OPEN-CL framework for Digitizer FPGA signal processing

Miguel Astrain

Universidad Politécnica de Madrid

Host scientist: Dr. Andreas Wagner, Head Nuclear Physics Division, HZDR
Contact: Arnd Junghans, HZDR

Outline

- About me
- Project context
- Development
  - HLS vs OpenCL
  - Xilinx HLS worksflow
  - Spdevices devkit adaptation
  - Testing framework
- Results
- Conclusions

About me

- **Universidad del País Vasco/Euskal Herriko Unibertsitatea**
  - **Physics graduate**
  - **Electronics Engineering graduate**
- **Universidad Politécnica de Madrid**
  - **MSc Nuclear Science and Technology 2016-2018**
  - Final Project: “Implementation and assessment of ITER fast plant interlock system methodology“
- **Universidad Politécnica de Madrid 2018-2021**
  - **Thesis:** Application of OpenCL to FPGA-based systems in EPICS.
    Contribution to the standardization of advanced instrumentation systems for Big Science experiments.
  - A methodology to standardize the development of FPGA-based high-performance DAQ and processing systems using OpenCL. *Fusion Engineering and Design Volume 155, June 2020, 111561*
  - Development of deep learning applications in FPGA-based fusion diagnostics using IRIO-OpenCL and NDS. *Fusion Engineering and Design Volume 168, July 2021, 112393*
  - Real-Time Implementation of the Neutron/Gamma discrimination in an FPGA-based DAQ MTCA platform using a Convolutional Neural Network. *IEEE Transactions on Nuclear Science (TNS)*
- **Max-Planck-Institut für Plasmaphysik in Garching Asdex Upgrade Tokamak**
  - *Developing the next generation of data acquisition system in the ASDEX Upgrade tokamak.*

Context

- From the primary ELBE electron beam a monoenergetic positron beam is created by pair production at a tungsten target.
- A pulsed positron source with high repetition rate, high intensity and selectable implantation energies.
- With this beam, measurements at surfaces and thin layers can be done performed with high depth resolution.
- **Positron-annihilations lifetime-spectroscopy (PALS)** measures the elapsed time between the implantation of the positron into the material and the emission of annihilation radiation.
- However, accurately measuring the lifetime of the positron inside the material is key to extracting the valuable data about the material defects.
- This leads to the need of improving signal processing capabilities.

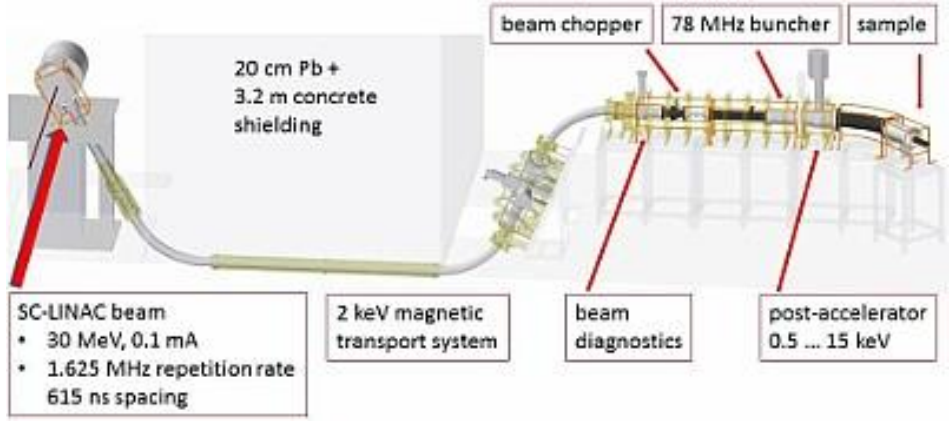


Fig 1: Schematics of the positron beam facility MePS.

Context

- The status of the tools on project start was a complete working CPU based processing solution, based in C++ and Qt, using in-house developments together with the Spdevices Linux drivers and API, named HidraPALS.
- The requirements were expected to rise as improvements to the data acquisition system were already beign discussed. The CPU algorithms could get too slow to provide results immediately.
- The proposal was then made to explore an FPGA based processing algorithm to accelerate it.

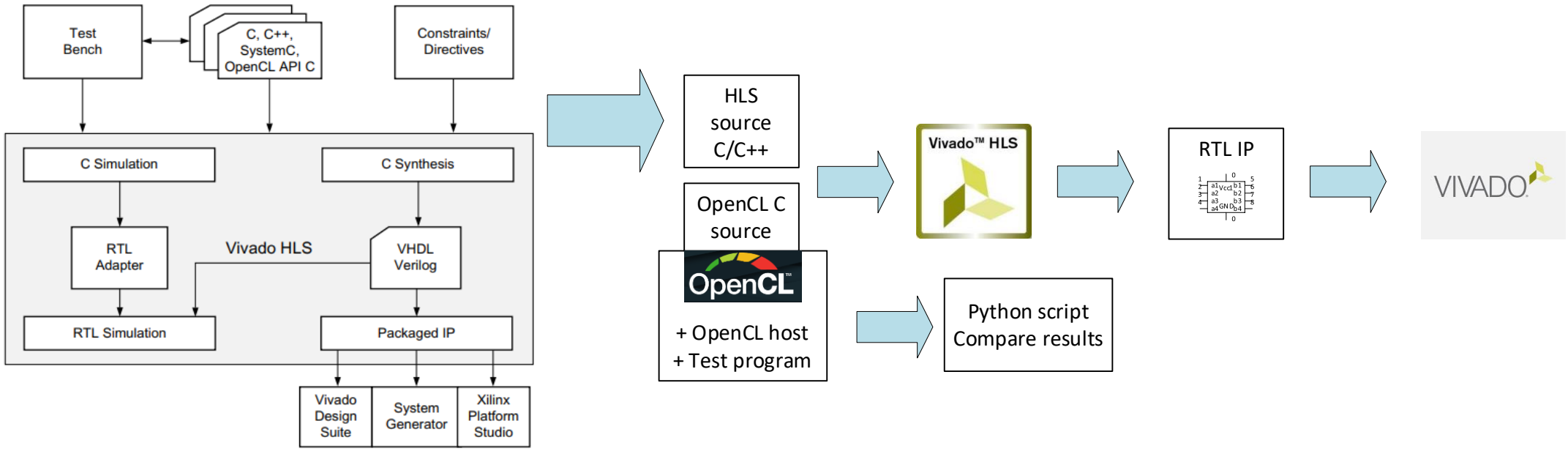
- The Data acquisition system was based in the SP devices adq14.
- Contains an FPGA capable of additional processing.
- Can we develop an algorithm and set the groundwork for easy FPGA processing with minimal HDL knowledge?

Development

- High level synthesis languages
- OpenCL™ (Open Computing Language) is an open, royalty-free standard for cross-platform, parallel programming of diverse accelerators found in supercomputers, cloud servers, personal computers, mobile devices and embedded platforms.
- Vivado® High-Level Synthesis included as a no cost upgrade in all Vivado HLx Editions, accelerates IP creation by enabling C, C++ and System C specifications to be directly targeted into Xilinx programmable devices without the need to manually create RTL.
- **They both enable the deployment of algorithms from C/C++ to the FPGA.**

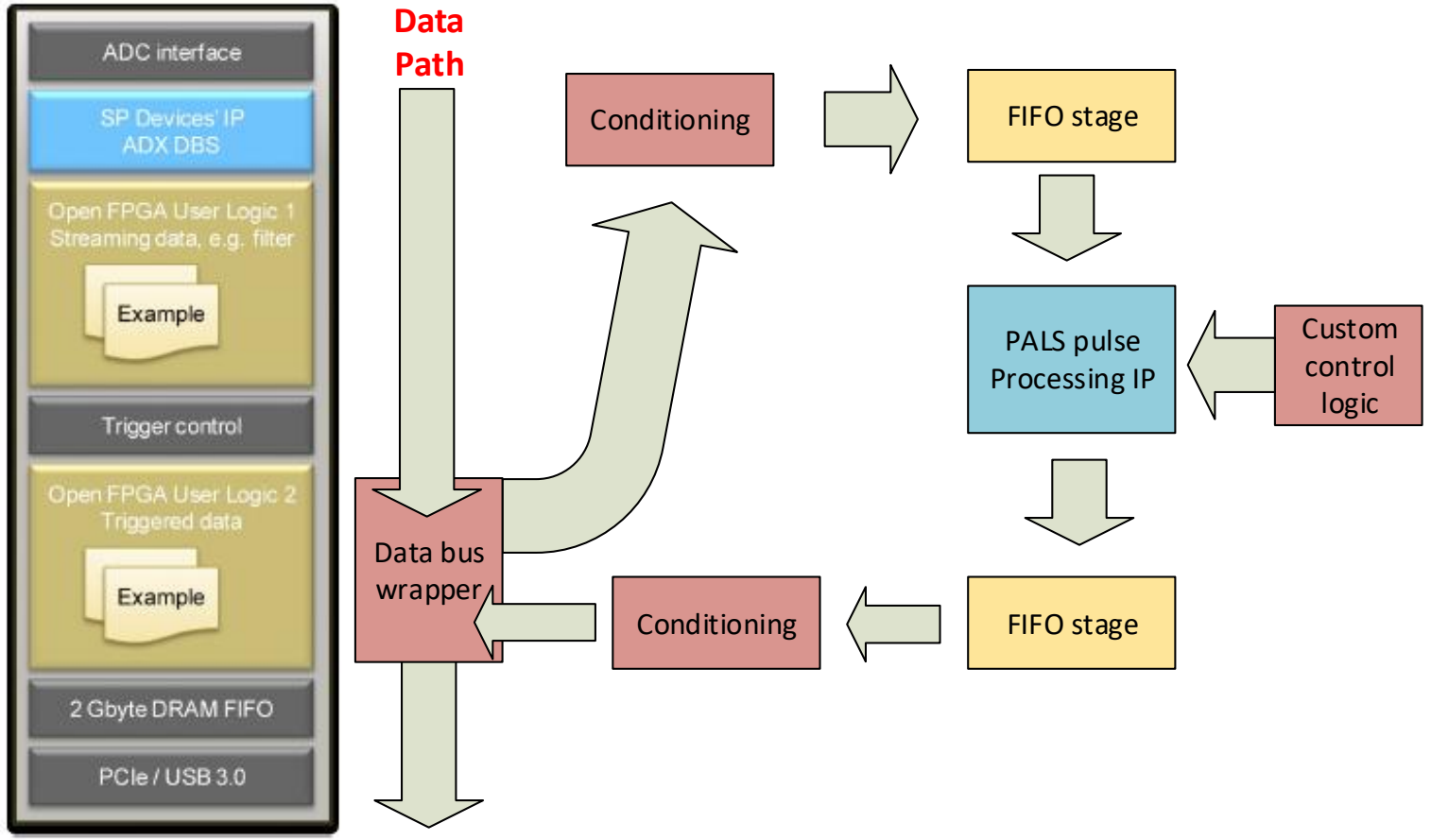
Development

- **However the SPdevices ADQ14 contains an FPGA with a custom firmware that is not designed to use OpenCL.**
- **HLS was selected for the IP development.**
  - **Note that OpenCL is still very compelling for prototyping and testing.**



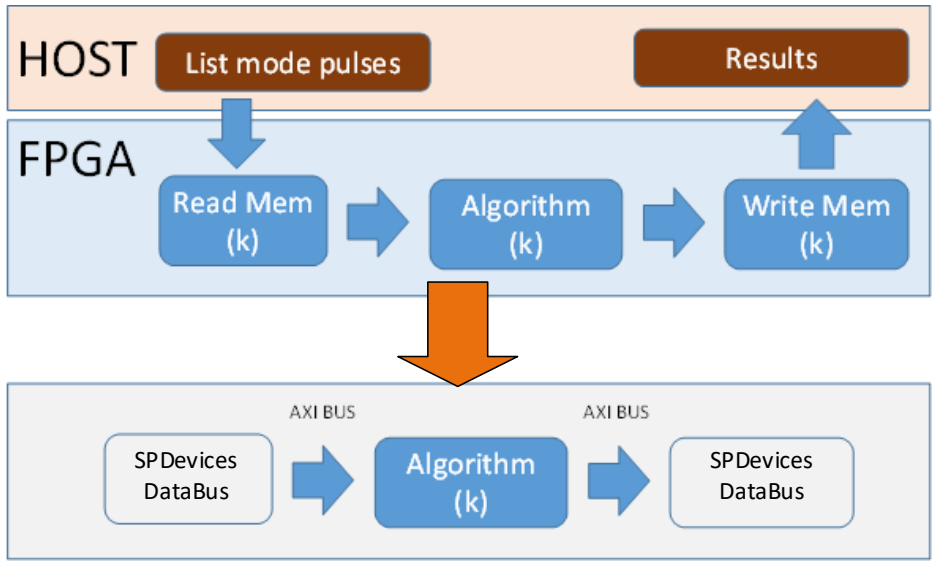
Development

- PALS processing integration (blue) diagram. The data bus wrapper is provided with SPDevices devkit. The custom control logic and the conditioning is done in Verilog (Red). The FIFO stages are added as Xilinx IP blocks (Yellow).



Development

- These developments did not yield a successful compilation of the SPdevices firmware.
- Comments later. But in short. The toolchain versions need to be all compatible.
  - VIVADO 2015.1 for the FPGA but 2019 HLS is not a good match.
- A testing platform was selected based in the Xilinx Alveo U200. A datacenter card that supports HLS and OpenCL programming models.



Results

- The algorithm was evaluated for FPGA for the first time. The algorithm has very little parallelizable tasks, and thus, it is sequential by nature.
- This gives advantages to the high clocks of the CPU, for example, the current CPU setup with an Intel Xeon clocks at 3 GHz.
- Meanwhile, the processing kernel uses a 300MHz clock.

Algorithm	Performance (Pulses / seconds channel)
Householder CPU (Intel Xeon 4c8t @ 3GHz)	~ 5 M
Householder OpenCL no-optimization	~ 500 k
Householder OpenCL dataflow-optimization (double)	~ 1,4 M
Householder OpenCL dataflow-optimization (float)	~ 1,8M

Results

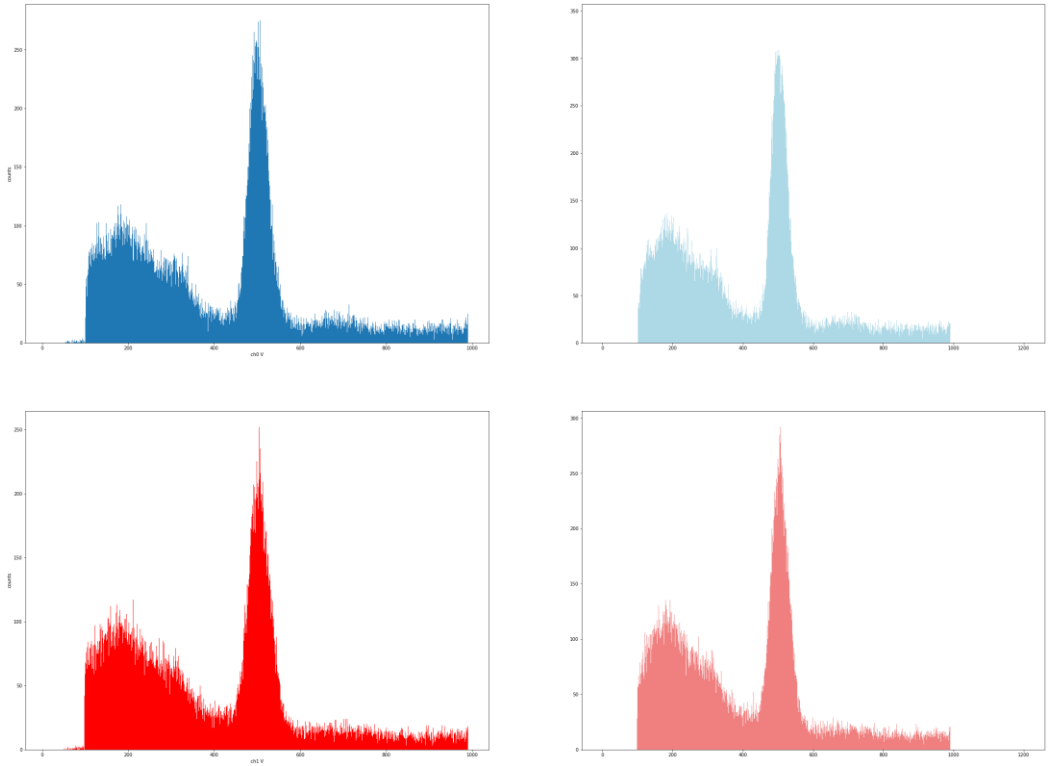
- The utilization of resources was compared to the SPDevices ADQ14 FPGA.

Firmware	FF	LUT	DSP	BRAM
Xc7k325tffg900-2	407,600	203,800	840	445
SPDevices Base + PD	48%	57%	53%	43%
Householder OpenCL dataflow-optimization (double)	16615 4,08%	15735 7,72%	50 5,95%	2 0,45%
Householder OpenCL dataflow-optimization (float)	14745 3,62%	14011 6,87%	50 5,95%	2 0,45%

- FPGA resource utilization. SPDevices firmware uses Xc7k325tffg900-2 target. Householder total resource usage is estimated for Alveo U200 target compilation, percentage is applied based in Xc7k325tffg900-2 total resources.

Results

- The database of pulses was validated against CPU results.
- Pulse maximum value distribution. To the left channel 0 (blue) and channel 1 (red) from the FPGA processed pulses. To the right channel 0 and channel 1 from the original experiment



Conclusions

- While the final goal was not reached, the results and insight gained towards this development cycle has been positive.
- Performance wise, the numbers do not look positive. However, the utilization of the FPGA is not very high, leaving a lot of room from performance improvements.
  - An FPGA can replicate the processing blocks to increase the throughput.
  - Memory efficiency can be increased with HLS code optimizations.
  - The CPU can only process data AFTER the data acquisition (DMA reception of a block of pulses), while the FPGA is working in between pulses.
- Is this it? Is this technology just not enough?

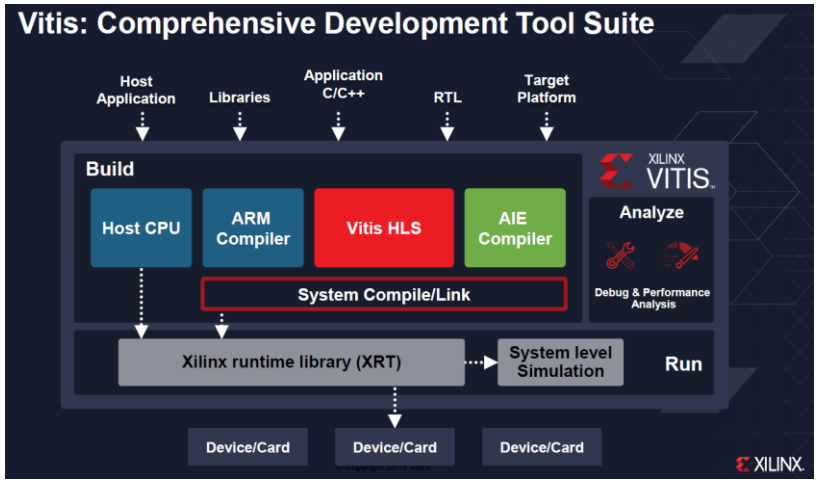
Conclusions

- The technology is being developed as we speak.
- Many of these advancements take years to mature.
- However, the main manufacturers are investing heavily in the development of the tools.

Intel:

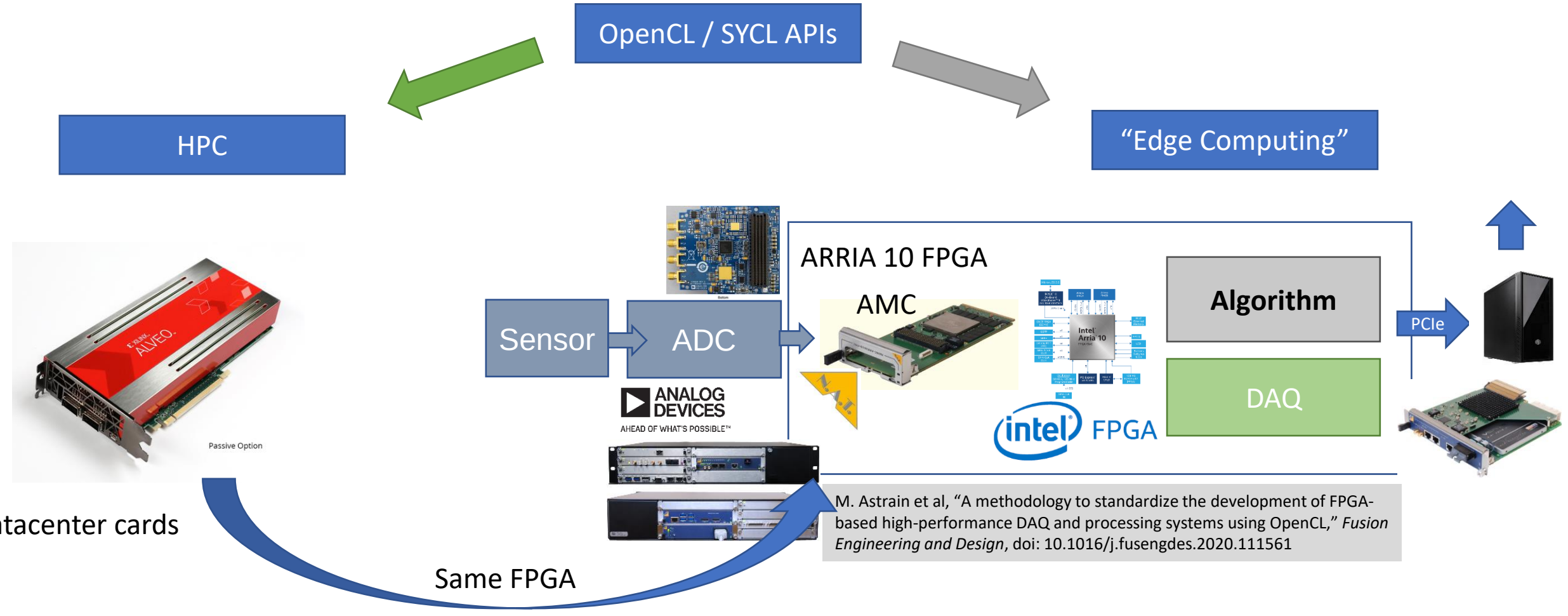
more info: https://www.alcf.anl.gov/aurora

Xilinx:



Will end up in products

… and then we will need to figure out how to work with it…



M. Astrain et al, "A methodology to standardize the development of FPGA-based high-performance DAQ and processing systems using OpenCL," *Fusion Engineering and Design*, doi: 10.1016/j.fusengdes.2020.111561

Thank you

OPEN-CL framework for Digitizer FPGA signal processing

This project has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 847594 (ARIEL).