

CONTROLS ARCHITECTURE FOR THE DIAGNOSTIC DEVICES AT THE EUROPEAN XFEL

O.Hensler, A.Aghababayan, L.Petrosyan, V.Petrosyan, K.Rehlich, DESY, Hamburg, Germany

Abstract

The European XFEL X-ray laser is a 3.4-km-long facility, which runs essentially underground and comprises three sites above ground. It will begin on the DESY site in Hamburg, Germany and runs mostly below surface to the XFEL research site, which is to be erected south of the town of Schenefeld. First electron beam is expected in 2014 in the injector and in 2015 through out the whole facility. The XFEL will be an electron accelerator and its electron beam will be used to create intense photon beam in long undulator sections to be used in various research fields.

For controlling all diagnostic devices like a toroid, BPM or BLM, it is planned to use the new MTCA.4 crate standard instead of VME. ATCA is an emerging standard from the Telecom Industry and adapted with the PICMG MTCA.4 branch for physics usage. The communication on the backplane utilizes the high-speed serial PCIe communication plus precise clock lines and SATA interface. The MTCA.4 hardware supports hot-plug mechanism and remote monitoring and control via IPMI over Ethernet. Some of the diagnostics will be connected to 16Bit ADCs with up to 125Mhz sampling rate from Struck Company or to an internal DESY development call DAMC2. The software architecture is based on the DOOCS control system known from the FLASH accelerator. The raw data from the ADCs will be read via DMA transfer by one server process. Then this raw data will be distributed locally on the CPU using a message passing system based on the ØMQ project. The receiving server processes in order to calculate these data into engineering units. Everything works in an event driven way.

The current hardware and software concept will be presented.

INTRODUCTION

At FLASH over the last 15 years, VME systems have been used for all kinds of readout and control of fast diagnostics like BPM, toroid or wirescanner. The more then 30 years old VME standard has several drawbacks e.g. the parallel bus, no standardized crate management and it is relative expensive. So, it was considered to move to a more modern crate standard for the European XFEL [1] project. Since FLASH [2] is the "test bench" for XFEL, it was decided to install the same hardware and software for the FLASH2 extension and migrate slowly the old VME systems to the μ TCA one.

Since 2002, the new Advanced Telecom Computing Architecture (ATCA) emerged on the market, which is based on PC technology and serial communication links.

In addition to ATCA, a smaller version of it, based on the ATCA mezzanine cards, called μ TCA was defined. This standard was extended even further to MTCA.4 [3], formerly known as μ TCA for Physics.

The reasons, why μ TCA was chosen for the XFEL are the following:

- ∞ Modern standard based on common PC technology
- ∞ Fast serial communication via PCIe
- ∞ Standardized crate management via IPMI
- ∞ Hot Swap capability
- ∞ Capable of redundant crate layout
- ∞ More cost efficient then VME

DOOCS

The Distributed Object Oriented Control System DOOCS [4] is the leading system for the FLASH accelerator and chosen for the XFEL. DOOCS is a common client/server control system and based on an object-oriented approach at the front-end/server and client/display side. It is mainly implemented in C++, but there is now a Java client-side implementation available called jDOOCS, which is the communication interface for the new display tool jDDD [5]. An interface for MATLAB clients is provided. The communication protocol is based on ONC Remote Procedure Calls (RPC), but efforts are on the way to replace it by the TINE [6] protocol. The client interfaces of EPICS and TANGO are integrated.

μ TCA HARDWARE

The μ TCA crate is a 19" form factor frame to accommodate the AMC cards. There are already several versions of these crates on the market, which allow sliding in the AMCs in horizontal or vertical orientation. One may purchase simple desktop test crates or full-featured 19" versions with 12 slots with the possibility of redundant fans, power supplies, CPUs or MCHs.

MTCA.4

For the XFEL project, the standard is enhanced to host vertical orientated AMC boards with double height and the possibility to attach Rear Transition Modules (RTM). These RTMs are used to connect IO signals and to allow signal-conditioning circuitry. Compared to 6U VME boards, it offers about 30% more PCB space. The idea of this standard extension is to add features, which are required especially for industrial and physics applications, like precise clock distribution over the backplane. This standard is called MTCA.4 and defined by the PICMG organization in 2011.

Used commercial AMC modules at DESY

- ∞ Several CPU modules from different vendors with an Intel or PowerPC CPU
- ∞ Hard disk carrier for 2.5" SATA drives
- ∞ IndustryPack module carrier for one or three modules
- ∞ SIS8300: 10-channel 125MSample 16bit ADC from Struck Company plus RTM analog interface
- ∞ MCH (MTCA Carrier Hub)
- ∞ Slow Analog and Digital IO
- ∞ CANbus, 2 lines
- ∞ Ethernet/Telco interfaces
- ∞ Power Supplies

Available in-house AMC modules at DESY

- ∞ XFEL timing system interface
- ∞ XFEL MPS (Machine Protection System) interface
- ∞ DAMC2, Virtex5 board with SFP connections
- ∞ LLRF controller
- ∞ LLRF RTM 1.3GHz down converter

Several more RTMs are in design phase at DESY.

IPMI CRATE MANAGEMENT

One of the key features of μ TCA is the ability to monitor and control the whole system by a well-defined management, independent from the CPU and the operating system, called Intelligent Platform Management Interface (IPMI). This crate management is done locally by the MircoTCA Controller Hub (MCH), which controls the power supplies, manages the 12V supply of every module including the CPU, enables the hot-plug functionality, takes care of the PCIe switching between AMC modules, has Ethernet switching capabilities and provides an IPMI interface to the external Ethernet. Every AMC module has to provide a well-defined set of information, like an ID, a serial number or its power consumption, otherwise it is not accepted by the MCH and no 12V is supplied. The IPMI protocol runs over Ethernet allowing external tools for monitoring. An IPMI channel to connect to the console output and to the BIOS of the CPU is under development.

A DOOCS server, running on a separate middle-layer computer, was developed to readout and control all the properties of the μ TCA crate. This server is able to detect the insertion or removal of AMCs, creates new server locations and starts monitoring this new card. The jDDD panels are then able to reflect these changes dynamically, so that these panels will always show the actually installed modules.

SERVER ARCHITECTURE

The software concept of front-end DOOCS servers running on a μ TCA crate will be explained by the

example of a toroid readout in the following. Very similar implementations are planned for BPM, BLM and other fast diagnostic devices, which require bunch resolved readout. To provide a transparent transition to the new server, a new facility was created in the DOOCS naming system, which allows keeping the same device, location and property names.

As operating system, Ubuntu Linux 12.04 Long Time Support (LTS) has been chosen and an installation service is provided to allow network booting and crash recovery.

In order to read the toroid signal, three DOOCS server will be involved and described here in the order of operation.

- ∞ The x1timer server is in charge of configuring the XFEL timer hardware through a Linux device driver interface. Via this driver, the server receives a SIGUSR1 interrupt just after a macro-pulse, which synchronizes the server with the accelerator repetition rate. After reading several information from the hardware, like the macro-pulse number or the timestamp, the server sends this data package to the SIS8300DMA server, using a message passing system.
- ∞ In order to configure the SIS8300 hardware, this server creates one location per 10-channel ADC module, which keeps all necessary hardware information. A 108MHz external clock clocks the ADCs, leading to a buffer size of 90000 16Bit words, which allows covering an 830 μ s long pulse. The SIS8300DMA server receives the data package from the x1timer via a callback function. Inside this callback function, the internal SIGUSR1 method is called to loop over all locations, where one location corresponds to one ADC board. Inside of this location, the DMA transfer is setup and the raw ADC data is stored in local 16bit arrays and tagged with the information from the x1timer server. This buffer is then passed to the messaging system to be pushed to the toroid server.
- ∞ At startup, the toroid server registers with one of the raw data channels of the SIS8300DMA server. After the raw data is read, this server sends the data to the toroid server in order to do the conversion to a charge reading. The data is reduced here from the 108MHz sampling to one data value per bunch. As FLASH runs with up to 9MHz bunch repetition rate, a float buffer of 7200 values will be created. In addition the toroid server is computing the number of bunches and archives the charge reading over time. This server does the Ethernet communication to the Data Acquisition system DAQ [7] and to the displays as well.

OMQ

In the former VME based systems, a similar architecture as above described is implemented, but it is using shared memory inter process communication (IPC) to pass the raw data from the DMA- to the toroid server.

The system is synchronized by the usage of semaphores, which can lead in problematic dead locks.

Using a message passing system like ØMQ [8], this problem is solved by design, as no locking mechanism is required.

ØMQ is an open source socket library with several language bindings for C, C++, Java, .NET, Python and others. The library may carry the messages locally via Inter Process Communication (IPC), via TCP or multicast.

The IPC functionality of ØMQ is integrated into the core server library of DOOCS now and implemented for several single and array like DOOCS data types. The sender process has to provide a data buffer and enable this sender functionality at start-up for the desired data types. The receiver process may connect to this property by using a standard DOOCS address string. A special class is provided to keep this string and attach the callback function. This string may be changed online to reattach to another channel.

It was easy to integrate ØMQ into the DOOCS system; first experience shows, that it runs very stable and after restart of a process in the chain, the reconnects are working fast and reliable.

STATUS

At the moment, several μ TCA crates have been already installed at the FLASH accelerator and are used in standard operation, mainly serving as interface to the old 9MHz FLASH timing system or for slow analog or digital IO read-out. For all these crates, IPMI middle-layer servers are setup for monitoring and control.

One μ TCA crate is permanently attached to the 9MHz timing plus the 108MHz external clock to allow parasitic operation and development of the toroid system with two toroid signals connected. On this computer, the above described software architecture has been installed. Some more timing information need to be passed through the messaging chain to adapt the readout to changes of the accelerator settings automatically.

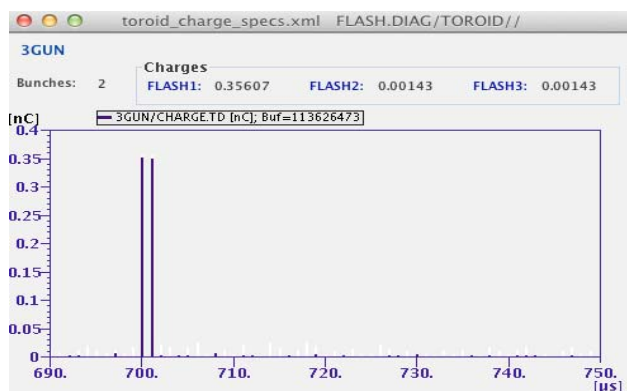


Figure 1: Toroid data display with 2 bunch, 0.35nC, 1 μ s repetition rate.

The software runs stable and the performance seems sufficient, even though a full equipped crate has not been

tested yet. The latency of the data flow from the driver through all three servers to the toroid server was measured and is around 3ms as shown in Figure 1.

Installation of the hardware and software will start in spring 2013 for FLASH2 and for the XFEL one year later.

SUMMARY

The XFEL fast diagnostics and controls will be based on μ TCA front-end systems. The required hardware has been designed and tested; the mass production and the purchasing of boards will start soon. The software architecture has been proven to work with toroid readout and will be adapted now for other systems.

REFERENCES

- [1] <http://xfel.eu>
- [2] <http://flash.desy.de>
- [3] <http://www.picmg.org>
- [4] <http://doocs.desy.de>
- [5] JDDD": A Java DOOCS Data Display for the XFEL
E.Sombrowski, K.Rehlich, ICAEPCS 2007
- [6] <http://tine.desy.de>
- [7] Evolution of the FLASH DAQ System, V.Rybnikov
et al, ICAEPCS 2009.
- [8] <http://www.zeromq.org>