ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP071

APPLICATION DEVELOPMENT ON CPCI-S.0 HARDWARE AT PSI

I. Johnson*, R. Biffiger, D. Felici, W. Koprek, R. Rybaniec, B. Stef and G. Theidel
Paul Scherrer Institut (PSI), CH-5232 Villigen PSI, Switzerland

Abstract

A Hardware and Software Toolbox is being created to accelerate the engineering of electronic components for large facility upgrades at the Paul Scherrer Institute. This Toolbox consists of modular hardware following the CPCI-S.0 standard, project base designs, firmware libraries and software packages. The goal is to simplify and accelerate developments with a set of compatible electronics, starting foundations, tools and modules.

INTRODUCTION

Typical applications at accelerator facilities require network controlled, low–latency (1 µs) systems. The PICMG Compact PCI Serial standard, CPCI-S.0 [1], offers an ideal platform to interconnect these real–time systems. Large facility upgrades at the PSI, SLS 2.0 and HIPA, are moving to this flexible, crate-compatible standard. This paper provides an overview of the hardware, embedded system, software and firmware developments that enable the efficient creation of applications.

CPCI-S HARDWARE

A crate with a CPCI-S.0 compatible backplane has been designed by PSI and is being manufactured by ELMA, Fig. 1. The backplane of this crate interconnects a system slot, 8 peripheral slots, rear slots and a power backplane which houses a system monitoring card, utility cards and power supplies. Multi–gigabit capable traces in both a full–mesh topology and star topology from the system slot provide routes for high–speed communication between the front slots. Ethernet and computer–compatible buses (PCIe, USB and SATA) are in the CPCI-S.0 pinout specification to enable standard–protocol data–transfer between the front cards. These multi–gigabit links can alternative be utilized for custom protocols. There is a rear slot behind each of the 8 peripheral slots. Multi–gigabit serial links, i2c and general purpose IO create a versatile connection between the front and rear cards. Rear slots extend the real–estate for analog and digital signal processing cards.

Adhering to CPCI-S.0 standard makes the crate also compatible with both commercial of–the–shelf cards. Companies like EKF offer a wide range of CPCI-S boards. Two commonly utilized commercial cards from EKF are the SC5 Festival (a High Performance CPU Board) and the SD1-DISCO (SATA Drive Carrier Board).

More than a handful of custom PSI cards have also already been produced:

Figure 1: Picture of CPSI-S crate with 5 custom (UFC) cards inserted into the left most peripherally slots, a System Monitoring card just right of the middle and 3 redundant power supplies on the right.

- CPSI_UFC: Universal FMC+ Carrier with a Zynq™ UltraScale+ MPSoC, HPC FMC+ slot, 48 multi–gigabit links and plenty of high speed GPIOs, Fig. 2.
- CPSI_CIO: Communication IO card with a Zynq™ UltraScale+ MPSoC for multi–gigabit, crate–external communication with two SFP+ and two QSFP+ slots, Fig. 3.
- CPSI_RTM_DAC: Rear Digital to Analog Converter card which contains two 16 bit 500 Msps DAC, two SFPs and an expansion connector.
- CPSI_RTM_FIO: Fixed Input/Output board with one SFP+, two 500 MHz SMA and eight LEMO connectors.
- CPSI_CM1: Crate control and monitoring card.
- Power Load Board: Card for measuring the crate's power capabilities and stability.
- Backplane Loopback Board: Card for testing all the data connection on the backplane (Full–Mesh and Star).

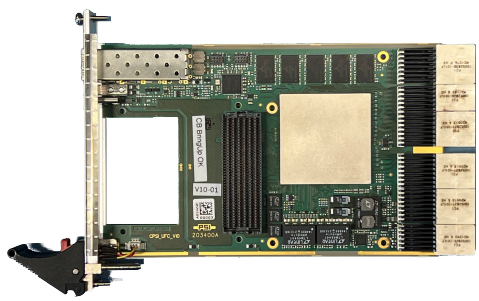

Figure 2: Picture of the Universal FMC+ Carrier (CPSI_UFC).

Some of these board are already being utilized in applications, see Section APPLICATIONS.

EMBEDDED ENVIRONMENT

Every CPSI-S custom front card contains a PSoC 62 micro–controller from Infineon. The software running on this micro–controller is responsible for the start–up and shut-

* ian.johnson@psi.ch

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP071

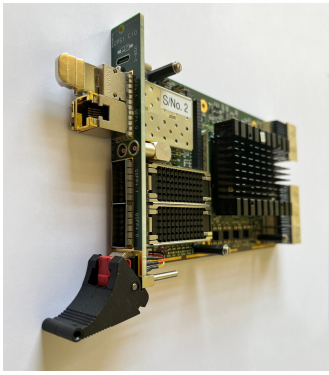

Figure 3: Picture of the Communication IO (CPSI_CIO).

down of the front, rear and potential mezzanine cards. It gathers identification and version information from these cards and provides it to the control interface (EPICS IOC) and the application. Applications can perform controlled start–ups by first verifying io–compatibility with the plugged rear and mezzanines cards, and then initiate the powering of the boards. During operation the micro–controller supervises various system–monitoring sensors (temperatures, currents, voltages) on the front, rear and mezzanine cards. This system information is also forwarded to the control interface.

Multiprocessor System on Chip

A AMD Zynq™ UltraScale+ MPSoC device is at the center of the main FPGA cards (UFC and CIO). MPSoC devices simplify real–time control and data acquisition systems by integrating an array of systems components into a single device. For example the ”EG” devices contain a Quad-core Arm Cortex-A53 Application Processing Unit (APU), a Dual-core Arm Cortex-R5F real-time processing unit (RPU), a Mali-400MP2 graphics processing unit (GPU) and Programmable Logic (PL). Linux, EPICS soft IOCs and custom software run directly on the APU, while the real-time signal processing is offloaded onto the RPU and PL.

Base designs provide advanced starting points for applications on the MPSoC devices. They are an environment of basic ready–to–use system and functional building blocks organized into two main layers: one for the Processing System (PS) and one for the Programmable Logic (PL). The former is a collection of the software packages that run within an Operating System. The latter, lower layer consists of a seed Vivado project, library of firmware components and ready–to–use modules.

Boot Sequence The Zynq Ultrascale+ MPSoC is booted in multi–stage sequence. First of all the Configuration Security Unit (CSU) checks which configuration option is selected. For all our hardware platforms we use booting from an SD card. The SD card contains a boot.bin file which is a collection of Power Management Unit (PMU) firmware, First Stage Boot Loader (FSBL), u–boot and default PL configuration. The CSU first loads the PMU firwmare, next the FSBL, then the default PL configuration and finally the FSBL loads the u-boot (stage 1 in Fig. 4). U-boot offers multiple possibilities on how the Linux is booted. In our applications Linux is downloaded from a boot server connected by Ethernet. After the Linux kernel is downloaded the u-boot passes control to Linux (stage 2 in Fig. 4). During booting process Linux mounts the root file system also stored on a boot server and application specific network drives. Once Linux is running the boot process is finished. The default PL configuration allows accessing from Linux peripheral devices located on the board and connections to the user pins of PL. For our systems we use Linux compiled with Yocto 4.

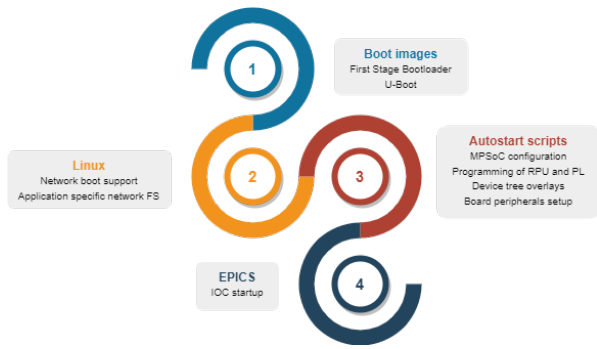

Figure 4: A diagram of the system initialization.

Application Initialization After booting the OS a collection of reusable bash Auto-Start scripts configure the PL and RPU, load device tree overlays, setup on–board peripherals and finally start EPICS soft IOCs. These scripts are written in a generic, reusable fashion to support multiple applications and board specific features. For example, device–tree–overlay scripts, that simplify the creation of high-level connections, are tailored to the component (GPIOs, I2C, Custom–Modules, etc.) not the specific instance. A description of the PL instantiated component (like: AXI address, numbers, names) is passed to the script that creates the logical–name binding for the user–space software drivers and tools. There are also scripts to configure the PL, RPU and chips on the board (like clock chips over i2c). Application specific Auto–Start scripts call these generic scripts to prepare the board for a specific application, stage 3 in Fig. 4. EPICS [2, 3] IOCs are started, stage 4 in Fig. 4, once the board has been configured for the application.

Software Modular software layers glue all levels of the system together and simplify the development and maintenance. On a higher level, maintained C libraries and python packages facilitate the application development process by providing top-level interfaces to the low level components. Python is installed on the Linux of the Zynq UltraScale+. A ZynqMon Python package was developed to create an object–oriented approach to the communication with the firmware components in PL and hardware peripherals. This package offers easily access to the AXI bus, I2C, UART,

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP071

SPI, physical memory, peripheral chips and PL IPs found in the firmware library. ZynqMon greatly simplifies the prototyping of hardware, firmware and applications.

A Report–Creator Python package has also been developed to automated the testing and report generation. Executables, found in this package, automatically call hardware–support and test functions from files in a specified directory. Test functions have access to the hardware and report objects through global instances that are created within the hardware–support files. As an example, in the CPSI_UFC board testing suite there are global instances that connect to the Linux command–line, serial terminal of the PSoC and ZynqMon. Test functions collect data, process information and verify measurements. Results are then passed to the global report object that combines the sections, tables, plots and results into pdf and html documents. These documents are stored on both the board's SD card and in the PSI inventory system.

Firmware A library of firmware components is available and maintained on GitHub [4] to accelerate the application development in the PL. These components are grouped into signal processing elements and commonly utilized blocks. The former, psi_fix library, is an array of fixed–point arithmetic entities, like filters and modulation cores. The latter, psi_common library, contains communication interfaces (AXI, I2C, SPI, …) and basic data processing elements like FIFOs and clock domain crossings. Code level instantiation of these library components aid the development of both the application specific modules (Fig. 5) and reusable IP–blocks.

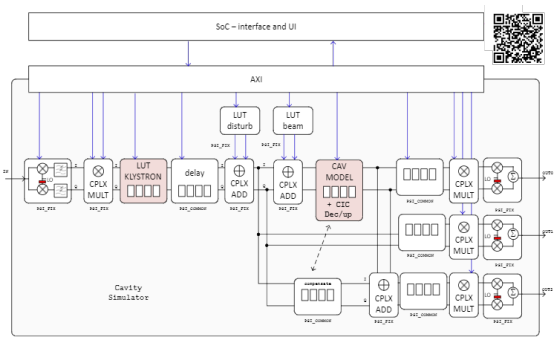

Figure 5: A synoptic of a RF cavity simulation created mostly with entities from the PSI Firmware library [4]. All components except the two application specific (LUT Klystron and CAV Model, in red) are reusable modules from the library. The QR code is a link to the GitHub psi-fix repository.

Firmware version identification, clock measure and a multi–gigabit transceivers test module are examples of reusable IP modules that can be simply instantiated and connected to the AXI bus, see Fig. 6.

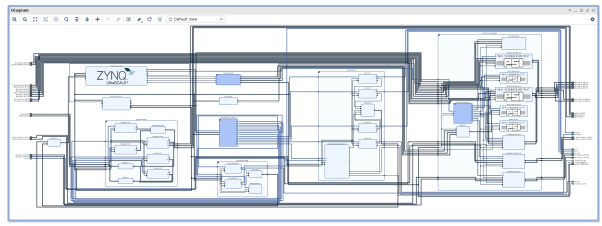

Figure 6: Vivado block design illustrating the instantiation of both AMD/Xilinx and custom IP blocks.

APPLICATIONS

The new processing platform is versatile and suitable for several applications around accelerator technologies, Fig. 7. As a generic electronic data acquisition CPSI_UFC (Fig. 2) and CPSI_CIO with custom boards or COTS cards (E.g. FMC) connect fast control and acquisition channels to the machine network. These cards are well suited for real–time applications like: Longitudinal Loss Monitors (LLM), Generic Fast Scope, Filling Pattern Monitors (FPM), Parametric Current Transformers (PCT), Beam Dump Controllers (BDC), Integrated Current Transformers (ICT) and Low–Level Radio Frequency (LLRF) systems.

Figure 7: Diagram of accelerator applications that utilized CPCI-S.0 Hardware at PSI.

As an example, the high performance Generic Fast Scope is composed of a GSps ADC FMC card mounted on a CPSI_UFC board. This specific configuration and base design are also the starting point for other accelerator application, like the FPM.

One of the main application for CPSI_CIO cards is the synchronous distribution of the SLS 2.0 accelerator timing (Timing Event Generator, Timing Fan–out and Timing Event Receiver). The timing signal will be fanned–out via the QSFP ports and received through the SFP+ port of CPSI_CIO boards (Fig. 3). The latter receiving cards contain the embedded event receiver and serve applications.

Furthermore, capabilities of the platform have been demonstrated by satisfying the complex requirements of the LLRF control system for the SLS 2.0 project. A CPSI_UFC, 250 MSps ADC and a 500 MHz DAC were combined in a CPSI-S crate to construct the complete LLRF signal processing system that ensures real–time cavity control, in phase

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP071

and amplitude. This system is currently in operation and successfully conditioning RF cavities.

CONCLUSION

A CPCI-S.0 toolbox is being constructed to increase the speed and efficiency of applications developed for the large facilities and accelerators at PSI, like HIPA and SLS 2.0. It is an ecosystem of compatible and ready–to–use components that creates the complete environment for electronics measurement and control systems. Standard interfaces (PCIe, USB, Sata, i2c, multigigabit serial links) on the backplane combined with the state–of–the–art electronics (AMD Zynq™ UltraScale+ MPSoC, custom electronics and COT cards) make the CSPI-S.0 a versatility hardware platform with extreme potential. A common network booted OS, base designs, software packages, firmware libraries and reusable IP modules provide engineers and scientist a quick start and allow them to focus on application specifics. Furthermore, a handful of initial accelerator applications have already demonstrated the capabilities and strengths of the toolbox.

ACKNOWLEDGEMENTS

Many thanks to the support from the PSI GFA/AEK department and NUM/LTP Electronics team. Also special thanks to D. Anicic, R. Ditter, M. Gloor, F. Hämmerli, B. Kalantari & L. Moser for their great contributions.

REFERENCES

[1] B. Kalantari, E. Johansen, W. Koprek, P. Pollet, and G. Theidel, "CompactPCI-Serial Hardware Toolbox for SLS 2.0", presented at the ICALEPCS'21, Shanghai, China, Oct. 2021, paper TUAL01, unpublished.

[2] EPICS, http://www.aps.anl.gov/epics/index.php

[3] T. Celcer *et al.*, "The SLS 2.0 Beamline Control System Upgrade Strategy", presented at ICALEPCS 2023, Cape Town, South Africa, 2023, paper TUPDP105, this conference.

[4] Firmware libraries, https://github.com/paulscherrerinstitute/psi_fpga_all