Filip Leonarski :: Beamline Data Scientist :: MX Data Group

**Jungfraujoch: a Data Acquisition and On-the-fly Analysis System for HDR MX**

17th March 2021
Presentation plan

- JUNGFRAU detector implementation at synchrotron MX beamline
- OpenCAPI based data acquisition
- Jungfraujoch data acquisition system detailed design
JUNGFRAU charge integrating detector

- As integrating detector, JUNGFRAU measures amount of energy deposited in a pixel by X-ray photons during exposure.
- To maximize both sensitivity and dynamic range, JUNGFRAU pixel can switch between the three modes during exposure:
  - G0 is used to measure precisely weak signal (up to 20 x 12.4 keV photons),
  - G1 has wider dynamic range (up to 700 x 12.4 keV photons), but with less sensitivity to single photons,
  - G2 has highest dynamic range (up to 11000 x 12.4 keV photons), but with lowest resolution.

Applications where JUNGFRAU is advantage

Photon counting detectors (PILATUS, EIGER) revolutionized MX and are providing excellent data quality in most cases. There are however applications that can benefit from **new generation integrating detectors**:

1. **Bright X-ray sources**
   - XFEL, diffraction limited storage ring, multilayer monochromator beamline
   - Short exposure, fast data collection
   - Sub-millisecond time-resolved crystallography, high throughput (3 seconds/dataset)

2. **Long wavelength X-ray crystallography**
   (≤ 6 keV)

For more details see: Leonarski et al., Nature Methods (2018)
http://dx.doi.org/10.1038/s41592-018-0143-7
1. JUNGFRAU must be operated fast (1-2 kHz) at continuous source, so generates large volume of data

2. JUNGFRAU read-out must be converted to become useful or even to allow frame summation (cannot add G1 and G2 values without putting them on single scale)

3. Calibration for dark-current (pedestal) needs to be regularly updated

**X06DA SLS**

JUNGFRAU is installed in a dry nitrogen housing to allow cooling to -15°C without condensation for synchrotron applications
Task: JUNGFRAU 10 Mpixel in 2022

- SLS 2.0 upgrade in 2023-2024

- Beamline X06DA was selected as a pilot project – it will get major refurbishment in 2022

- One of developments is tiled JUNGFRAU detector for native-SAD applications – producing up to 46 GB/s

- This requires data acquisition system that will provide kHz performance

- Need smooth integration of the detector into beamline

- 4 Mpixel system will be installed this summer
Year 2018: Conventional Hardware

- The most powerful single server available from PSI vendor
  - 4 CPU socket, 1.5 TB RAM, NVMe SSD drives and Mellanox fiber ethernet
    network cards

- Hours of software development, profiling and performance tuning

- Outcome: the server can handle JUNGFRAU 4Mpixel at 1 kHz (and with on-the-fly manner), but not more

- Bottleneck: memory bandwidth of CPU is too small for conversion procedure

Year 2019: Task-specific Hardware

- Experiment with task specific architecture
- IBM POWER9 showed great numbers for I/O and memory throughput in Summit and Sierra supercomputers
- First option: offload conversion on GPU
- Second option: use IBM-specific interface for FPGAs
- POWER9 server arrived at PSI late autumn 2019 and development started
- At first sight FPGA was considered too complex to implement...
Field Programmable Gate Arrays

• Integrated circuits that can be configured by customer

• Consists of large number of programmable elements (logic gates) and routes

• FPGA can be connected directly (via transceivers) to input/output, like 100G ethernet

• They are also offered as boards that can be installed in a computer

• Very powerful, but significant effort in development, due to need of specific hardware design skills and hardware description languages
Year 2020: ... 5 months later FPGA is collecting protein diffraction images

March 2020
Thaumatin images acquired and converted on FPGA and saved on POWER9 system
Why it worked so fast?

Thanks to OpenCAPI
Virtual address space (VAS) is what software developers are used to

- VAS is key concept in computer architecture
- Integral element of all modern computing architectures and multitasking operating systems
  → for x86 protected mode from 80286 (improved in 80386)
- Hides complexity of physical memory from software developer
- CPU memory management unit and operating system kernel do a lot behind the scenes

OpenCAPI makes FPGA design much easier for software developers thanks to access to VAS

- Only available on IBM POWER CPU architecture
- Allows to attach accelerators, like FPGA, to CPU with full access to virtual address space
- This can be combined with tools to compile C++ code into FPGA hardware design language
- All communication between FPGA and CPU is implemented with familiar memcopy function or array/pointer semantics
- OpenCAPI has much lower entry barrier for software developer and is easier to verify correctness, as compared to traditional hardware interfaces, like PCI Express

**Xilinx QDMA** is a robust but highly complex solution for PCI Express – used to interface FPGAs with x86 AMD and Intel CPUs
Up to **50 GB/s acquisition and data analysis** in a single 2U IBM POWER9 server with 1-4 FPGA boards

FPGA board with OpenCAPI interface

- Data acquisition
- Initial data analysis
- Pre-compression

(2.5 Mpixel/board for JF)
Jungfraujoch (3454 m a.s.l.) – highest railway station in Europe

Modular design

- Stream of data handled by successive cores doing work in parallel → **throughput and latency of each core is determined by the hardware design**
- Extra stages can be relatively simply added, option to bypass cores
- Limited buffering on the way in on-chip memory (around 1 frame)
Ethernet UDP/IP core

Processes ethernet packets from network, ignores unnecessary packets, reads frame header to get frame number, module number, etc.
Pedestal correction core

This cores is responsible for calculating moving average of detector frames. Calculated value is used as dark current (pedestal) for subsequent frames.
Gain correction core

This core translates JUNGFRÄU read-out into units of energy or photon counts. It benefits from very fast HBM2 memory within the FPGA (460 GB/s). Data leaving this core can be used for processing by data analysis software.
**Frame summation core (work in progress)**

As data that left gain correction core are on linear scale, they can be summed to reduce downstream data rate, if lower frame rate is needed, as compared to detector.
**Strong pixel finder core**

This is first step of spot finding algorithm (for example COLSPOT). It identifies pixels that are stronger than given number of standard deviations of their neighborhood.

Note – streaming implementation can only accommodate algorithms that depend on local neighborhood of the pixel. Algorithms that require knowledge of full image must be implemented on CPU or GPU.
Bitshuffle core (work in progress)

FPGAs are bit order agnostic. Therefore exchanging bit order in popular compression prefilter is pretty much for free on FPGA.
Host memory write

Address in host memory buffer is calculated and data forwarded to host memory via OpenCAPI. Additional image statistics are saved as well.
Jungfraujoch Board FPGA
Year 2021: Japan

- Detector and data acquisition system was sent in Autumn for an experiment in Photon Factory, KEK

- More than 1,000 datasets collected for protein targets, most with long wavelength (2.7-3.3 Å)

- Already few real-life native-SAD structures solved with JUNGFRAU and Jungfraujoch

- Files generated by Jungfraujoch pass NXmx validation and can be opened/analyzed with most MX viewers/data processing tools

- Due to pandemic, detector support and development (including deployment of new FPGA design and software is done fully remotely from Switzerland)
Real-life crystal native-SAD with JUNGFRAU 4M and Jungfraujoch DAQ

- Laser shaped crystal
- Space group P3$_1$21
- Amino acids: 1040
- 42 sulfurs (mostly MET)
- Solvent content: 40%
- Wavelength: 3.0 Å (4.1 keV)
- Turns: 5 x 360° (different kappa; 100°/s)
- Open shutter time: 5 x 3.6 = 18 second
- Detector frame rate: 1000 Hz
- Solution: SHELXC -> SHELXD -> Crank2

Beamline: BL-1A KEK

Acknowledgements
Tristan Wagner Group
(Max Planck Institute for Marine Biology, Bremen)
Naohiro Matsugaki (Photon Factory)
Jungfraujoch is a fully integrated DAQ solution for kilohertz frame rate JUNGFRAU detectors – deployed on X06DA beamline this summer for user operation.

FPGA design is not as difficult as it seems – just need right tools to do that, like for example OpenCAPI.
Acknowledgements

**MX Group (PSI)**
- Vincent Olieric
- Takashi Tomizaki
- Chia-Ying Huang
- Sylvain Engilberg
- Justyna Wojdyła
- Meitian Wang

**Detector Group (PSI)**
- Aldo Mozzanica
- Martin Brückner
- Carlos Lopez-Cuenca
- Bernd Schmitt

**Science IT (PSI)**
- Leonardo Sala

**Controls (PSI)**
- Andrej Babic
- Leonardo Hax-Damiani

**SLS management (PSI)**
- Oliver Bunk

**Photon Factory, KEK**
- Naohiro Matsugaki
- Yusuke Yamada
- Masahide Hikita

**MAX IV**
- Jie Nan
- Zdenek Matej

**Uni Konstanz**
- Kay Diederichs

**LBL**
- Aaron Brewster

**DLS**
- Graeme Winter
- DIALS Team

**ESRF**
- Jerome Kieffer

**IBM Systems (France)**
- Alexandre Castellane
- Bruno Mesnet

**InnoBoost SA**
- Lionel Clavien
Single photon counting

- Low-noise detection (no readout noise)
- Non-uniform response (“corner effect”)
- Count-rate limited (“pile up”)
- Threshold calibration is less optimal at low energy (< 6 keV)

Charge integrating

- No instantaneous count-rate limitation
- Uniform response
- Dark current (“readout noise”)
- Dynamic range limited
Conversion of detector units to photon counts in JUNGFRAU is as following

\[ \text{photons} = g \times (\text{ADU} - p) \]

Where \( p \) – dark current, \( g \) – conversion factor from ADU to photon counts (or eV)

- \( p \) and \( g \) constants are pixel specific and mode specific
- I.e. each pixel has 6 associated constants (G0: \( g_0, p_0 \), G1: \( g_1, p_1 \), G2: \( g_2, p_2 \))
- It is not expensive in terms of calculation complexity (few GFLOPS), but loading correction constants requires few hundredth GB/s throughput
Reliability and verification tools embedded in the design

C++ verification (minutes) tests functionality – code is compiled with GCC
FPGA simulation (hours) tests that hardware implementation is correct
Hardware verification (days) dedicated functionality in the design for debug and performance assessment

C++ verification and FPGA simulation are part of an automated test pipeline