Distributed and Parallel Real-time Control System Equipped FPGA-Zynq and EPICS Middleware

Sangil Lee silee7103@ibs.re.kr, C.W. Son scwook@ibs.re.kr, H.J. Jang lkcom@ibs.re.kr,
Rare Isotope Science Project, Institute for Basic Science, Daejeon, South Korea

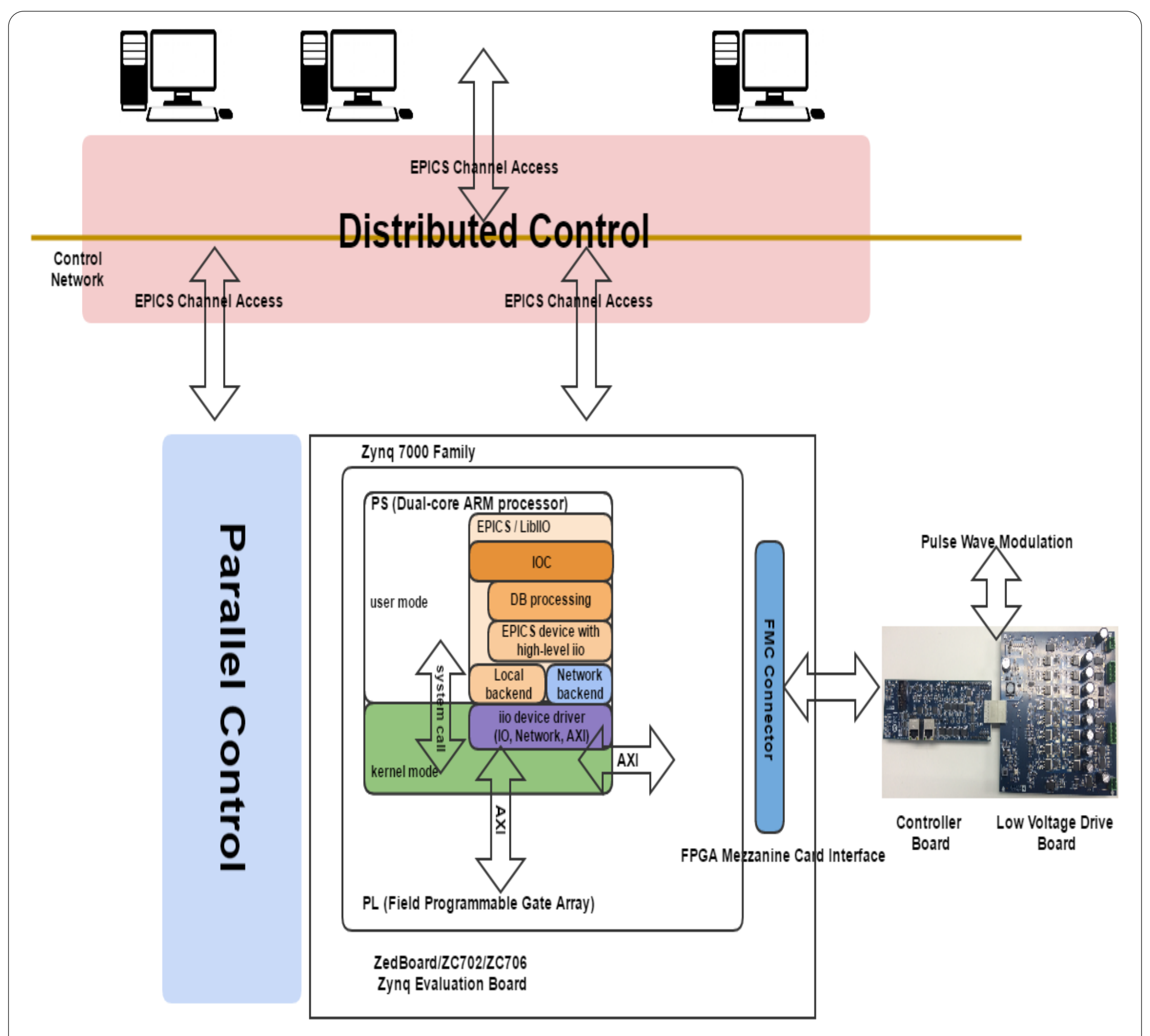
Abstract

Zynq series of Xilinx FPGA chips are divided into Processing System (PS) and Programmable Logic (PL), as a kind of SoC (System on Chip). PS with the dual-core ARM Cortex?A9 processor is performing the high-level control logic at run-time on linux operating system. PL with the low-level Field Programmable Gate Array (FPGA) built on high-performance, low-power, and high-k metal gate process technology is connecting with a lot of I/O peripherals for real-time control system. EPICS (Experimental Physics and Industrial Control System) is a set of open-source-based software tools which supports for the Ethernet-based middleware layer. In order to configure the environment of the distributed control system, EPICS middleware is equipped on the linux operating system of the Zynq PS. In addition, a lot of digital logic gates of the Zynq PL of FPGA-Zynq evaluation board (ZedBoard) are connected with I/O pins of the daughter board via FPGA Mezzanine Connector (FMC) of ZedBoard. An interface between the Zynq PS and PL is interconnected with AMBA4 AXI. For the organic connection both the PS and PL, it also used the linux device driver for AXI interface. This paper describes the content and configuration of the distributed and parallel real-time control system applying FPGA-Zynq and EPICS middleware.

RAON Introduction

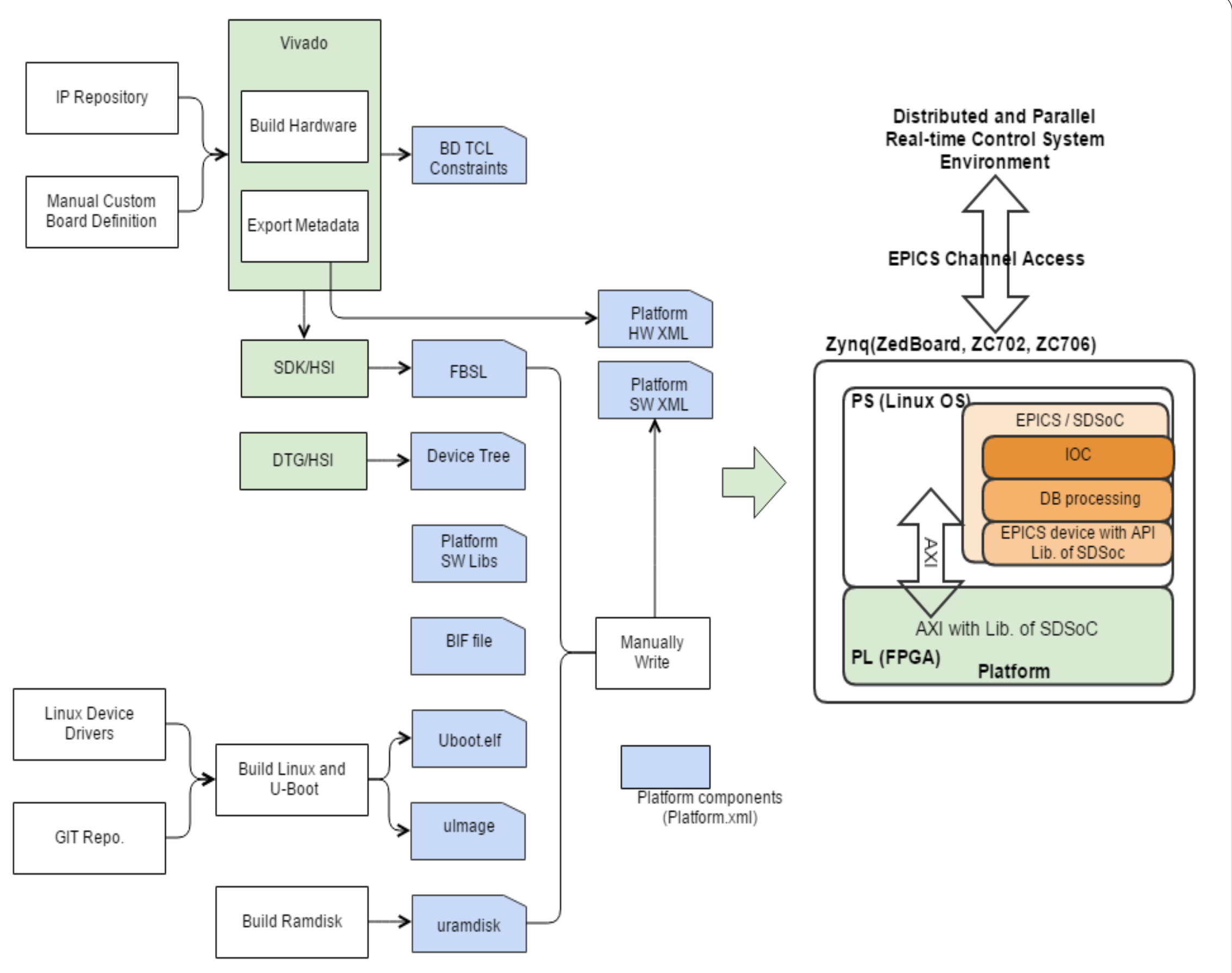
The RAON[1] is a new heavy ion accelerator under construction in South Korea, which is to produce a variety of stable ion and rare isotope beams to support various researches for the basic science and applied research applications. To produce the isotopes to fulfill the requirements we have planed the several modes of operation scheme which require fine-tuned synchronous controls, asynchronous controls, or both among the accelerator complexes.

Environment for Distributed and Parallel Control System



- Distributed Control Environment
  - Linux OS on Zynq PS (ARM Processor)
    - ∗ ARM Cross Compile Tool Chain (arm-linuxgnueabihf)
    - ∗ Linux Kernel Source (Linaro)[2]
    - ∗ Bootloader (BOOT.BIN: FSBL.elf, U-Boot.elf, uImage, Zynq.bif, User.bit) [3]
    - ∗ Board Support Package (Linux Device Tree)
    - ∗ Root File System (Busybox)[4]
  - Linux Device Driver : Interface between PS and PL is through the AXI of AMBA
  - EPICS Framework [5]
    - ∗ Base R3.14.15.2
    - ∗ Control System Studio (CSS)
  - Libiio Library of Analog Device[6, 7]
- Parallel Processing Control Envrionment (ZC706)
  - Zynq PL (FPGA)
  - FPGA VerilogHDL by Vivado (Used the HDL code of Analog Devices)
  - Controller and Low Voltage Drive Board of Analog Devices
  - Controller communicates with ZC706 via FMC Connector
  - FMC Connection: XADC, Digital I/O, 2xGB Ethernet
  - XADC and digital I/O signals to LVDB
  - PWM Signal Generation through MOSFET Gate Drivers
  - External Power: 12 ∼24 DC to LVDB

Software Defined System on Chip (SDSoC) Environment



The SDSoC[8] environment is an Eclipse-based Integrated Development Environment (IDE) for implementing embedded systems using the zynq programmable SoC platform. The SDSoC environment includes support for the ZC702, ZC706, MicroZed, ZedBoard and Zybo development boards featuring the Zynq-7000 AP SoC.

Using SDSoC platform can implement as follows:

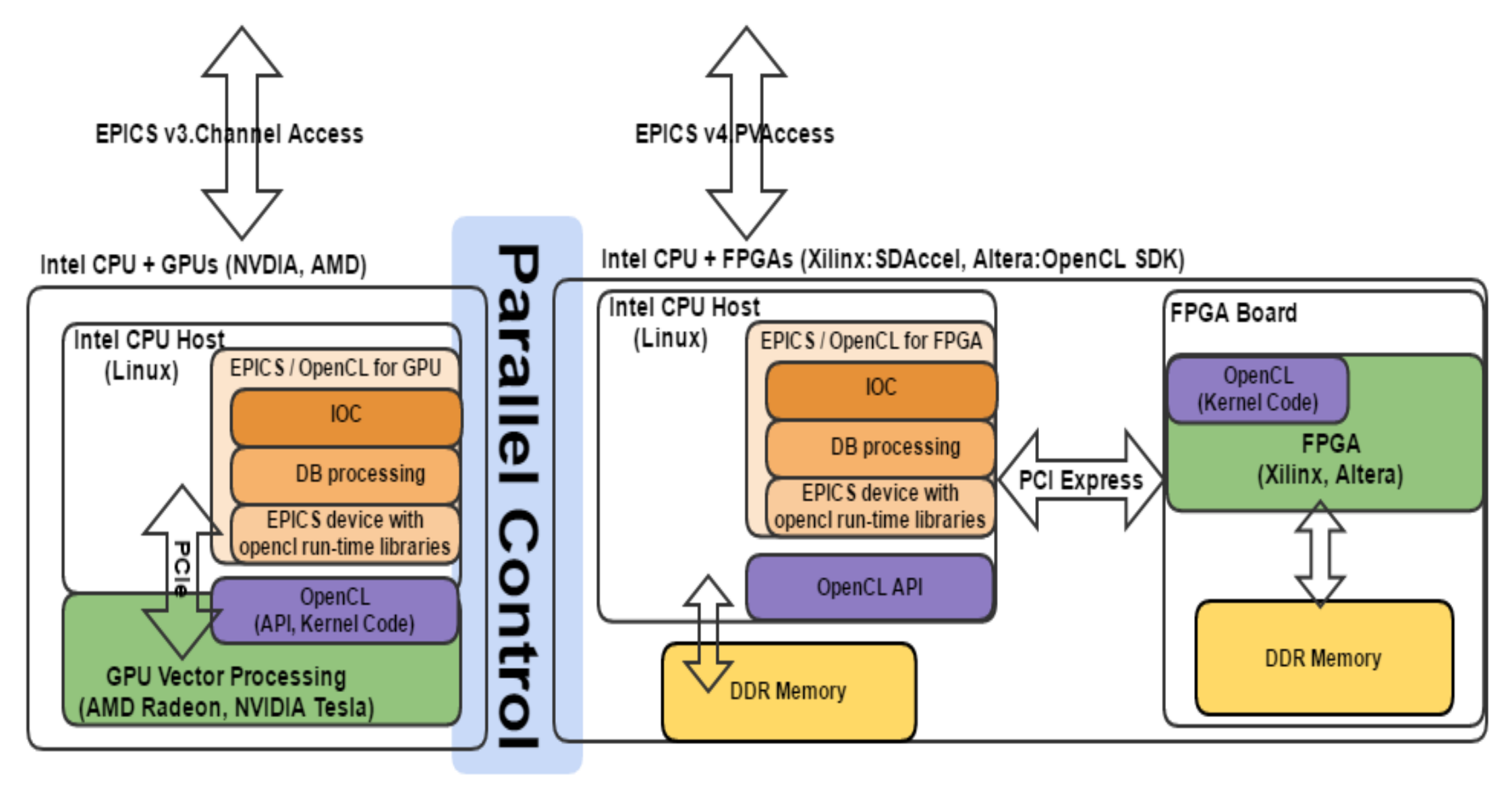
- Hardware and Software Architecture
- Application Context including Linux OS
- Bootloader
- Root File System
- High Level Libraries and Applications for Interface with PL

In addition, EPICS Framework on Linux Built using SDSoC :

- EPICS Base R3.14.15.2
- EPICS Device Suport Routine with High Level Library Interface which developed and deployed with SDSoC

Future Plan

In order to connect the distributed processing system and parallel processing system more effectively, it should provide a unified interface layer for the specific various parallel processing hardwares. For the purpose of developing the unified parallel processing interface layer it can be used Open Computer Lanaguage (OpenCL)[9].



OpenCL[10, 11] is a framework or programming model for writing programs that execute across heterogeneous platforms consisting of CPUs, GPUs, DSPs, FPGAs and other processors or hardware accelerators and was developed by the Khronos group.

- Hardwares Support for OpenCL for Parallel Control Environment
  - GPU: AMD Radeon or NVIDIA GeForce
  - FPGA: Xilinx - SDAccel or Altera - OpenCL SDK
- Abstract layer using OpenCL for Parallel Processing
  - OpenCL (1.2 or 2.0)
  - Host Code: OpenCL API-Serial, Sequence code
  - Kernel Code: Parallel Code for GPUs, FPGAs
  - EPICS Device Support IOC code + Host Code

Adopting EPICS v4, it will be interfaced large amounts of data and complex data to be processed in the parallel processing of a variety of experimental devices.

- EPICS Version 4[12] for Distributed Control Environment
  - Normative Types - Collection of Structured Data Types [13]
  - pvCommon, pvData, pvAccess, pvaSrv Modules

Conclusion

Big scientific experiment facilities such as accelerators, nuclear fusion, and telescope are required by the unity of the heterogenous distributed control system and the fast response according to its characteristic. These requirements for ***the distributed and parallel environments*** must be satisfied at the same time. If the distributed control system using EPICS makes a connection with the high speed parallel processing of FPGA, it is possible to improve the performance and efficiency of the control system. ***Zynq SoC*** can be considered as an ideal device to satisfy with ***the distributed and high-speed parallel control system*** at the same time. More and more, the control environment of devices used in the big science experiment of the future will be used the distributed processing environment combined with the parallel processing environment. Therefore, it expect to be increased the requirements for the unified software abstraction layer to operate the heterogenous parallel proccessing hardwares."FPGA-based heterogeneous system (CPU + FPGA) using the OpenCL standard has a significant time-to-market advantage compared to traditional FPGA development using lower level hardware description languages (HDLs) such as Verilog or VHDL"[11]. As a result, utilizing the ***OpenCL standard*** on the heterogeneous parallel processing hardwares may offer significantly ***higher performance*** and ***the unified abstract layer*** at much lower power.

Acknowledgement

This work is supported by the Rare Isotope Science Project funded by Ministry of Science, ICT and Future Planning **(MSIP)** and National Research Foundation **(NRF)** of KOREA.

References

[1] Y. K. Kwon, *et. al*,"Status of Rare Isotope Science Project in Korea", Few-Body Syst 54, 961-966, (2013).
[2] Linaro Document website, https://en.wikipedia.org/wiki/Linaro
[3] U-Boot Document website, http://www.denx.de/wiki/U-Boot
[4] Busybox Document website, http://www.busybox.net/
[5] EPICS website, http://www.aps.anl.gov/epics/
[6] Analog Devices website, http://www.analog.com
[7] Industrial I/O Document website, https://wiki.analog.com/resources/tools-software/linux-software/libiio
[8] SDSoC Document, http://www.xilinx.com/support/documentation/sw_manuals/xilinx2016_ sdsoc-platforms-and-libraries.pdf
[9] OpenCL, https://www.khronos.org/opencl/
[10] OpenCL Definition, https://en.wikipedia.org/wiki/OpenCL
[11] OpenCL on Altera, https://www.altera.com/en_US/pdfs/literature/wp/w 01173-opencl.pdf
[12] EPICS Version 4, http://epics-pvdata.sourceforge.net/
[13] EPICS Version 4 Normative Types, http://epics-pvdata.sourceforge.net/docbuild/normativeTypesCPP/tip/ documentation/ntCPP.html