ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA133

MicroTCA.4 INTEGRATION AT ESS: FROM THE FRONT-END ELECTRONICS TO THE EPICS OPI

J.P.S. Martins, S. Farina, J.H. Lee, D. Piso, European Spallation Source, Lund, Sweden

Abstract

The European Spallation Source (ESS) is a collaboration of 17 European countries that is building a leading neutron research center in Lund, Sweden. The ESS facility will have the most powerful neutron source in the world, providing 5 MW of beam power. The Integrated Control Systems Division (ICS) is responsible for all the control systems for the whole facility. For the accelerator control, ICS will provide different hardware platforms according to the requirements of each specific system. For high performance systems, demanding high data throughput, the hardware platform is the MicroTCA.4 standard. This work presents the software stack that makes the integration of a high-end MicroTCA hardware into the ESS Control System, with the implementation details of the FPGA firmware framework, kernel and userspace drivers, EPICS driver and finally the EPICS IOC that integrates the MicroTCA boards.

FRONT-END TO USERSPACE

For the ESS Control System, MicroTCA platform has been chosen as standard hardware for systems that demand fast acquisition rate and online processing. With the flexibility provided by Advanced Mezzanine Cards (AMC) carriers and a selection of mezzanine boards, this platform can meet the most user requirements at the facility. ICS plan is to provide a reduced standard set of hardware in order to optimize the long term support and maintenance of the software and hardware that comprises the MicroTCA ecosystem.

Data Acquisition Hardware

The main board of this set of hardware [1] is the AMC carrier IFC1410, equipped with a Xilinx Kintex Ultrascale FPGA and a QorIQ T2081 CPU, manufactured by IOxOS Technologies. The IFC1410 also has two HPC VITA 57.1 slots for FPGA Mezzanine Card (FMC) units. The combination of FMC cards and Rear Transition Modules (RTM) to the AMC board gives the flexibility to configure an acquisition system. The on-board T2081 CPU allows a user to run an EPICS IOC directly on the board, thus increasing the modularity of the hardware platform in the sense of control system integration and distribution. The Figure 1 shows the IFC1410 board, without any FMC attached to it.

For analog signals the main FMC boards supported by ICS are ADC3110, ADC3111 and ADC3117, from IOxOS Technologies. The ADC3110 implements eight analog inputs using 16-bit ADC up to 250 MSPS with AC coupled input stages. The ADC3111 has the same characteristics as the ADC3110 but with DC coupled input stage. The ADC3117 implements 20 analog inputs (DC coupled) using a 16-bit ADC up to 5 Msps and also has two analog outputs driven by a 16-bit DAC with a range of ± 10V.

Figure 1: IOxOS IFC1410 MicroTCA board.

The IFC1420 AMC is designed for applications that require analog signal conditioning that is not possible to implement in an FMC card, due to the reduced area available, but can be achieved using a Rear Transition Module (RTM). The IFC1420 has the same FPGA and CPU model of the IFC1410 but with a special mezzanine card that occupies one of the FMC slots and has 10 analog input channels routed to the RTM interface.

Currently ESS has been using the SIS8300 digitizer board, from Struck [2]. This board is a MicroTCA AMC carrier with on-board 10-channel ADC (16-bit @ 125 MSPS) and two 16-bit DAC for fast-feedback implementations, also routed to the RTM interface.

FPGA Firmware Application

The basic firmware application that is available for the IFC1410 FPGA is a waveform digitizer that acquires analog signals from the FMCs attached to the board. This application can be used for standard data acquisition on the CPU side and can be modified to add user-specific digital processing blocks and functions. The main features of the basic application are:

- Waveform digitizer with data buffering using the FPGA memory blocks.
- Waveform digitizer with storage in external DDR3 memory.
- Independent trigger function for both FMCs.
- Parallel acquisition for all channels.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA133

Interface between CPU and FPGA

The data transfer between CPU and FPGA in data acquisition systems is generally done using PCIe interface, which enables high throughput rates and distributed architecture. On the IFC1410 the embedded CPU has a PCIe x4 Gen3 link connected to the FPGA. The FPGA design kit provided by IOxOS uses a proprietary framework [3]. The main feature of its architecture is a high-speed interconnection among all the entities that interface to the FPGA: FMC slots, on-board DDR3 SDRAM, AMC backplane and the PCIe endpoint linked to the CPU.

This framework implements a memory mapped model with segregated I/O space (IO_Bus) and memory space (TMEM) that enables the CPU of the IFC1410 communicate with the FPGA application using the PCIe link:

- The IO_Bus mechanism acts as a register mapped structure on the PCI BAR and can be used for basic 32-bits read/write transactions.
- The TMEM memory space transactions are used by the DMA operations to transfer bulky data from the DDR3 connected to the FPGA up to the CPU kernel space.

Kernel and Userspace Drivers

The basic software layer provided by IOxOS as part of the support package for its MicroTCA boards is a kernel module and a userspace API. The kernel module implements the facilities of a device driver for data acquisition systems: register read/write operations; buffer allocation; DMA transfers and access to communication buses inside the FPGA infrastructure. The userspace API encapsulates the system calls, providing a higher-level interface for the clients of the library.

On the top of those software layers, ICS is developing a library to provide generic data acquisition interface to the firmware application running on the FPGA. This library abstracts away the differences between the FMC cards providing a unified set of common functions for arming the device for analog conversion, wait for acquisition to complete and reading out the acquired samples. By keeping this standard interface, the support development of new FMC modules is optimized, as they can be quickly deployed for the high layers applications.

BOOT ENVIRONMENT

ESS Control System is based on ESS EPICS Environment (EEE) [4], which implements dynamic linking of compiled EPICS modules at runtime. On the MicroTCA crates, the CPUs that run the IOC process should be a diskless device that gets its root filesystem and mounts the EEE pool of files using NFS service.

For the IFC1410/20 cards the boot procedure is the standard Linux embedded system approach. As these boards have both CPU and FPGA, they need support for managing the FPGA bitstreams that will be flashed on the FPGA in addition to the kernel image and root filesystem for the CPU. The boot process is configured using U-Boot bootloader scripts. The U-Boot script is retrieved from the server using TFTP and sourced on the bootloader environment, instructing the board on where it should find the necessary files to start-up the system. The bootloader will use TFTP to fetch the FPGA bitstream from the server, flash it on the chip, get the kernel image, mount the root filesystem and start Linux. This architecture provides good flexibility to bring up new boards and test different FPGA bitstreams.

EPICS INTEGRATION

ESS Control System [5] will be based on EPICS middleware, thus all the MicroTCA-based systems will export its high level functionalities to the EPICS infrastructure. The use of new electronics like MicroTCA boards recently released on the market requires the development of new EPICS drivers to integrate those devices.

Early Implementation

Currently, the EPICS drivers for the MicroTCA boards at ESS were developed using the NDS3 [6] framework, developed by Cosylab as a C++ set of classes. NDS is built on top of Asyn [7] thus offering direct compatibility with EPICS device support layer.

Drivers for EPICS based on NDS3 implements the device as a hierarchical tree structure, automating the creation of objects associated to EPICS records, with specific read and write functions.

AreaDetector for Digitizers

The collaboration with EPICS community is one of the main goals of the ICS strategy for the ESS Control System. Following this premise, the use of widely used EPICS modules is encouraged, so we can benefit from the support of the community and contribute to it. AreaDetector [8] is the standard for 2D detectors and its facilities can be easily used to develop driver support for digitizer cards. Therefore, AreaDetector was chosen as the standard framework for the digitizer cards at ESS Control System. AreaDetector is also built on top of Asyn, which is also widely used in EPICS community. The architecture of AreaDetector allows the creation of C++ classes that can implement the basic controls of the hardware and can be inherited by new class definitions for specific applications on the construction of the EPICS IOCs. AreaDetector also provides out of the box plugins for statistics, conversion and other data processing operations on the acquired arrays from the hardware. Currently, ESS has developed support for the Struck SIS8300 using AreaDetector for some beam instrumentation applications, showing the viability of the framework. The EPICS support for the IFC1410/20 boards will be redesigned and also developed using AreaDetector. A layered diagram of the complete software stack is depicted in Figure 2.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA133

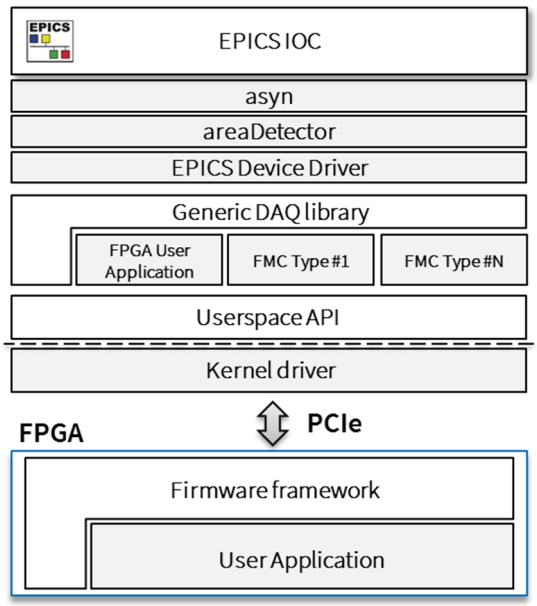

Figure 2: Software stacks of MicroTCA integration at ESS.

User Interfaces

ESS will use Control System Studio (CSS) [9] as main tool to create and execute user interfaces for the facility control system. ESS uses a site-specific version of CSS but also participates on the development of CSS project for the EPICS community. The latest releases of CSS include the Display Builder plugin, a new framework that replaces OPI Builder (BOY) and adds new features to CSS, while still keeping compatibility with BOY displays. The Figure 3 shows a standard user interface in CSS plotting the analog signals acquired with an IFC1410 and ADC3110 FMC.

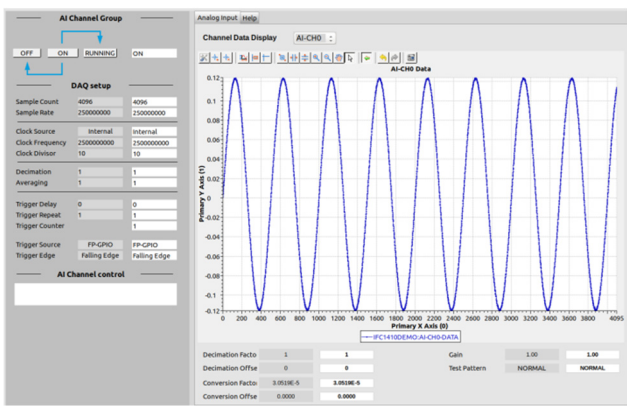

Figure 3: User interface running in CSS plotting analog signals acquired with IFC1410.

SUMMARY AND OUTLOOK

A description of the current status for integration of MicroTCA hardware into ESS Control System was presented. MicroTCA standard provides a powerful electronic platform for digital and analog signal processing requires careful design of all the layers that comprises the full system.

ICS will provide a reduced set of hardware with flexible configuration possibilities, to fulfill the requirements of applications demanding fast acquisition and processing for ESS facility. The IFC1410 and IFC1420 AMC carrier boards are the main components of this hardware set. These cards have both embedded CPU and FPGA, enabling this platform to be highly distributed. The EPICS driver support for these boards will be based on AreaDetector framework, to benefit from the EPICS community support and the plugins for data processing already available.

The main goal of the integration strategy for MicroTCA systems at ESS is to enable the use of this cutting edge technology along the entire facility while offering good support and long term maintenance with the available resources of ICS Division. There are already MicroTCA crates fully integrated with EPICS IOCs running and timing receivers triggering acquisitions, but it is still a under development project that will receive improvements and new features.

REFERENCES

[1] S. Farina, D. Piso, J. H. Lee, J. P. S. Martins, “MicroTCA Generic Data Acquisition Systems at ESS”, presented at ICALEPCS’17, Barcelona, Spain, October 2017, paper TUAPL01, this conference.

[2] M. Kirsch, “Kintex Ultrascale based MTCA digitizer cards, the SIS8300-KU and beyond”, lecture given at 5th MicroTCA workshop for industry and research, Hamburg, Germany, December 2016.

[3] J. Bovier, P. Cristin, and J. F. Gilot, “TOSCA II User’s Guide Architecture”, IOxOS Technologies internal manual, 2011.

[4] T. Korhonen, N. Claesson, J. H. Lee, ”EEE – The ESS EPICS Environment”, lecture given at Spring EPICS Collaboration Meeting, Lund, Sweden, May, 2016.

[5] T. Korhonen et all, “Status of the European Spallation Source Control System”, presented at *ICALEPCS’15*, Melbourne, Australia, October, 2015, paper FRB3O02.

[6] NDS3 Docs, `https://cosylab.github.io/nds3`

[7] Asyn: Asynchronous Driver Support for EPICS, `www.aps.anl.gov/epics/modules/soft/asyn`

[8] M. Rivers, “AreaDetector Status and Updates”, lecture given at EPICS Collaboration Meeting, Oak Ridge, USA, September, 2016.

[9] K. Kasemir, M. Grodowitz, A. Carpenter, C. Rosati, “CSS and Display Builder Update”, lecture given at Spring EPICS Meeting, Kurri, Japan, May, 2017.