Facility-Wide Synchronization of Standard FAIR Equipment Controllers

Stefan Rauch, Wesley Terpstra, Wolfgang Panschow, Matthias Thieme, Cesar Prados, Marcus Zweig, Mathias Kreider, Dietrich Beck, Ralph Bär

GSI, D-64291 Darmstadt, Germany

Abstract

The standard equipment controller under development for the new FAIR accelerator facility is the Scalable Control Unit (SCU). It is designed to synchronize and control the actions of up to 12 purpose-built slave boards, connected in a proprietary crate by a parallel backplane. Inter-crate coordination and facility-wide synchronization are a core FAIR requirement and thus precise timing of SCU slave actions is of vital importance.

The SCU consists primarily of two components, an x86 COM Express daughter board and a carrier board with an Altera Arria II GX FPGA, interconnected by PCI Express. The x86 receives configuration and set values with which it programs the real-time event-condition-action (ECA) unit in the FPGA. The ECA unit receives event messages via the timing network, which also synchronizes the clocks of all SCUs in the facility using White Rabbit. Matching events trigger actions on the SCU slave cards such as ramping magnets, triggering kickers, etc.

Non-deterministic execution time is a potentially much more serious problem. For example, if a kicker executes an action a few nanoseconds too late, the beam might be lost. However, not all actions require the same precision, and it may make sense to trade accuracy for flexibility in some situations. Fortunately, the most common equipment controller in FAIR, the Scalable Control Unit (SCU), has several possibilities for executing actions. This paper outlines the timing requirements of various accelerator components in FAIR and explores the alternatives which could meet them.

Use Cases

- main frontend controller for the FAIR project
- different slaves for different use cases
  - Adaptive Control Units (ACU) for power supplies
  - FPGA Interface Board (FIB) for Radio Frequency (RF) control
  - Kicker modules controlled by IFK via MIL-STD-1553 based field bus

Introduction

In the FAIR control system, a data master issues high-level commands to control accelerator devices. The front-end controllers in the system react to relevant commands, issuing appropriate actions to their hardware components. Depending on the action to be taken, there are different timing requirements to be met.

- commands carry absolute execution timestamp
- time limit for front-end controllers for receiving commands
- depending on the time for processing an action
- tradeoff between responsiveness and planning ahead

Execution Alternatives

- FPGA
  - can be programmed to generate output on 8ns phase aligned clock edge
  - with fine delay card down to 1ns
  - only source of jitter in PLL of the FPGA and inherent inaccuracy of White Rabbit
- LM32
  - FPGA triggers soft-CPU via interrupt
  - software generates appropriate action
  - delay from switch time to interrupt context and running the software
  - jitter from cache behaviour and on-chip bus access
- Atom-Kernel
  - FPGA interrupt directly handled in kernel
  - delays equal to LM32 + PCIe bridge delay
  - more error caused by Linux kernel
- Atom-Userspace
  - FPGA interrupt delivered to userspace
  - adds delay by context switch
- FESA
  - interrupt is translated to an action using threads
  - this increases number of context switches

Conclusion

The measured times as presented in Figure 2 must be reviewed in the context of different use-cases. As an example, ramping of magnets must be done synchronously. Here, a guaranteed synchronicity of 10-20ps must be achieved for ring machines like the SIS18 and the SIS100. Another example is the control of kicker magnets, which requires at least 3ns precision and can only be done with FPGA Hardware Description Language (HDL). Software on the COM Express module may only be used for cases, where hard real-time is not required. None of the solutions involving the CPU on the COM Express module fulfill those requirements, as long as the use of real-time Linux as operating systems is a stringent requirement for software tools like FESA.

For hard real-time the options are FPGA HDL or LM32 software. Here, FPGA HDL provides nanoseconds timing while LM32 software provides a better flexibility. To avoid stringent limitations for future developments of the FAIR accelerator complex, standard FAIR equipment controllers like the SCU should be designed to support hard real-time on the nanoseconds scale. If flexibility during runtime is required, the ideal solution could be a combination of both options, where LM32 software creates the action patterns that are phase aligned with high precision by FPGA HDL.