency of a switch is on the order of a few μs.

Timing Receiver Nodes

Asterisk supports four TR types. EXPLODER is a standalone TR for digital I/O and allows configuration via USB or via the timing network. VETAR and PEXARIA are VME and PCIe modules. They can be configured via their host system bus and support IRQs. The most important for equipment control is the SCU. This is an embedded system and operated in custom crates of 3U height. It includes a SCU carrier board with on board TR, dedicated piggy boards, and a Com-Express module hosting the front-end software. As a very important feature, it provides a connection to the slave modules in the same crate via the so-called SCU-bus on the backplane. Slave modules implement I/O to external hardware like power supplies or radio-frequency systems.

INTEGRATION WITH THE OVERALL CONTROL SYSTEM

CRYRING

CRYRING is the first system that requires an integration of all components of the FAIR control system. Towards the application layer and settings management, the Data Master has implemented a prototype FESA class.

At the TRs, a first version of an interface towards FESA has been implemented. Frankly speaking, this is only a hack in FESA core. However, it already allows configuring the TR and to receive IRQ and the data associated to a timed action of the ECA. As a demonstration, a FESA class is implemented which, first, receives all ECA actions received via the host bus system and, second, publishes relevant information via the Controls MiddleWare (CMW) [11]. This information about actions triggered by the GMT is distributed and available to all applications of the control system.

The department of Beam Instrumentation has successfully integrated TRs provided by the GMT into PCIe and VME systems. Three FESA classes have been implemented demonstrating the integration of GMT, FESA and control system equipment in FECs [12].

INTEGRATION WITH DAQ SYSTEMS

The timing receivers developed for the GMT have two features that are of interest for DAQ systems.

The timestamp latch unit (TLU) allows timestamping with a granularity of 1 ns. This feature is of interest for timestamping events or even detector signals. This can be applied

ISBN 978-3-95450-146-5

to DAQ systems with high trigger rates exceeding 100 kHz. Long term tests over weeks have been passed successfully. This demonstrates the robustness of software, drivers and SoC Wishbone architecture of the TR.

Another feature is clock and timestamp fan-out provided by the form factors PCIe, VME and standalone. A 200 MHz clock is phase locked between all TRs connected to the timing network. A 100 kHz clock is phase locked to the 200 MHz clock and provides encoded timestamps that are sent in between the 100 kHz clock ticks. Due to the distributed nature of the GMT, this feature allows not only for timestamping but, when combined with dedicated electronics, for the measurement of time differences with a precision in the two digit picoseconds range. As an example, this can be used for Time-of-Flight measurements for particle identification.

For more details on the integration of GMT and DAQ please refer to [13].

ISSUES

Although the integration of DAQ and GMT systems shows high stability even on long term operation, this picture changes as soon as additional Ethernet traffic is allowed in the timing network. A loss of White Rabbit lock has also been observed by other users of White Rabbit elsewhere and has been investigated by our colleagues from CERN. It has been reported, that this issue is improved with a new release of the switches software and gateware [14].

The present implementation of front-end software typically claims a resource of the TR exclusively. This results in a failure to share unique resources like IRQs between two or more userland application. Along the same line, access to digital I/O on TRs or access to the SCU backplane bus can effectively not be shared amongst different applications. These and other issues triggered the development of a software framework, codenamed *SaftLib*, which is presently being defined.

Some important features of the GMT are not yet implemented. Examples are *robustness* of timing messages, *redundancy* of the timing network and *priority encoding* in White Rabbit switches.

CONCLUSION AND OUTLOOK

Important features of the GMT have been implemented. A Data Master exists. The SCU is integrated. TRs in the form factor PCIe and VME have been developed and integrated in the FECs of the Beam Instrumentation group as well as DAQ systems. The GMT is *ready for installation* and integration with CRYRING equipment will be started once the required infrastructure, such as power, racks, network and cable trays, has been installed at CRYRING.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the CERN White Rabbit Team, the driving force behind the development of White Rabbit PTP. Furthermore the authors would like to thank our GSI colleagues from the departments of Beam Instrumentation, Experiment Electronics and Radio-Frequency for their help. We thank Peter Moritz (†) and Sabine Voltz (†).

REFERENCES

[1] D. Beck et al., "The New White Rabbit Based Timing System for the FAIR Facility", FRIA01, Proceedings of PCaPAC (2012) Kolkata, India.

[2] J. Fitzek et al., "Settings Management within the FAIR Control System Based on the CERN LSA Framework", WEPL008, Proceedings of PCaPAC (2010) Saskatoon, Canada.

[3] Al. Schwinn et al., "FESA3 - The New Front-End Software Framework at CERN and the FAIR Facility", WECOAA03, Proceedings of PCaPAC (2010) Saskatoon, Canada.

[4] J. Serrano et al., "The White Rabbit Project", TUC004, Proceedings of ICALEPCS (2009) Kobe, Japan, 2009.

[5] M. Kreider et al., "New developments on the FAIR Timing Master", FPO022, *These Proceedings*, PCaPAC (2014) Karlsruhe, Germany.

[6] S. Rauch et al., "Facility Wide-Synchronization of Standard FAIR Equipment Controllers", WEPD48, Proceedings of PCaPAC (2012) Kolkata, India.

[7] Wishbone B4, specification: `http://cdn.opencores.org/downloads/wbspec_b4.pdf`

[8] W.W. Tersptra et al., "Inexpensive Scheduling in FPGAs", TCO301, *These Proceedings*, PCaPAC (2014) Karlsruhe, Germany.

[9] W.W. Terpstra, "The Case For Soft-CPUs in Accelerator Control Systems", THCHMUST05, Proceedings of ICALEPCS (2011) Grenoble, France.

[10] M. Kreider et al., "Open Borders for System-on-a-Chip Buses: A Wire Format for Connecting Large Physics Controls", Phys. Rev. ST Accel. Beams 15 (2012) 082801.

[11] V. Rapp et al., "Controls Middleware for FAIR", WCO102, *These Proceedings*, PCaPAC (2014) Karlsruhe, Germany.

[12] H. Bräuning, GSI (2014) private communication.

[13] N. Kurz et al., "White Rabbit Applications for FAIR Experiments", GSI Scientific Report 2013 (2014) to be published.

[14] Eighth White Rabbit Workshop, Geneva, 2014: `http://www.ohwr.org/projects/white-rabbit/wiki/Oct2014Meeting`

ISBN 978-3-95450-146-5