Gateware/Software Architecture

Greg Daniluk (BE-CEM)

May 2023

Abstract: This document describes the gateware and software architecture of DI/OT (Distributed I/O Tier)-based systems. The goal is to facilitate DI/OT adoption and avoid duplication of similar developments by offering a basic set of services. The architecture ensures coexistence of centrally provided monitoring/management/communication services with the application-specific gateware and low-level, real-time software.

Versions

Version	Date	Comment
v0.1	11-04-2023	General architecture description - first draft for comments
v0.2	15-05-2023	Draft after internal CEM-EDL review

Contents

1 Introduction

1.1 Distributed I/O Tier hardware kit

Distributed I/O Tier (DI/OT) is a generic, reusable hardware platform for custom electronics at CERN. It is based on a 3U crate compatible with industrial standard CompactPCI-Serial (CPCI-S.0) and comes in two flavors: radiation-tolerant and non-radiation-tolerant.

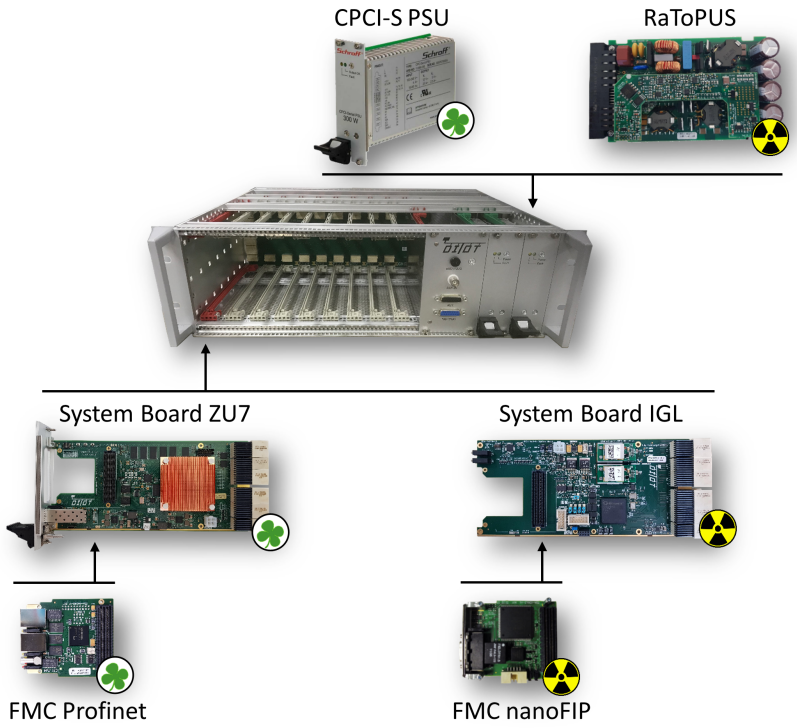

Figure 1: Distributed I/O Tier hardware kit

The full DI/OT ecosystem depicted in fig. 1 consists of the following application-independent modules:

- **DI/OT 3U crate** - common design for both radiation-exposed and radiation-free applications, electrically compatible with the CPCI-S.0 standard, hosts a fully passive DI/OT backplane;
- **Fan tray** - common for both radiation-exposed and radiation-free applications, connects to the DI/OT crate front panel and supports the PMBus diagnostics interface;
- **RaToPUS** - radiation-tolerant, 100W power supply that produces 12V and 5V for the DI/OT crate and implements a diagnostic interface;
- **CPCI-S power supply** - non-radiation-tolerant, 300W power supply that produces 12V and 5V for DI/OT and supports the PMBus diagnostics interface;

- **System Board IGL** - radiation-tolerant crate controller featuring a Microchip Igloo2 Flash-based FPGA and a Low-Pin-Count FMC connector to host a communication mezzanine;
- **System Board ZU7** - non-radiation-tolerant crate controller featuring a Xilinx Zynq Ultrascale+ MPSoC, 1/10G Ethernet, White Rabbit support and a Low-Pin-Count FMC;
- **FMC nanoFIP** - radiation-tolerant FMC mezzanine implementing a WorldFIP agent;
- **FMC Profinet** - non-radiation-tolerant FMC mezzanine implementing a Profinet IO device;

Each DI/OT-based application is made of the 3U crate and a selection of modules (power supply, System Board, communication mezzanine) depending on whether radiation-tolerance is a requirement or not (Fig. 2). A DI/OT instance is further customised by adding application-specific Peripheral Boards and FPGA gateware on the System Board.

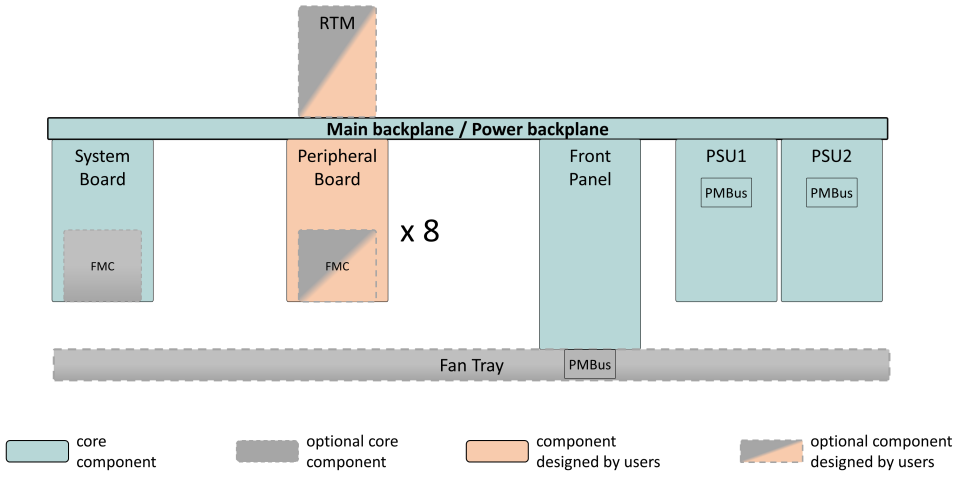

Figure 2: Components of a DI/OT-based system

1.2 System on a Chip

System on a Chip (SoC) is a general term that describes an Integrated Circuit (IC) that combines various components and peripherals, that historically would need to be placed as separate ICs on a PCB together with the main processing unit. A SoC can be for example: multiple ARM processors and peripherals like I2C, SPI, GPU all integrated in a single IC package; or multiple ARM processors and FPGA fabric integrated in a single IC package. Xilinx/AMD Zynq7 and Zynq Ultrascale+ are examples of SoC devices.

The DI/OT hardware kit benefits from the flexibility of SoC architecture by making use of Xilinx Ultrascale+ MPSoC (main processing device of the System Board ZU7) and supporting HydRA (an FPGA-based radiation-tolerant System-on-Chip) for System Board IGL.

1.3 Supported connection architectures

Regardless of the DI/OT variant (radiation-tolerant or non-radiation-tolerant) the connection architecture to interface with the rest of the control system is very similar for both:

1. DI/OT in radiation-exposed areas is always an extension of a traditional FEC (Fig. 3). It never interfaces directly with the Technical Network. It is controlled from a process running on a FEC over a radiation-tolerant fieldbus (e.g. WorldFIP). All application data as well as monitoring is passed over that single, radiation-tolerant fieldbus.

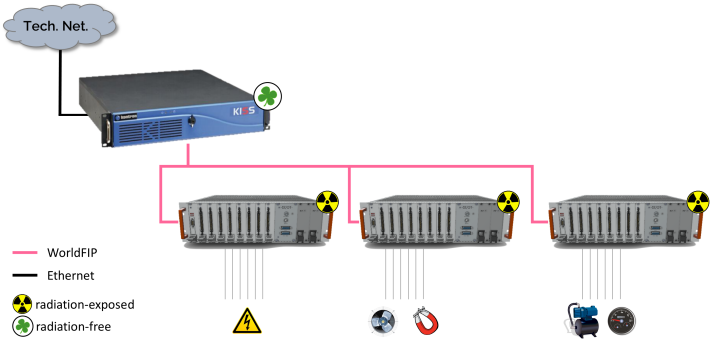

Figure 3: DI/OT in radiation-exposed area

2. DI/OT in radiation-safe areas is also an extension of a traditional FEC (fig. 4) but in this case, it is a standalone device connected directly to the Technical Network. It is managed and exchanges application data with a FEC using a standardised Ethernet protocol. If needed, this architecture enables the DI/OT crate also to communicate with other CERN services directly through Ethernet to provide monitoring and configuration data. It can run user-defined processes directly in the ARM processor of the Zynq Ultrascale+ MPSoC and interface over the Technical Network with user-defined services running on a FEC.

   In case there is a specific requirement to have a deterministic and/or high-throughput communication link between a DI/OT and a FEC, a dedicated link from a System Board can be established with the help of an FMC with SFP ports. This dedicated link could be for example a 10Gbps Ethernet or PCI-Express-over-fiber.

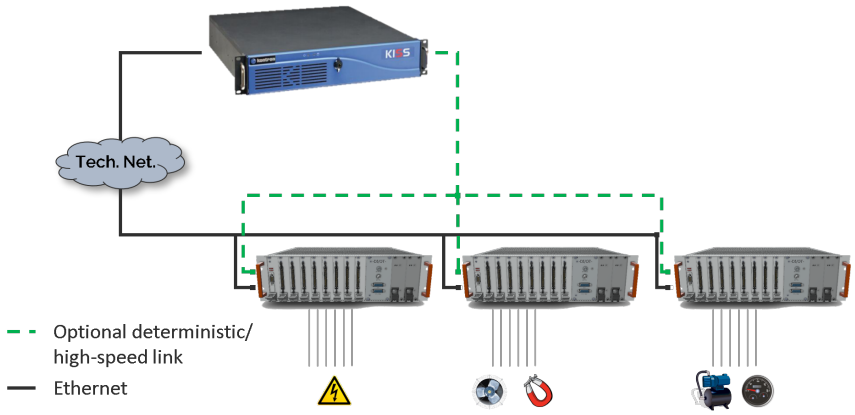

Figure 4: DI/OT in radiation-safe area

2 General SoC architecture

The goal of the common SoC architecture for DI/OT-based systems is to facilitate DI/OT adoption and avoid duplication of independent gateware and software developments by offering a basic set of services. These services are then implemented as part of the reference designs for both System Boards. They are meant to be a starting point for every future DI/OT-based application. These reference designs can be used ”as is” since they provide a set of basic functionalities to interact with a DI/OT crate, or can be further expanded/customised with application-specific software and FPGA design.

Regardless of whether we consider a radiation-tolerant (based on Igloo2) or a non-radiation-tolerant (based on Zynq US+) variant of the System Board, the general architecture of the reference design can be split into: *Processing System* and *Programmable Logic* (figure 5). The former uses processors (ARM or soft-core RISC-V) available on both platforms to run various software services. The latter consists of HDL IP cores to implement interfaces and deterministic data processing for the *Processing System*.

2.1 Processing System

The *Processing System* is split into two parts: *Management CPU* and *Application CPU*, with a possibility to exchange data between them. The *Management CPU* is responsible for all ”housekeeping” tasks. It runs DI/OT reference software implementing the following services:

- **Fieldbus communication** with a Front-End Computer through Ethernet (non-rad-tol System Board) or WorldFIP (rad-tol System Board).
- **Monitoring** of the whole crate including: detection of Peripheral Boards, Power Supplies, Fantray, firmware versions running on all these modules, temperatures, current consumption, etc.
- **Remote firmware/gateware update** for the Management CPU (fail-safe), Application CPU, Programmable Logic, Peripheral Boards, DI/OT Fantray.
- Fail-safe, **remote power cycle** in case of unrecoverable freeze

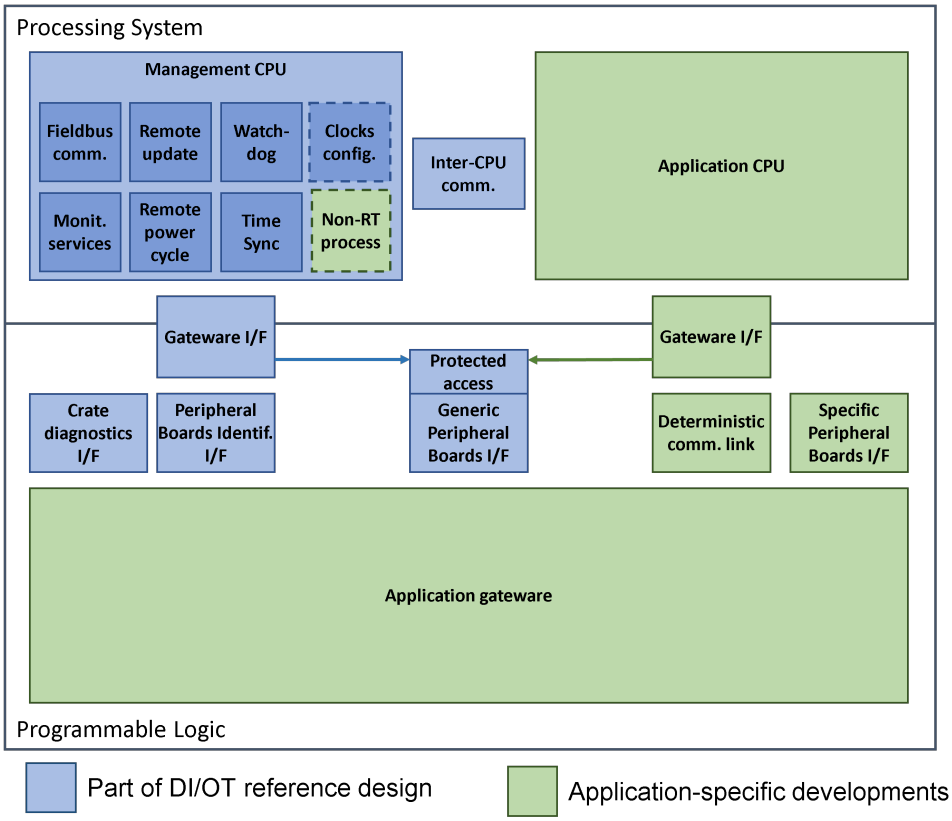

Figure 5: General DI/OT SoC architecture

- **Watchdog** to prevent freezes of the whole crate
- **Time synchronisation** to the reference provided by a Front-End Computer and distributed over the Fieldbus link[1].
- (non-rad-tol System Board only) **Clocks configuration** to allow generation of user-defined clocks for Peripheral Boards and FMC mezzanine.
- (non-rad-tol System Board only) Running **non-real-time user processes** inside the Linux operating system

The *Application CPU* (or CPUs) is fully available to the user to run real-time tasks coded in bare-metal C. The architecture allows also to exchange data between the fieldbus link, *Crate management CPU* and *Application CPU* (*Inter-CPU communication* in figure 5).

The *Processing System* can communicate with IP cores implemented inside the *Programmable Logic* (*Gateware I/F* in figure 5). However, to ensure there are no access conflicts that could compromise the real-time operation of the *Application CPU*, each of the processing units (*Management CPU*, *Application CPU*) has its own instance of the gateware interface/bus.

[1]In the case of WorldFIP, a simple synchronisation is ensured by WorldFIP Master by starting each cycle on reception of an externally provided trigger signal. No additional time-of-day information is distributed over the bus.

Management Gateware Interface allows the *Management CPU* to access IP cores inside the *Programmable Logic* that are provided together with the DI/OT reference gateware. These cores interface with various crate diagnostic peripherals (temperature, current sensors, PMBus, etc.) as well as the I^2C bus used to read the identification EEPROMs of DI/OT boards.

Application Gateware Interface allows the *Application CPU* to access all application-specific IP cores implemented inside the *Programmable Logic*. For example application-specific deterministic communication link, interfaces with Peripheral Boards or any other data processing cores.

The reference gateware provides also a generic interface to access Peripheral Boards (*Generic Peripheral Boards I/F* in figure 5). This IP core can be accessed by both *Application CPU* and *Management CPU*. However, if the *Application CPU* is used to run real-time tasks, it always has priority to access this interface over the *Management CPU*. In fact, the latter is blocked from accessing *Generic Peripheral I/F* during normal operation, and should first request access to the *Application CPU* through the inter processor communication scheme (*Inter-CPU comm.* in figure 5). In the case where an application doesn't use the *Application CPU* but relies only on running non-real-time processes on the *Management CPU*, that processor has exclusive access to the Peripheral Boards interface.

2.2 SoC architecture of the non-rad-tol DI/OT

The internal architecture of the Xilinx Zynq Ultrascale+ MPSoC (fig. 6) naturally maps onto the general DI/OT SoC architecture described in previous sections.

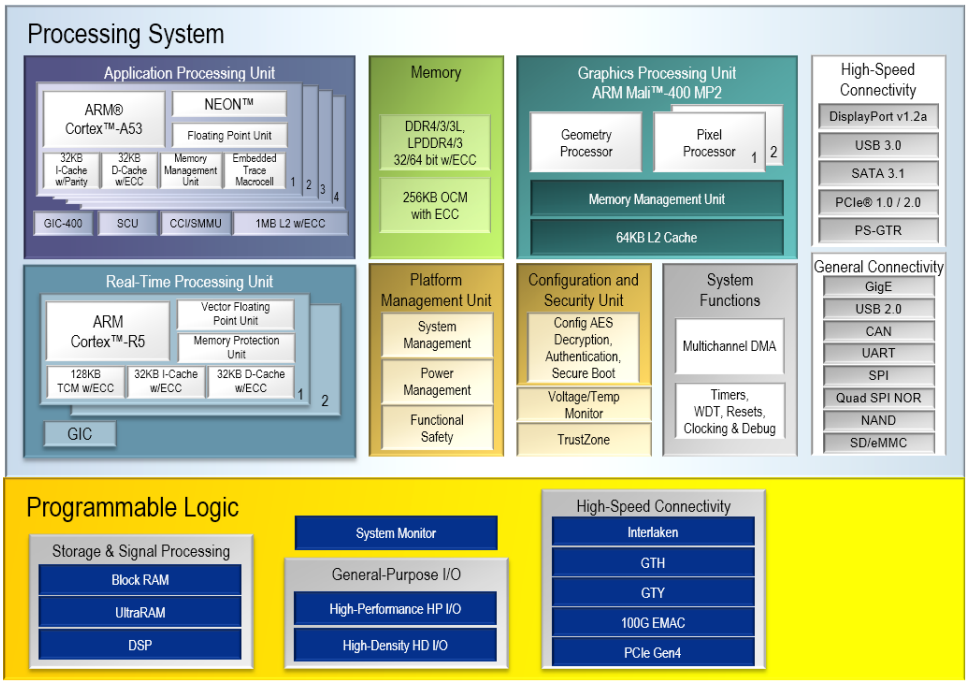

Figure 6: Xilinx Zynq Ultrascale+ internal architecture
Source: *https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html*

The following table summarises the mapping of components from the general DI/OT SoC

architecture to internal modules of the Xilinx Zynq Ultrascale+ architecture.

DI/OT SoC architecture component	Zynq US+ component	Description
Management CPU	Two Cortex-A53 cores running Linux operating system	
Application CPU	Two Cortex-A53 cores or dual-core Cortex-R5	
Fieldbus communication	**GEM** - PS Gigabit Ethernet Controller with standardised DI/OT Ethernet protocol	**??**
Monitoring	User-space processes and dedicated drivers in Linux on the Management CPU	
Remote update	User-space processes to program Application CPU, send bitstream and trgger update of Peripheral Board FPGA. Xilinx Virtual Cable tool for Jtag-over-Ethernet to Peripheral Boards.	
Remote power cycle	User-space tool in Linux on the Management CPU + fail-safe mechanism based on pattern recognition by an Ethernet PHY chip	**??**
Watchdog	**SWDT** - System Watchdog Timers of Zynq US+ controlled by a user-space process in Linux on the Management CPU	**??**
Time synchronisation	**NTP** (Network Time Protocol) or **PTP** (Precision Time Protocol) daemon on the Management CPU	**??**
Clocks configuration	Programming user-defined configuration to the clock generator chip at boot of the Management CPU	**??**
Non-real-time processes	User-provided binaries/python scripts running on Linux (Management CPU) without any real-time guarantee	
Management Gateware Interface	AXI HPM0 - AXI Master interface of Zynq PS	
Application Gateware interface	AXI HPM1 - AXI Master interface of Zynq PS	

2.3 SoC architecture of the rad-tol DI/OT

The same general DI/OT SoC architecture can also be applied to the HydRA-based reference design for the radiation-tolerant System Board (fig. 7).

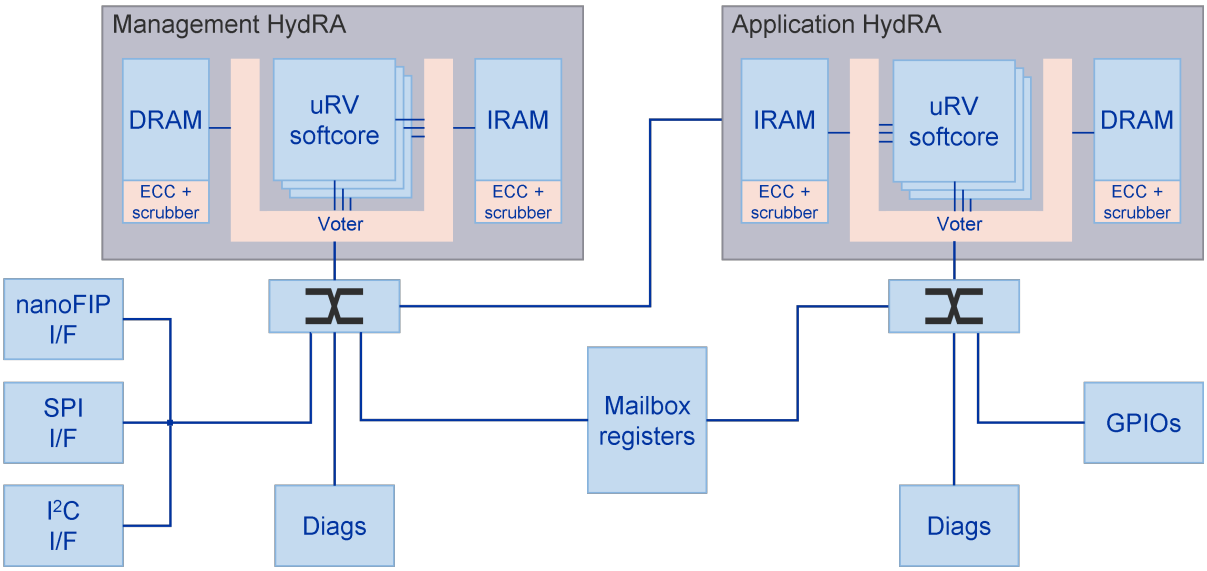

Figure 7: Rad-tol System Board reference FPGA design

HydRA is a radiation-tolerant implementation of the Micro RISC-V (*uRV*[2]) soft-core processor. A single HydRA core is made of a triplicated *uRV*, an instruction memory (IRAM) and a data memory (DRAM). Both memories are implemented using Block RAMs of the System Board's Igloo2 FPGA. Since Block RAM cannot be easily triplicated like flip-flops inside an FPGA, an Error Correction Code (ECC) with a scrubber is implemented to ensure the integrity of the data stored in both IRAM and DRAM. In addition, a combinatorial voter arbitrates all accesses from the triplicated *uRV* to the IRAM, DRAM and Wishbone bus (which is used for accessing other FPGA peripherals).

The HydRA-based reference design for the radiation-tolerant System Board includes two instances of HydRA: the Management HydRA and the Application HydRA. Each of these can access other FPGA IP Cores through a Wishbone bus. The Management HydRA has access through a Wishbone crossbar to:

- **nanoFIP I/F** - IP Core implementing communication with nanoFIP FMC mezzanine
- **SPI I/F** - SPI master used for accessing an external SPI flash
- **I²C** - I²C master used for accessing diagnostics and monitoring data (on-board sensors of the rad-tol System Board, sensors of DI/OT crate power supplies, identification EEPROM of all Peripheral Boards)
- **Diags** - Diagnostics IP Core of HydRA providing statistics about detected and corrected bit errors, *uRV* processor recoveries, etc.

[2] `https://ohwr.org/project/urv-core/wikis`

- **IRAM** of Application Hydra - to load/update an application program

The Application HydRA has access through a Wishbone crossbar to:

- **GPIOs** - IP Core implementing communication with Peripheral Boards over the DI/OT backplane
- **Diags** - Diagnostics IP Core of Application HydRA providing statistics about detected and corrected bit errors, *uRV* processor recoveries, etc.

Additionally, these two HydRAs can exchange data with each other through the Mailbox registers IP Core.

The following table summarises the mapping of components from the general DI/OT SoC architecture to internal modules of the HydRA-based reference design:

DI/OT SoC architecture component	**Rad-tol System Board ref design component**	**Description**
Management CPU	Management HydRA	
Application CPU	Application HydRA	
Fieldbus communication	nanoFIP interface IPcore accessible over Wishbone bus	**??**
Monitoring	Part of bare-metal C task on Management Hydra and accessing I^2C peripherals through $I^2Cinterface$ IPcore	
Remote update	JTAG bit-banged by nanoFIP FMC	
Remote power cycle	Part of bare-metal C task on Management Hydra	**??**
Watchdog	Reset signal generated by nanoFIP FMC at reception of predefined WorldFIP frame	**??**
Time synchronisation	Only synchronisation through WorldFIP Master starting each cycle on the reception of externally provided trigger signal.	**??**
Management Gateware Interface	Wishbone Master of Management Hydra	
Application Gateware interface	Wishbone Master of Application Hydra	

A Appendix

A.1 DI/OT backplane topology

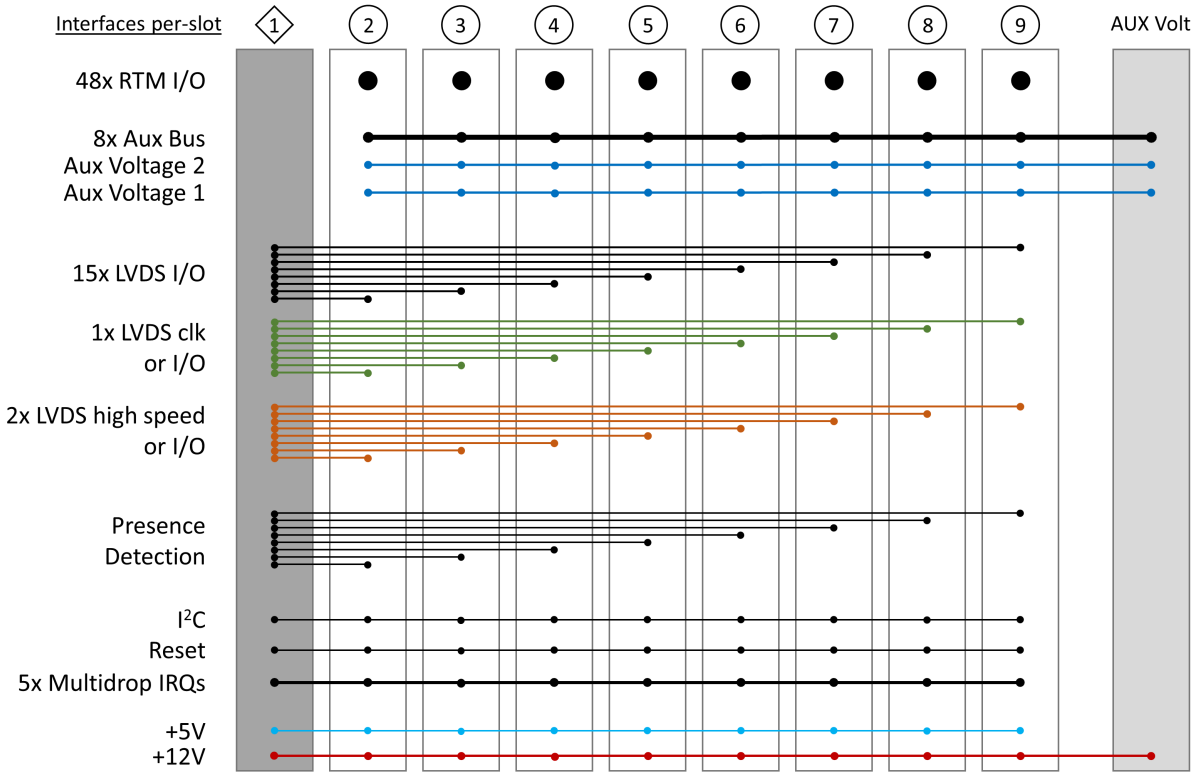

Figure 8: DI/OT backplane topology

A.2 Peripheral Boards: electrical requirements

- *RST_N* (pin H2 of backplane connector P1) is a reset signal, active low, pulled-up on the System Board to 3V3 (**TODO:** on rad-tol SB it's pulled-up to PPERIPH).
- *SHARED_BUS_0* (pin I2 of backplane connector P1) is PGOOD Open-Drain signal from the System Board. Peripheral boards should pull it up to STBY 5V (delivered from pin B1 of the P1 backplane connector) and use as enable pin for local DC/DC converters. The pull-up to STBY 5V ensures that the Peripheral Board is enabled (powered) if not deliberately shut down by the System Board.
- *SERVMOD* (pin K2 of P1) - shall be pulled down to GND through a 200Ohm resistor (**TODO:** can we use a larger resistor value? Motor driver board has 2k2). System Board reads it to identify in which slots there is a Peripheral Board plugged. System Board also drives this signal to 3V3 (**TODO:** PPERIPH on rad-tol SB) to enable a "Service mode". In service mode an identification EEPROM is connected to the shared I2C bus, and several P1 I/Os are used as JTAG lines. **TODO:** add here an example schematics.

A.3 Differences between DI/OT and traditional CERN FronEnd Computers (FECs)

The introduction of a new hardware platform can raise questions about the differences with respect to traditional FrontEnd Computers (FECs) used at CERN - Industrial PCs, VME, MTCA, PXIe. The main characteristics of the DI/OT platform that make it different from FECs are:

- Radiation-tolerance
- More PCB space for application-specific electronics than in the usual FMC-kit - the DI/OT crate is able to host up to 8 Peripheral Boards of 220mm x 100mm.
- Can host hundreds of digital/analog I/Os and connect all of them with low and deterministic latency to the FPGA of the System Board.
- Can serve simple applications without the need to implement complex communication techniques between System and Peripheral Boards.
- Provides dedicated CPUs to run bare-metal software in a fully deterministic way.
- The above-mentioned 4 features make it a perfect solution for real-time, low-latency processing of many I/O channels

References

[1] F.Vaga, ”FPGA Interface Design Documentation”, `https://ohwr.org/project/fpga-dev-id/wikis`

[2] ”24AA025E48 EEPROM irradiation report”, `https://edms.cern.ch/document/2684144/1`

[3] ”NanoFIP JTAG Master Controller feature”, `https://ohwr.org/project/nanofip/wikis/WP10`

[4] ”NanoFIP functional specification”, `https://edms.cern.ch/document/1107940/6`