GigaFRoST (Gigabyte Fast Read-Out System for Tomography): Controls and DAQ System Design
WHY

GigaFRoST?
• Pco.dimax sensor & head board
  • 2016 x 2016 pixels
  • Sub 20 um pixel size
  • kHz frame rate
  • 12 ADCs
• Custom made auxiliary electronics
• Continuously acquire & stream up to 7.7 GB/s to the backend server
• Providing a live image preview stream (handled by the backend server)
GigaFRoST architecture

Head Board
- ADC NW
- ADC NE
- ADC SW
- ADC SE

Data Board (x2)
- RAM calibration data
- FPGA FX100T FF1136
- PHY 10 Gbit ethernet
- SFP+

Control Board
- RAM calibration data
- FPGA FX70T FF665
- SFP

Connections:
- 8 x 10Gb/s fiber optic connections
- 12V DC
- TTL signals

Backend server
Beamline console

Head board & sensor

Control board
Data readout board
Camera control & image acquisition

- EPICS IOC running on the PPC of the control board
  - Control of camera parameters, system monitoring, coordination of image acquisition as well as some DAQ configuration (for backend server) via a caQtDM GUI
  - Support for ao, ai, bo, bi and waveform records
  - Epics driver + CS independent lower level driver talking to the API
  - 2-step writing to and reading from the camera
  - Direct reset / shutdown of the camera or reset of individual readout boards or even FPGAs
  - In Sync flag (EPICS PVs vs. camera parameters)
DAQ architecture

• **Backend server**
  – 2 x CPU (each 14 cores, 256 GB RAM)
  – Assembling the images (each CPU ½ of the image)
  – Specialized ring buffer architecture for high-frequency memory access
  – 2 x 56 Gbit/s InfiniBand connections downstream
  – ZMQ data stream + live image preview stream

• **File writing and storage**
  – Subscribing to the ZMQ data stream
  – Packaging images into HDF5 files
  – Storing on a GPFS file storage server

• **Current limitations and expandability**
  – Writing data from the backend server downstream is currently limited to approx. 2.2 GB/s
Supporting infrastructure for image acquisition sync

Aerotech fast rotation stage

Aerotech WIN PC – EPICS IOC

EPICS channel access

EPICS IOC

Beamline network

Beamline console

‘Signal box’

TTL

GigaFRoST

TTL

Other external devices

Serial link

Firewire

EPICS channel access
My thanks go to

- **R. Mokso & C. Schlepuetz** from Tomcat BL for driving the project
- **G. Theidel** for FPGA FW, Linux OS and GigaFRoST API
- **E. Schmid** for HW design
- **D. Zimoch** for providing EPICS core for selected architecture
- **H. Billich** for developing the backend server (& **A. Babic** for continuing his work)
- **S. Ebner & T. Zamofing** for setting up the HDF5 file writing and ZMQ streaming infrastructure