www.fpgafais.com

Concepts for PANDA DAQ system

Dr Grzegorz Korcyl

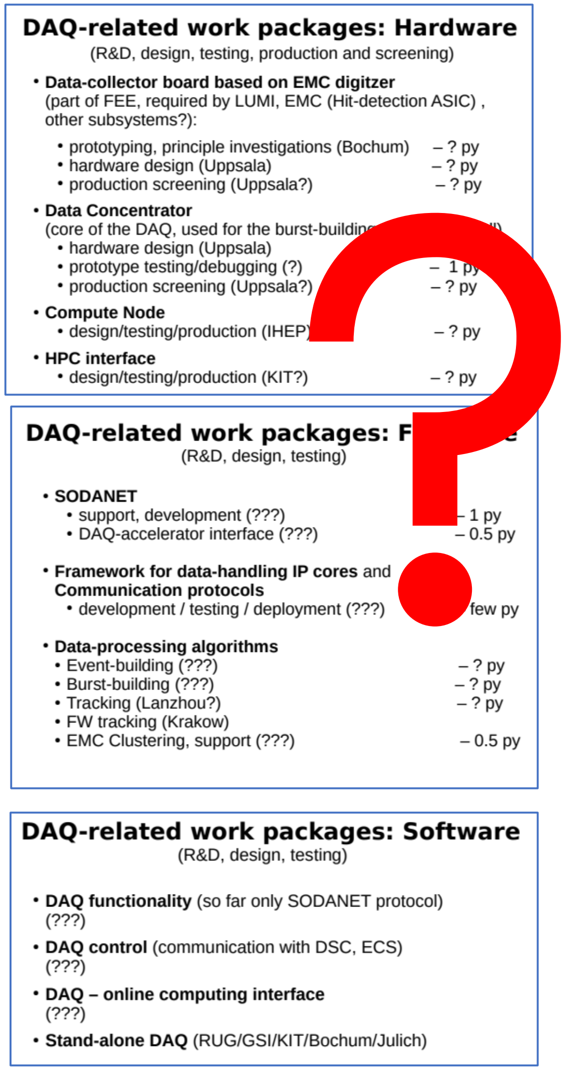
Department of Information Technologies

Jagiellonian University, Cracow

9 December 2019, Kraków

PANDA DAQ problems

- Constant shortage of manpower
- Slow, expensive and risky hardware development cycles
- Many custom hardware components, standards, protocols
  - In fact simillar boards: many links, powerful FPGA, differences in memory, clocking, system interfaces
- Time-scale, time-shifts, delays



DAQ-related work packages: Hardware
(R&D, design, testing, production and screening)

- **Data-collector board based on EMC digitzer**
  (part of FEE, required by LUMI, EMC (Hit-detection ASIC) , other subsystems?):
  - prototyping, principle investigations (Bochum) – ? py
  - hardware design (Uppsala) – ? py
  - production screening (Uppsala?) – ? py
- **Data Concentrator**
  (core of the DAQ, used for the burst-building [illegible]l)
  - hardware design (Uppsala)
  - prototype testing/debugging (?) – 1 py
  - production screening (Uppsala?) – ? py
- **Compute Node**
  - design/testing/production (IHEP) – ? py
- **HPC interface**
  - design/testing/production (KIT?) – ? py

DAQ-related work packages: F[illegible]e
(R&D, design, testing)

- **SODANET**
  - support, development (???) – 1 py
  - DAQ-accelerator interface (???) – 0.5 py
- **Framework for data-handling IP cores** and **Communication protocols**
  - development / testing / deployment (???) few py
- **Data-processing algorithms**
  - Event-building (???) – ? py
  - Burst-building (???) – ? py
  - Tracking (Lanzhou?) – ? py
  - FW tracking (Krakow)
  - EMC Clustering, support (???) – 0.5 py

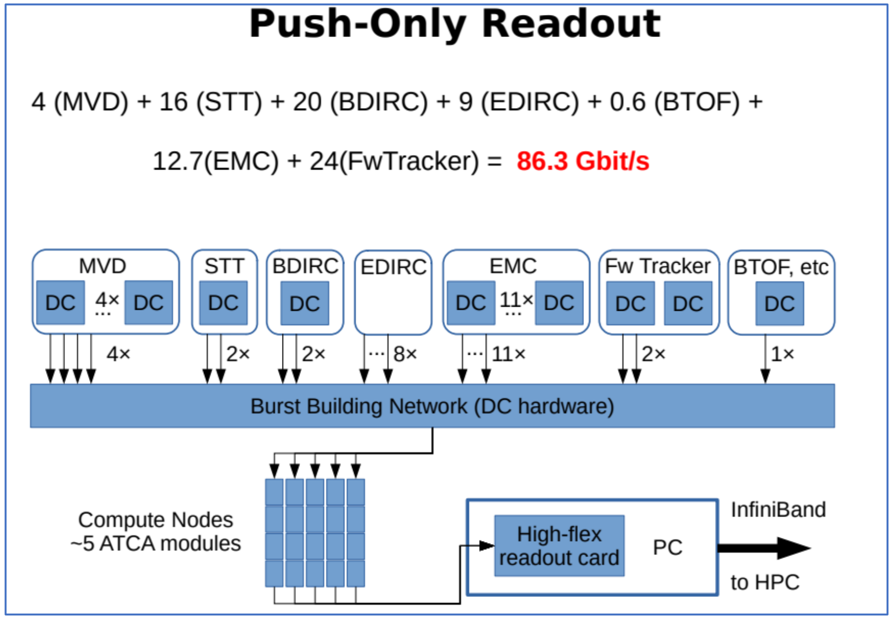
DAQ-related work packages: Software
(R&D, design, testing)

- **DAQ functionality** (so far only SODANET protocol)
  (???)
- **DAQ control** (communication with DSC, ECS)
  (???)
- **DAQ – online computing interface**
  (???)
- **Stand-alone DAQ** (RUG/GSI/KIT/Bochum/Julich)

M. Kavatsyuk

Custom elements mosaic

- Hardware
  - Subsystems FEE
  - Data Concentrators
  - SODANet distribution
  - Compute Nodes
  - High-Flex cards
- Protocols
  - Subsystem internal
  - SODANet
  - Raw 8/10b
  - InfiniBand
- Systems, interfaces
  - Standalone
  - ATCA
  - AMC
  - PCIe



Push-Only Readout
4 (MVD) + 16 (STT) + 20 (BDIRC) + 9 (EDIRC) + 0.6 (BTOF) +
12.7(EMC) + 24(FwTracker) = 86.3 Gbit/s
MVD
DC 4× DC
STT
DC
BDIRC
DC
EDIRC
EMC
DC 11× DC
Fw Tracker
DC DC
BTOF, etc
DC
4×
2×
2×
8×
11×
2×
1×
Burst Building Network (DC hardware)
Compute Nodes ~5 ATCA modules
High-flex readout card
PC
InfiniBand
to HPC

M. Kavatsyuk

- Firmware/software
  - Subsystem preprocessing
  - Protocols stacks
  - Burst building
  - Analysis algorithms
  - Drivers

Development issues

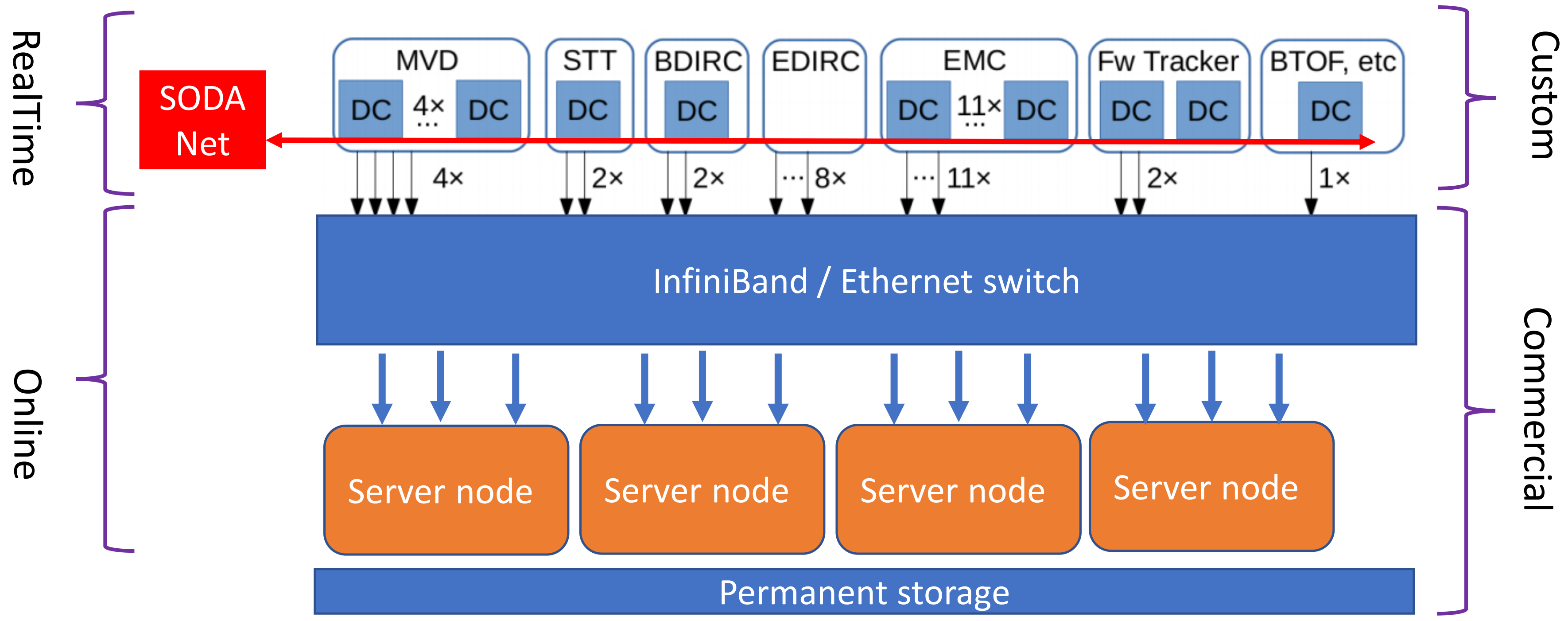
- Limited numer of experts in both HW and FPGA development
- Hardware:
  - Long development cycles
    - Expensive, time consuming iterations
  - Risky
    - Withdrawn, discontinued technologies (HMC on DC!)
    - Long-term investment (PANDA Day-0 still couple years ahead)
- Firmware:
  - Complex algorithms are hard to migrate to HDL
  - Long debugging cycles
  - Difficult evaluation

Proposed solution

- Escape to commercial hardware at the earliest stage
  - Long-term manufacturer support
  - Support from experts/community
  - Easily upgreadable with new HW/SW releases

- Select technologies with high-level development
  - Avoid reinventing the wheel (protocols, data handling, etc.)
  - Focus on algorithmics
  - Merge offline and online analysis procedures
  - Accelerate development and debugging cycles
  - Involve non-experts into the developers group

Proposed DAQ scheme

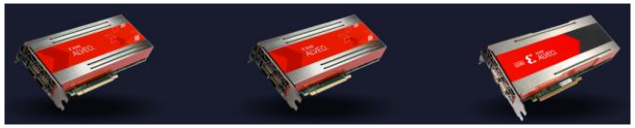


- Predicted 100 Gbps is not a challenge anymore
- Fast technology development in HPC sector in networking and processing

- DC:
  - SODANet distribution / slow control
  - Data assembly / zero suppersion / filtration / compression
  - Common routing protocol application (IB, Eth)
- Server nodes:
  - Complete SuperBursts assembly
  - High-level algorithms online but not in real-time

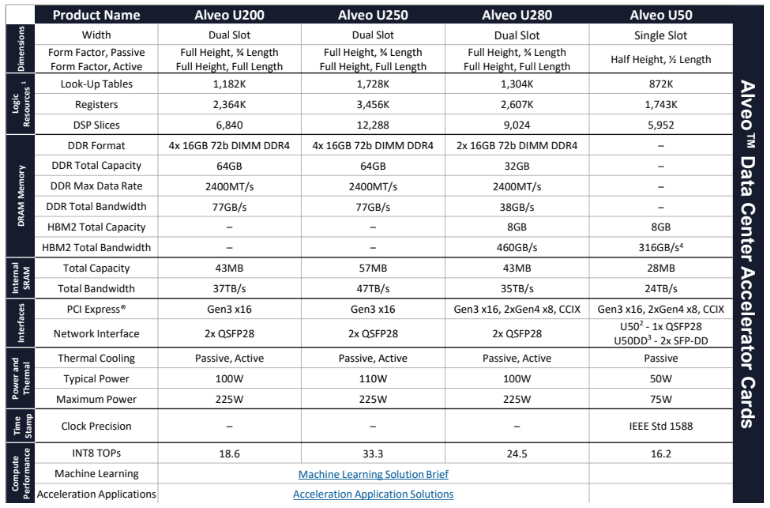
Server nodes

- Server nodes equipped with commercial but powerful hardware (CPU/GPU/FPGA)
  - No requirement to work in real-time regime
  - Easily scalable, easily upgradeable and easy to maintain
  - Trend in experiments: CBM, Alice
- Growing protfolio of FPGA, PCIe accelerator cards, e.g. Xilinx Alveo



- Complete package:
  - Hardware:
    - Powerful FPGA (Virtex U+ variants)
    - Embedded HBM/external DDR4
    - QSFP28 interfaces
    - PCIe Gen3 x16, CCIX
  - Firmware/Software:
    - basic FW shell
    - SW drivers
    - OpenCL support

Xilinx.com

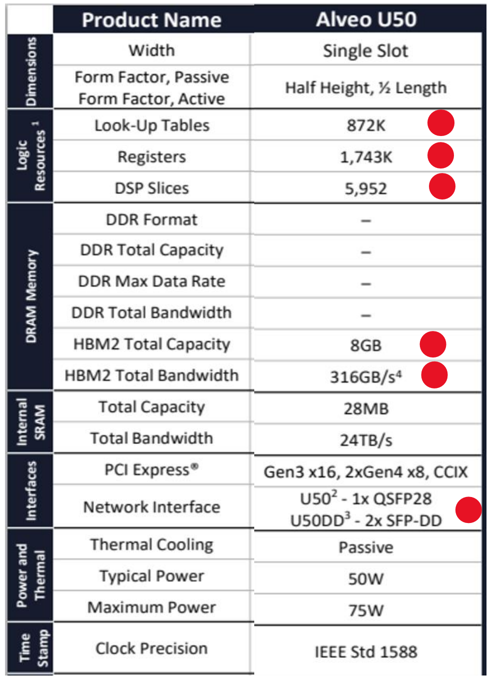


Alveo™ Data Center Accelerator Cards

	Product Name	Alveo U200	Alveo U250	Alveo U280	Alveo U50
Dimensions	Width	Dual Slot	Dual Slot	Dual Slot	Single Slot
	Form Factor, Passive Form Factor, Active	Full Height, ¾ Length Full Height, Full Length	Full Height, ¾ Length Full Height, Full Length	Full Height, ¾ Length Full Height, Full Length	Half Height, ½ Length
Logic Resources [1]	Look-Up Tables	1,182K	1,728K	1,304K	872K
	Registers	2,364K	3,456K	2,607K	1,743K
	DSP Slices	6,840	12,288	9,024	5,952
DRAM Memory	DDR Format	4x 16GB 72b DIMM DDR4	4x 16GB 72b DIMM DDR4	2x 16GB 72b DIMM DDR4	–
	DDR Total Capacity	64GB	64GB	32GB	–
	DDR Max Data Rate	2400MT/s	2400MT/s	2400MT/s	–
	DDR Total Bandwidth	77GB/s	77GB/s	38GB/s	–
	HBM2 Total Capacity	–	–	8GB	8GB
	HBM2 Total Bandwidth	–	–	460GB/s	316GB/s[4]
Internal SRAM	Total Capacity	43MB	57MB	43MB	28MB
	Total Bandwidth	37TB/s	47TB/s	35TB/s	24TB/s
Interfaces	PCI Express®	Gen3 x16	Gen3 x16	Gen3 x16, 2xGen4 x8, CCIX	Gen3 x16, 2xGen4 x8, CCIX
	Network Interface	2x QSFP28	2x QSFP28	2x QSFP28	U50[2] - 1x QSFP28 U50DD[3] - 2x SFP-DD
Power and Thermal	Thermal Cooling	Passive, Active	Passive, Active	Passive, Active	Passive
	Typical Power	100W	110W	100W	50W
	Maximum Power	225W	225W	225W	75W
Time Stamp	Clock Precision	–	–	–	IEEE Std 1588
Compute Performance	INT8 TOPs	18.6	33.3	24.5	16.2
	Machine Learning	Machine Learning Solution Brief			
	Acceleration Applications	Acceleration Application Solutions			

Alveo advantages

- Costs
  - No HW development costs
  - Logic resources / $
  - Example: Alveo U50 – **2 500$**
    - Similar resources (Digikey, depends on speed grade):
      - Virtex U+ XCVU33P – **20 000 - 40 000$** (0.9 MLUTS, 2880 DSP, 8 GB HBM)
    - Similar price:
      - Kintex U XCKU040 – **2 500$** (0.8 MLUTs, 1970 DSP, no HBM)
    - FPGA on Data Concentrator:
      - Kintex U+ XCKU15P – **4 000 – 8 000$** (1.1 MLUTs, 1970 DSP, no HBM)
    - FPGA on High-Flex:
      - Virtex 7 XC7V330T-2 – **4 000 $** (0.3 MLUTs, 1120 DSP, no HBM)



	Product Name	Alveo U50	
Dimensions	Width	Single Slot	
	Form Factor, Passive Form Factor, Active	Half Height, ½ Length	
Logic Resources [1]	Look-Up Tables	872K	●
	Registers	1,743K	●
	DSP Slices	5,952	●
DRAM Memory	DDR Format	–	
	DDR Total Capacity	–	
	DDR Max Data Rate	–	
	DDR Total Bandwidth	–	
	HBM2 Total Capacity	8GB	●
	HBM2 Total Bandwidth	316GB/s[4]	●
Internal SRAM	Total Capacity	28MB	
	Total Bandwidth	24TB/s	
Interfaces	PCI Express®	Gen3 x16, 2xGen4 x8, CCIX	
	Network Interface	U50[2] - 1x QSFP28 U50DD[3] - 2x SFP-DD	●
Power and Thermal	Thermal Cooling	Passive	
	Typical Power	50W	
	Maximum Power	75W	
Time Stamp	Clock Precision	IEEE Std 1588	

P. Marciniewski, „A data concentrator for the PANDA experiment", 2019

M. Caselle, „High-speed, low-latency readout system with real-time trigger based on GPUs", 2016

Alveo advantages

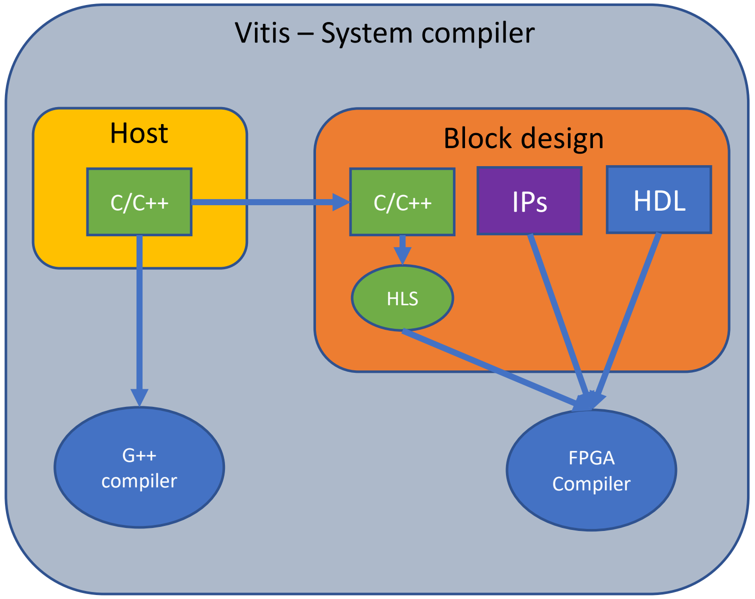
- Integrated development environment
  - Entire project in C/C++
    - **Abstract code structures (structures, classes)**
    - **OO mechanisms: operators overload, inheritance, templates**
    - **No dynamic memory management (on HW)**

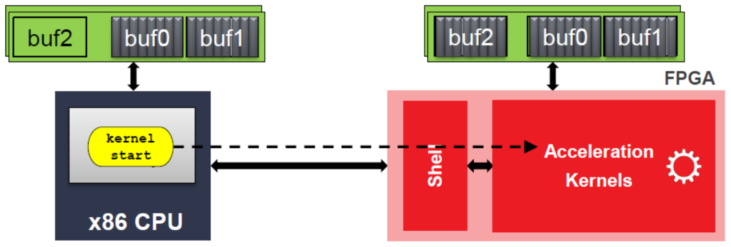
  - **Incorporate HDL IP Cores**

  - Host <–> Kernel architecture
  - Main function – starting point on the host
  - Kernel – hardware accelerated function

  - **Algorithm debugging on C++ level**

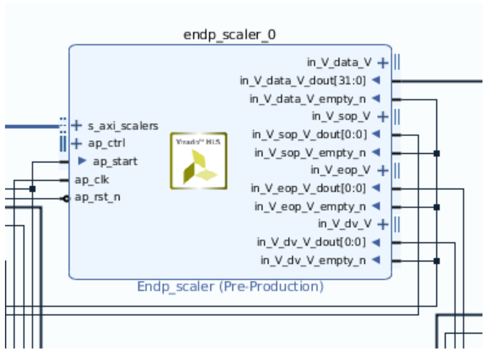
  - Encapsulates all other tools and compilers
  - Multi-level evaluation: g++, hw emulation, hw

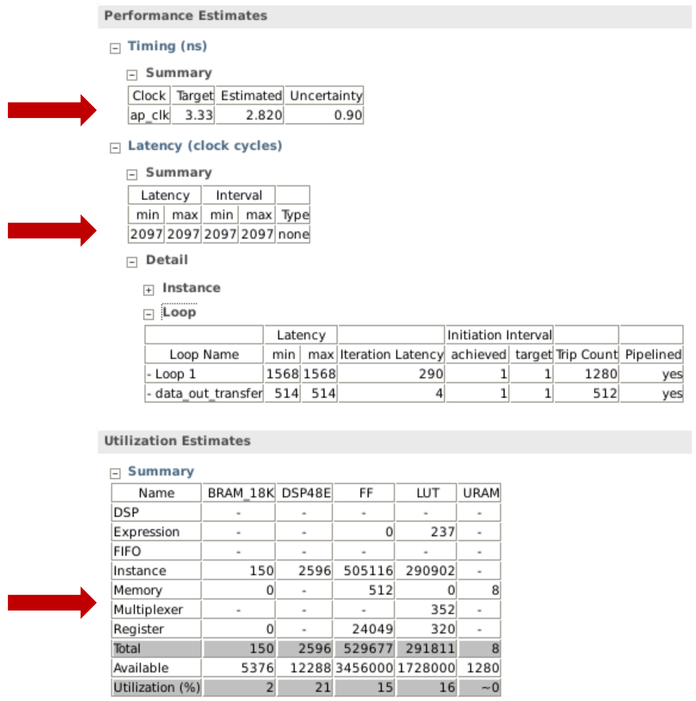




Development in HLS

- Single function – single component
  - Function arguments become component interface
  - Function body translated into logic
  - Results analysis with a set of reports
    - Timings, resources
  - Compilation process controlled with a set of #pragmas
  - **High level component evaluation**





Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	3.33	2.820	0.90

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
2097	2097	2097	2097	none

Detail

Instance

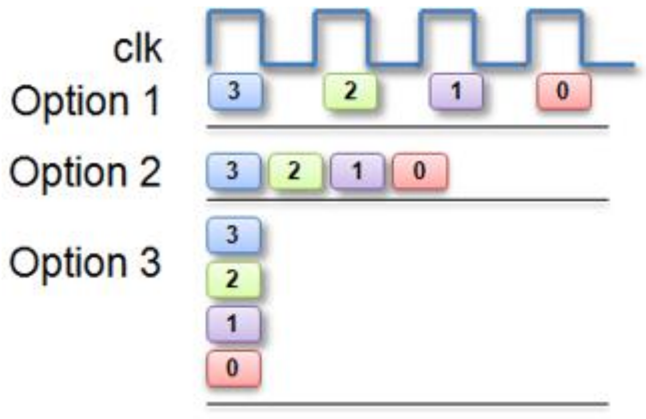
Loop

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Loop 1	1568	1568	290	1	1	1280	yes
- data_out_transfer	514	514	4	1	1	512	yes

Utilization Estimates

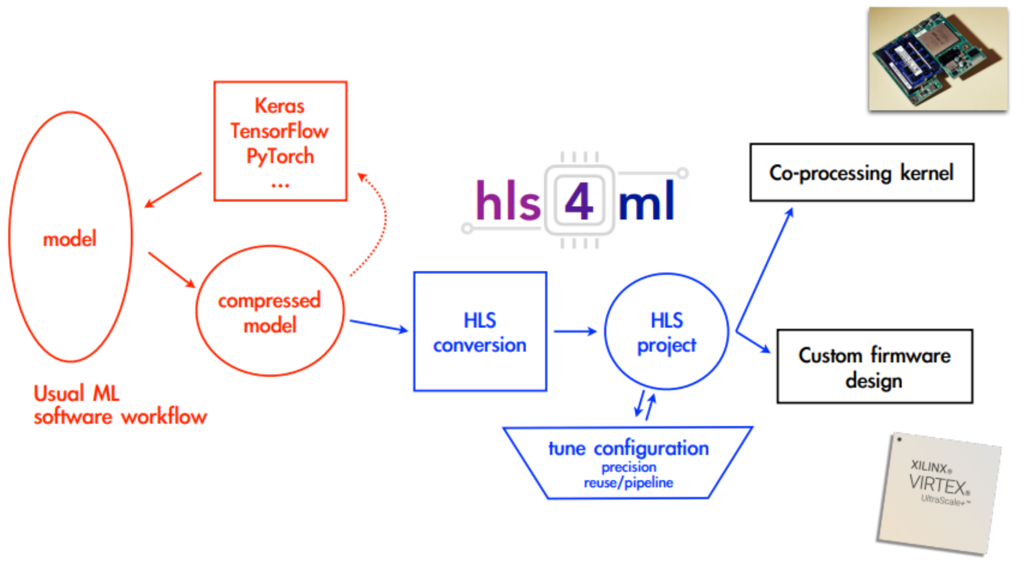
Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	237	-
FIFO	-	-	-	-	-
Instance	150	2596	505116	290902	-
Memory	0	-	512	0	8
Multiplexer	-	-	-	352	-
Register	0	-	24049	320	-
Total	150	2596	529677	291811	8
Available	5376	12288	3456000	1728000	1280
Utilization (%)	2	21	15	16	~0



HLS Example

- Entire flow for Neural Network implementation on FPGA within a single HDL module
- Advantages: fixed latency, controlled level of parallelism, custom data types
- Used for example in L1 trigger at CMS CERN „Artificial Intelligence Accelerates Dark Matter Search", Xilinx Case Study 2019
- Import trained NN model to FPGA with just few mouse clicks
- MNIST network example implemented by students during FPGA Summer Camp 2019



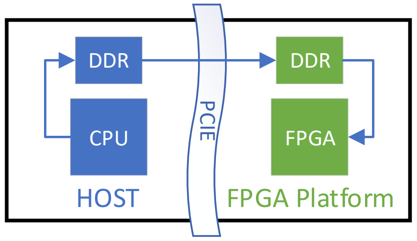
Python C++ Bitstream

hls4ml

Alveo application example

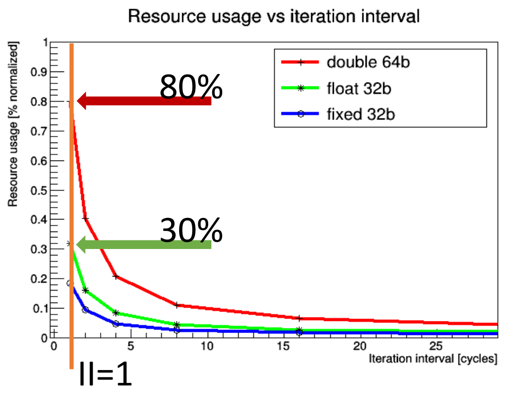
- Conjugate Gradient as HPC benchmark
    - Host prepares data sets and streams to the accelerated kernel
    - 1464 floating point (IEEE754) numerical operations per single set
    - Kernel implemented with II=1, latency 142 at 300 MHz
    - **2x faster than Intel Xeon Phi 64-core, 1.7 GHz**
    - **Reaching performance of Nvidia V100 (only 15% of PP!)**

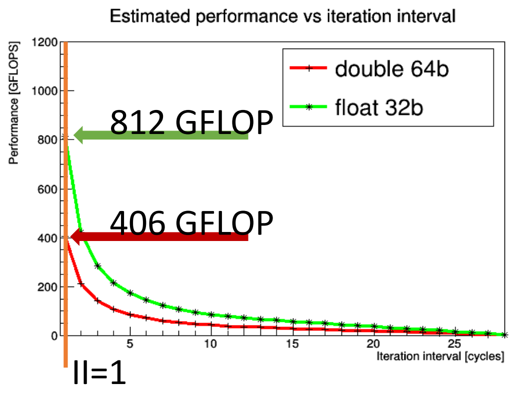
    - **Not a single HDL line written**

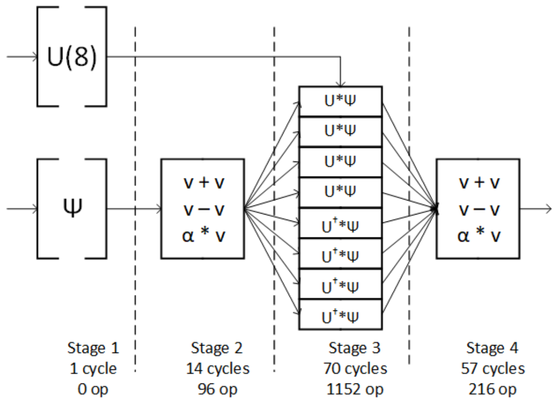


S. Durr, *„Three Dirac operators on two architectures with one piece of code and no hassle”*, LATTICE2018, arXiv:1808.05506v2, Nov. 2018

E. Berkowitz, et. Al.. „Simulating the weak death of the neutron in a femtoscale universe with near-exascale computing”, 2019



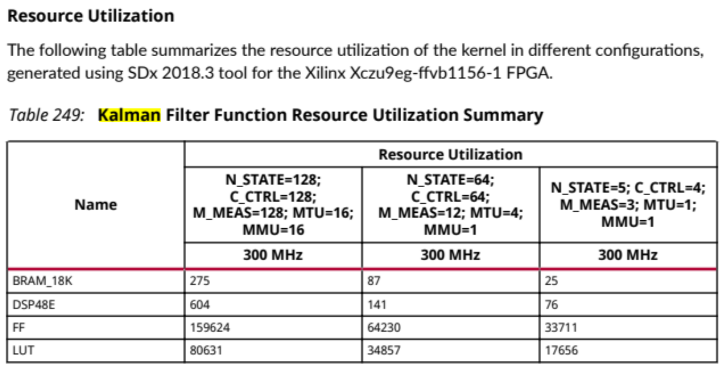




1464 operations
Within 142 clock cycles
With II=1

Algorithms on Alveo

- Many data analysis algorithms already developed by physicists
  - Migrate from C/C++/Python/Root/PandaRoot to C++/OpenCL
  - C++/OpenCL runs also on GPUs – interesting research
  - Can be done by people without strong FPGA background
  - Easy and fast evaluation with simulated/collected data
  - Growing library of HW-optimised libraries/functions
    - E.g. OpenCV – Kalman Filter
  - **Easily maintenable**
  - **Easily upgradeable**



Resource Utilization

The following table summarizes the resource utilization of the kernel in different configurations, generated using SDx 2018.3 tool for the Xilinx Xczu9eg-ffvb1156-1 FPGA.

Table 249: **Kalman Filter Function Resource Utilization Summary**

Name	Resource Utilization		
	N_STATE=128; C_CTRL=128; M_MEAS=128; MTU=16; MMU=16	N_STATE=64; C_CTRL=64; M_MEAS=12; MTU=4; MMU=1	N_STATE=5; C_CTRL=4; M_MEAS=3; MTU=1; MMU=1
	300 MHz	300 MHz	300 MHz
BRAM_18K	275	87	25
DSP48E	604	141	76
FF	159624	64230	33711
LUT	80631	34857	17656

- Hardware available for tests on cloud providers (Nimbix, Amazon)
- Cluster available at ETH Zurich (registration required)
- **Xilinx will soon open Alveo cluster for universities – early access for UJ**

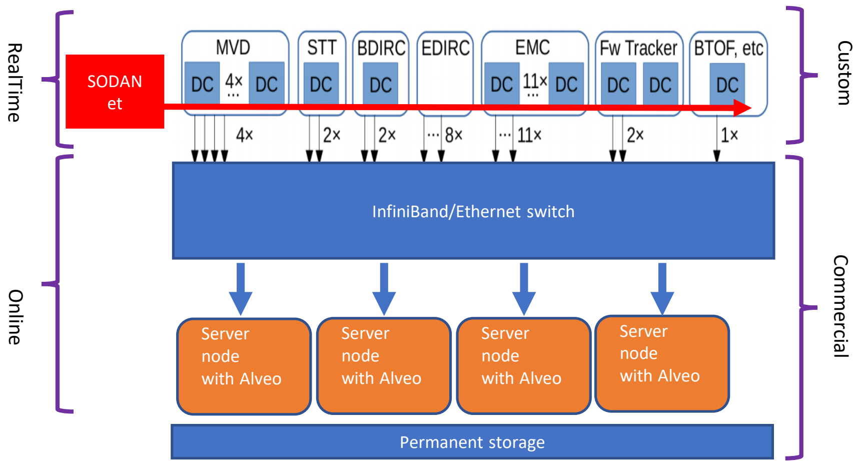
Proposed PANDA DAQ

- Keep subsystems FEE, synchronized with SODANet

- Exit the real-time path on Data Concentrator level and enter standard network

- Push-only architecture – total 100 Gbit/s
  - Commercial network and switch
  - Route SuperBurst fragments to Alveo farm

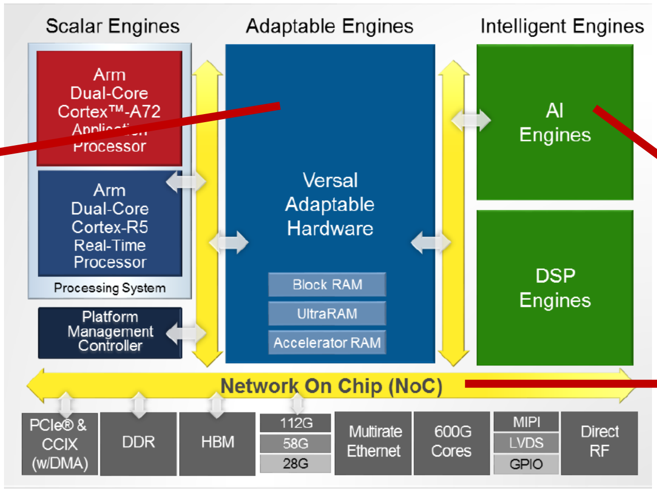
- Alveo farm
  - Accumulate data in HBM and reassemble SB
  - Perform complex algorythmics
  - Store the results



What is next to come

- 7nm Versal Architecture

General CLB resources:
- 4x larger blocks (32 LUTs)
- Less routing
- Higher frequencies



Interconnected vector SIMD:
- 1GHz frequency
- 512b floating point vector
- Local/shared memory
- Streamlined designs

High bandwidth interconnect

Xilinx.com

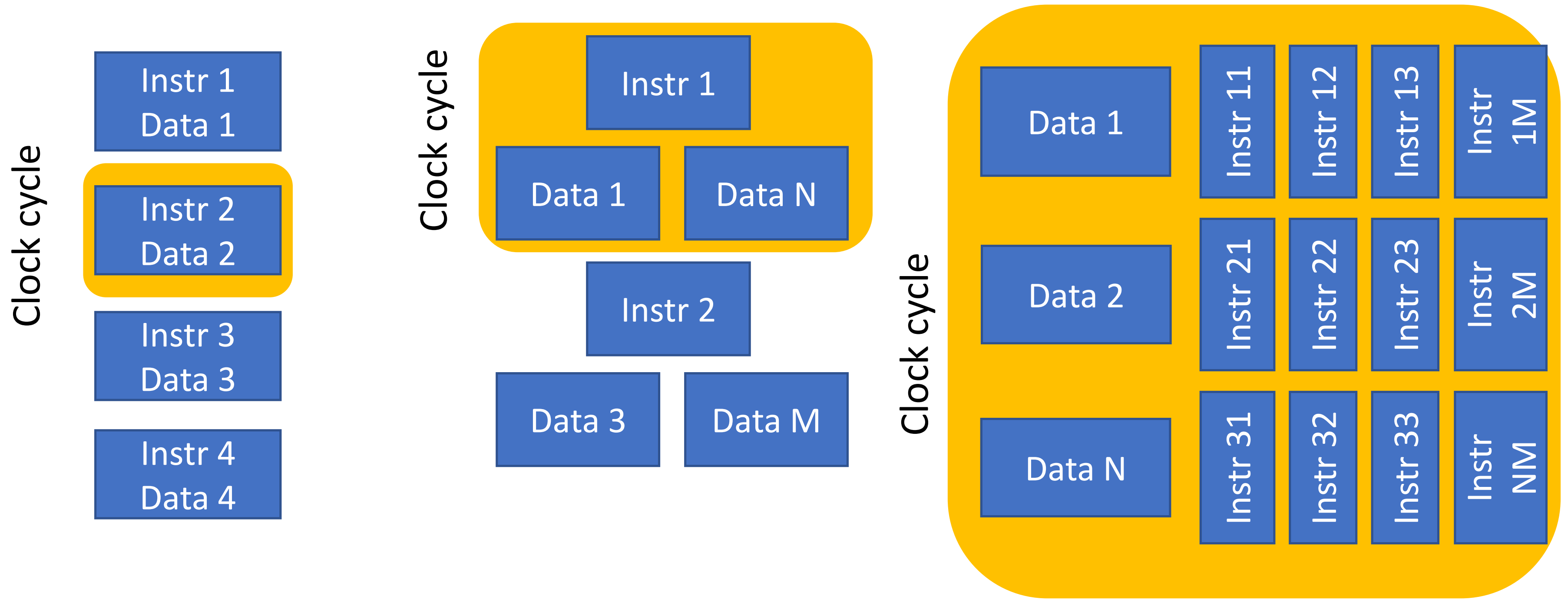
- New Alveo cards will be released soon: 3 versions per year

Research plans

- Continuation of research on Conjugate Gradient on FPGA
  - 3 published papers, 1 under review, application for ISC Frankfurt 2020
  - Strong support from Xilinx
  - Early access to software and hardware
- Application for a scientific grant
  - Development of an energy-efficient and adaptable supercomputing module for HPC
  - Dedicated for establishment of young research groups
  - Decision Q3/2020, starting Q1/2021, duration 2 years
  - 350 000 EUR (60 000 EUR for hardware)
  - 4 team members
- Evaluation of algorithms for PANDA
  - Migration of FT tracking based on Kalman filtering to HLS and OpenCL
    - Evaluation on Alveo, performance comparisons to GPU
    - Reactivate research group at the Technical University in Cracow

Backups

CPU vs GPU vs FPGA



- CPU
  - Single Instruction Single Data per core
  - Fixed instruction set
  - Multiple cores
  - High clock freq.
  - Operating system
- GPU
  - Single Instruction Multiple Data
  - Fixed instruction set
  - High clock freq.
  - Memory access
  - Accelerates CPU
- FPGA
  - Flexible architecture
  - Massive parallelism
  - Streamlined processing
  - Low clock freq.
  - Instant memory access
  - Standalone platforms

Different approach

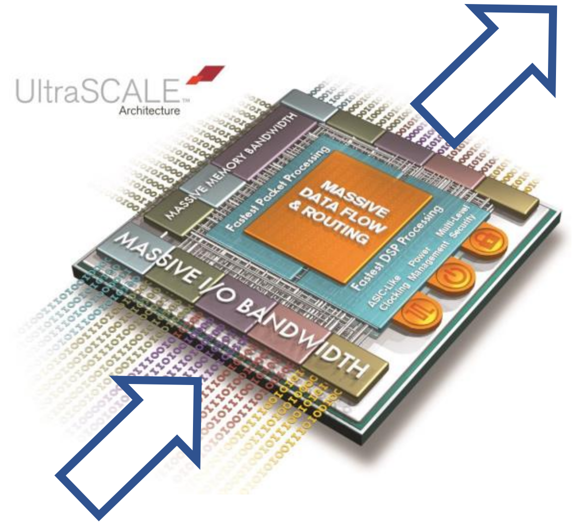
Instead of adapting the program to a given architecture

Let’s design the architecture that performs the task in the most efficient way

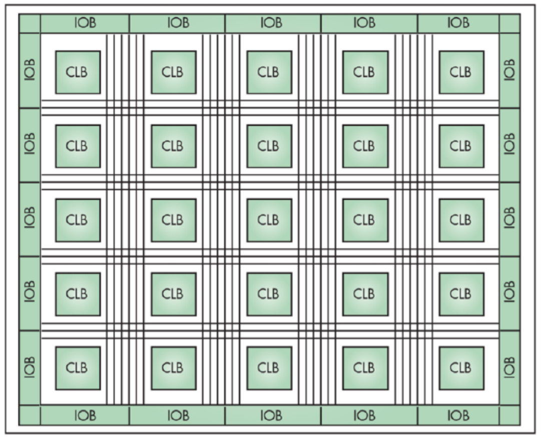
Especially when algorithms evolve much faster than hardware

What are FPGAs

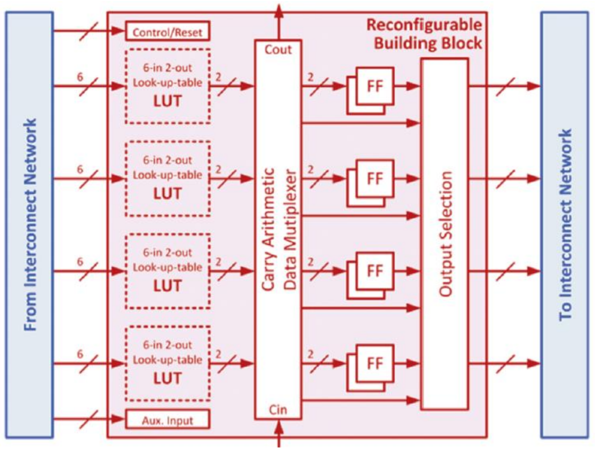
- Field Programmable Gate Arrays
  - Devices for processing digital data streams
  - Adaptable computing resources
  - Reconfigurable at any time

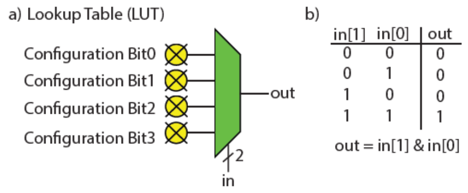


Arrays of Configurable Logic Blocks



Basic Configurable Logic Block


From Interconnect Network
Control/Reset
Reconfigurable Building Block
6-in 2-out Look-up-table LUT
Cout
Cin
Carry Arithmetic Data Mutiplexer
FF
Output Selection
Aux. Input
To Interconnect Network
6
2



in[1]	in[0]	out
0	0	0
0	1	0
1	0	0
1	1	1

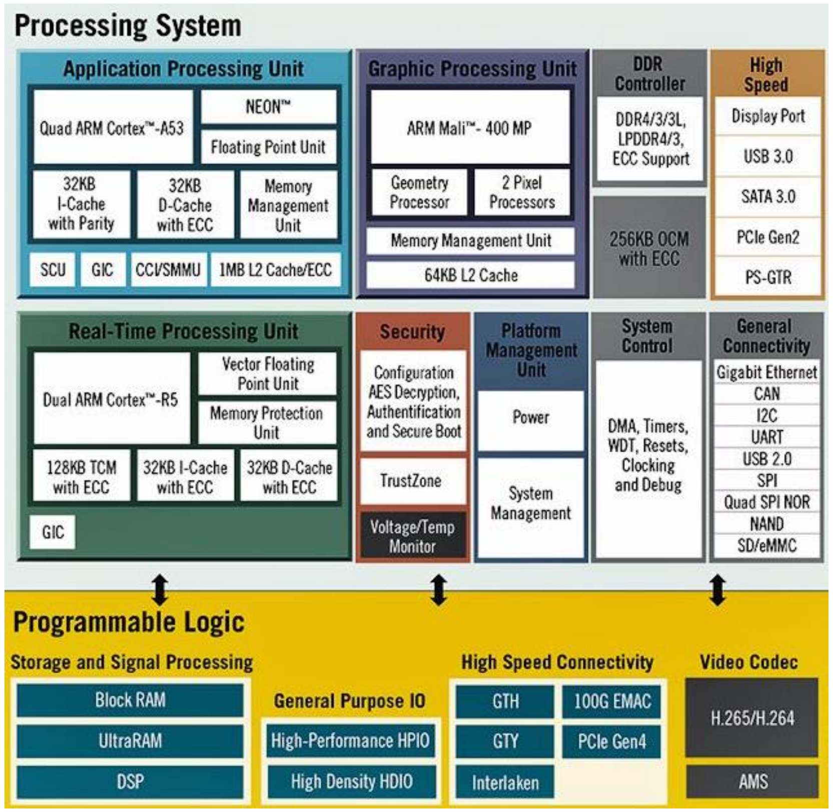
out = in[1] & in[0]

R. Kastner, J. Matai, S. Neuendorffer „Parallel Programming for FPGAs"

www.electronicdesign.com

What are FPGAs

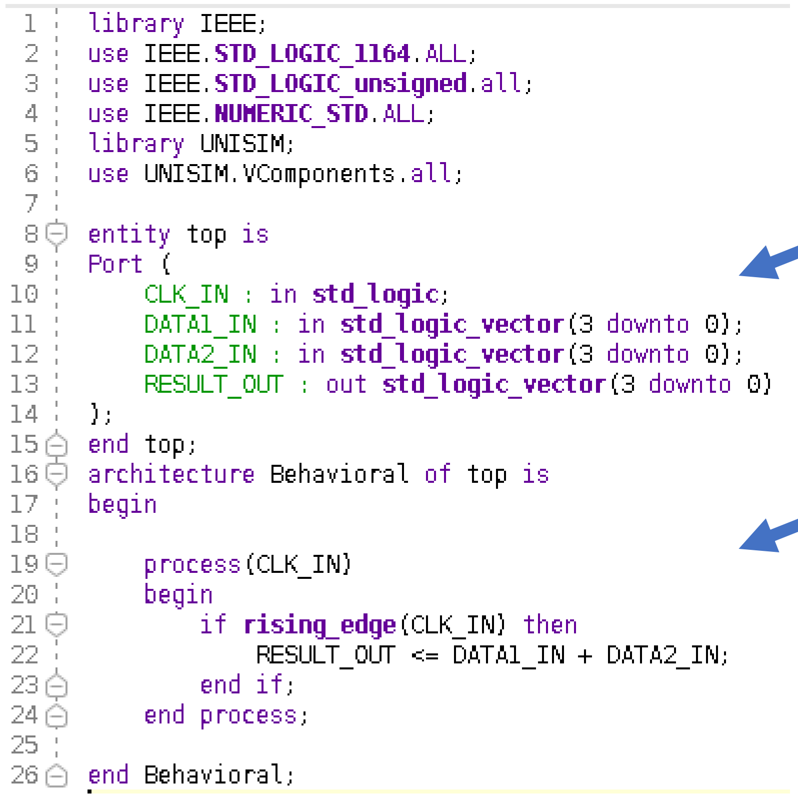
- Much more than just CLBs:
  - Memory blocks
  - DSP block (hard multipliers)
  - Multigigabit transceivers
  - Clock managers
  - Hard protocols and codecs
  - Ext. memory controllers
  - ADC/DAC
- Complete System-On-Chip:
  - PowerPC/ARM
  - Ext. Memory controllers
  - Multiple I/O controllers
  - Fast interconnect



Xilinx Zynq MPSoC - infrastructure

Xilinx.com

An example – source code



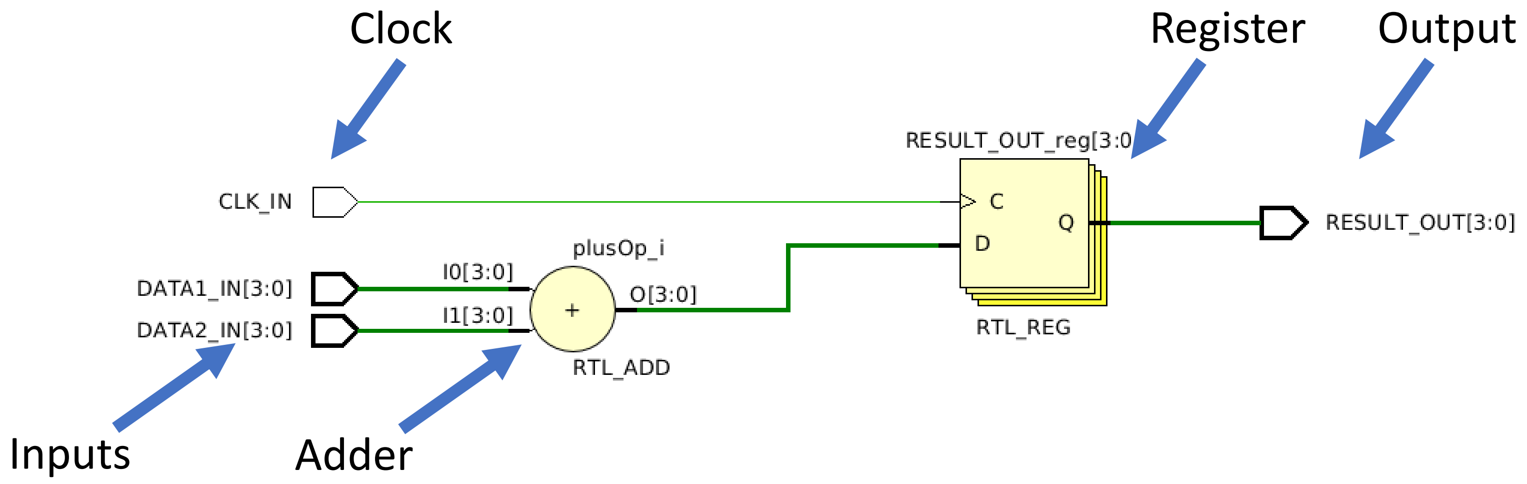
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.all;
use IEEE.NUMERIC_STD.ALL;
library UNISIM;
use UNISIM.VComponents.all;

entity top is
Port (
    CLK_IN : in std_logic;
    DATA1_IN : in std_logic_vector(3 downto 0);
    DATA2_IN : in std_logic_vector(3 downto 0);
    RESULT_OUT : out std_logic_vector(3 downto 0)
);
end top;
architecture Behavioral of top is
begin

    process(CLK_IN)
    begin
        if rising_edge(CLK_IN) then
            RESULT_OUT <= DATA1_IN + DATA2_IN;
        end if;
    end process;

end Behavioral;
```

Top-level interface – hardware pins, relates to the board design

Logic that describes the relation between input and output ports

An example – elaborated design

- Functional schematic of the design



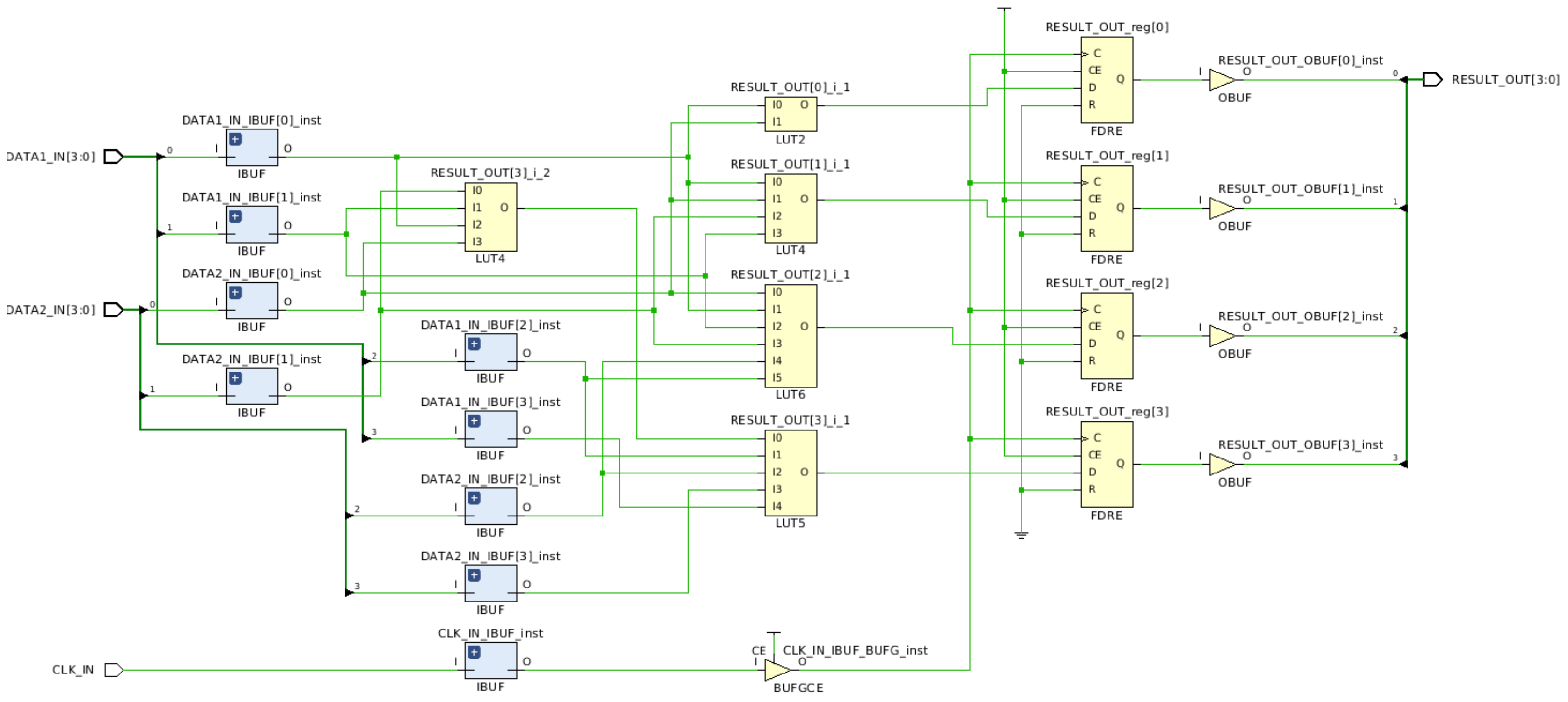
Function has:

- Initiation interval = 1 clock cycle
- Latency = 1 clock cycle

When CLK_IN = 100 MHz
This logic will achieve 100 MOps/s

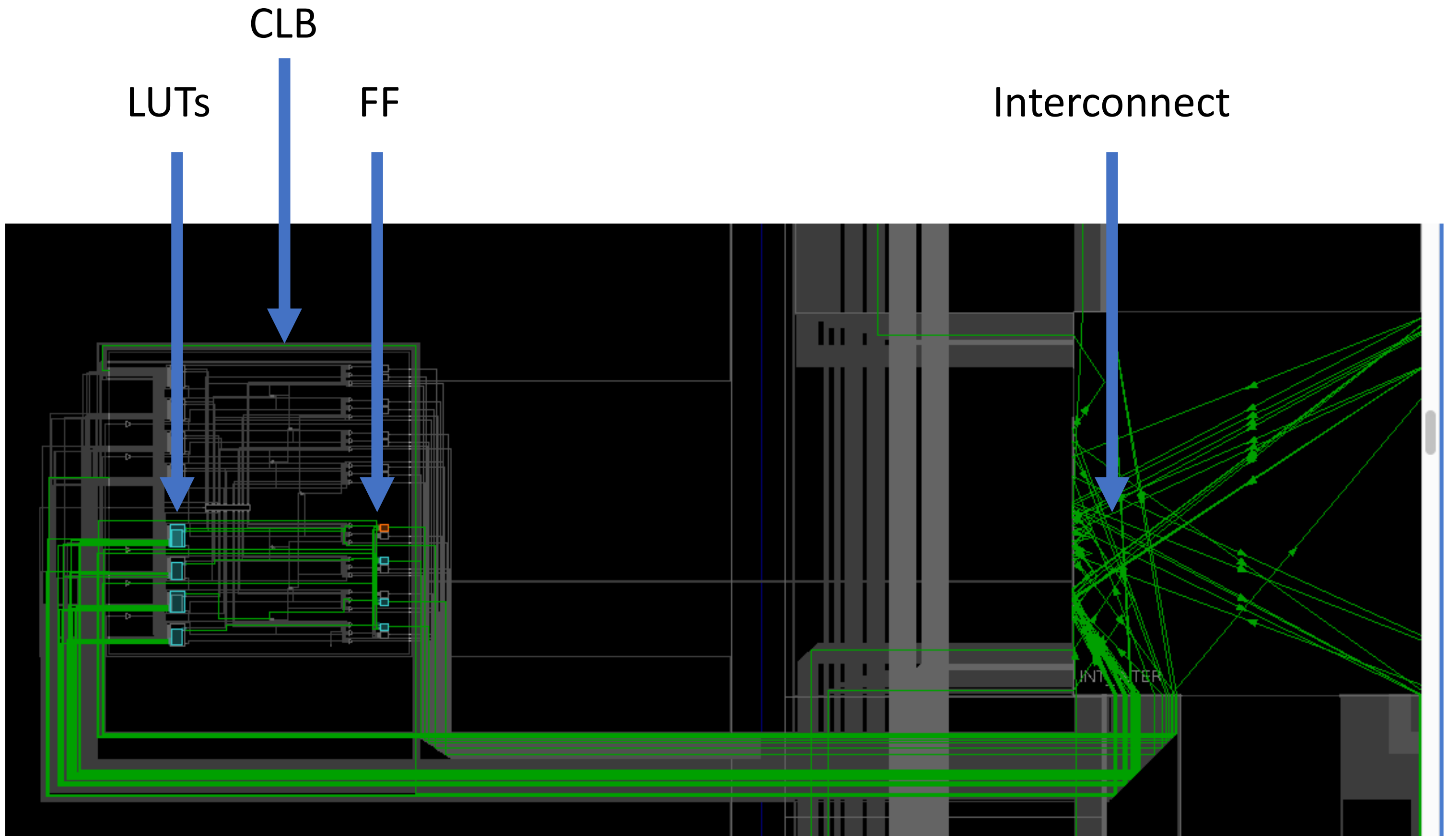
An example – synthesized design

- Functional schematic mapped into device resources

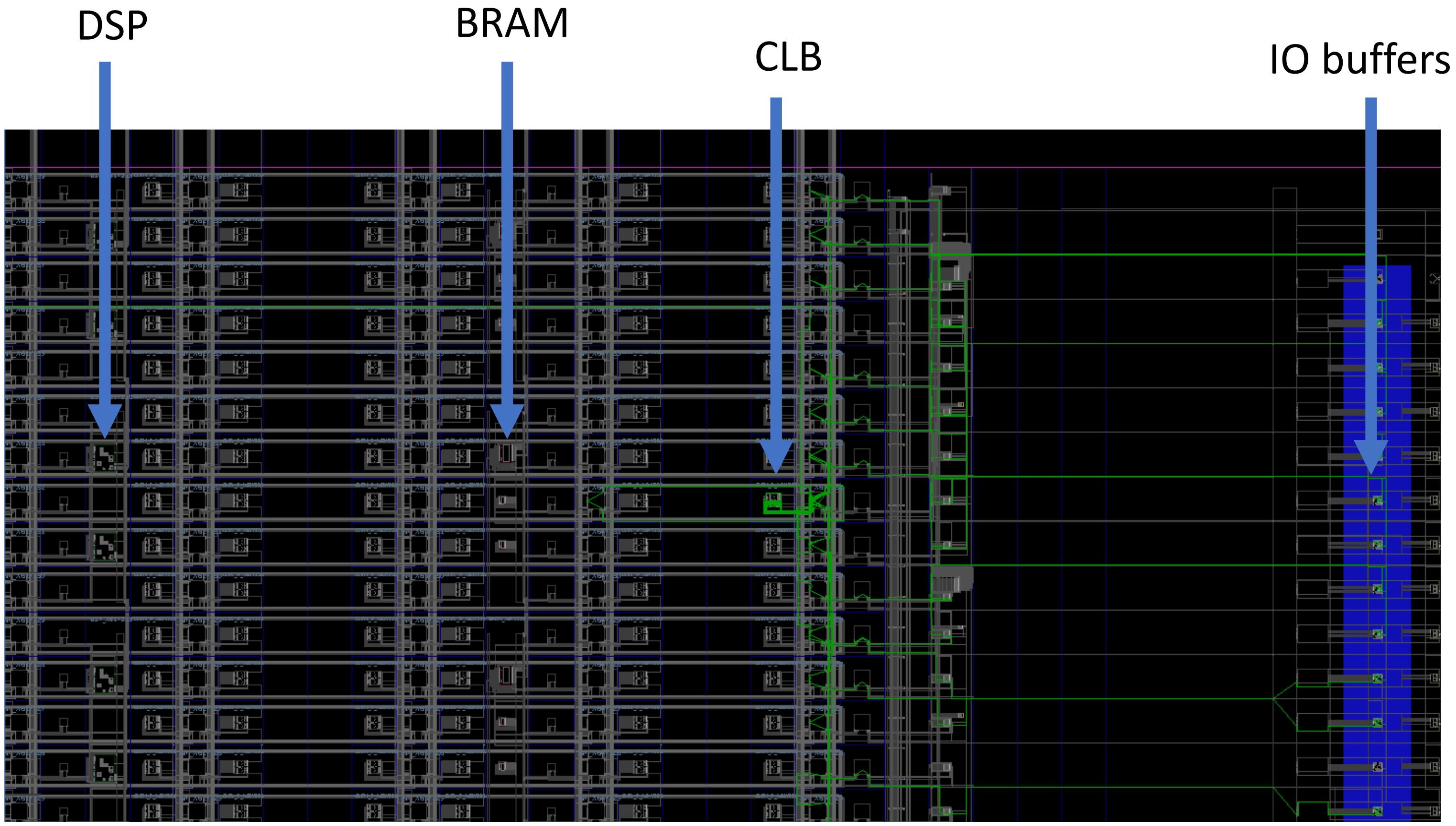


An example – implemented design

- Logic resources placed and connected in the device

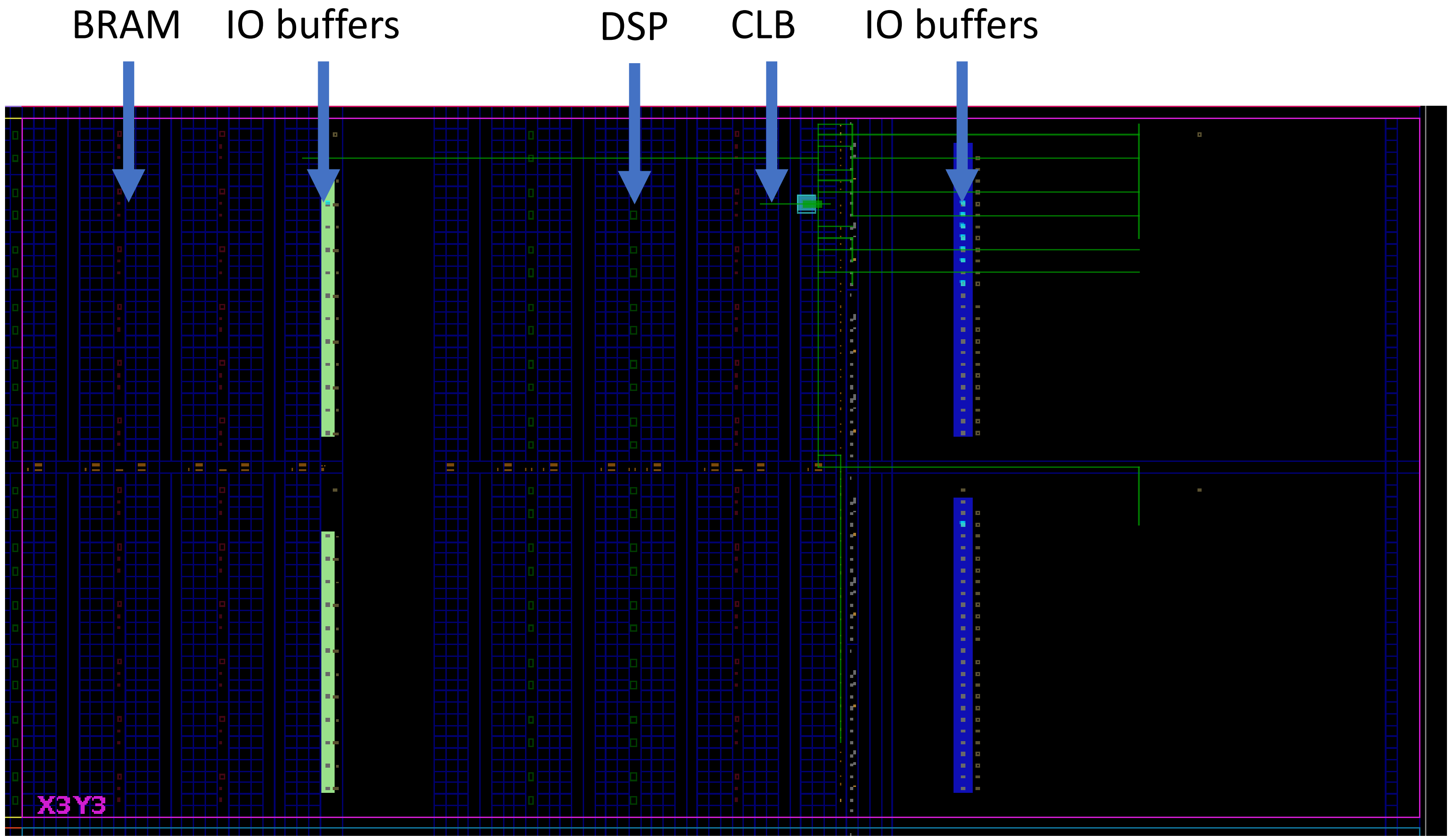


An example – implemented design

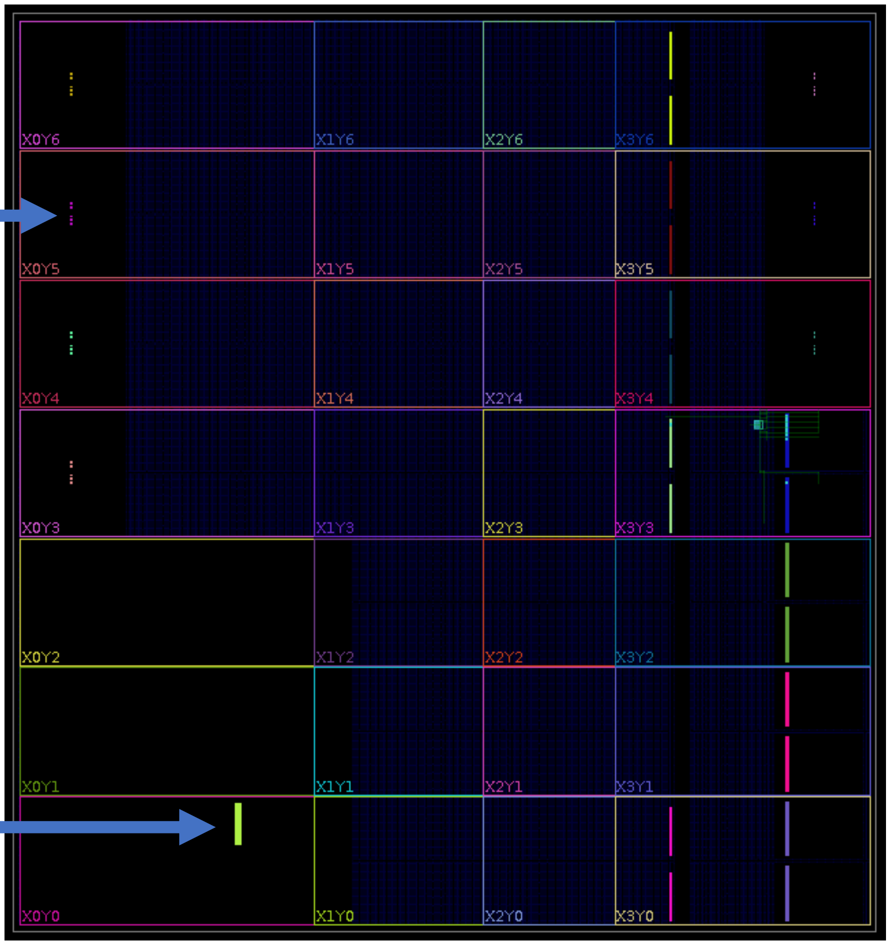


An example – implemented design

- Clock Region



An example – implemented design

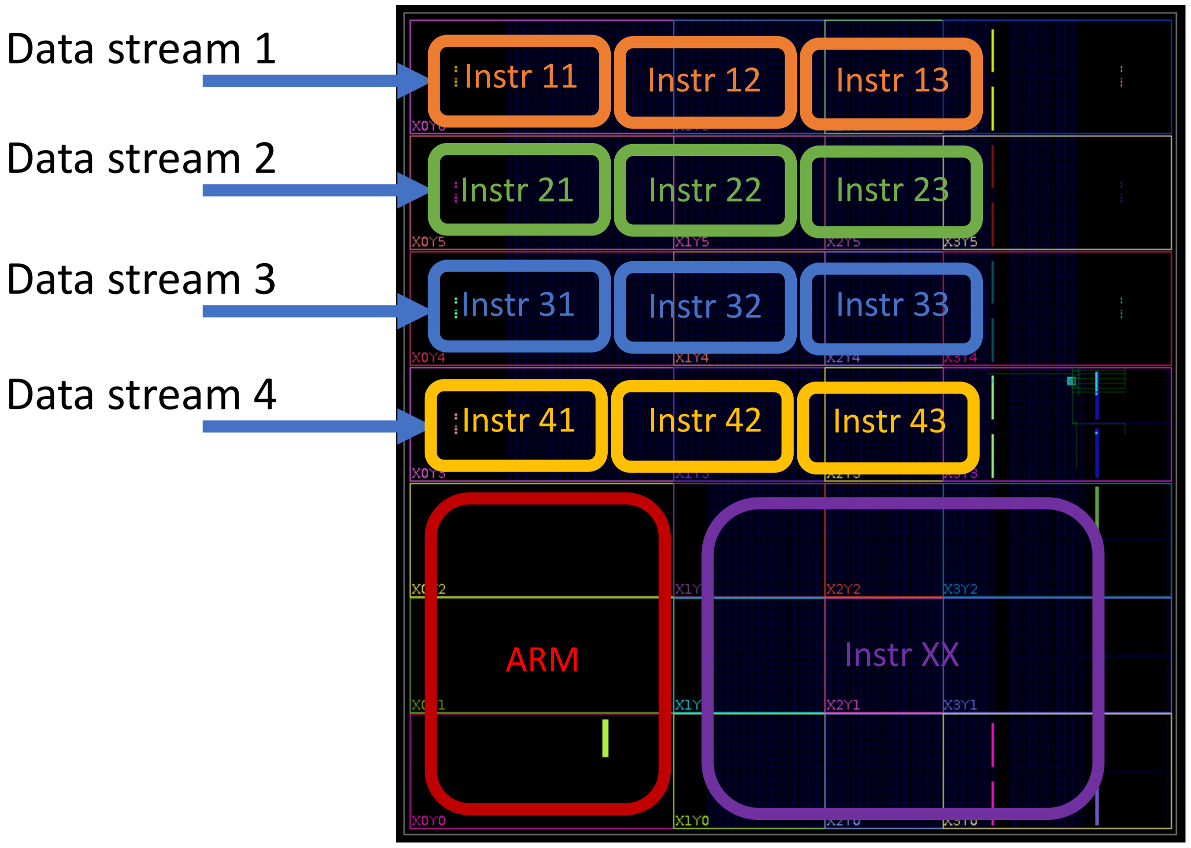


MGT

ARM

Utilization:
LUT: 4 out of 274 080
FF: 4 out of 548 169

Natural parallelism and streamlined processing



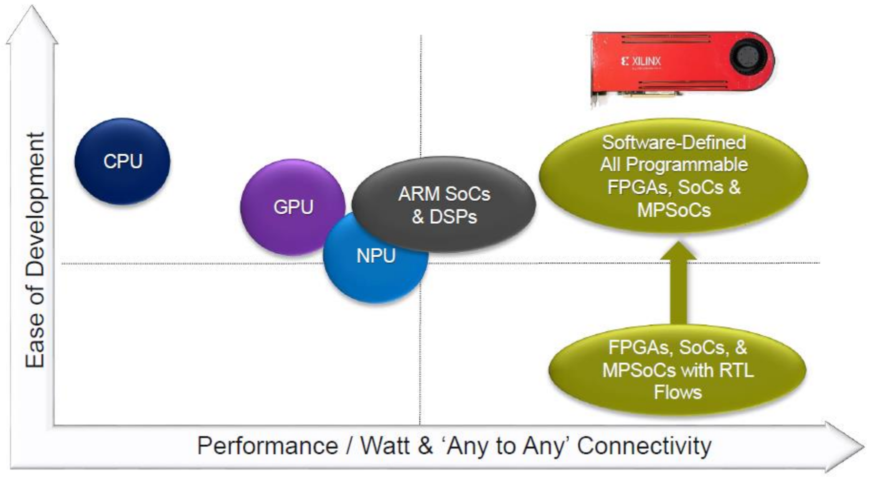
Advantages

- Architecture adapted for specific function
- Custom data types – resource optimization
- Instant memory access to BRAMs
- Real time - control over each clock cycle
- Natural parallelism
- Streamlined processing

- Perfect solution for:
  - Multi-channel sensor readout and processing
    - E.g. detector channels readout, multiple video streams processing, networking, …
  - Sensor Fusion
  - Monitoring and control in real time
    - E.g. mission critical control
  - High Performance Computing
    - E.g. matrix multiplicaiton, neural network inference, …

Disadvantages

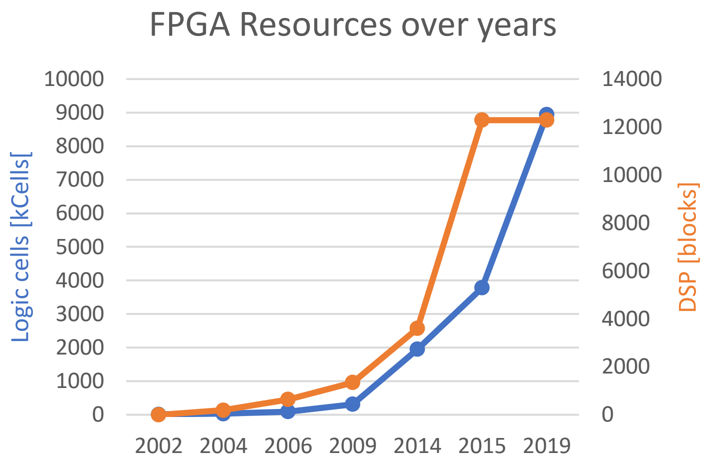
- Low level design
- Difficult firmware development
- Time consuming development and evaluation cycle
- High costs of single devices and of hardware development


Ease of Development
Performance / Watt & 'Any to Any' Connectivity
CPU
GPU
NPU
ARM SoCs & DSPs
Software-Defined All Programmable FPGAs, SoCs & MPSoCs
FPGAs, SoCs, & MPSoCs with RTL Flows
XILINX

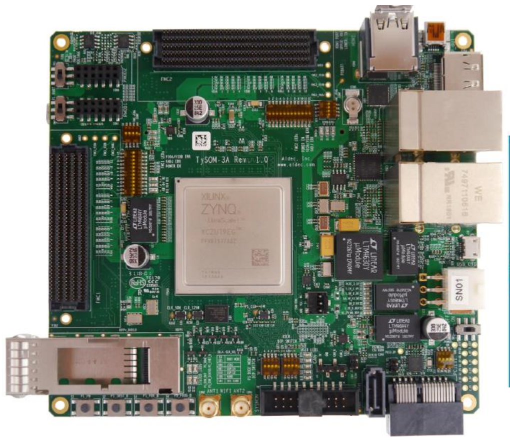
Xilinx.com

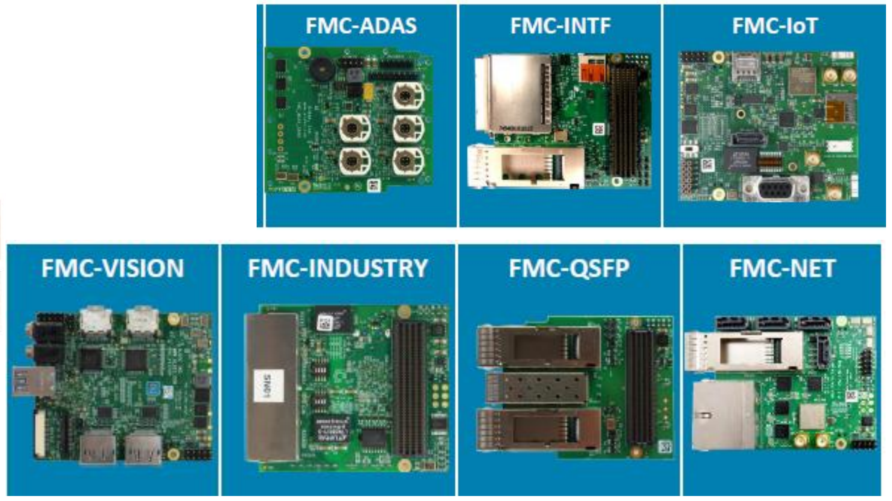
Why now

- Continuous increase of available resources
- Wide range of hardware platforms on the market
- Growing library of IP Cores
- Novel design development techniques

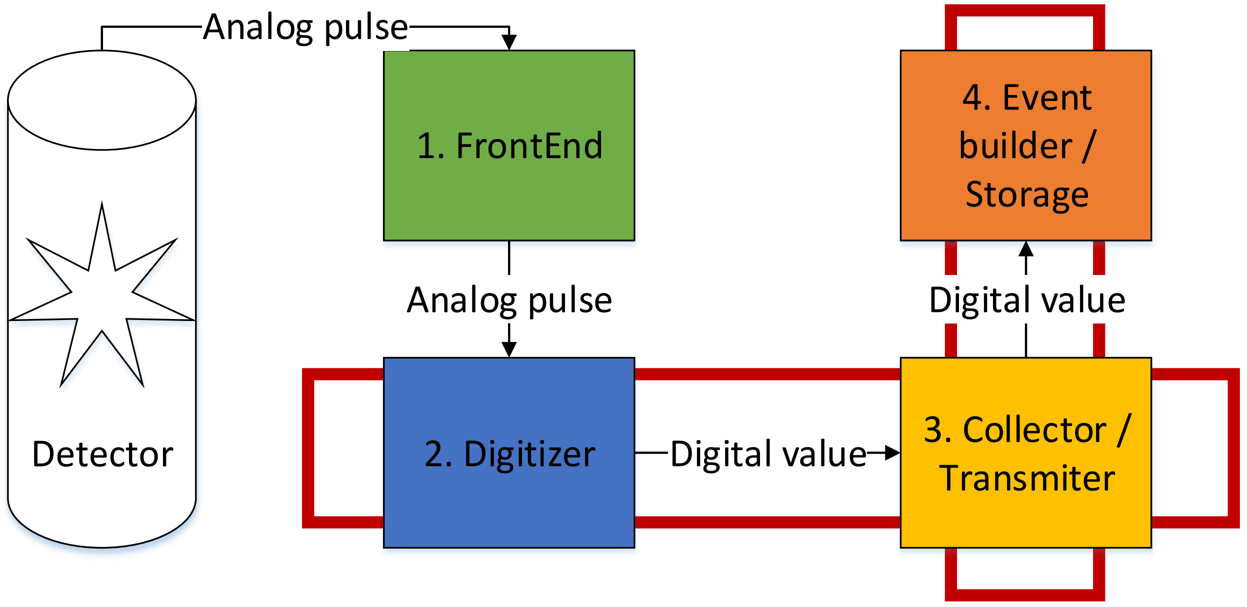


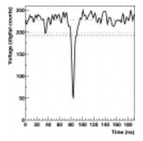
Example:
Aldec TySOM-3A-ZU19EG
10x14 cm
1,1 kCells
2k DSP
70 Mb memory
ARM COrtexA53
4 Gb DDR4
QSFP+, 2x RJ45, 4x USB, 2x HDMI
2x FMC

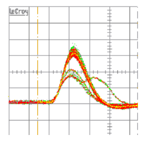




Processing pipeline



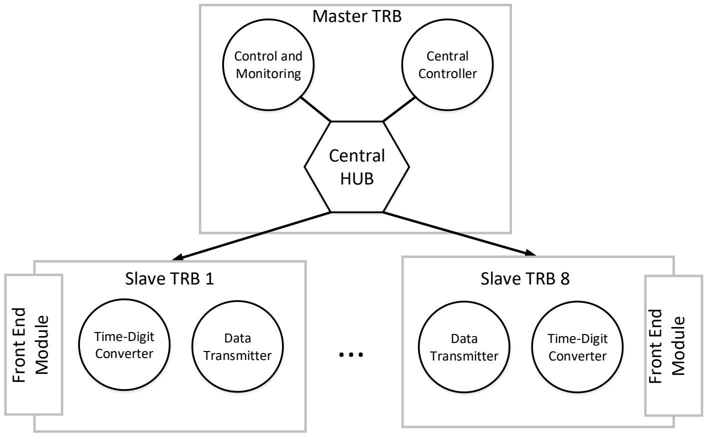




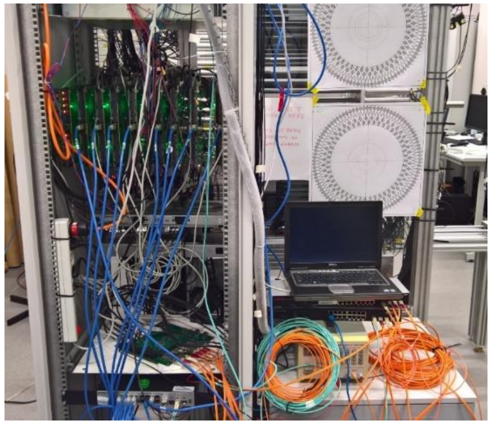
```
0000  ff ff ff ff ff ff 00 00
0010  01 1c 0b 00 00 00 ff 11
0020  ff ff c3 50 c3 50 01 08
0030  00 02 01 03 00 00 00 00
0040  00 02 06 01 00 02 06 02
```

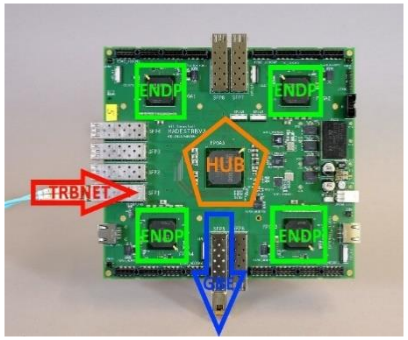
JPET Readout System

- Entirely based on FPGAs
  - Front End boards
    - Analogue signal discrimination
  - Digitizers / Data collectors
    - TRBv3 boards
    - TDC in FPGA
  - Data processing and visualization
    - Controller board – continuous readout
    - Event by event processing
  - Custom hardware, custom protocols



G. Korcyl, M. Kajetanowicz, P. Moskal, M. Pałka „A system for acquisition of tomographic data"
Patent nr.: US20160209522A1, WO/2015/028594





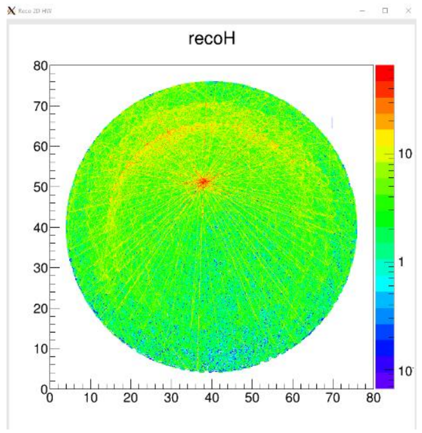
Traxler, M.; Korcyl, G.; Bayer, E.; Maier, L.; Michel, J.; Palka, M.
„A compact system for high precision time measurements (<14 ps RMS) and integrated acquisition for a large number of channels",
JINST 10.1088/1748-0221/6/12/C12004

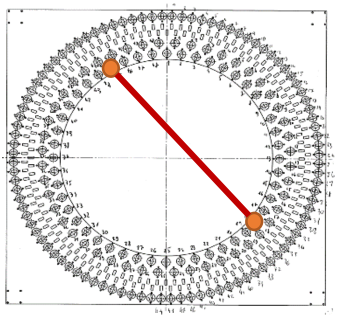
JPET processing pipeline

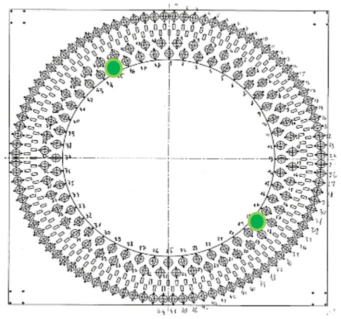
- Data processing steps
  - Data units reception and assembly
  - Extraction of timing data
  - Application of detector geometry
  - Application of calibration parameters
  - Search for time coincidences
  - Filtration
  - Construction of histogram and visualization

```
0000  ff ff ff ff ff ff 00 00
0010  01 1c 0b 00 00 00 ff 11
0020  ff ff c3 50 c3 50 01 08
0030  00 02 01 03 00 00 00 00
0040  00 02 06 01 00 02 06 02
```

Hit1: ch 1, 115 ns, TOT 5 ns
Hit2: ch 2, 116 ns, TOT 7 ns
…

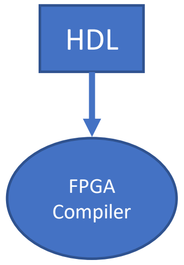




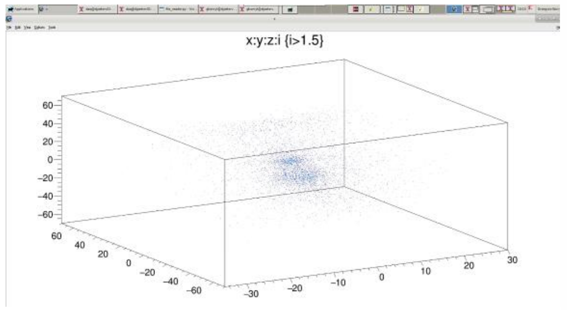


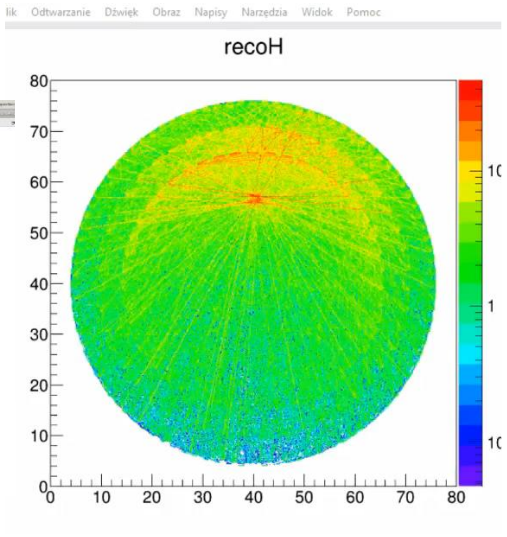
JPET visualization

- Resource utilization: 75% (Zynq045)
- Performance: 42 Mhits/s
- Power consumption: 20 W
- About 1 year of development
- Almost entire design in low level HDL



G. Korcyl, et al. JPET "Evaluation of single-chip, real-time tomographic data processing on FPGA SoC devices"
IEEE Transactions on Medical Imaging, vol. 37/11, May 2018





Shift in technology availability

- Wide variety of hardware components:
  - Evaluation boards, HW companies (Trenz, HiTech, Aldec, …)
  - FPGA Mezzannine Cards (FMC, Vita standards), …
- Well established standards, protocols and IP Cores:
  - AXI-based ecosystem, Aurora connectivity, …

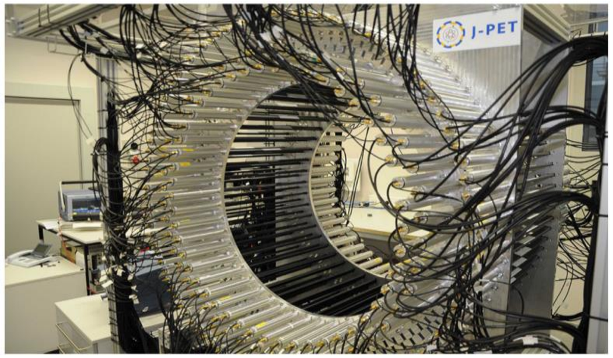
- Xilinx VCU108
  - Virtex Ultrascale U095
    - 1.1 M logic cells
    - 768 DSP
    - 60 Mb BRAM
    - 64 MGT (32 GTH / 32 GTY)
  - 8 GB DDR4
  - 2x FMC HPC 10xGTH

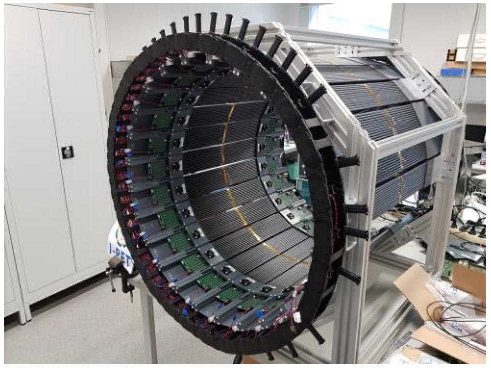
- Xilinx ZCU102
  - Zynq MPSOC ZU9EG
    - 0.6 M logic cells
    - 2520 DSP
    - 32 Mb BRAM
    - 24 MGT
  - ARM Cortex A53
  - 4 GB DDR4
  - 2x FMC HPC 16xGTH

- HTG FMC 10x SFP+

Modular JPET readout

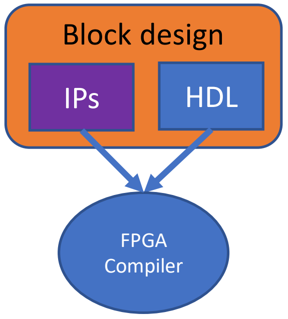
	JPET	Modular JPET	
Scintillators	192	312	**1.6x**
Analog channels	1536	4992	**2.9x**
Digitizers	32	48	**6x**
Logic [k cells]	350	5400	**15.4x**
Memory [Mb]	19	272	**14.3x**
DSP	900	3972	**4.4x**
ARM cores	2	4 + 2x RT + 1x GPU	**>2x**

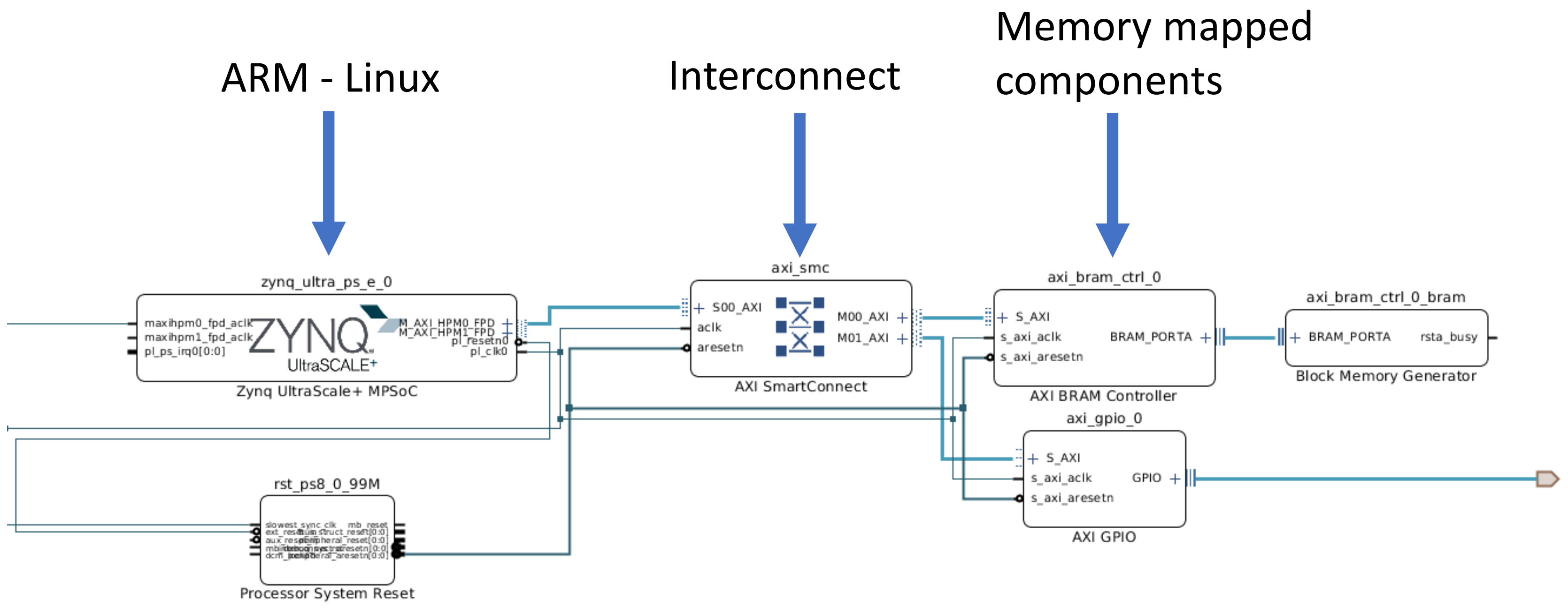




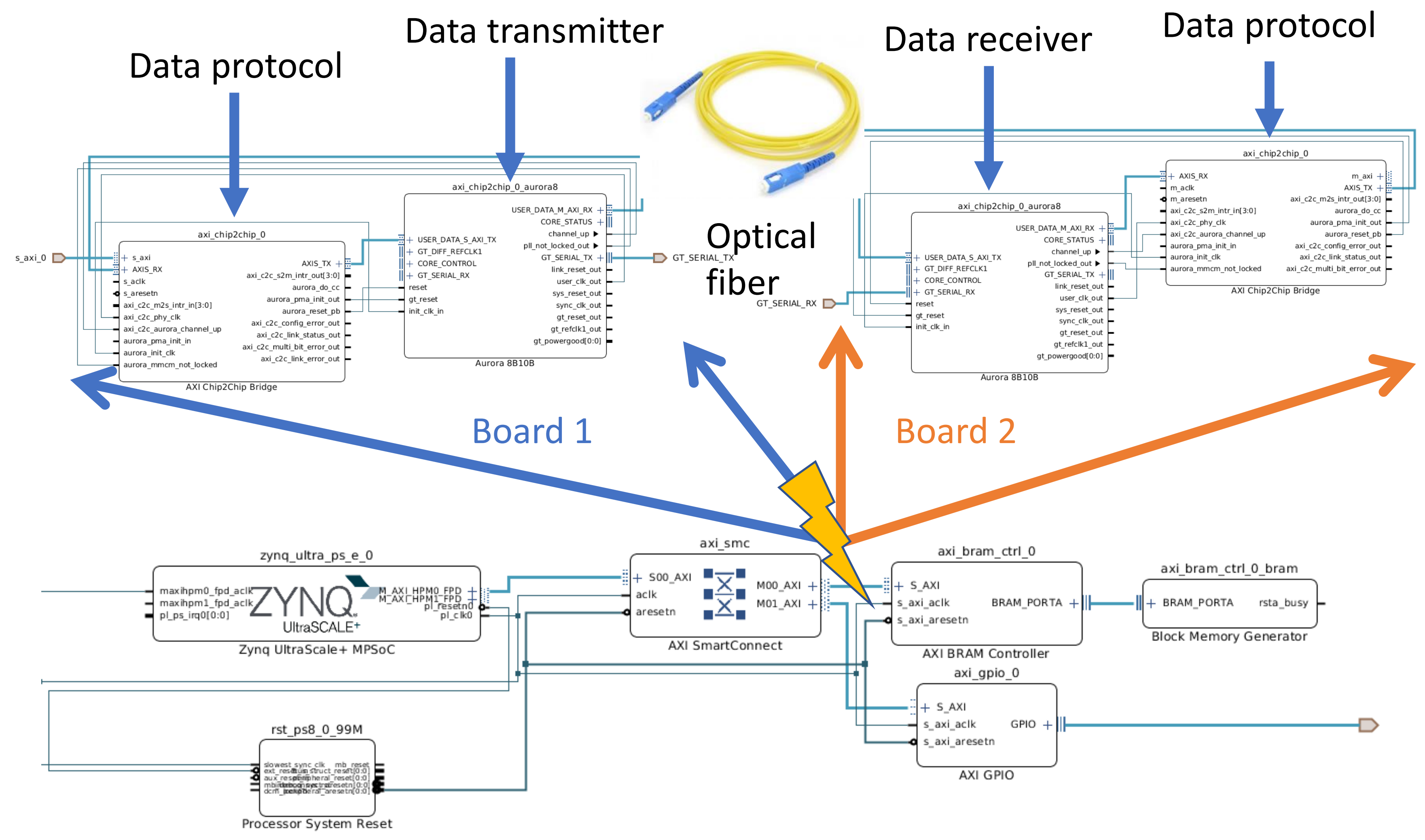
Communication infrastructure

- Standard AXI interfaces and building blocks
- All components mapped into Linux physical memory
- All components accessible from software level on Linux
- Any HDL component can be encapsulated with AXI layer



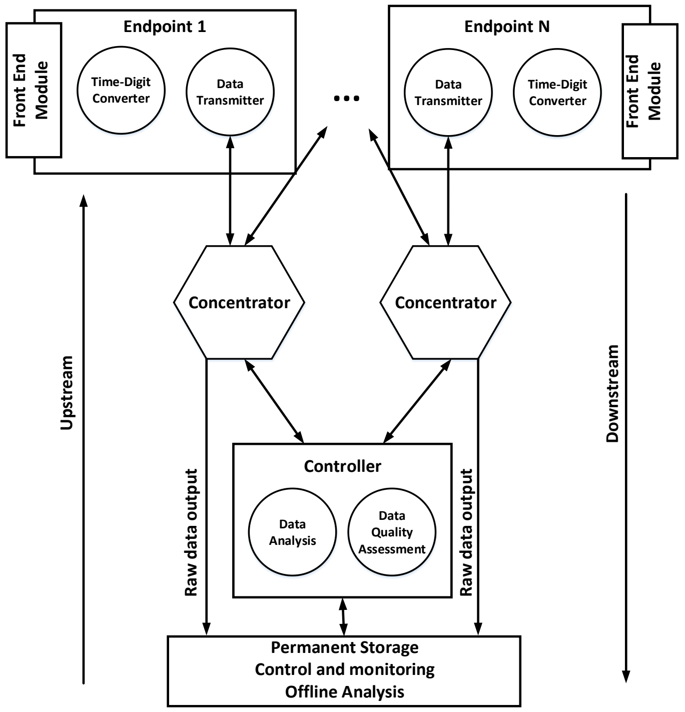


Communication infrastructure



System design

- Basic system infrastructure design and setup in a single day
  - 48 endpoint boards (Artix 7)
  - 4 concentrator boards (VCU108)
  - 1 controller board (ZCU102)
- Optical connections Aurora 8b/10b, 5Gbps, full-duplex
- Synchronous links – precise time synchronization of all elements
- Each logical component in the system is addressable and accessible from the software on controller board
- Readout data stream mixed with control and monitoring messages
- 10 Gigabit Ethernet UDP module



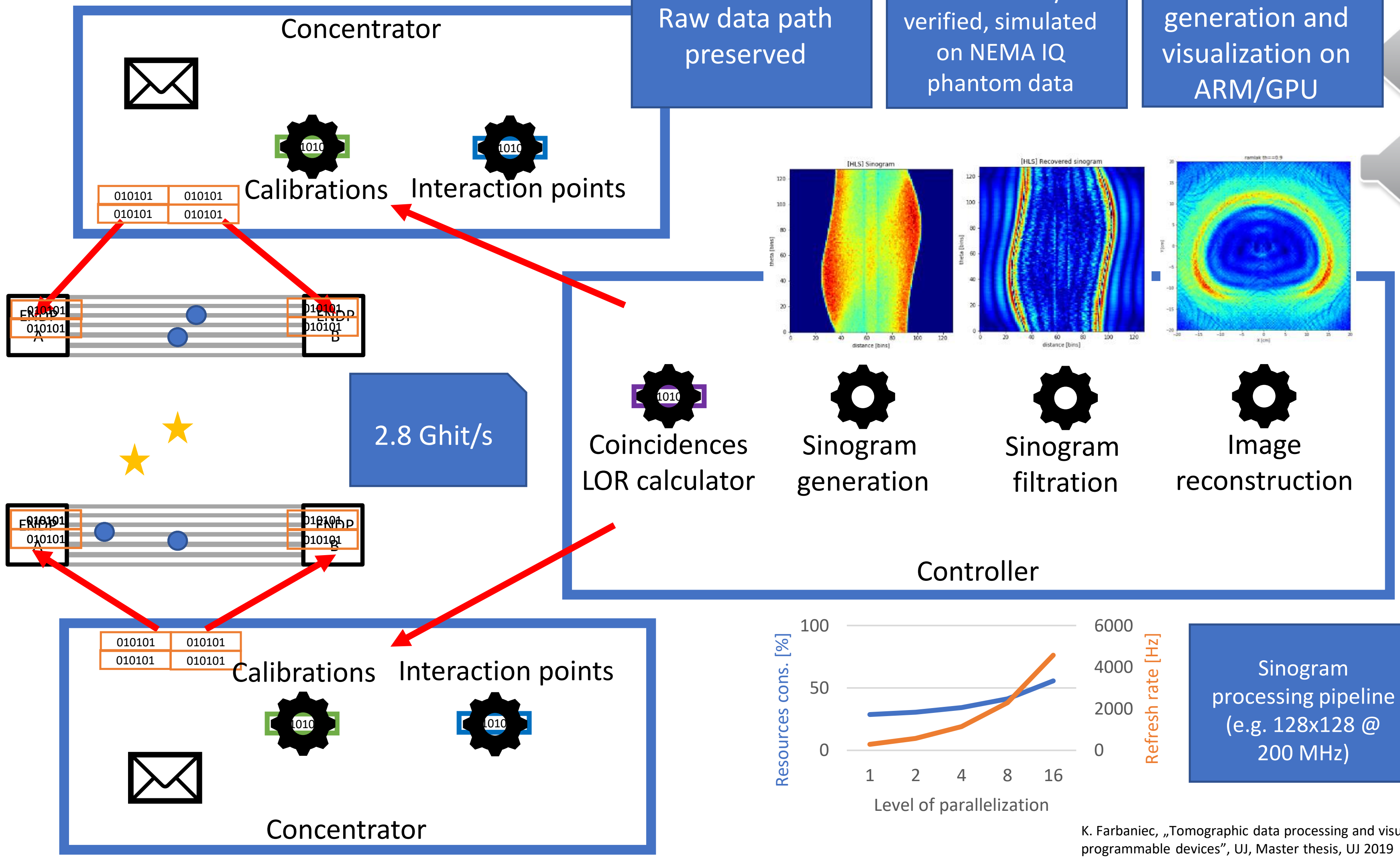
Full processing pipeline

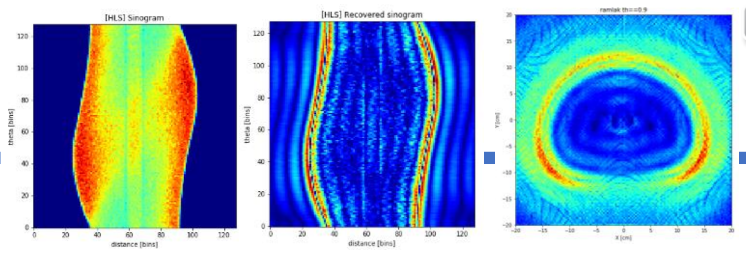
Raw data path preserved

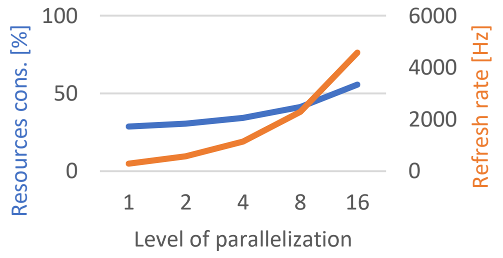
Numerically verified, simulated on NEMA IQ phantom data

Final image generation and visualization on ARM/GPU

2.8 Gbit/s







Sinogram processing pipeline (e.g. 128x128 @ 200 MHz)

K. Farbaniec, „Tomographic data processing and visualization on programmable devices", UJ, Master thesis, UJ 2019

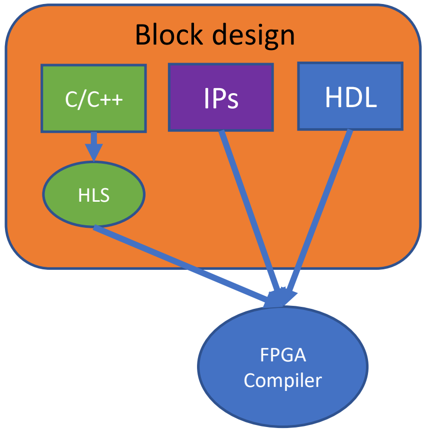
Novel development techniques

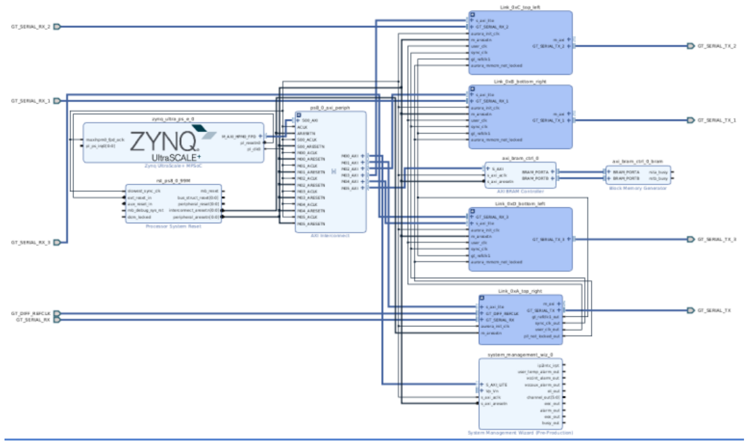
- Reduce HDL logic development to minimum
  - Time consuming, requiring experience, error-prone process

- Block designs
  - Drag-and-Drop, connection automation, library of ready to use, configurable components (IP Cores)
- **High Level Synthesis**
  - **Component development in C/C++/OpenCL**
  - **Compilation into AXI encapsulated HDL IP Core**

- Algorithmic/data processing components in HLS
- Hardware interfacing in HDL

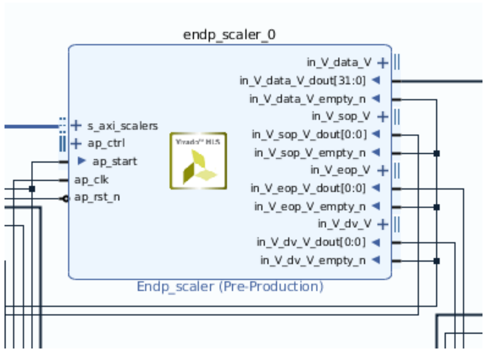
- Full flow, integrated environmets

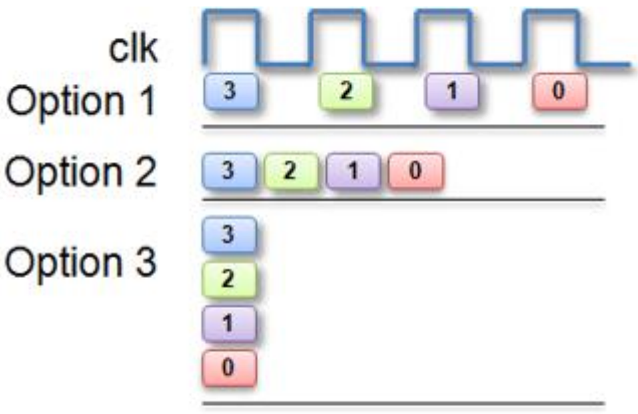


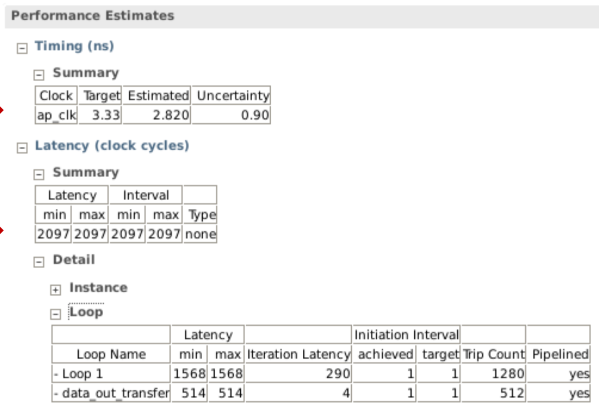


Development in HLS

- Single function – single component
  - Function arguments become component interface
  - Function body translated into logic
  - Results analysis with a set of reports
    - Timings, resources
  - Compilation process controlled with a set of #pragmas
  - **High level component evaluation**







Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	3.33	2.820	0.90

Latency (clock cycles)

Summary

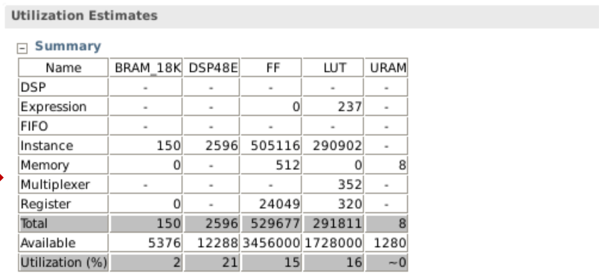
Latency		Interval		
min	max	min	max	Type
2097	2097	2097	2097	none

Detail

Instance

Loop

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Loop 1	1568	1568	290	1	1	1280	yes
- data_out_transfer	514	514	4	1	1	512	yes



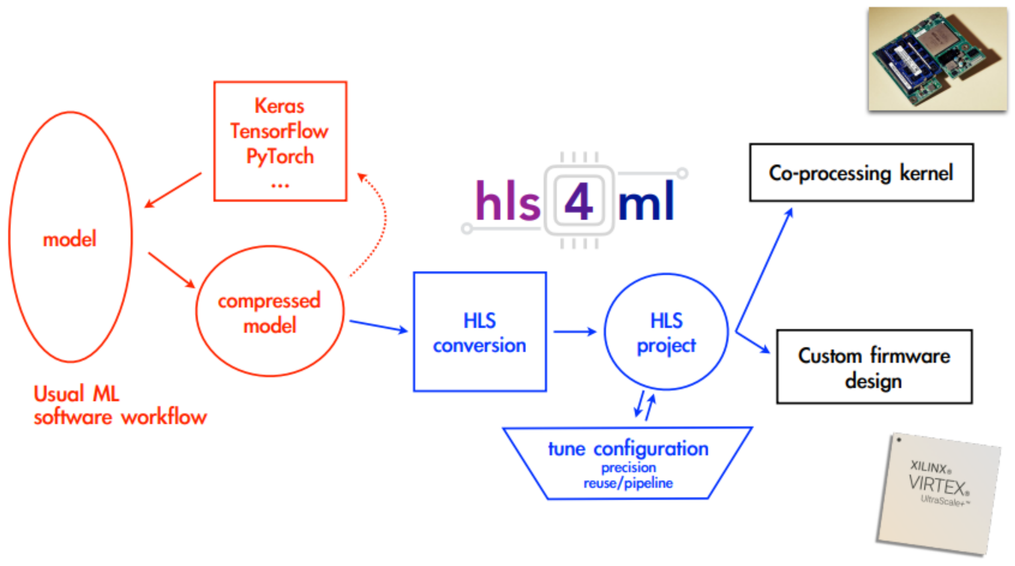
Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	237	-
FIFO	-	-	-	-	-
Instance	150	2596	505116	290902	-
Memory	0	-	512	0	8
Multiplexer	-	-	-	352	-
Register	0	-	24049	320	-
Total	150	2596	529677	291811	8
Available	5376	12288	3456000	1728000	1280
Utilization (%)	2	21	15	16	~0

HLS Example

- Entire flow for Neural Network implementation on FPGA within a single HDL module
- Advantages: fixed latency, controlled level of parallelism, custom data types
- Used in L1 trigger at CMS CERN „Artificial Intelligence Accelerates Dark Matter Search", Xilinx Case Study 2019
- Import trained NN model to FPGA with just few mouse clicks
- MNIST network example implemented by students during FPGA Summer Camp 2019



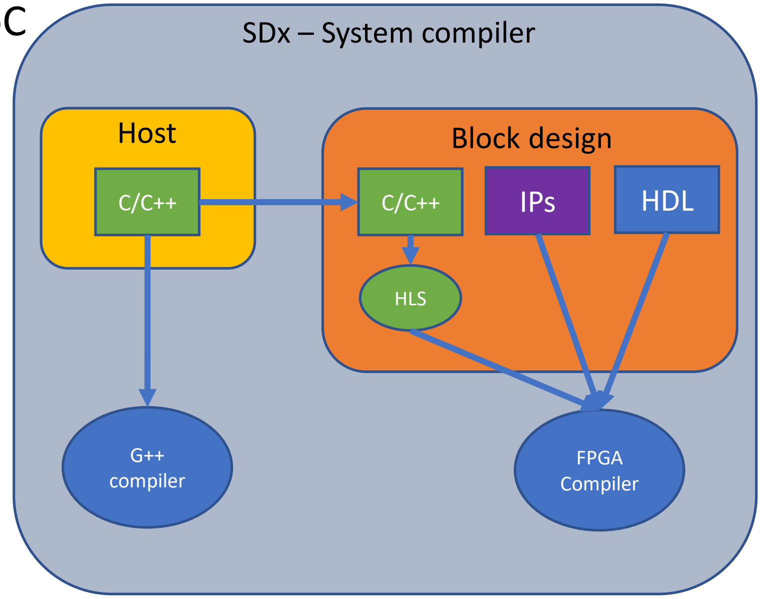
Python C++ Bitstream

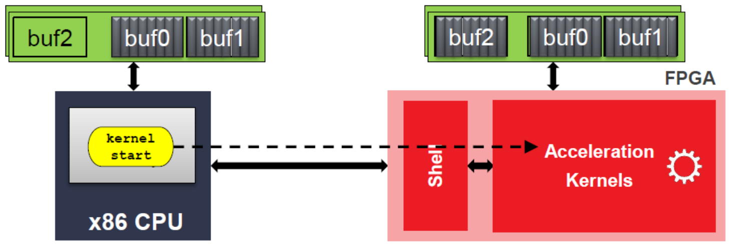
hls4ml

Software Defined Environment

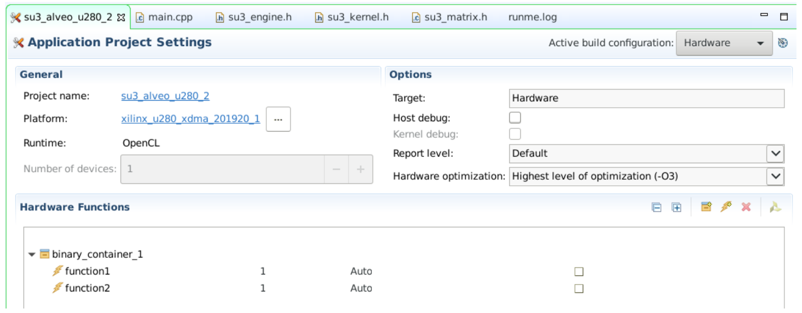
- Large selection of hardware platforms on market
  - Standalone boards – System-on-Chip devices – SDSoC
  - Accelerator boards – PCIe enabled - SDAccel

- Development environment
  - Entire project in C/C++
  - Host <–> Kernel architecture
  - Main function – starting point on the host
  - Kernel – hardware accelerated function
  - Encapsulates all other tools and compilers
  - Multi-level evaluation: g++, hw emulation, hw





Xilinx.com



Application Project Settings Active build configuration: Hardware

General

Project name: su3_alveo_u280_2
Platform: xilinx_u280_xdma_201920_1
Runtime: OpenCL
Number of devices: 1

Options

Target: Hardware
Host debug: ☐
Kernel debug: ☐
Report level: Default
Hardware optimization: Highest level of optimization (-O3)

Hardware Functions

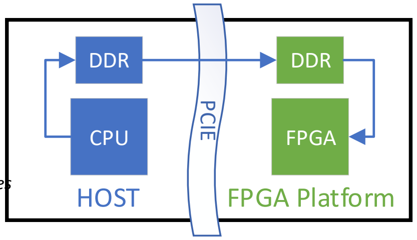
binary_container_1			
function1	1	Auto	☐
function2	1	Auto	☐

SDx Example

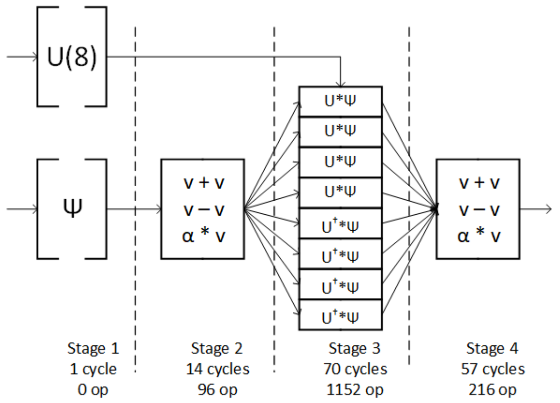
- Conjugate Gradient as HPC benchmark
  - Host prepares data sets and streams to the accelerated kernel
  - 1464 floating point numerical operations per single set
  - Kernel implemented with II=1, latency 142 at 300 MHz
  - 2x faster than Intel Xeon Phi 64-core, 1.7 GHz
  - **Reaching Nvidia V100 (only 15% of PP!)**

  - Not a single HDL line written

S. Durr, „*Three Dirac operators on two architectures with one piece of code and no hassle*", LATTICE2018, arXiv:1808.05506v2, Nov. 2018

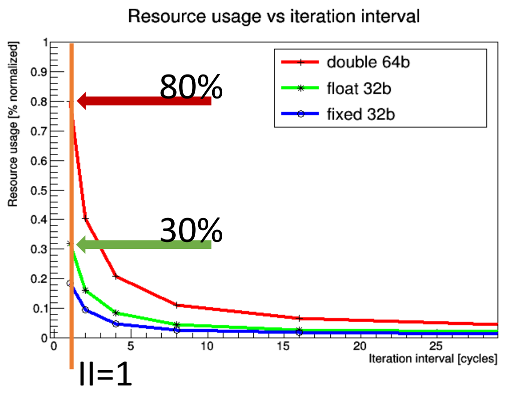


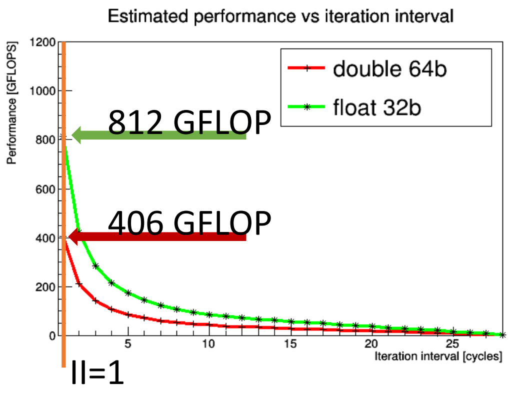
E. Berkowitz, et. Al.. „Simulating the weak death of the neutron in a femtoscale universe with near-exascale computing"



1464 operations
Within 142 clock cycles
With II=1

- Close cooperation with Xilinx





SDx Example

- Conjugate Gradient implementation
  - Alveo U280
    - 8GB of integrated HBM memory
      - 32x channels: 512b @ 300 MHz each (460GBps)
    - QSFP connector for large scale systems (t.b.d.)
    - 32 GB externel DDR4
    - 1 M LUTs, 9k DSP, 960 URAM
    - 250W
  - Alveo U50 (to be released)
    - No DDR4, 75W

  - **Host and kernel code in C++**
    - **Abstract code structures (structures, classes)**
    - **OO mechanisms: operators overload, inheritance, templates**

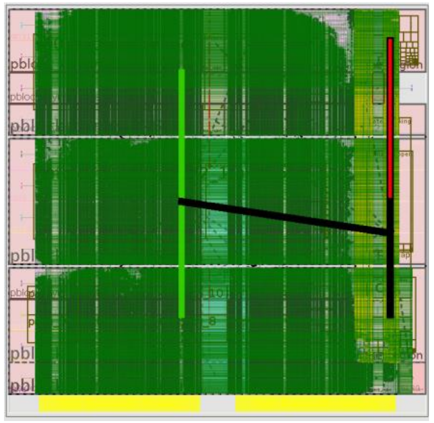
  - OpenCL bindings for C++
    - System definition
    - Data transfer mechanisms

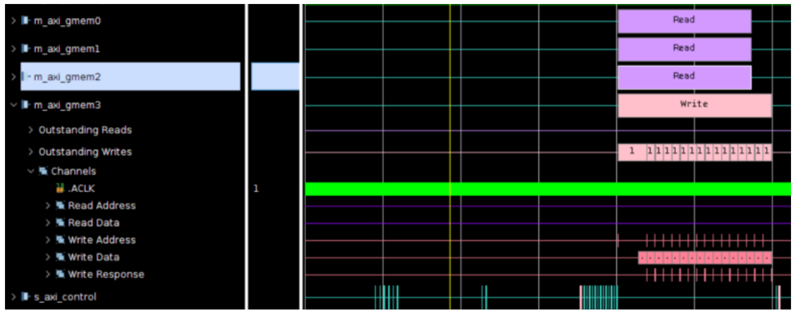
- 3 kernel instances, 70% device usage
- Executed on hardware (Nimbix cloud)

Xilinx.com

1 kernel instance =
2x JPET processing pipeline

3 instances + data movement infrasctructure

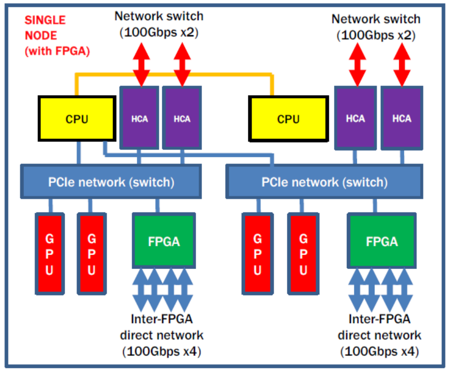




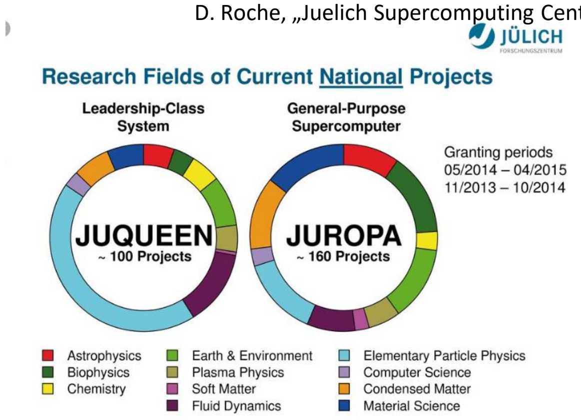
G. Korcyl, P. Korcyl, *„Towards Lattice Quantum Chromodynamics on FPGA devices”*,
Computer Physics Communications 2019.107029, Nov. 2019

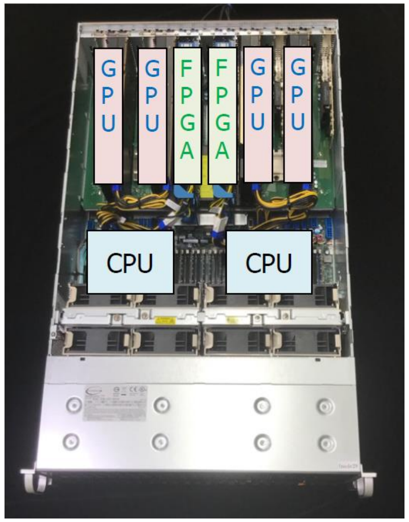
HPC Example

- Cygnus – Center for Computational Sciences, Tsukuba, Japan


SINGLE NODE (with FPGA)
Network switch (100Gbps x2)
Network switch (100Gbps x2)
CPU
HCA
HCA
CPU
HCA
HCA
PCIe network (switch)
PCIe network (switch)
GPU
GPU
FPGA
GPU
GPU
FPGA
Inter-FPGA direct network (100Gbps x4)
Inter-FPGA direct network (100Gbps x4)

D. Roche, „Juelich Supercomputing Centre"





T. Boku, „Japanese Supercomputer development and hybrid accelerated supercomputing"

Xilinx.com

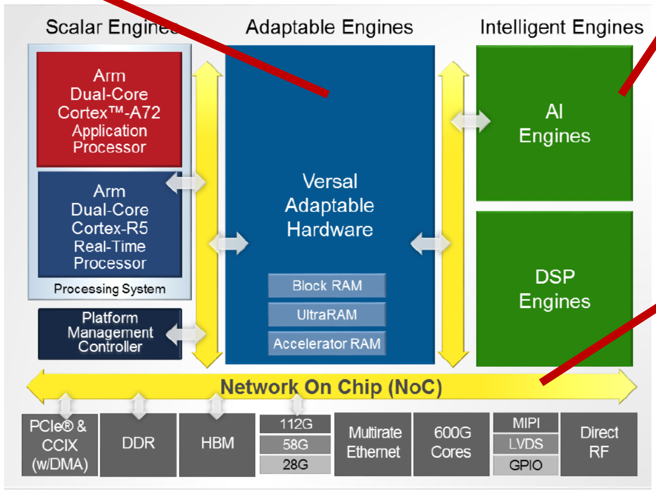
What is next to come

- 7nm Versal Architecture

General CLB resources:
- 4x larger blocks (32 LUTs)
- Less routing
- Higher frequencies

Interconnected vector SIMD:
- 1GHz frequency
- 512b floating point vector
- Local/shared memory
- Streamlined designs

High bandwidth interconnect

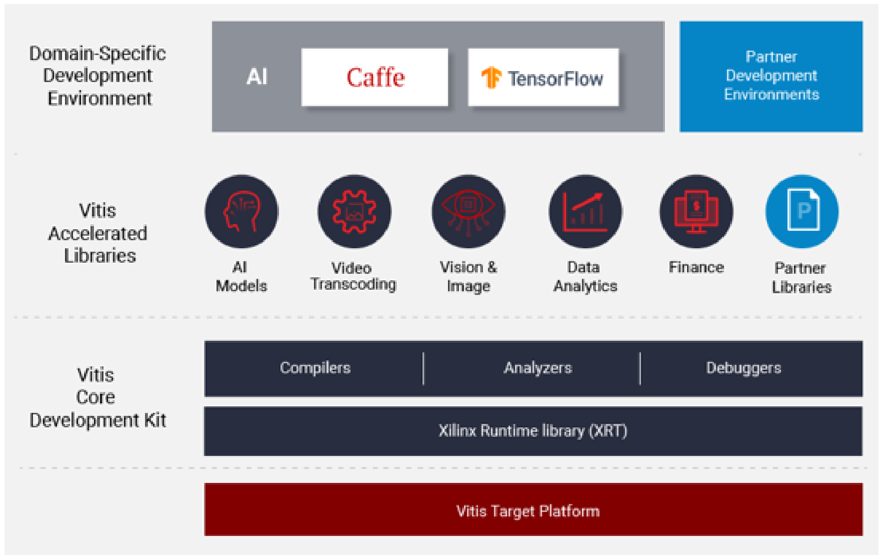


Xilinx.com

What is next to come

- Vitis Unified Software Platform
  - Integration of compilers
  - Integration of libraries
  - Python/C/C++ entry language

- HPCG project imported, compiled and evaluated on Nimbix one day after release
- Working with Xilinx on including our HPCG implementation in the set of libraries


Domain-Specific Development Environment
AI
Caffe
TensorFlow
Partner Development Environments
Vitis Accelerated Libraries
AI Models
Video Transcoding
Vision & Image
Data Analytics
Finance
Partner Libraries
Vitis Core Development Kit
Compilers
Analyzers
Debuggers
Xilinx Runtime library (XRT)
Vitis Target Platform

Xilinx.com

Summary

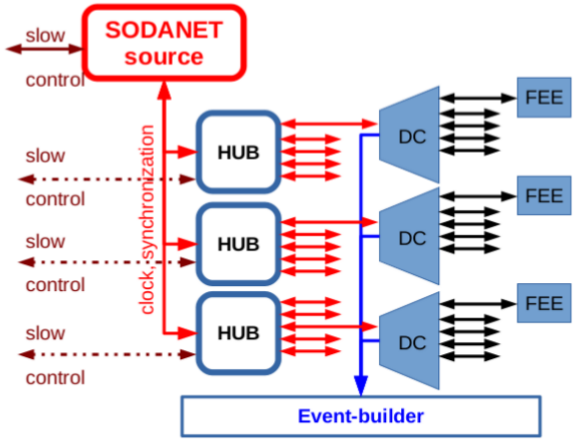
- Disadvantages:
  - Low level design
    - Integrated ARM cores / softcore processors
  - Difficult firmware development
    - High Level Synthesis
  - Time consuming development and evaluation cycle
    - Accelerated evaluation with C-based tests
  - High costs of single devices and of hardware development
    - Wide-range of products available on market

- FPGAFAIS group
- FPGA Summer Camp
- Sympozjum FPGA

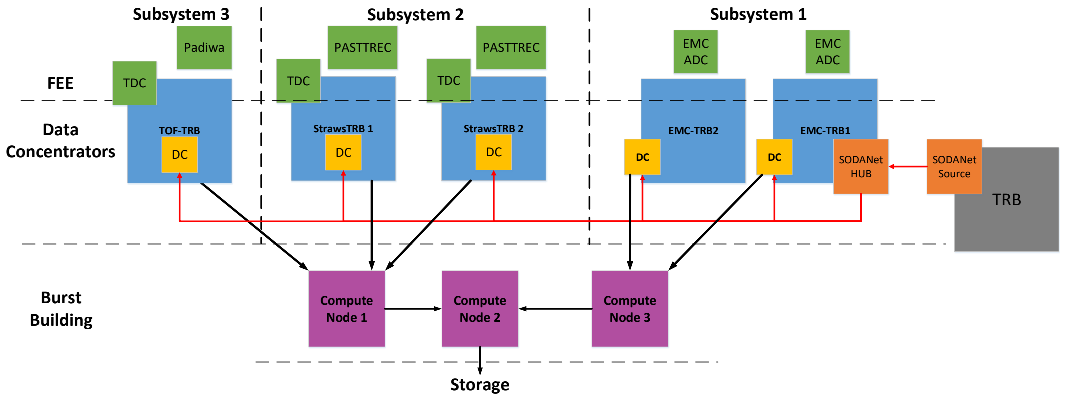
www.fpgafais.com

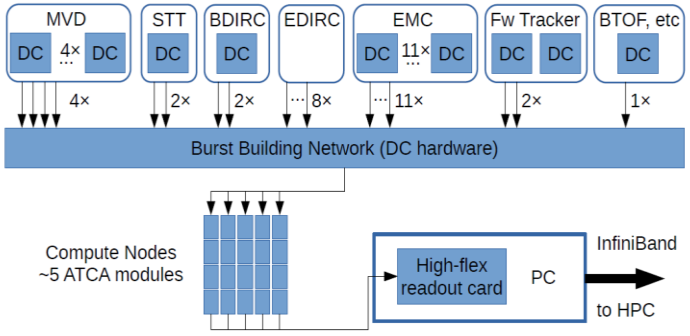
Future of PANDA

- Launch in 2025
- Cracow involved in:
  - Straw trackers readout
  - Data acquisition system
- DAQ – playground for new concepts
- Successful beamtime at Juelich 2019



M. Kavatsyuk

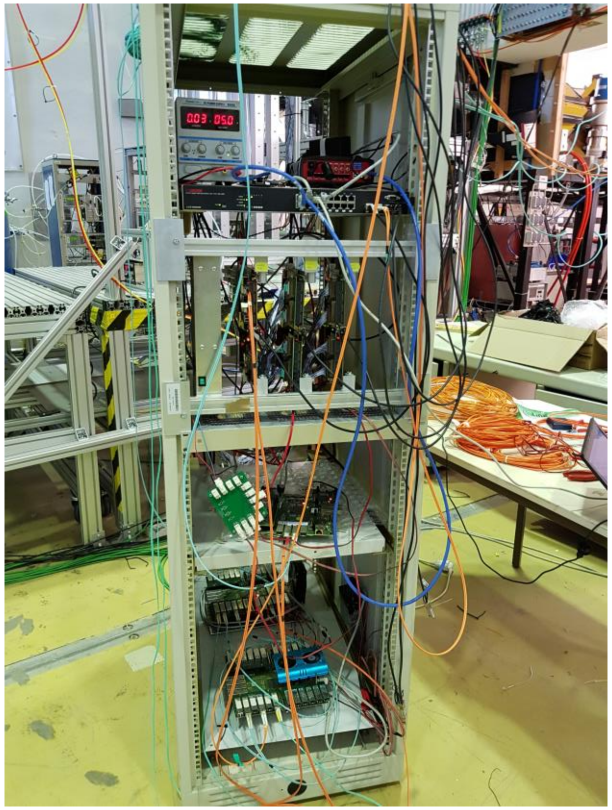




M. Kavatsyuk

Future of PANDA

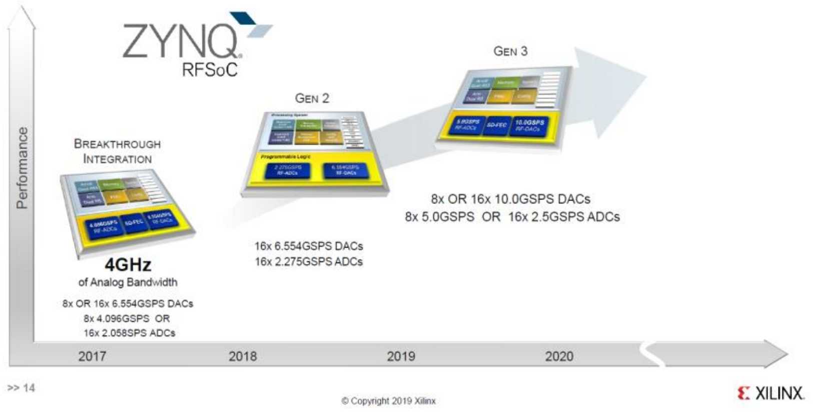
- Basic readout system accepted, evaluated and verified
- Upgrade readout for Straw trackers founds secured
  - Production and evaluation of ASICs
    - ASIC – AGH, cards - UJ
  - Development of next generation of readout system
    - TRB platform – cooperation with GSI
    - Time measurement
    - Communication and synchronization features
  - Development of online tracking procedures
    - FPGA / GPU



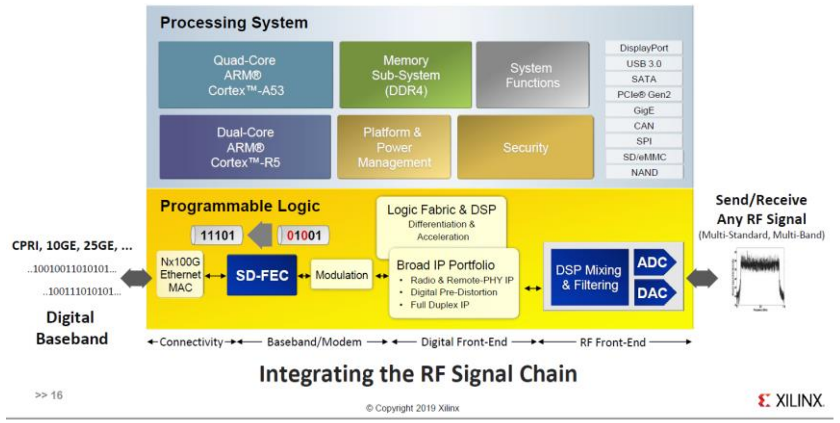
- https://www.youtube.com/watch?v=PnDv_iij5Po

What is next to come

- ZYNQ RFSoC
  - Integrated DAC and ADC channels



ZYNQ RFSoC
Performance
Breakthrough Integration
4GHz
of Analog Bandwidth
8x OR 16x 6.554GSPS DACs
8x 4.096GSPS OR
16x 2.058SPS ADCs
Gen 2
16x 6.554GSPS DACs
16x 2.275GSPS ADCs
Gen 3
8x OR 16x 10.0GSPS DACs
8x 5.0GSPS OR 16x 2.5GSPS ADCs
2017
2018
2019
2020
>> 14
© Copyright 2019 Xilinx
XILINX



Processing System
Quad-Core ARM® Cortex™-A53
Memory Sub-System (DDR4)
System Functions
Dual-Core ARM® Cortex™-R5
Platform & Power Management
Security
DisplayPort
USB 3.0
SATA
PCIe® Gen2
GigE
CAN
SPI
SD/eMMC
NAND
Programmable Logic
11101
01001
Logic Fabric & DSP
Differentiation & Acceleration
CPRI, 10GE, 25GE, ...
..10010011010101...
..100111010101...
Digital Baseband
Nx100G Ethernet MAC
SD-FEC
Modulation
Broad IP Portfolio
Radio & Remote-PHY IP
Digital Pre-Distortion
Full Duplex IP
DSP Mixing & Filtering
ADC
DAC
Send/Receive Any RF Signal
(Multi-Standard, Multi-Band)
Connectivity
Baseband/Modem
Digital Front-End
RF Front-End
Integrating the RF Signal Chain
>> 16
© Copyright 2019 Xilinx
XILINX