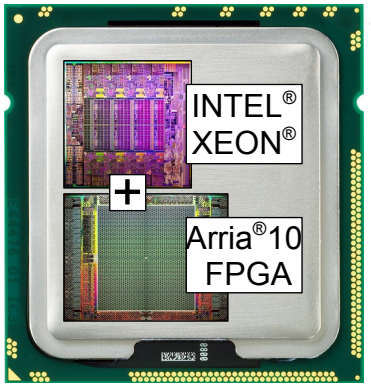
FPGAs as co-processors for reconstruction

Christian Färber
CERN Openlab Fellow
LHCb Online group



On behalf of the LHCb Online group and the HTC Collaboration

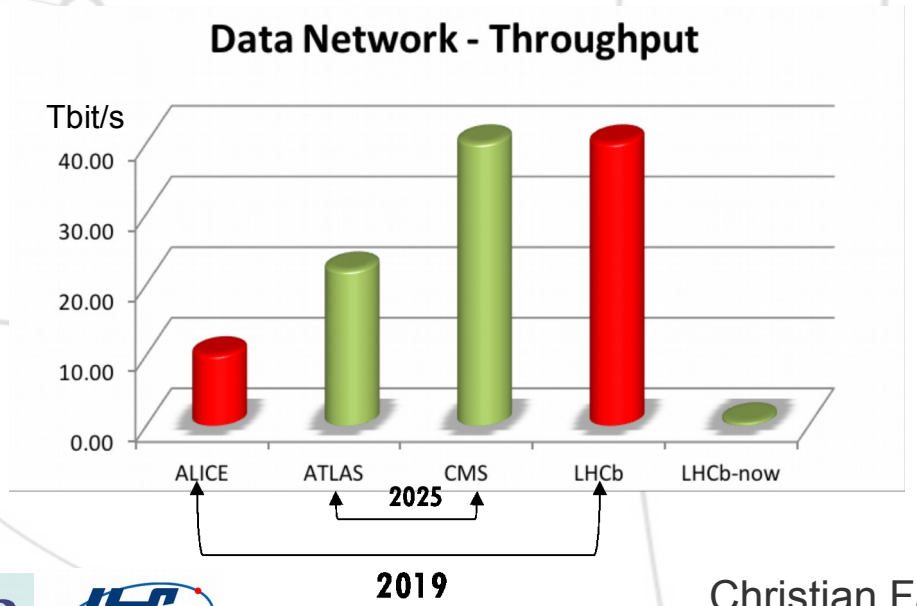
Joint WLCG and HSF Workshop 2018, Naples
28.03.2018

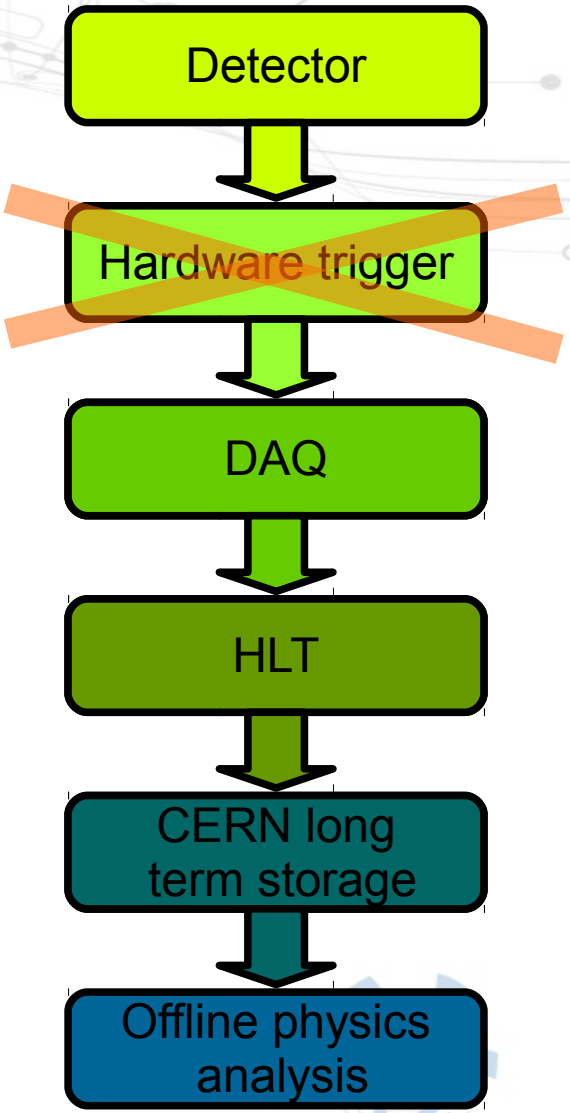
HTCC

- High Throughput Computing Collaboration
- Members from Intel® and CERN LHCb/IT
- Test Intel technology for the usage in trigger and data acquisition (TDAQ) systems
- Projects
  - Intel® KNL computing accelerator
  - Intel® Omni-Path Architecture 100 Gbit/s network
  - Intel® Xeon®+FPGA computing accelerator

Future Challenges

- Higher luminosity from LHC
- Upgraded sub-detector Front-Ends
- Removal of hardware trigger
- Software trigger has to handle
  - Larger event size (50 KB to 100 KB)
  - Larger event rate (1 MHz to 40 MHz)



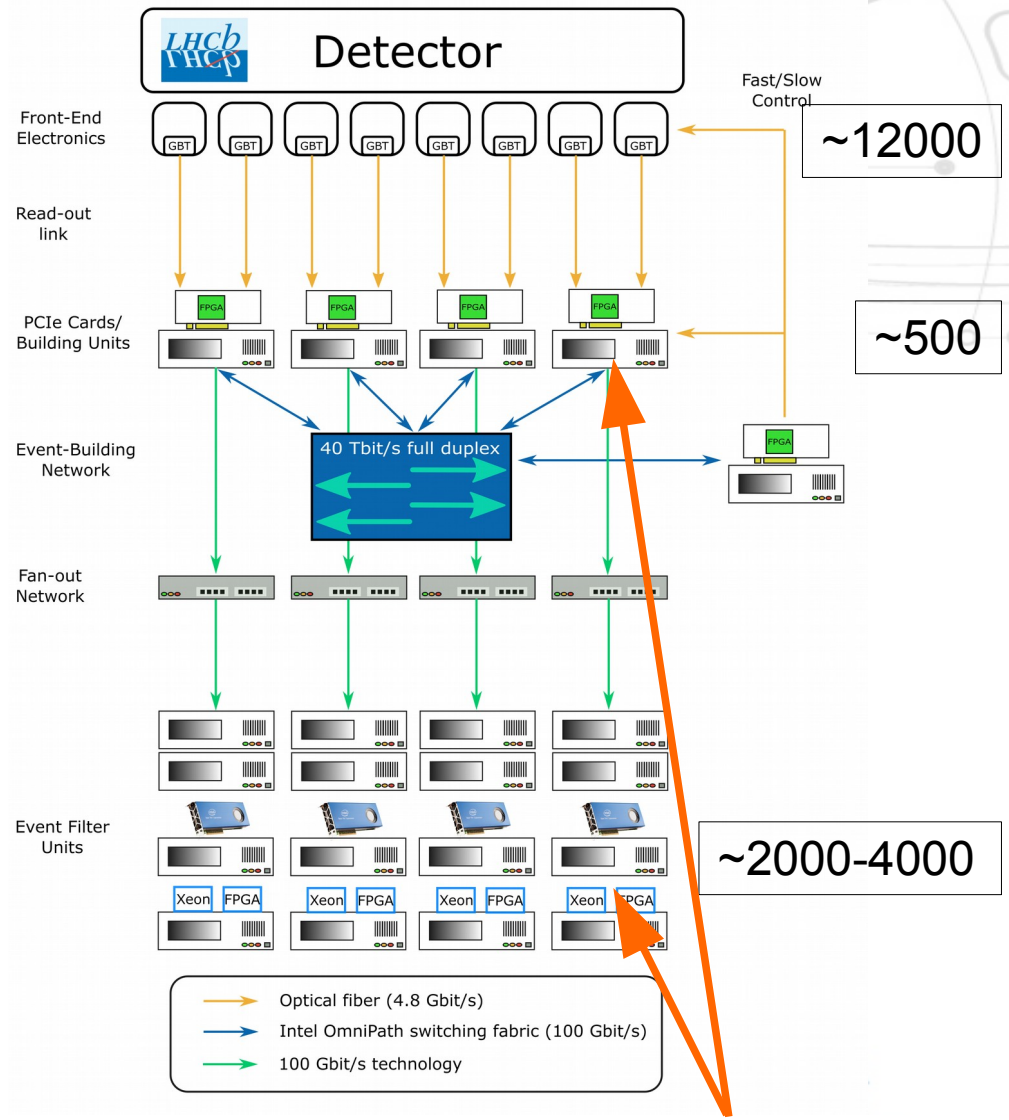


Upgrade Readout Schematic

- Raw data input ~ 40 Tbit/s
- EFF needs fast processing of trigger algorithms, different technologies are explored.

- Test FPGA compute accelerators for usage in:
  - Event building
    - Decompressing and re-formatting packed binary data from detector
  - Event filtering
    - Tracking
    - Particle identification

- Compare with: GPUs, Intel® Xeon Phi™ and other compute accelerators



Which technologies?

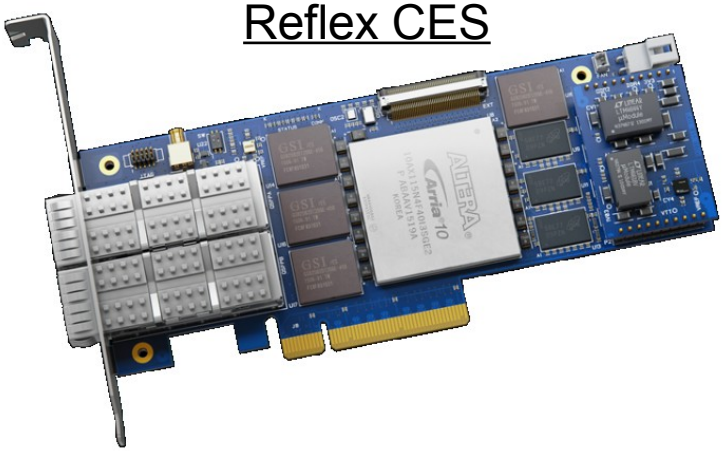
FPGAs as Compute Accelerators

- Microsoft Catapult and Bing
  - Improve performance, reduce power consumption
- Reduce the number of von Neumann abstraction layers
  - Bit level operations
- Power only logic cells and registers needed
- Current test devices in LHCb
  - Nallatech PCIe with OpenCL
  - Intel® Xeon®+FPGA

FPGA compute accelerators

- Typical PCIe 3.0 card with high performance FPGA
  - NIC or GPU size
- On board memory
  e.g. 16 GB DDR4
- Some cards have also network e.g. QSFP 10/40 GbE,…
  - More flexible than GPUs
- Programming in OpenCL
  - OpenCL compiler → HDL
- Power consumption below GPU, price higher than GPU
- Use cases: Machine Learning,
  Gene Sequencing,
  Real-time Network Analytics

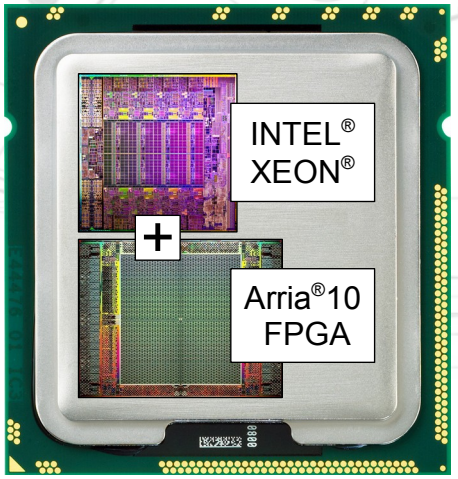
Reflex CES



Nallatech

Becoming a product this year!!!

Intel® Xeon®+FPGA with Arria® 10 FPGA



- Multi-chip package including:
  - Intel® Xeon® E5-2600 v4
  - Intel® Arria® 10 GX 1150 FPGA
    - 427'200 ALMs, 1'708'800 Registers, 1'518 DSPs
- Hardened floating point add/mult blocks (HFB)
- Host Interface: Bandwidth target 5x higher than Stratix® V version
- Memory: Cache-coherent access to main memory
- Programming model: Verilog, soon also OpenCL

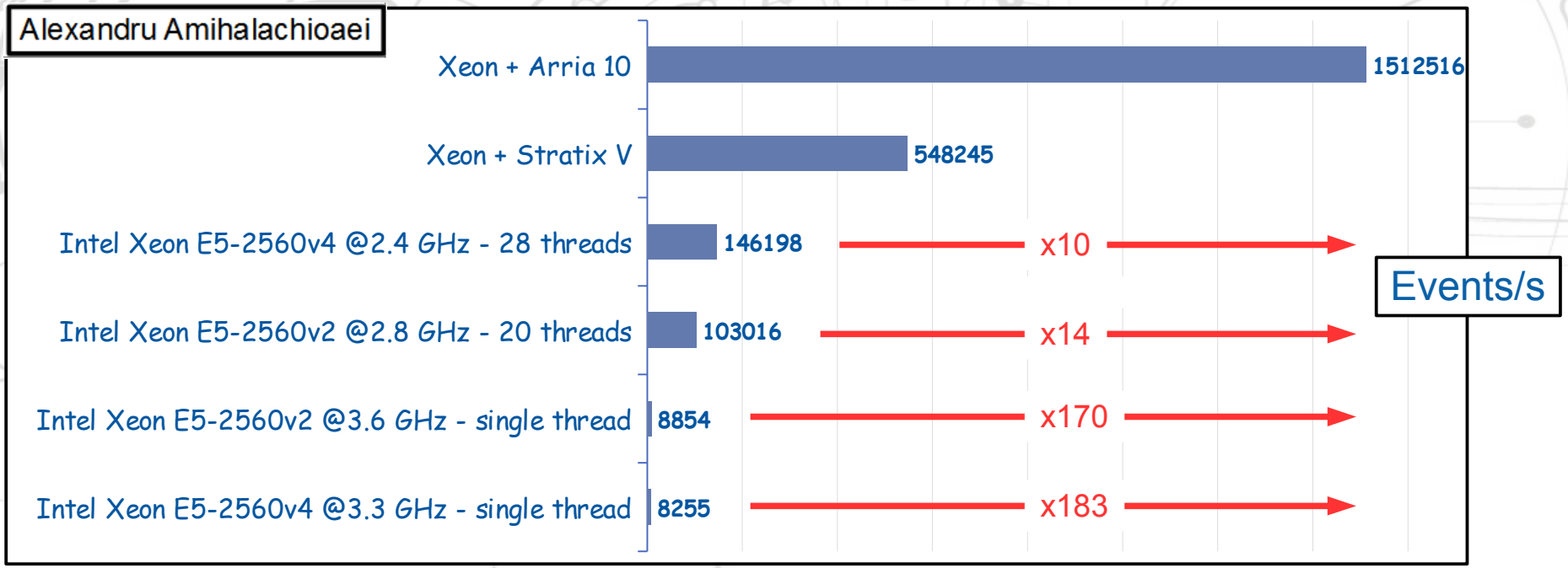
Test case: LHCb Calorimeter Raw Data Decoding

- Two types of calorimeters in LHCb: ECAL/HCAL
- 32 ADC channels for each FEB of 238 FEBs
- Raw data format:
  - ADC data is sent using 4 bits or 12 bits
  - A 32 bit word stores information about which channel has short/long decoding

LHCb Calorimeter raw data bank

Control word (9b) (Figure 18)	Crate (5b)	Card (4b)	Length ADC (7b)	Length trigger (7b)
Trigger bit pattern (32b)				
Zero padding	Trigger (8b)	Trigger (8b)	Trigger (8b)	
ADC bit pattern (32b)				
ADC low	ADC long (12b)	ADC long (12b)	ADC (4b)	
Zero padding at the end	ADC long (12b)	ADC high (8b)		

Results Calorimeter Raw Data Decoding: BDW+Arria10

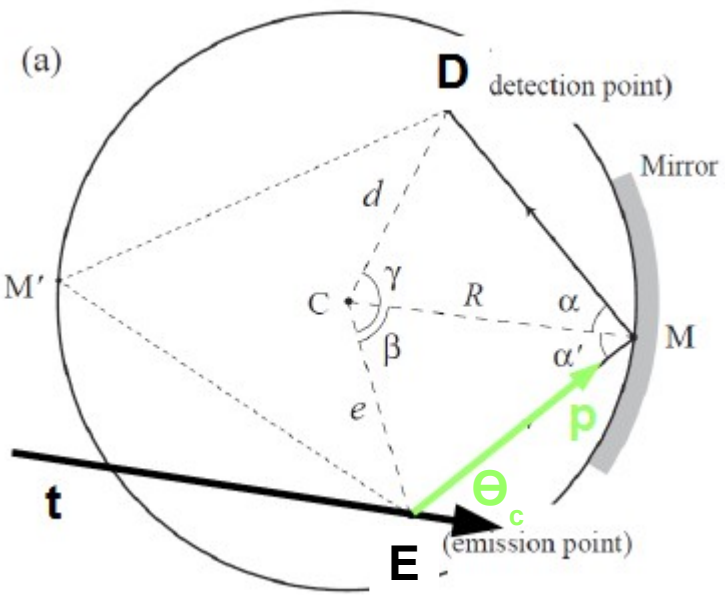


- The higher bandwidth of the newest Intel® Xeon®+FPGA results in an impressive acceleration of a factor 180

FPGA Resource Type	FPGA Resources used [%]	For Interface used [%]
ALMs	57	18
DSPs	0	0
Registers	19	5

Test Case: RICH PID Algorithm

- Calculate Cherenkov angle Θ_c for each track **t** and detection point **D**, not a typical FPGA algorithm
- RICH PID is not processed for every event, processing time is too long!



Reference: LHCb Note LHCb-98-040

Calculations:

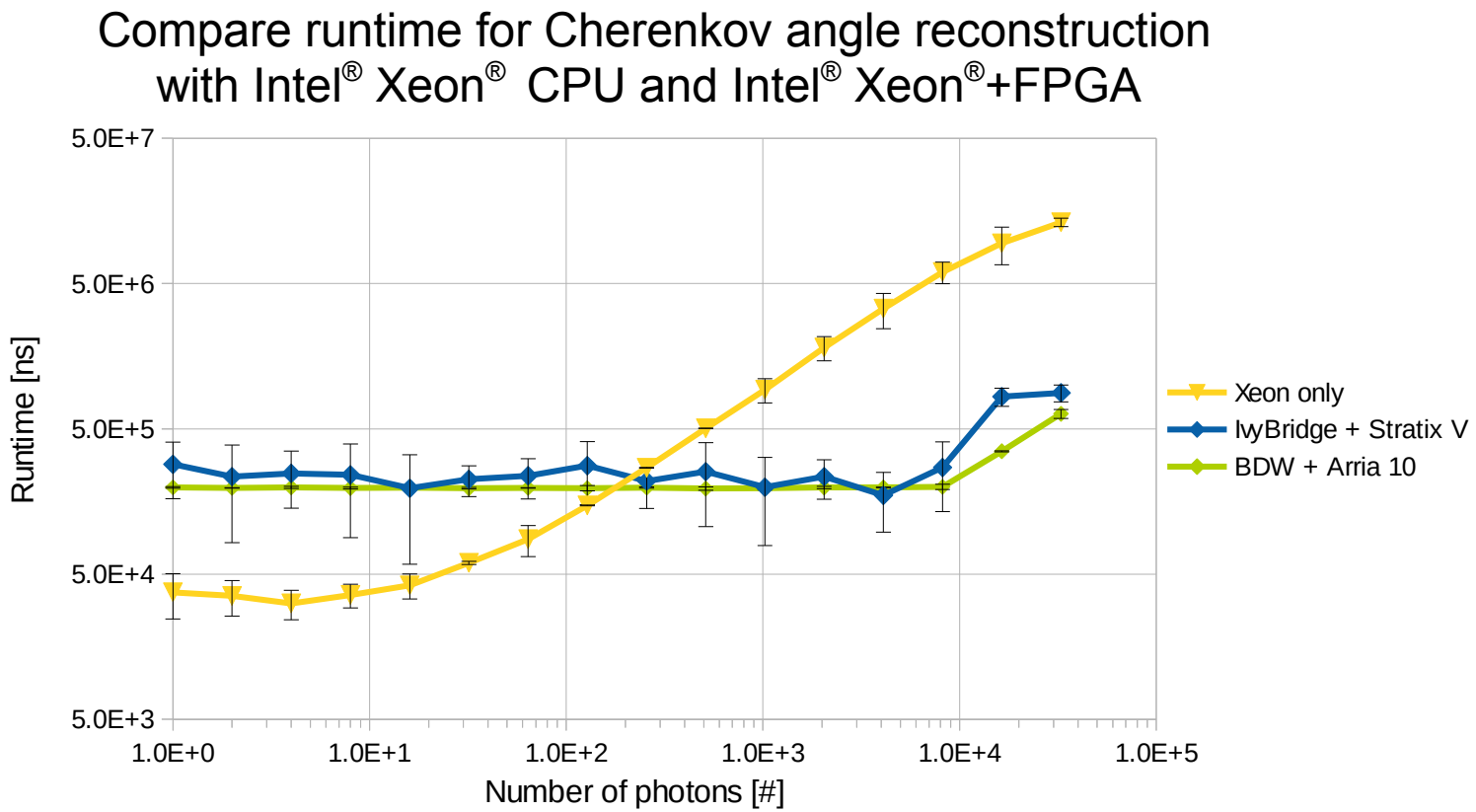
- solve quartic equation
- cube root
- complex square root
- rotation matrix
- scalar/cross products

Implementation of Cherenkov Angle reconstruction Arria 10

- 259 clock cycle long pipeline written in Verilog
  - Stratix V blocks ported using HFB: complex square root, rot. matrix, cross/scalar product,...
- Pipeline running with 200MHz → 5ns per photon
  - With Arria 10 GT FPGA 400 MHz possible
- FPGA resources:

FPGA Resource Type	FPGA Resources used [%]	For Interface used [%]
ALMs	32	18
DSPs (HFBs)	15	0
Registers	12	5

Intel® Xeon®+FPGA Results



- Acceleration of up to factor 35 with Intel® Xeon®+FPGA
- Theoretical limit of photon pipeline: a factor 64 with respect to single Intel® Xeon® thread, for Arria® 10 a factor ~ 300
- Bottleneck: Data transfer bandwidth to FPGA, caching can improve this, tests ongoing

Compare Verilog - OpenCL

- Development time

  2.5 months – 2 weeks

  3400 lines Verilog – 250 lines C

- Performance

  Cube root : x35 – x30

  RICH : x35 – x26

- FPGA resource usage Stratix® V

RICH Kernel

FPGA Resource Type	Verilog RTL FPGA Resources used [%]	OpenCL FPGA Resources used [%]
ALMs	88	63
DSPs	67	82
Registers	48	24

OpenCL

Faster

Easier

Comparable performance

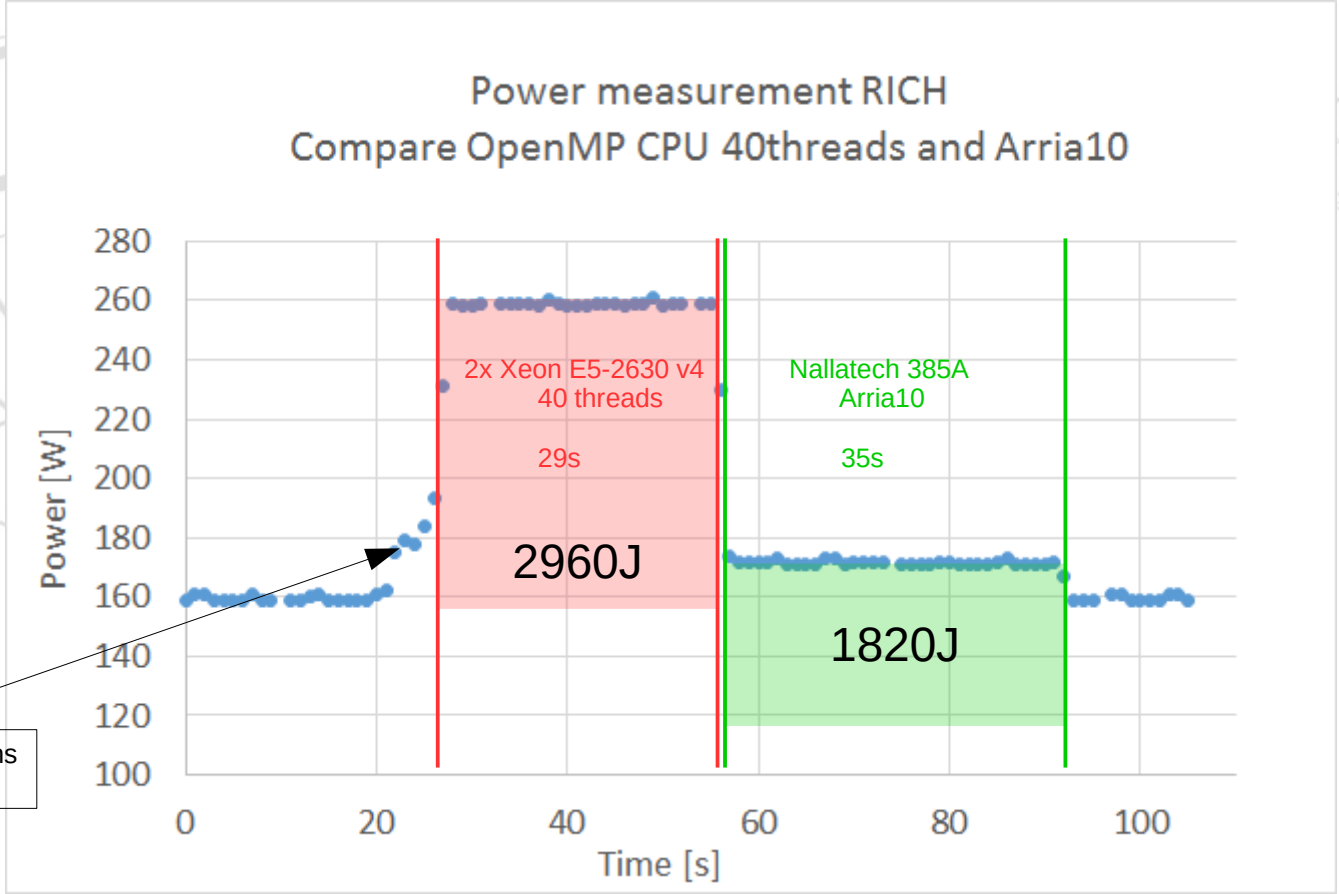
Similar resource usage

Nallatech 385A Board

- FPGA: Intel® Arria® 10 GX 1150 FPGA
  - 427'200 ALMs, 1'708'800 Registers
  - 1'518 DSPs
- Programming model: OpenCL
- Host Interface: 8-lane PCIe Gen3
  - Up to 7.9 GB/s
- Memory: 8 GB DDR3 SDRAM
- Network Enabled with (2) QSFP 10/40 GbE ports
- Power usage: full FPGA firmware ~ 40 W

(CERN techlab)

RICH with Nallatech 385A

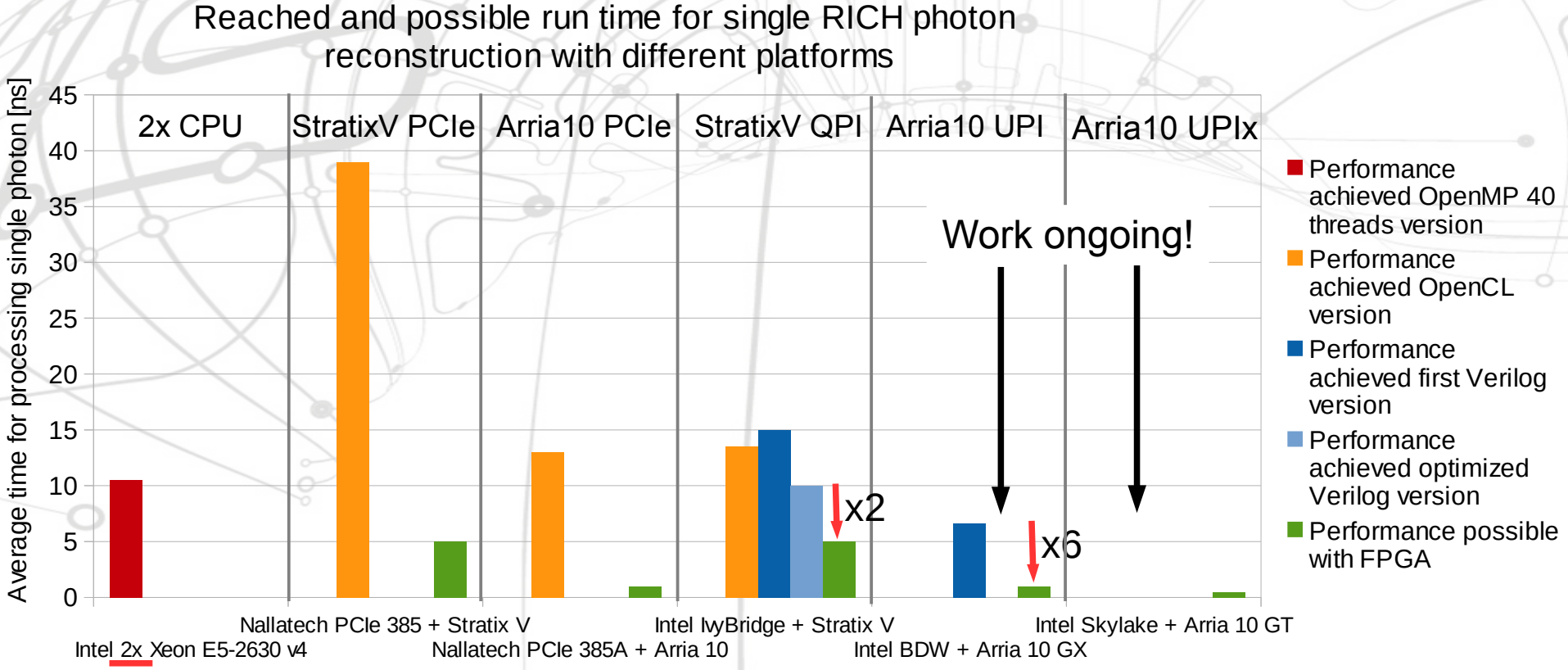


Create random photons
single thread

16777216 random photons
Multi loop factor: 160
Used CPU threads: 40

FPGA uses 1.6x less energy

Reached and possible run time for RICH photon reconstruction



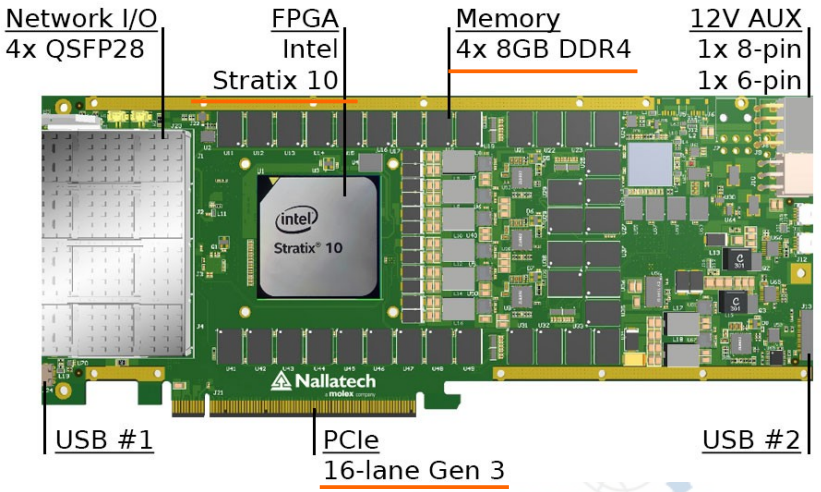
- The difference between reached and possible time is due to the limitation by the bandwidth between CPU and FPGA, in both cases the FPGA could process the photons faster. The same case is with the PCIe accelerator, but even worse
- The bandwidth gap could be reduced by caching, for RICH kernel possible
- Between Ivy Bridge and BDW the bandwidth improved by a factor 2

Future Tests

- Implement additional CERN algorithms
  - Tracking - Kalman filter, CNNs
  - Christoph Hasse works on Velo tracking
- Compare performance with Intel® Xeon®+FPGA system with Skylake + Arria® 10 FPGA
  - Waiting for missing software and firmware
  - Power measurements
- Longterm

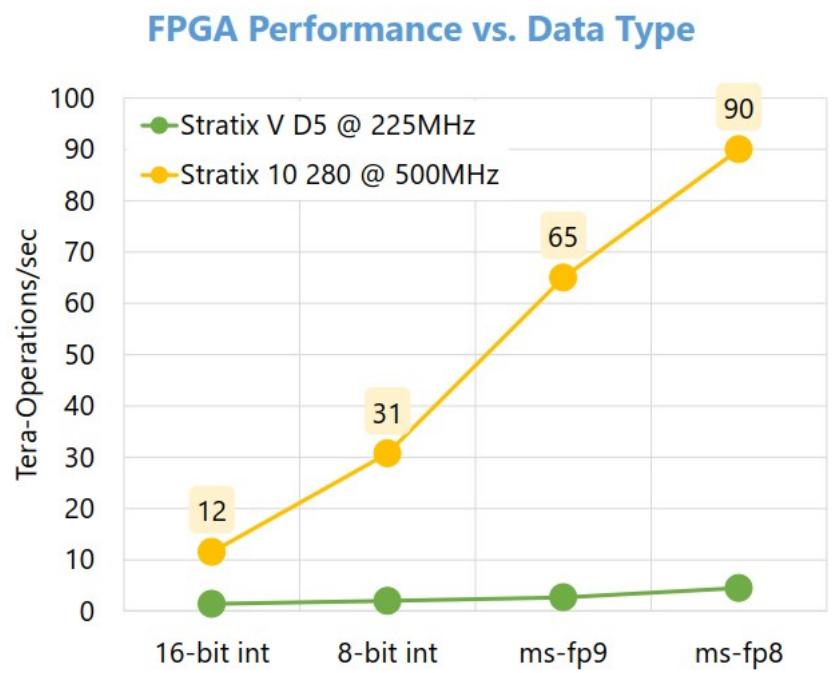
  Measurements of Stratix10 PCIe accelerators and Intel® Xeon® + Stratix10

Nallatech:520 ~10 TFlops



Optimizations for CNN Inference

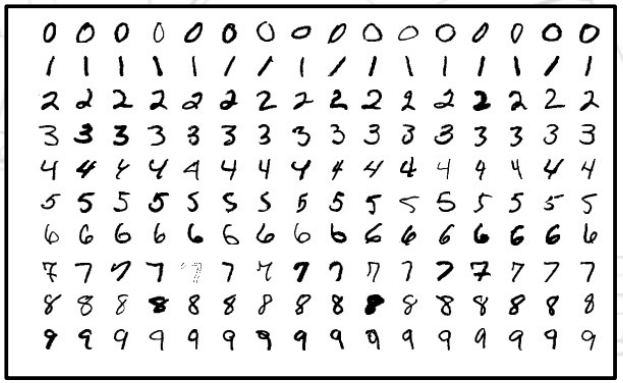
- Pruning
- Quantization
- Advantage of using precision as needed on FPGAs
- For FPGAs BNNs very interesting



Source: FPGA Datacenters - The New Supercomputer, Andrew Putnam – Microsoft Catapult_ACAT_2017_Public

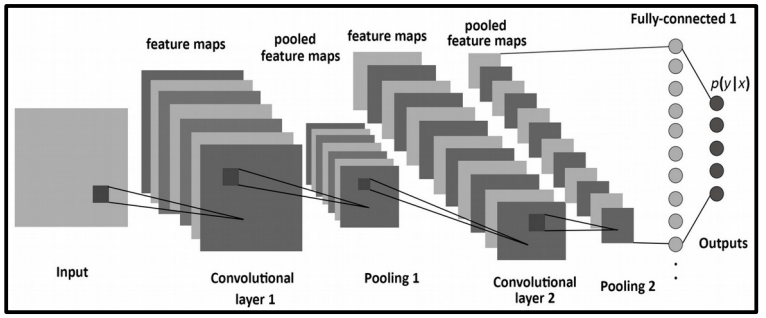
Work done already at CERN

- MNIST optimization for FPGA inference
  - Weights 32bit → 11bit → 2bits?
  - Block RAM memory architecture and adder multiplier optimization



- CMS is investigating CNNs on FPGAs for future L1 Trigger

https://indico.cern.ch/event/686137/attachments/1575876/2488495/Harris_PCH_FPGA_ML_14_12.pdf



- FPGA compute acceleration interesting for Monte Carlo production (e.g.Geant V)

https://indico.cern.ch/event/567550/timetable/#20170824.detailed

FPGA development

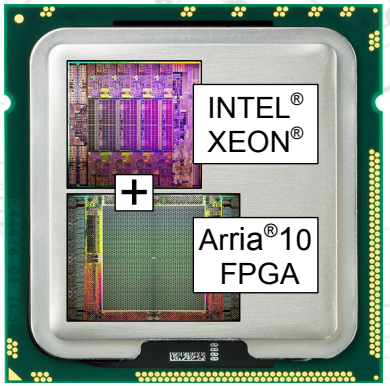
- FPGA potential for general compute acceleration increased a lot with Arria10 and the hardened floating point DSP blocks
  - Future FPGAs will have sev. 10'000 of these DSPs (nowadays already ~6k)
- FPGA transceivers will make huge bandwidth into chip possible, tightly coupled to RAM
- Programming model is changing now to using mostly HLS and OpenCL even for standard FPGA designs
  - Intel recommends to use HLS for Stratix10

Challenges to use FPGA accelerators

- Compute heavy blocks have to be identified to be ported to the FPGA
- For PCIe accelerators an off-load model is used (larger latency)

  → Intel® Xeon® + FPGA advantage (streaming)
- Kernel size limited by FPGA resources
  - Intel will change programming time from O(s) to O(us) in the future, which makes kernel swapping during runtime practical
-

Summary

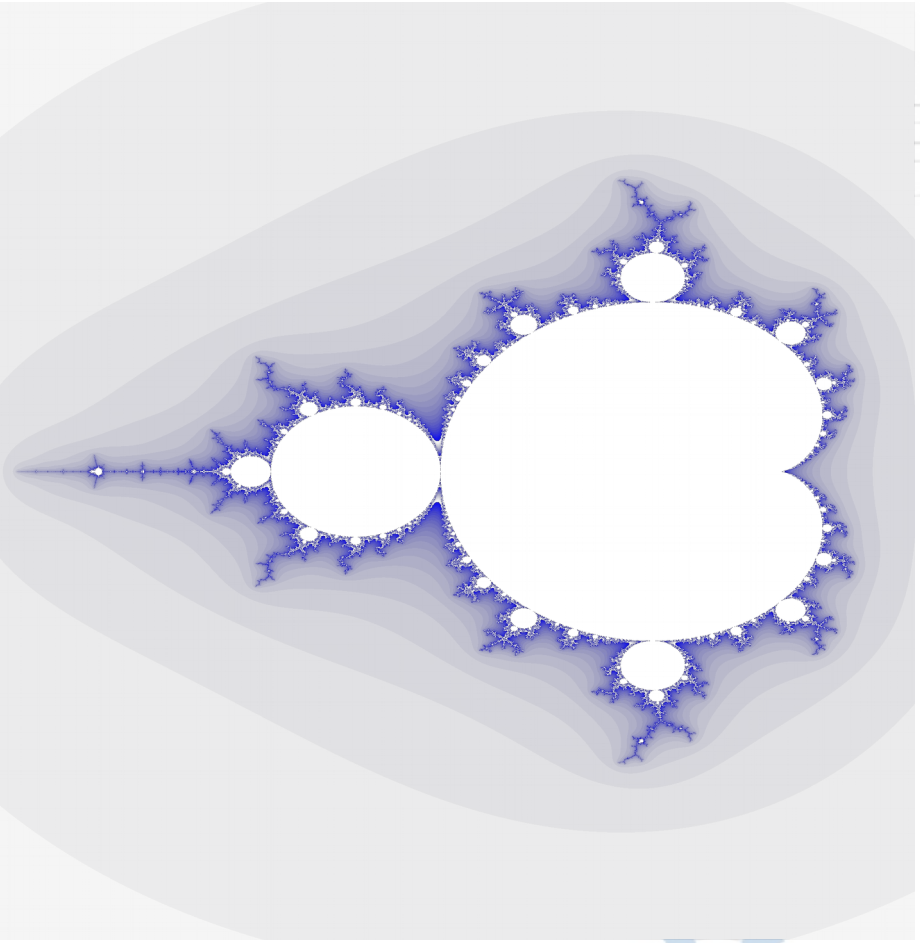


- Results are very encouraging to use FPGA acceleration in the HEP field
- Comparing the energy consumption with CPUs show better performance for FPGAs (getting a greener CERN computing ?)
- Programming model with OpenCL very attractive and convenient for HEP field, HLS now also available
- Also other experiments want to test the usage of the Intel® Xeon®+FPGA with Arria10
- High bandwidth interconnect coupled with Arria® 10 FPGA suggests excellent performance per Joule for HEP algorithms! Don't forget Stratix® 10 … !

Thank you

Mandelbrot on Intel® Xeon®+FPGA

- Mandelbrot with floating point precision
  - Implemented 22 fpMandel pipelines running at 200 MHz, each handles 16 pixels in parallel (total: 352 pixels)
  - FPGA is x12 faster than Intel® Xeon® running 20 threads in parallel
  - Used 72/256 DSPs
  - Reuse of data on FPGA high



Sorting with Intel® Xeon®+FPGA

- Sorting of INT arrays with 32 elements
  - Implemented pipeline with 32 array stages
  - FPGA sort is up to x117 faster than single Xeon® thread
  - Bandwidth through the FPGA is the bottleneck