A flexible data acquisition system for the MEDIPIX family detectors

N. V. Biesuz[1], J. A. Alozy[2], R. Bolzonella[1,3], M. Campbell[2], V. Cavallini[1,3], A. Cotta Ramusino[1], M. Fiorini[1,3], X. Llopart-Cudie[2], G. Romolini[1,3]

1 - INFN sezione di Ferrara, 2 - CERN, 3 - Università degli Studi di Ferrara

16th Pisa Meeting on Advanced Detectors - La Biodola, Isola d'Elba, May 26-June 1, 2024

The Medipix Family ASICs

The CERN MEDIPIX Collaboration developed a family of read-out ASICs for particle imaging and detection development.

The Medipix ASIC family comprises 4 generation of ASICs.

The latest iterations are:
- Timepix4 *[1]*:
  - 512x448 pixel matrix, 55 μm pitch, data-driven readout, ToT (1.56ns bin) and ToA (195ps bin), for timing applications
- Medipix4 *[2]*:
  - 320x320 pixel matrix, 75 → 150 μm pitch, 1→ 8 thresholds, for imaging applications
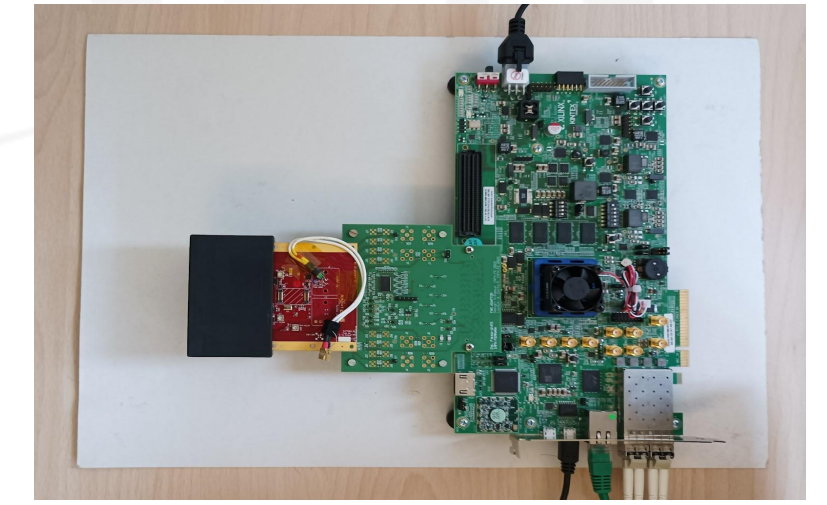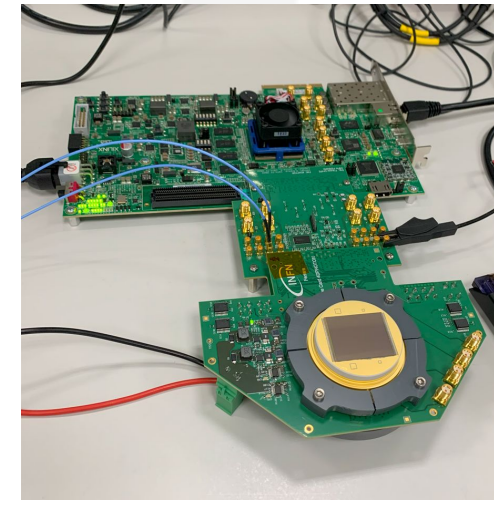
INFN and Timepix4

DAQ developed in the context of the 4DPHOTON *[3]* project:
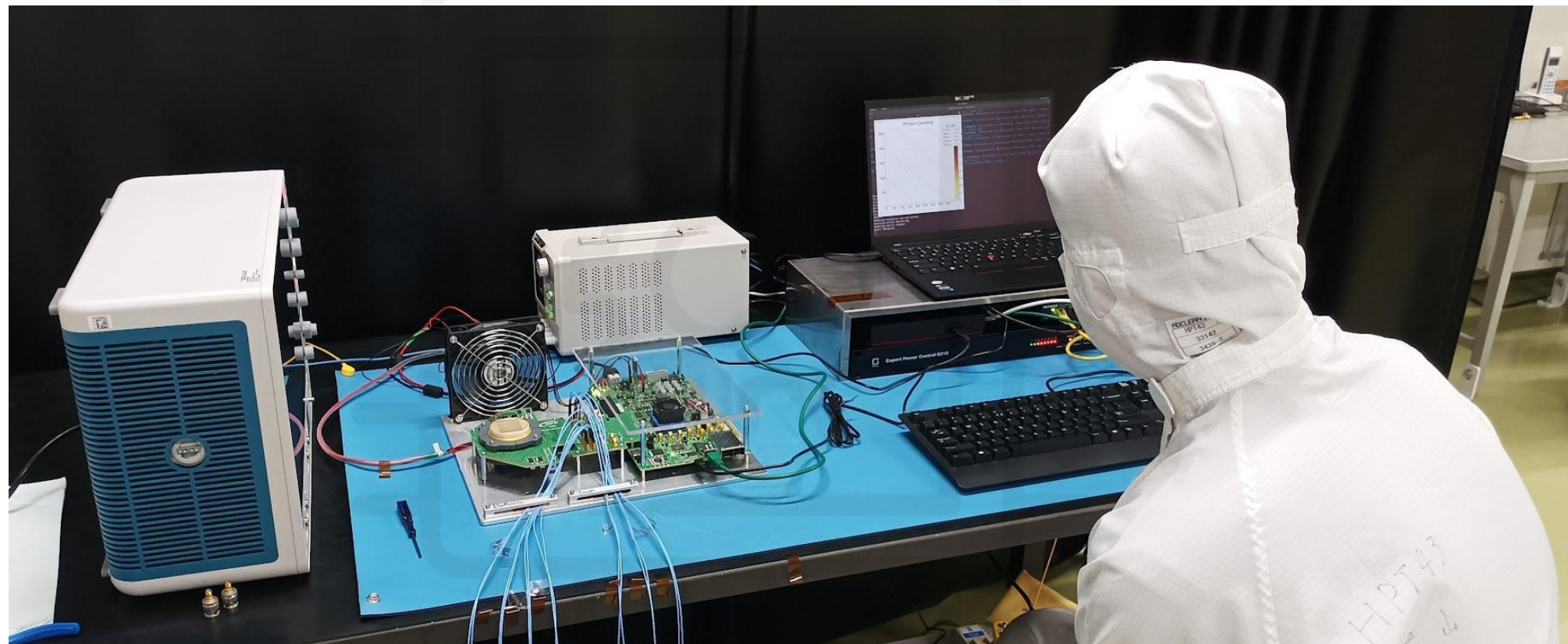- Development of a single photon detector with integrated front-end.

DAQ also used by INFN in other activities:
- bump bonded semicond. sensors for medical use and tracking

There is a need for a common and flexible DAQ system

The Hardware

Develop "general purpose" control card/DAQ
- Based on "consumer" hardware (Xilinx KCU105)
- Use standard protocols
  - 1G ethernet for configuration data from controller
  - 10G ethernet for detector data to storage
  - FMC VITA 57.1 for detector communication (Physical only)
- Enough resources for minimal pre processing and monitoring
- Open source and open hardware
  - Released under open licence: EUPL 1.2 Licence

The Firmware

Top level schema for the Timepix4 case

Top level schema for the Medipix4 case

The DAQ system allows for operation with:
- Timepix4 ASIC (timing applications)
- Medipix4 ASIC (imaging applications)

Main difference → "fast" readout:
- Timepix4: 16 MGT @10.24 Gbps (64b/66b)
- Medipix4: 16 synchronous links @320 MHz DDR

Firmware is fully configurable to support different detector cards using custom peripherals
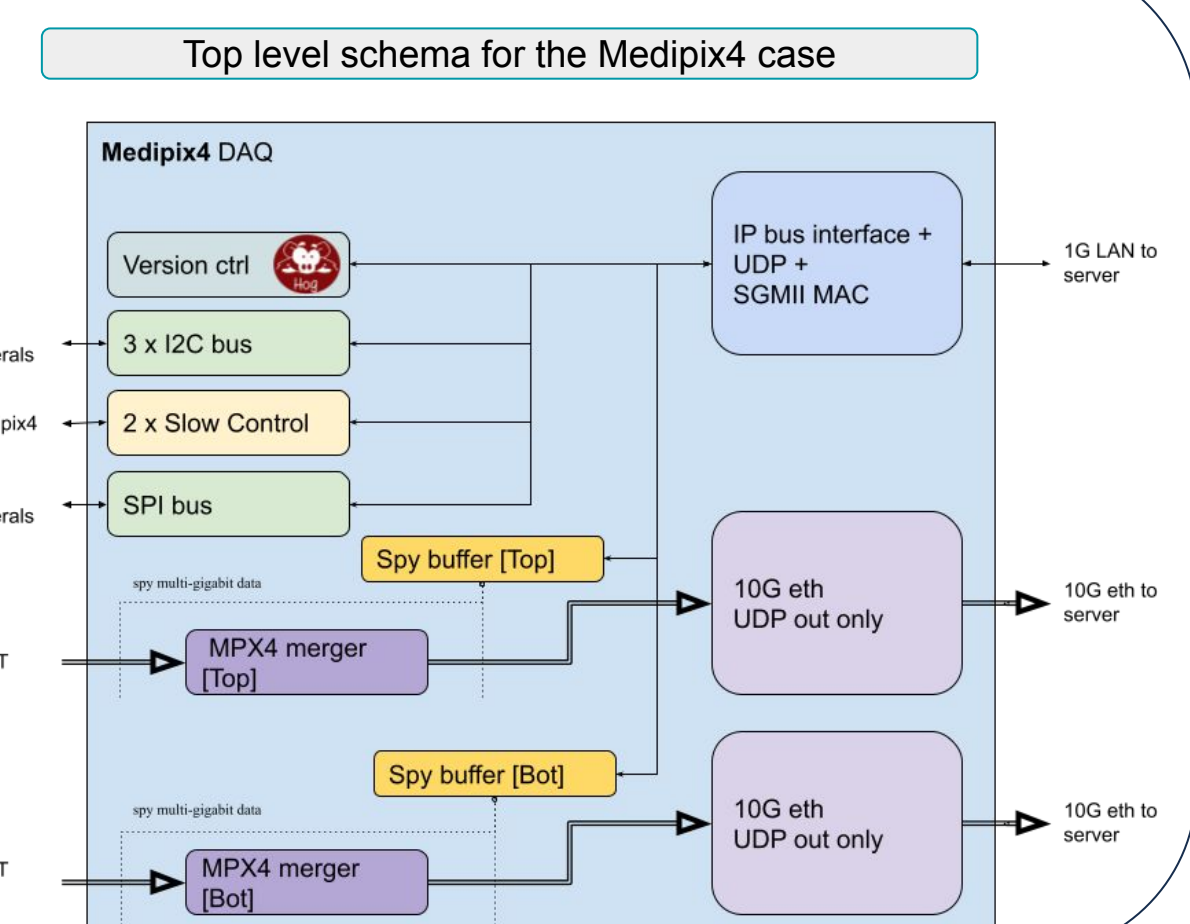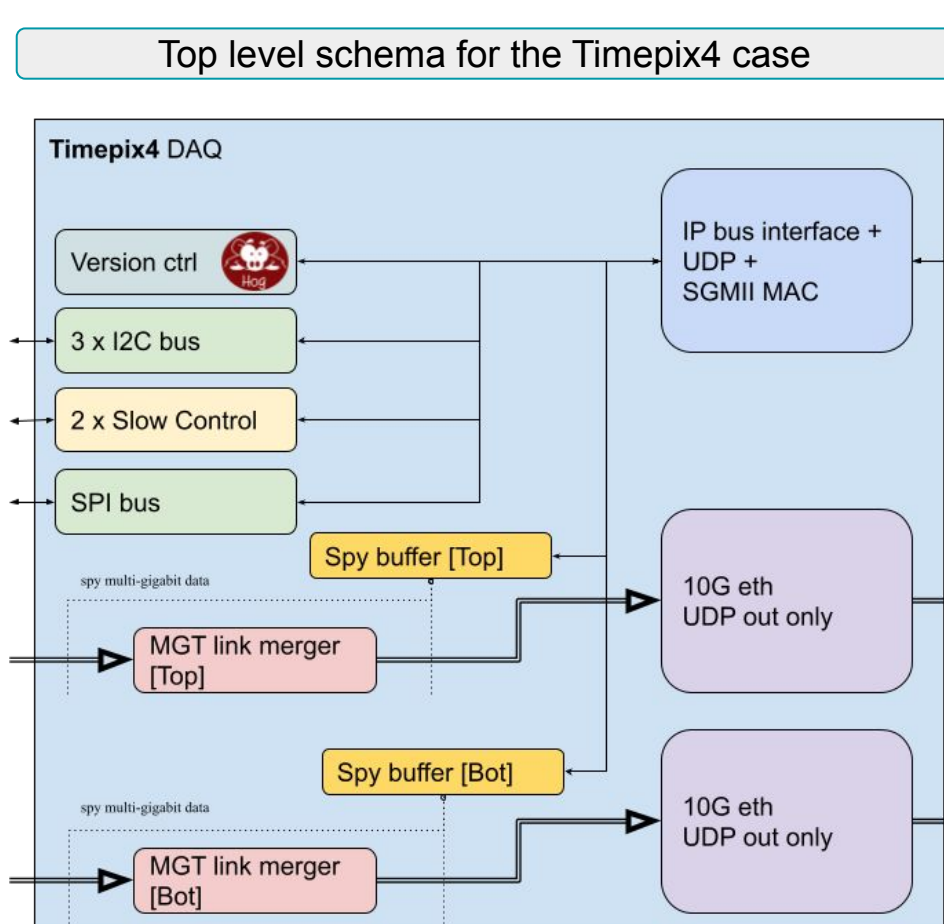
Reduce, Reuse, Recycle

Use HDL on git (Hog) *[4]* to handle project in git:
- existing HDL sources included as git submodules
  - ➤ ipbus *[5]*   (from github.com/ipbus)
  - ➤ 10G/1G UDP-IP stack   (from github.com/alexforencich)
  - ➤ common modules: (from baltig.infn.it)
    - I2C master, SPI master, reset generators, debouncers, …

Keep it open source → user can add modules at firmware level

The software

DAQ system is complemented by a software toolset (C++)
- Allows for both
  - Low level access (single commands)
  - High level routines (equalization, calibration, data acquisition, etc..)
- Compatible with other DAQ systems
- Scalable to multi-DAQ systems

Create DAQ object — Configure — Analyse

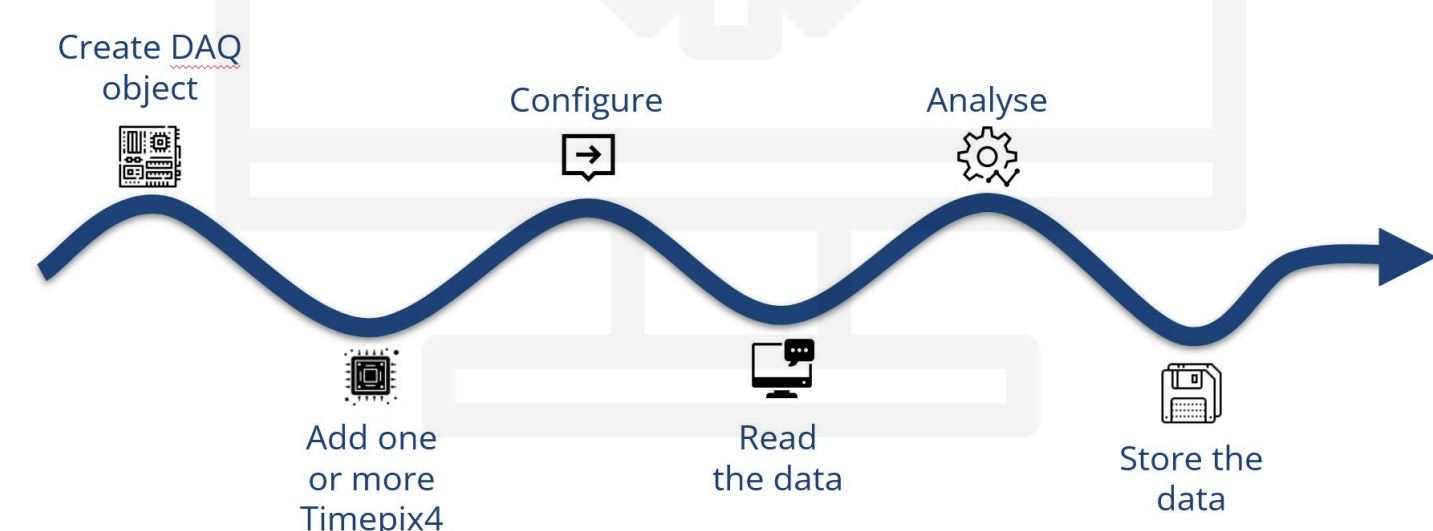Add one or more Timepix4 — Read the data — Store the data

Future developments

DAQ development still ongoing, planned improvements are
- Full support of the MEDIPIX4 ASIC
- Addition of Partial Reconfiguration regions in the data-path
  - Allow for data filtering, online clustering in hardware
- Exploit Hog for instantiation of peripherals masters

References

1. X. Llopart et al., 2022 JINST 17 C01044, DOI: 10.1088/1748-0221/17/01/C01044
2. V. Sriskaran et al., 2024 JINST 19 P02024, DOI: 10.1088/1748-0221/19/02/P02024
3. M.Fiorini et al., 2018 JINST 13 C12005, DOI: 10.1088/1748-0221/13/12/C12005
4. N.V. Biesuz et al., JINST 16 (2021) 04, T04006, DOI: 10.1088/1748-0221/16/04/T04006
5. C. Ghabrous Larrea et al., JINST 10 (2015) 02, C02019. DOI: 10.1088/1748-0221/10/02/C02019

With the exception of institutional logos, background icons are a courtesy of flaticon.