PAUL SCHERRER INSTITUT PSI

Hardware Concept for SLS 2 LLRF Systems

LLRF 2017

Kristian Ambrosch, Ernst Johansen, Mario Jurcevic, and Roger Kalt

Abstract

For the upgrade of the Swiss Light Source in 2023, we propose the concept of a new hardware platform for the LLRF system that is based on our experience with the SwissFEL LLRF. For SwissFEL the concept of having both CPU and FPGA on the same board allowed for a compact design. Here the FPGA was used for the fast pre-processing of the signals and the CPU was used both, for the real-time application running the control loop and the EPICS-based Accelerator Control System (ACS). However, the latency of the PCI-Express connection between FPGA and CPU would be a bottle neck for fast feedback loops if the feedback should run on the CPU. Furthermore, sharing resources between EPICS and real-time applications complicates the software design. Using plain Linux as a common operating system for EPICS and real-time applications makes it difficult to predict the latency of interrupt handling for real-time applications. Thus, we present a new concept using a Multi-Processor-System-On-Chip (MPSoC) using the novel Xilinx Ultrascale+. On such a platform EPICS applications can run on a Quad-Core Cortex-A53 ARM CPU, while providing a separate Dual-Core Cortex-R5 for real-time applications. This enables us to run real-time applications on a bare metal system where the latency of each interrupt can be determined, while a Linux operating system for EPICS applications runs on a separate CPU. The interconnection between the CPUs and the programmable logic is provided by an AXI interface, which enables a low latency transmission of small data blocks. First ideas and concepts of a LLRF system based on such a platform will be given.

Current LLRF Platform – IFC 1210

For the SwissFEL we are using the IFC 1210 VME carrier board for FMC mezzanine cards from IOxOS.

Here, a P2020 PowerPC is used in combination with a Virtex-6 FPGA. The connection between CPU and FPGA is established via PCI-Express. The VME bus is connected via an internal VME to PCI-Express bridge that is hosted on the FPGA, which is supported by the TOSCA design environment.

Lessons learned – Software

During the design phase of this board the emphasis was put on hardware layout, while the software implementation was not the main focus. However, it turned out that the way the VME bus was connected, the access to the VME controller required additional effort on the driver side to keep the access thread safe and therefore multi-threading compatible.

Apparently our Contrls Section did not have the resources to design such kind of Linux kernel driver. Therefore, we had to contract Denx to develop this driver for us, which resulted in a good, but also very costly solution.

Lessons learned – Communication

While the connection between CPU and FPGA using PCI-Express might be a very powerful solution for the transmission of big data blocks, it comes at the disadvantage of a high latency. In LLRF applications the measurement data exchanged between FPGA and CPU is far smaller than the graphics applications PCI-Express was designed for.

Our measurements showed that we have 50µs-300µs minimum latency for any data transmission between FPGA and CPU. For the SwissFEL the Master board requires 500µs and the Slave board 850µs for the transmission of a single pulse measurement into the user space memory of the LLRF RT-App. While this was sufficient for our 100 Hz application, it leaved room for improvement.

Another aspect that could be optimized was the access to the Rear Transition Module (RTM). Here the connection was established using a separate Spartan-6 FPGA, to cope with the limited PIN count of the chosen Virtex-6. This resulted in additional latency for the access to the RTM, which was good enough for the original purpose, but made this board less versatile when using it for different applications in other facilities.

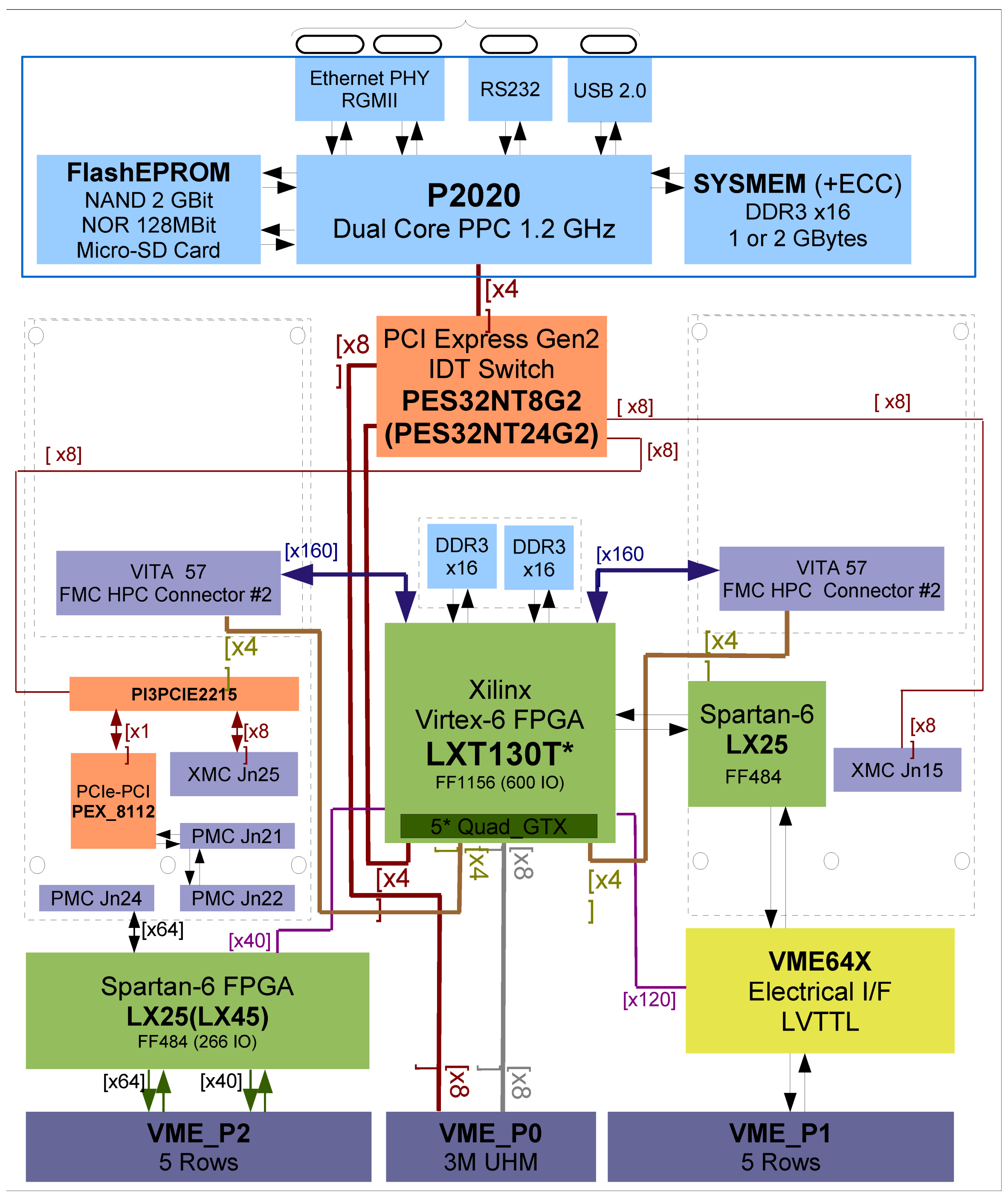

Figure 1: IFC 1210 Block Diagram
Source: IOxOS

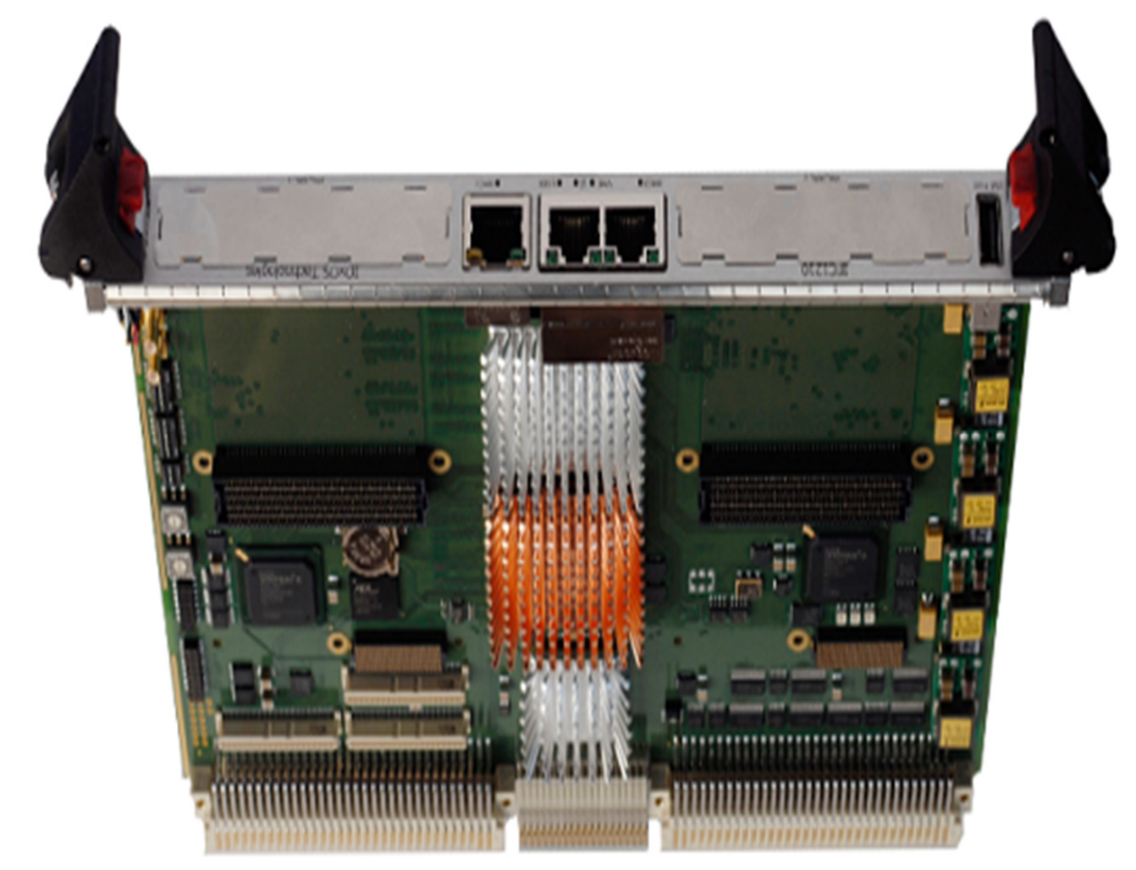

Figure 2: IFC 1210 Board

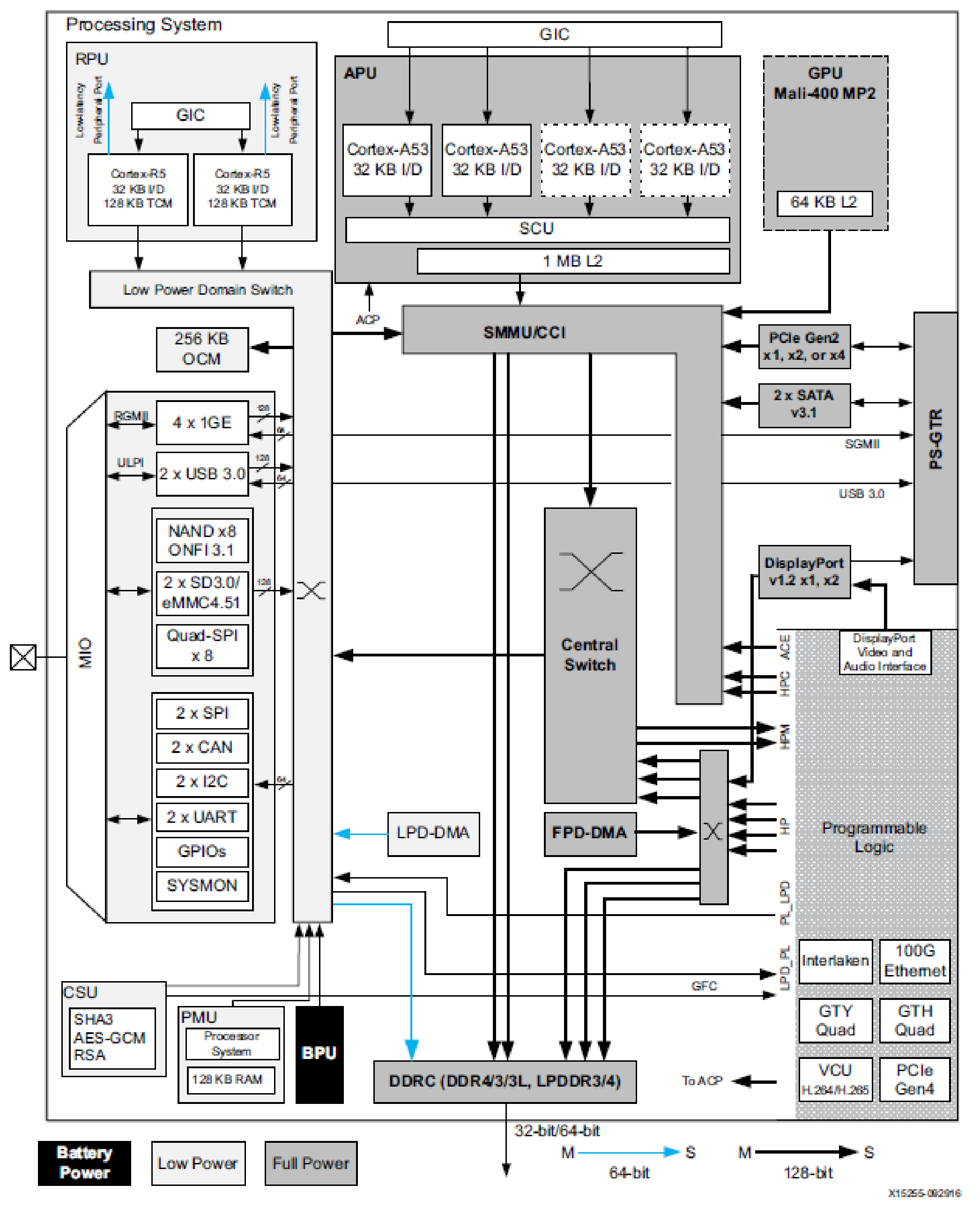

Figure 3: Zynq Ultrasacle+ Architecture
Source: Xilinx

A Single Chip Solution – Xilinx Zynq Ultrascale+

For the SLS 2 upgrade in 2023, we propose to reduce the complexity of our board design, the software design and the overall hardware costs, by using Zynq Ultrascale+ Multi-Processor SoC from Xilinx.

AXI-Bus

Here, the CPUs and the Programmable Logic (PL) are connected via an AXI bus. This enables a low latency data transmission even for single byte transmissions.

Application Processing Unit

Running the EPICS control system on a dedicated Cortex A-53 CPU allows us to decouple the real-time application from the non-real-time part. Here, we are evaluating Peta-Linux, which offers full Xilinx support, and Yocto, which is widely used in industrial applications.

Real-Time Processing Unit

A dedicated Cortex A5 CPU will be used for the LLRF RT-App. In difference to the IFC board, where the RT-App was running on custom build Linux, where a dedicated core was reserved to ensure real-time behaviour, the decoupling of control systems and RT-App allows for a bare metal implementation. Hence, we expect a significantly reduced jitter for interrupt handling and therefore a much better real-time behavior than in our Linux application.

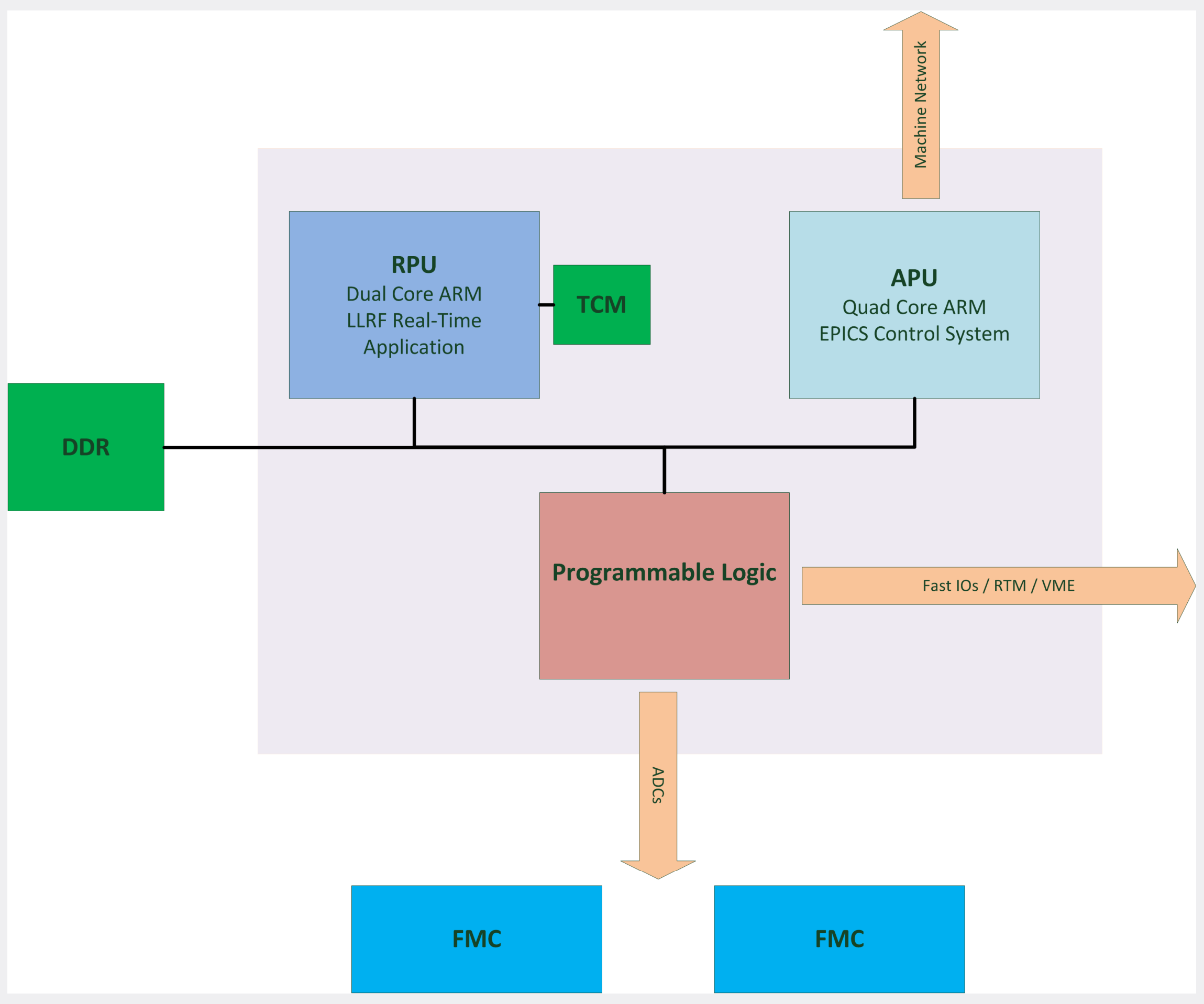

Figure 4: Block diagram of our processing system

Tightly Coupled Memory (TCM)

Two separate 128kB memory blocks allow us to separate the memory access for real-time data transmission between RPU and PL. The communication with the control system can be established using the DDR3 controller, which is shared by both CPUs.

Is VME still a viable solution?

An extensive analysis of the SwissFEL and SLS electronics made apparent that in-crate communication is a thing of the past. While we where filling 21-slot crates in 2001, our SwissFEL LLRF system consists of three boards that communicate via Ethernet.

Contrary to this, the data exchanged by Fast Feedback Networks is growing significantly. Here, the connected systems are spread over the whole machine, making it impossible to host them in a single crate. At PSI, our approach is to connect these systems via SFP+, using either PCI-Express with bus mastering (BPMs) or MGTs (SwissFEL Virtual RF Stations).

This allows us to focus on crate compatibility of all PSI facilities, e.g. using an optional AXI-VME bridge, while providing high-troughput solutions at the same time.

Just a LLRF System?

At PSI we are putting a high effort on increasing our efficiency while maintaining multiple large research facilities in parallel. While it proved to be cost efficient to adapt the board designs to the application when encountering appropriate piece counts, the effort for the board support was often disregarded. In the past this lead to the deployment of different chips for different application.

However, the effort to build and maintain a custom Linux, the driver development for different platforms as well as interfacing CPU and ADCs on the FPGA requires a significant amount of work.

Therefore, we are evaluating the deployment of the Xilinx Zynq Ultrascale+ as a new standard platform for our real-time applications at PSI.

Conclusion / Outlook

First evaluations of the Zynq Ultrascale+ showed promising results regarding the PL-CPU communication latency, throughput and ease of implementation.

In the upcoming months we plan to perform further benchmarking of this platform, including the software delivered from Xilinx, to ensure that is suitable for all our applications that can be covered using an FMC carrier board.

If these benchmarks meet our expectations, we expect first prototypes to be available in 2020 and deployment of the new LLRF system in 2023.

Abbreviations:
MPSoC (Multi Processor System on Chip) / AXI (Advanced eXtensible Interface) / APU (Application Processing Unit) / RPU (Real-time Processing Unit) / TCM (Tightly Coupled Memory) / VME (Versa Module Eurocard) / PL (Programable Logic)

Revision 1, AK84, 17.10.2017, LLRF17