Wir schaffen Wissen – heute für morgen

Paul Scherrer Institut

Timo Korhonen, Damir Anicic, Guido Janser, Mario Jurcevic, Ernst Johansen, Babak Kalantari, Roger Kalt, Michael Laznovsky, Lionel Schebacher, Thomas Schilcher, Dirk Zimoch

Modern Technology in Disguise

Contents

- Introduction
- Technology choices
- Firmware development
- Applications
- Conclusions

Introduction

- VME-based standard solutions at PSI with traditional architecture
- VME crates with CPU (PowerPC) boards
- VME cards as needed: ADC, DAC, Motion control, scalers, etc.
- Bus throughput limited, shared resource does not scale
- Limited possibilities to extend this traditional architecture

- We need to introduce new technology but cannot abandon our existing systems
- Hundreds of deployed systems that run stable and fine

- What do we need, really?
- Compatibility with the existing infrastructure
- Data bandwidth, scalability, connectivity
- FPGA for interfacing and processing
- Maximal reuse of developments (limited manpower)

Technology choices: requirements

- We need a board with **industry standard interfaces.**
  - We can benefit from commercially available boards.
  - Custom developments are potentially interesting for wider use.
- **FPGA** is a requirement for modern systems
  - Custom circuit interfacing
  - Algorithm acceleration, fast real-time processing
- We also need a board with **easy programmer accessibility**
  - FPGA programming is time-consuming and the number of FPGA programmers is limited
  - Development on a regular CPU is easier for most people
  - Software and firmware work can be decoupled for faster development
- We need more **data bandwidth.**
  - Adopting a new standard (e.g. μTCA) would need a whole new infrastructure. Do we really need to change?
  - Add parallel data paths to increase throughput

Technology choices

- Industry standard interfaces
  - FMC (VITA 57.1.) for FPGA applications
    - Standard interfaces and form factor for direct FPGA connection
    - Several applications can be covered with one base infrastructure
  - XMC and PMC standards have a lot of available cards
- Easy programmer accessibility: a CPU
  - A regular CPU that can work stand-alone
  - Moderate performance and power consumption
- Data bandwidth: use parallel data paths
  - Use VME P0 connector to provide a data path parallel to VME
  - PCI express (Gen 2) is the most common interconnect architecture today
- Our resolution:
  - Start a co-development with the company IOxOS SA (www.ioxos.ch)
    - They proposed an architecture that suits our needs
  - Intensive co-operation in hardware, software and firmware development

Technology choices

CPU Freescale P2020 PowerPC

- 1.2 GHz dual core

Not the fastest CPU on the market, but (probably) best performance/watt

- Long-term availability

FPGA Virtex-6 LX130T

(or other variants with the same footprint)

- TOSCA-II firmware package: Network on-chip

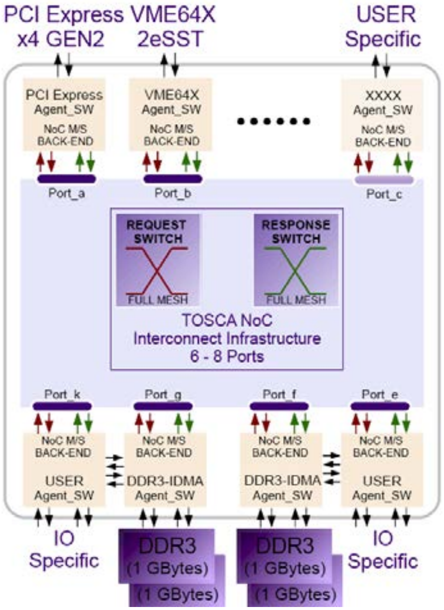
512 MB shared memory

PCI EXPRESS®

The most common interconnect today

Memory mapped. Scalable (no bus bottleneck)

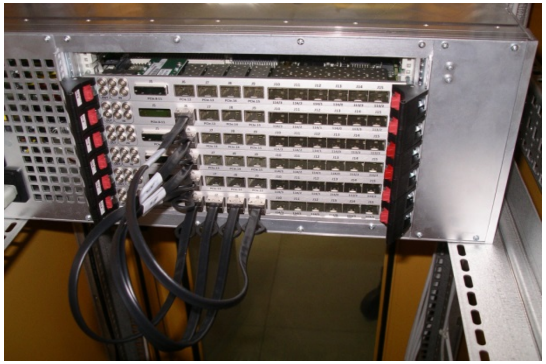
- Easy to extend – e.g., to an external computer



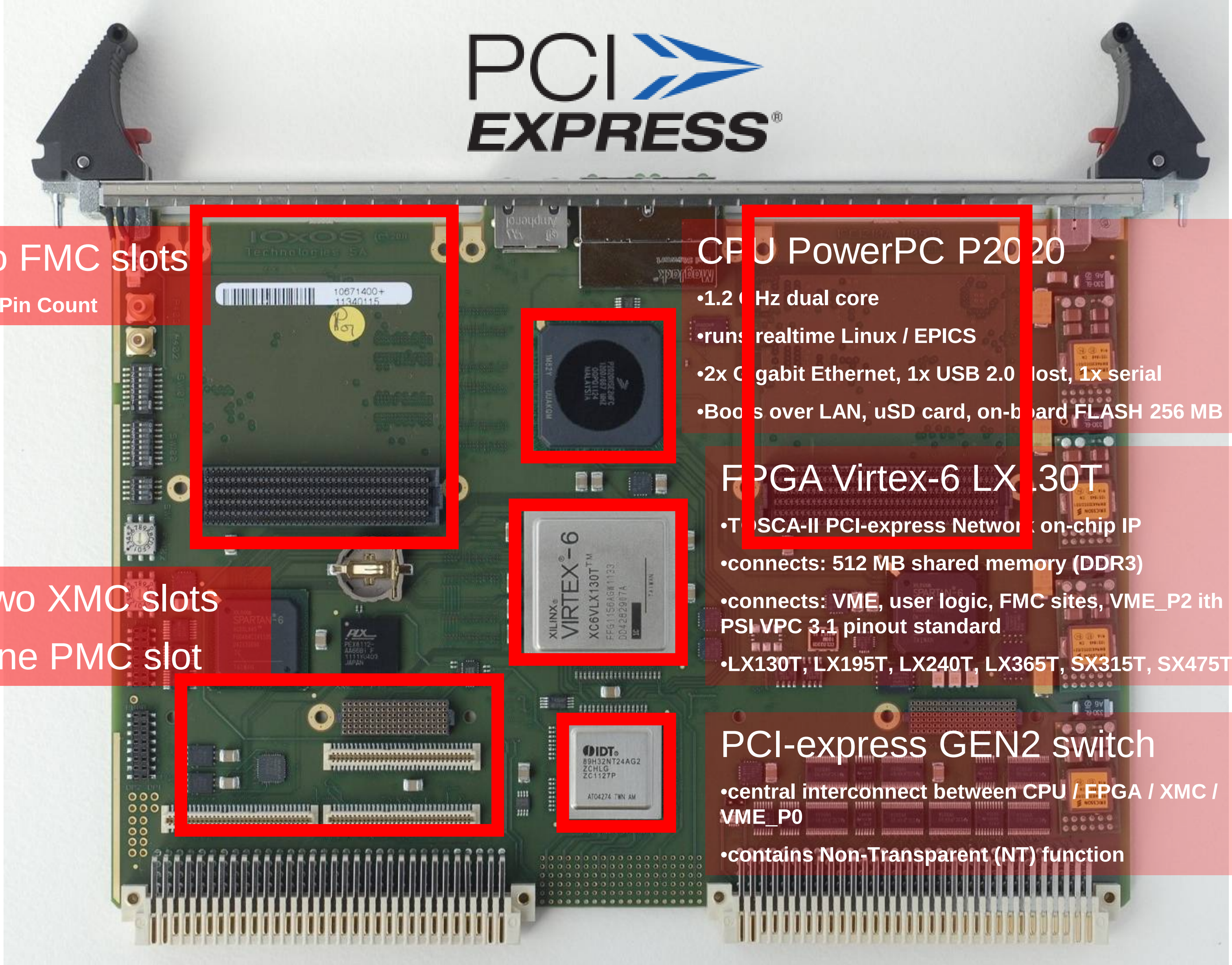
(diagram courtesy IOxOS)

- P0 UHM connector: a detail - but an important one
- -supports fast serial lines like PCI express, 10GBE

-can be used in standard VME64x crates



Board Resources



Two FMC slots

- High Pin Count

- Two XMC slots
- One PMC slot

CPU PowerPC P2020

- 1.2 GHz dual core
- runs realtime Linux / EPICS
- 2x Gigabit Ethernet, 1x USB 2.0 Host, 1x serial
- Boots over LAN, uSD card, on-board FLASH 256 MB

FPGA Virtex-6 LX130T

- TOSCA-II PCI-express Network on-chip IP
- connects: 512 MB shared memory (DDR3)
- connects: VME, user logic, FMC sites, VME_P2 ith PSI VPC 3.1 pinout standard
- LX130T, LX195T, LX240T, LX365T, SX315T, SX475T

PCI-express GEN2 switch

- central interconnect between CPU / FPGA / XMC / VME_P0
- contains Non-Transparent (NT) function

FMC Card examples

- The number of FMC cards on the market is rapidly increasing
- Some ADC FMCs that we have tried
  - IOxOS 3110/3111 (co-development)
    - 8-channel, 250 MSPS, 16-bit ADC
  - 4DSP, Curtiss-Wright FMCs with similar specs
  - D-tAcq (in evaluation)
    - 16/18-bit, 4 channel, up to 2 MSPS, programmable voltage ranges
- Several FMC projects from Open Hardware project (www.ohwr.org)
- 10 Gigabit Ethernet interfaces
- Floating-point computation accelerators (DSPs)

And many more. See e.g. VITA FMC website (http://www.vita.com/fmc.html)

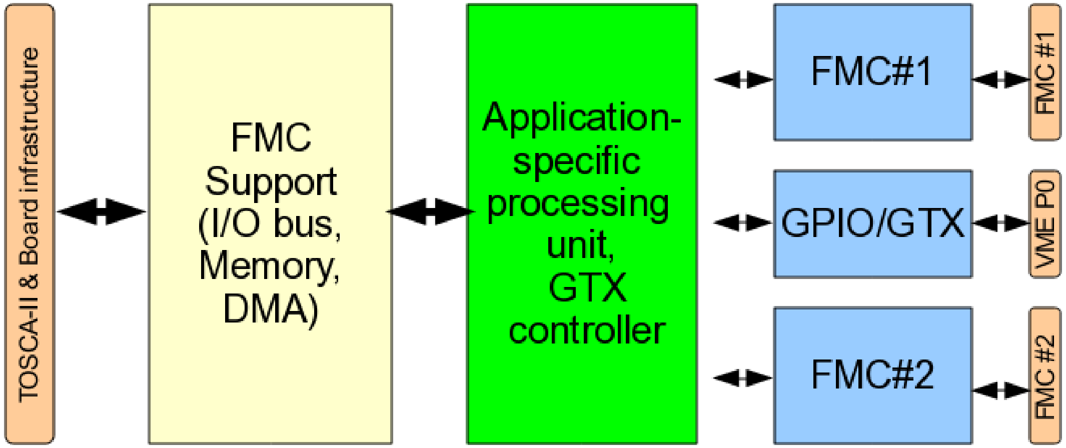
- Our experiences
  - With careful design, size of FMC does not limit analog performance
  - It takes 1-2 months from an experienced FPGA programmer to fully interface one card (first time takes longer)
  - More and more cards appear – technology improves continuously

Software

- Embedded Linux with RT-Preempt patch
  - Fully open-source, from kernel.org
  - Drivers to support board infrastructure provided by IOxOS
- EPICS on top of the Linux OS
  - Development on a standard Linux box with cross-compiler
  - EPICS runs in user space
  - Kernel drivers provide user-space access to memory allocation, interrupts and DMA engines
- CPU core allocation for special user programs
  - CPU affinity and high process priority
- Memory-mapped access to board resources
  - FPGA user-specific firmware
  - VME, Shared RAM
- A lot of monitoring and hardware control features
  - Temperatures, power consumption, settings via I2C

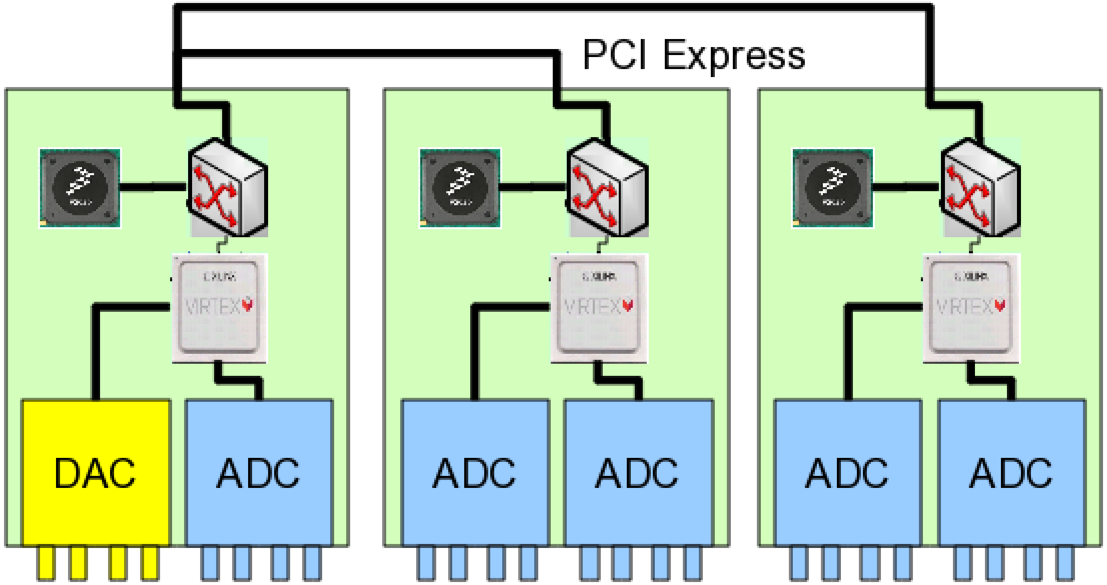
FPGA Development

- FMC standard defines the electrical and mechanical interfaces, but
  - Firmware structures are up to the implementer
  - Each vendor typically has its own approach
- Without a common structure, each FMC integration would be from scratch
  - We defined a structure of modules to maximize FPGA code reuse
  - Embedded in the TOSCA-II infrastructure
  - Defined register layout for common functions (board, firmware IDs, etc.)
  - Mix application-specific and FMC-specific parts
    - Same application logic for several ADCs or
    - different applications for one FMC


TOSCA-II & Board infrastructure
FMC Support (I/O bus, Memory, DMA)
Application-specific processing unit, GTX controller
FMC#1
GPIO/GTX
FMC#2
FMC #1
VME P0
FMC #2

Applications

- Low-Level RF controls
  - Many ADC channels, a couple of DAC channels
  - FPGAs do data preprocessing
  - Data transfer between several boards use PCI express
  - Feedback loops running on one main card
  - EPICS can run on all boards



Applications

- Digital Power Supply Controller
  - PSI digital power supply control over fiber links
  - Each board is an EPICS IOC
  - Fast orbit feedback interfaces direct through FPGA

- Generic Digitizer (“Oscilloscope Application”)
  - ADC with trigger functions in firmware
  - Combination with timing hardware
  - Same application, different front-ends

- EtherCAT master controller
  - EtherCAT master software stack on the CPU
  - Integration of a real-time capable fieldbus with event timing

Conclusions

The IFC_1210 is the new standard controls platform at PSI

- Provides the basis for a wide spectrum of applications
- High bandwidth but still backwards compatible
- Ample FPGA resources, lots of reusable firmware blocks
- Fully open source Linux OS, root file system and bootloader
- Comprehensive EPICS integration

A number of advantages:

- Easy upgrades of older systems without the need of full replacement of hardware and cabling
- The most valuable know-how is in portable components:
  - Firmware, software is not dependent on board form factor
- Provides a long-term development roadmap
- Fruitful collaboration with industrial partner
- No compromises in performance or flexibility

Thank you for your attention!