QUEST FOR THE NEW STANDARD PSI IOC PLATFORM

D. Anicic, Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

Abstract

With its four accelerator facilities the Paul Scherrer Institut (PSI) has already several decades of control system Input Output Computer (IOC) experience. The technology is moving forward fast. The older hardware is becoming obsolete: it is slow, consumes too much power, does not match new computing, networking and bus technologies, and replacements can no longer be purchased as models have been discontinued. All this forces us to opt for a new "standard" IOC platform with increasing regularity. What used to be twenty years, became ten, and is now tending towards five years. Here we present past and possible future IOC platforms which we are investigating. Feedback from the conference would be highly appreciated.

PSI ACCELERATOR FACILITIES

The Paul Scherrer Institut (PSI) operates four accelerator facilities. The oldest of these, the High Intensity Proton Accelerator (HIPA) has already been in operation for more than four decades. It has undergone several beam intensity, hardware and software upgrade and rejuvenation processes. The second oldest, Swiss Light Source (SLS) has been operated for almost two decades and is going to be upgraded to SLS2 soon. Third, the Proton Scanning (PROSCAN) medical cancer-treatment facility, with its superconducting, compact, medical cyclotron (COMET) has been successfully treating patients for a decade, and has recently been extended with the third Gantry area. And the newest, Swiss Free Electron Laser (SwissFEL) is just coming in operation.

Although that was not the case in the past, nowadays all facilities are based on the same or similar hardware. There is a common VME bus system in use, with mostly MVME5100 VxWorks and IFC1210 RTLinux [1] Input Output Computers (IOCs). The common control system is EPICS.

MOTIVATION

Our existing processing platforms are getting gradually old and we also have to respond to availability and performance issues. For some of the used components manufacturers have already issued the end of life notices. Some CPU applications are already at the limits of their resource usage, like processing power and memory and bus throughput and the FPGA applications are exceeding available size.

All, especially older, facilities are constantly going through hardware rejuvenation, due to the lack of the spare parts, being upgraded to match higher processing and precision demands, or are simply been extended or adapted to the new user demands.

Probably the biggest demand is driven by the SLS2 upgrade. This should start in year 2020, and the New Processing Platform (NPP) should provide a solution until then.

The NPP project was therefore initiated in December 2017, and the working group began investigating use cases and possible implementations.

GOAL

The design and implementation of the NPP is seen as the necessity, and it has to provide the platform that can commonly be used in all facilities and should equally serve as base in all application areas. Because it is hardly possible and meaningful to use single identical platform for all use cases, it is desirable to provide for scalable solution. Direct consequence of the common platform is optimal usage of available resources. Sharing the manpower, know-how, development tools and expertise between different groups can greatly speed up our work and lower the costs.

USED PLATFORMS HISTORY

HIPA has initially been based on Digital Equipment's PDP-11 and later on hprt743 single board VME computers with HP-RT OS, but they have been replaced, since around year 2000, with MVME5100 on LynxOS. Custom home-made control system was used.

PROSCAN has always been running on MVME5100, initially on LynxOS, with custom home-made control system, same as HIPA.

HIPA and PROSCAN control systems have then been replaced with VxWorks running EPICS [2] as control system, in scope of control system standardisation project at PSI.

SLS is running EPICS since beginning, initially on MVME2300 VME computers. Later MVME5100 were added, and for a few demanding applications MVME6100, too.

For SwissFEL, new platform has been chosen. In cooperation with Swiss company IOxOS [3], the IFC1210 was introduced. Decision was to keep the VME bus, and new was that two FMC cards could be plugged in, and user code could be implemented in on-board FPGA. For many applications FPGA app using one or two FMC modules was enough, and VME bus was only used to supply electrical power.

At the same time it was decided to go for PREEMPT_RT patched Linux (RTLinux) as the operating system, and EPICS is still our control system.

Now the time has come for a new platform.

REQUIREMENTS

As with any other technology, the time comes to upgrade processing platform to the higher level. Several

Copyright © 2018 CC-BY-3.0 and by the respective authors

groups in PSI have been expressing wishes for faster, bigger, more up to date processing capabilities. That's how NPP came to life.

Different groups have already looked for what could come next. The preliminary task, to get the requirements showed that we would have to cover a wide range, from simple and cheap to advanced and expensive. The involved groups, Low Level Radio Frequency (LLRF), PROSCAN Patient Safety System (PaSS), HIPA diagnostics, Digital Beam Position Monitors (DBPM), and Digital Power Supply, together with general purpose controls system group with EPICS IOC produced the following boundary conditions:

- Low latency loops (ADC -> DAC below 1µs)
- General purpose I/O (20 or more)
- Supports VME form factor
- Has dedicated Programmable Logic (PL) memory
- FPGA has at least 200K Logic Elements
- At least 8 gigabit links
- Two Gigabit Ethernet links
- Long term availability
- Costs and effort savings
  - Can reuse existing FMC cards
  - Can reuse VME transition boards
  - Can reuse developed software and firmware

The CPU processing power should well exceed existing IFC1210 performance.

Although the VME was specified as boundary condition, there are many applications which can do without it. It is not to be strictly taken into account.

What was already visible from discussions is that almost everybody was thinking of Xilinx Zynq UltraScale+ MPSoC [4] chips as the base, and one way or the other reach the goal to satisfy own requirements.

Anyway, some other possibilities had to be investigated, too.

POSSIBLE WAYS TO GO

One very obvious and less-effort possibility is to use what is already available. Here we think of IOxOS IFC1410/IFC1420 and IFC1211 boards.

The more preferred would be the use of Silicon on Module (SoM). This solution provides for better scalability and diversity of bus or non-bus carrier boards or embedded systems.

Similar to SoM would be a possibility to design own boards based on UltraScale+ MPSoC. This would be the most expensive and the most resource exhaustive solution. But we would not have to start from scratch, we could base it on Digital Beam Position Monitor (DBPM3) for which the development has already started. This solution does not provide either VME or MTCA.4 [5] bus, at least in this moment.

Already Available Platforms

PSI has started collaboration with European Spallation Source (ESS) in Sweden as In-Kind contributor. ESS decision was to use MTCA.4 as crate standard. IOxOS has developed IFC1410/IFC1420 boards for them. PSI role is to deliver these boards, together with software and firmware packages to ESS. So we could use already working solution. It incorporates T2081 processor and Kintex UltraScale FPGA and provides, as IFC1210, two FMC slots. Initial considerations came to conclusion that switching from VME to MTCA.4 would be radical change. Also, full operating MTCA.4 crate is much more expensive than VME crate. Additionally it seems that MTCA.4 has quite small user and provider community, and most of all there is not much know-how and expertise in PSI. Change would also have consequence on existing front-end electronics, which would have to be redesigned and developed. So, the tendency was to stay with VME, and keep an eye on trends and possible growing availability of MTCA.4 processing and I/O boards.

IOxOS has also developed the IFC1211 board, which is the same as IFC1410, but in VME form factor. That could be very simple replacement for existing IFC1210, but offering only slightly better performance and bigger FPGA resources. This idea was discarded immediately, because many user groups would like to have more than that.

The main penalty by using IOxOS boards is that system firmware is combined with user firmware in the same FPGA. This greatly reduces the number of FPGA Logical elements available for user applications.

On the other hand, it is not Zynq UltraScale+ based, what many users like to have.

SoM Concept

This concept integrates MPSoC, memory and communication interfaces on a pre-engineered printed circuit board (PCB). It represents the complete computer with FPGA, lacking only outside world connectivity. It can be simply plugged into any (compatible) carrier board to give complete solution. It can also be used directly embedded into custom system.

For this we have investigated the Swiss company Enclustra [6] Mercury+ XU1 module mounted on Mercury+ PE1 base board.

The Zynq UltraScale+ seems to be quite powerful and available in different speed and size grades. We would probably go for a quad core Cortex-A53 APU (Application Processing Unit), with dual core Cortex-R53 RPU (Real-time Processing Unit) and 350K Logic elements FPGA.

The main advantage of Zynq UltraScale+ solution is that APU, RPU and FPGA have access to the same memory over AXI bus. Besides low latencies, and therefore higher access speeds (compared to PCIe bus on IFC boards), there is no need to copy data between processing units.

Copyright © 2018 CC-BY-3.0 and by the respective authors

The drawback of this solution is that it would be quite challenging to implement single slot VME or MTCA.4 boards, with proper cooling. Namely, the MPSoC would most probably need own heat dispenser, increasing the board thickness to double-slot.

SoC Solution

We have used the original Xilinx ZCU102 reference board implementation for testing. It was mainly used as test project for evaluating the Zynq UltraScale+ usability and performance.

There are some groups in PSI which are tending towards using SoC on their own custom designed boards. DBPM3 is one of them. Its main purpose was to implement beam position monitors for coming SLS2. It is still in design phase, and that gives us opportunity to rethink and change design to better fit the needs of other groups and applications, too.

Because SoC is going to be directly soldered to the carrier board, there should be no problem of having it as single slot implementation, with proper cooling, making it better choice than using SoM.

The drawback is that it would probably cost more, and the current design does not provide neither VME nor MTCA.4 bus connectivity and form factor.

Decision

The decision is still not made. It is hard to separate from VME, and give up on expertise and re-use of existing VME resources and also know that costs would be higher. On the other hand, we should not give up on possibility to stride the new ways, if indeed the time is ripe.

The final decision is to be made soon.

SOFTWARE

Regardless of the path we take, any new processing platform would also need an operating and control system.

It is quite clear and never taken in consideration to use anything else then EPICS as control system. RTLinux would most probably be used as operating system.

To get the boards booted one also needs the u-boot, board matching device tree, kernel and Linux root file system.

U-boot and Device Tree

U-boot is boot loader usually used on platforms other than Intel x86. It is mostly provided by the manufacturer, because it has to be adapted to match the board. For the same reason the device tree also comes from manufacturer.

For the IFC1410/IFC1420 boards, IOxOS provides the u-boot and device trees. Although it does the job as expected, from our experience it never does exactly what we want. So we have reconfigured and modified the u-boot to match our needs.

For Xilinx reference board we have used the Xilinx petalinux as base. It provides the u-boot and device tree, which we have also reconfigured and modified for our needs.

Enclustra provides the build-root based system. It comes with u-boot and device tree for their boards. Same as for the other boards we have reconfigured and modified the u-boot.

We have not done anything yet for the DBPM3 board, because there is still no prototype available. As being our own custom design, we will have to provide the u-boot and device tree ourselves. This is going to be a challenge, because we have not done such board bring-ups from scratch before. The main idea is to make it as much as possible the same as either Xilinx reference board or like Enclustra boards.

Kernel

The Linux kernel is free and open source code. It supports a lot of architectures and microprocessors. We have expected that there will be not much problems with it. That unfortunately did not happen to be the case.

The problem is that not all drivers are available in the original, mainline, “vanilla” kernel source code. The manufacturers are promising and working on driver integration into mainline kernels, but this happens quite slowly. For that reason we have to stick, at least for beginning, to the manufacturer provided sources.

The second reason is that we would like to use the RT patch, which is not supported by manufacturers. So, if there is no RT patch available for a particular kernel provided by manufacturer, we have to either integrate additional manufacturer drivers into some other kernel, or re-work the RT patches for the manufacturer provided kernel.

For IOxOS boards we have already had kernel v4.1.8-rt8 used for IFC1210 boards. The only additional drivers, the “Tosca” suite, were developed for us by an external company in cooperation with IOxOS. We have ported Tosca drivers to work with IFC1211, and IFC1410/IFC1420 boards, too.

Zynq UltraScale+ boards from Xilinx and Enclustra provide basically the same kernel v4.9.0. We have patched them with PREEMPT_RT 4.9.0-rt1. This seems to work, but we will probably go for newer v4.9 version, with newer RT patch. The biggest problem with this two kernels is that they are not quite the same. Although, we assume, the Enclustra has taken Xilinx kernel sources, they have modified them to work with own boards.

Our idea was to support only one kernel for all Zynq UltraScale+ boards. So we have taken the same kernel sources from the Yocto project with additional Xilinx meta-layer. Unfortunately, it turned out that this kernel, again, differs from both Xilinx and Enclustra kernels. The problem is mainly related to Ethernet driver, but there are also differences in other drivers (clock, flash subsystem and some others). Although we managed to get both platforms boot the same kernel, there are still some limitations. We still have to work on better solution. This step will be done as soon as we start working on kernel for our custom Zynq UltraScale+ board.

Copyright © 2018 CC-BY-3.0 and by the respective authors

Root File System

From different sources we have received information that using manufacturer root file system is never the best idea. Manufacturers usually provide just basic Linux operating system, just to get the boards booting, and have fast hands-on on using it. Although such basic system can also be reconfigured and rebuilt to support additional features, sooner or later there will be something missing. They almost never provide all possible Linux services, libraries or tools. It would be much better to build own Yocto or build-root Linux system.

For that reason we have chosen to use Yocto as our Linux root file system for Zynq UltraScale+ platforms. We use only Yocto generated root file system, and simply remove any kernel, u-boot and device tree files, which we build afterwards separately.

For the IFC1410/IFC1420 boards we are using original Freescale provided QorIQ-SDK-V2.0, which is anyhow Yocto based, and provides everything we have needed. But we also use only root files system, and we build u-boot, device tree and kernel separately.

Just to mention that our IOCs are always diskless machines, booted over network, with read-only root-over-nfs mounted root file system. Although Yocto has an option to produce the read-only system, from our experience it never does it properly. So, our root file systems are after installing always adapted manually to really work as read-only root-over-nfs.

EPICS

Most IOCs in all four facilities are running EPICS version 3.14.12. The new facilities, such as the upcoming SLS2 and possibly also new projects in other facilities, should introduce newer version of EPICS, most probably EPICS 7. We already have initial installations of EPICS versions 3.15.5 and 3.16.1 as fallback option for EPICS version 7.0.1 (former EPICS4), which is also available. The latest versions are compiling and are still being tested. Not all drivers have been installed yet. Of course, nothing was really tested in operation. The first tests will be done soon in SwissFEL, because there are some users willing to make the step forward.

All these EPICS versions and drivers are available for all used and possible upcoming platforms: MVME5100, IFC1210, IFC1211, IFC1410/IFC1420 and also for any future Zynq UltraScale+ based boards.

The only exception is that PV access (for EPICS 3.16.1 and 7.0.1) is not available for older VxWorks based platforms, but they will anyhow not be used in future.

PERFORMANCE MEASUREMENTS

Performance is not the main factor on choosing our new platform, but it is still significant one. So we have performed different “benchmarks”, just to compare what we have and what we can get. Here we present some benchmarking results as comparison of IFC1210 (dual-core), IFC1211 (octa-core, same as IFC1410/IFC1420) and Zynq Ultrascale (quad-core) boards. All of them are running on 1.2 GHz clock, but with different RT kernels. The performed tests actually measured single core performance. Also the new boards could also run on higher frequencies (IFC boards on 1.8 GHz, UltraScale+ on 1.5 GHz). Also to mention is that UltraScale+ boards also come with the additional single- or dual-core Real-Time CPU (RPU), which could be used to increase the RT performance.

All the tests have always been executed on idle machine, not to influence each other. The tests also do not, as any other benchmark, give really real overall application performance. That’s why we do not present here exact numbers. Approximate numbers will be given here as triplet, for {IFC1210, IFC1211, UltraScale+}.

Cyclictest

Cyclictest [7] is also influenced by used kernel. We have observed mostly identical times of in average 10 microsecond latencies on all three platforms. They slightly deviate in maximum latency, {95, 155, 55}.

iPerf

iPerf [8] measures Ethernet throughput. All three platforms have gigabit Ethernet, and all achieve 930 – 960 Mbit/s as server, and 900 – 925 Mbit/s as client. The CPU usage varies from 60 to 100 percent of one core.

Linpack

Linpack [9] demonstrates the floating point performance for basic mathematical operations. Approximate values are: {220, 520, 530} MFLOPS.

memcpy

Simple, C library ‘memcpy’ function test. It demonstrates not only CPU, but also RAM bus speed. The values are: {200, 500, 2000} MB/s.

Math Operations

Table 1 compares the speed (in microseconds per operation) of various single floating point operations for three different platforms.

Table 1: Single Floating Point Operational Speeds

Platform: / Operation	IFC1210 (µs/op)	IFC1211 (µs/op)	UltraScale+ (µs/op)
sqrt	0.335	0.215	0.050
pow	0.780	0.655	0.360
exp	0.345	0.325	0.150
log	0.765	0.325	0.150
atan/asin/acos	0.350	0.290	0.160
sin/cos	0.280	0.290	0.135

Copyright © 2018 CC-BY-3.0 and by the respective authors

CONCLUSIONS

Although the future hardware platform have not been chosen yet, it is quite clear that it will be Zynq UltraScale+ based. This decision and start of developments will start very soon.

We are ready from the software point of view. Of course there is still much work to do, but we are confident that software will not be the show stopper.

ACKNOWLEGDMENTS

This paper has been based on work of several groups in PSI. The author's contribution was in bringing up the RTLinux with root file system and drivers, and providing for some performance tests.

The author wishes to thank all groups taking part in specifying, analyzing and testing the different aspects of NPP, especially the NPP representatives, namely B. Kalantari (project leader), W. Koprek, K. Bitterli, E. Johansen, K. Ambrosch, D. Zimoch, R. Kalt, B. Ronner, B. Keil, G. Marinkovic, J. Raabe, M. Brückner, M.Eichin, and G. Theidel.

REFERENCES

[1] RTLinux, https://rt.wiki.kernel.org/.

[2] EPICS, http://epics.org/.

[3] IOxOS, https://www.ioxos.ch/.

[4] Xilinx Zynq UltraScale+ MPSoC, https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html

[5] MTCA.4, https://www.picmg.org/openstandards/microtca/.

[6] Enclustra, https://www.enclustra.com

[7] Cyclictest, https://wiki.linuxfoundation.org/realtime/documentation/howto/tools/cyclictest/start

[8] iPerf, https://iperf.fr/.

[9] Linpack, http://www.netlib.org/linpack/.

Copyright © 2018 CC-BY-3.0 and by the respective authors