EPICS Collaboration Meeting June 2019

Methodology to standardize the development of FPGA-based intelligent DAQ and processing systems on heterogeneous platforms using OpenCL

M. Astrain[1], M. Ruiz[1], S. Esquembri[1], A. Carpeño[1], E. Barrera[1], J. Vega[2]

[1]Instrumentation and Applied Acoustic Research Group, Universidad Politécnica de Madrid, Madrid, Spain

[2]Laboratorio Nacional de Fusión, CIEMAT, Madrid, Spain

GRUPO DE INVESTIGACIÓN EN INSTRUMENTACIÓN Y ACÚSTICA APLICADA

miguel.astrain@i2a2.upm.es

UPM BIG SCIENCE

POLITÉCNICA

Technical University of Madrid

- With nearly 40.000 students and a annual budget of 400Mє, UPM is one the leading technical universities in Spain. Its organized in 20 engineering schools and 200 Research Groups

- UPM one of the top Spanish Universities in the European Research Ranking

- I2A2-UPM, research and development group working in advanced instrumentation systems:
  - Embedded platforms using SoCs, FPGAS, Embedded Linux
  - Software development including development of EPICS device supports/drivers
  - Collaborator of ITER CODAC since 2009
  - Specific training in Embedded Systems, FPGA and SoCs development
  - Engineers from UPM working in the EPICS community (FRIB and ITER)

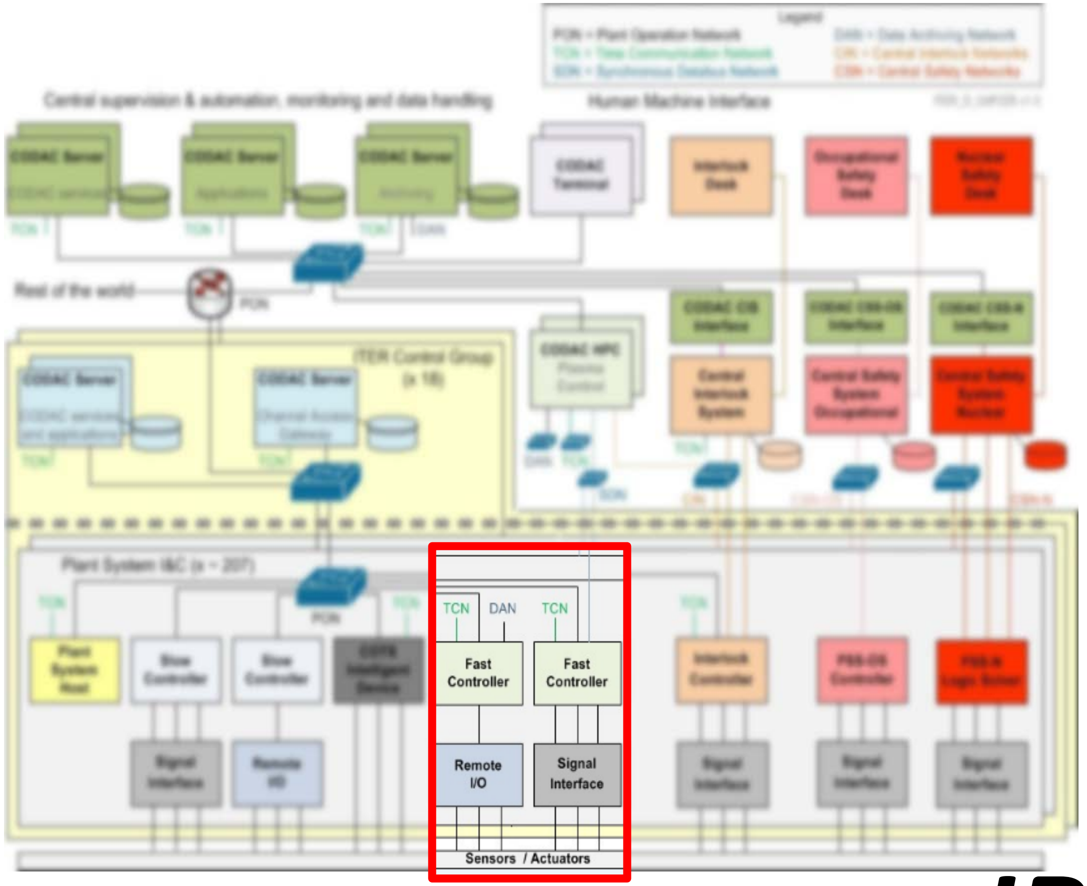
OUTLINE

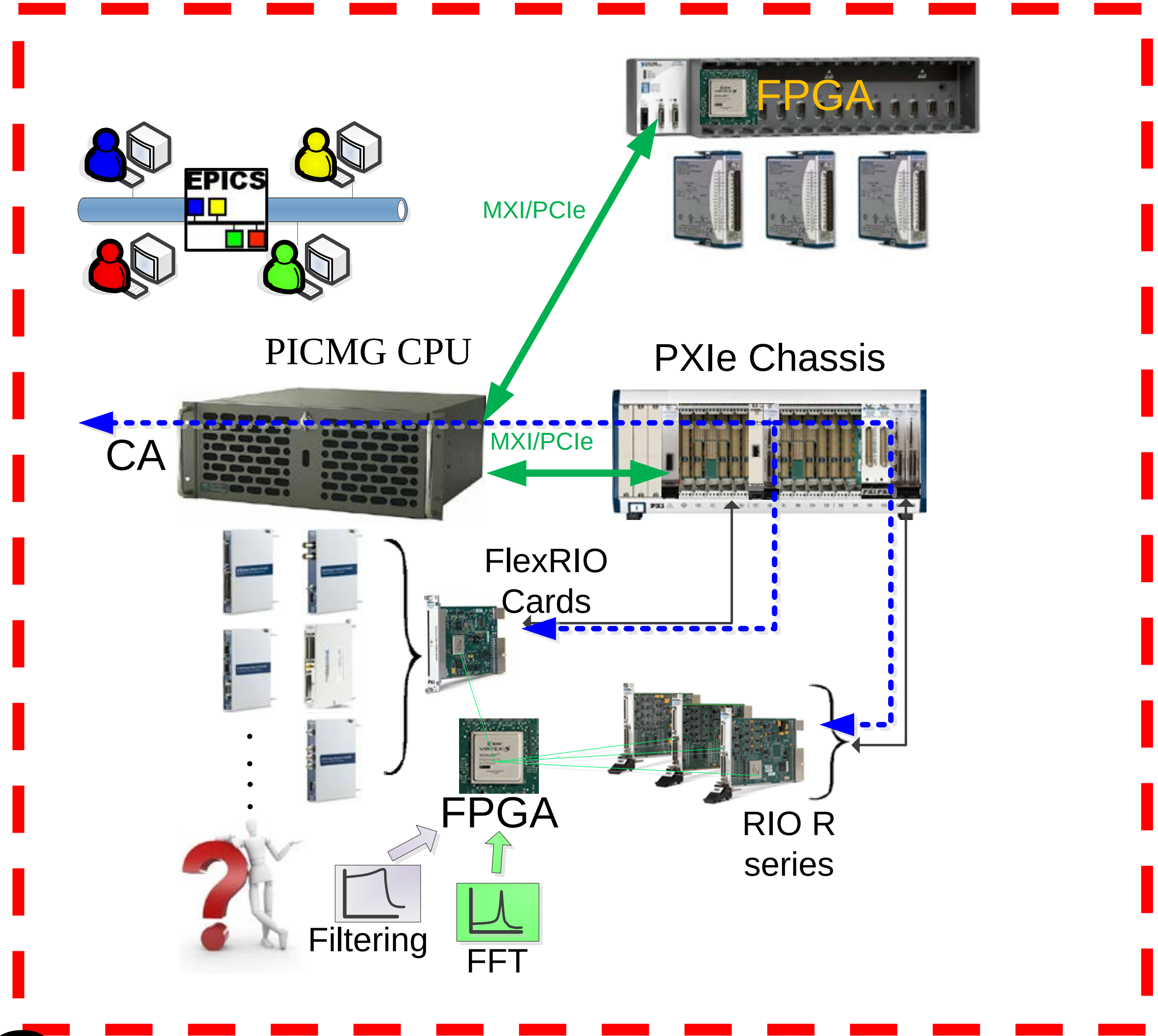
- Motivation
- System Architecture
- OpenCL standard
- Development cycle
- Results
- Conclusions

Motivation

- Simplify the development and integration of advanced DAQ systems DAQ system using FPGAs
  - Can we avoid or minimize the use of Hardware Description Languages?
  - Can we standardize the DAQ functionality to shift development effort only to the data processing?
- Standardize the integration of EPICS device drivers
  - Can we standardize the software to simplify the development/maintenance of applications?
    - asynDriver
    - Nominal Device Support
    - Same API and Kernel Linux Device Driver for all hardware devices (using OpenCL).

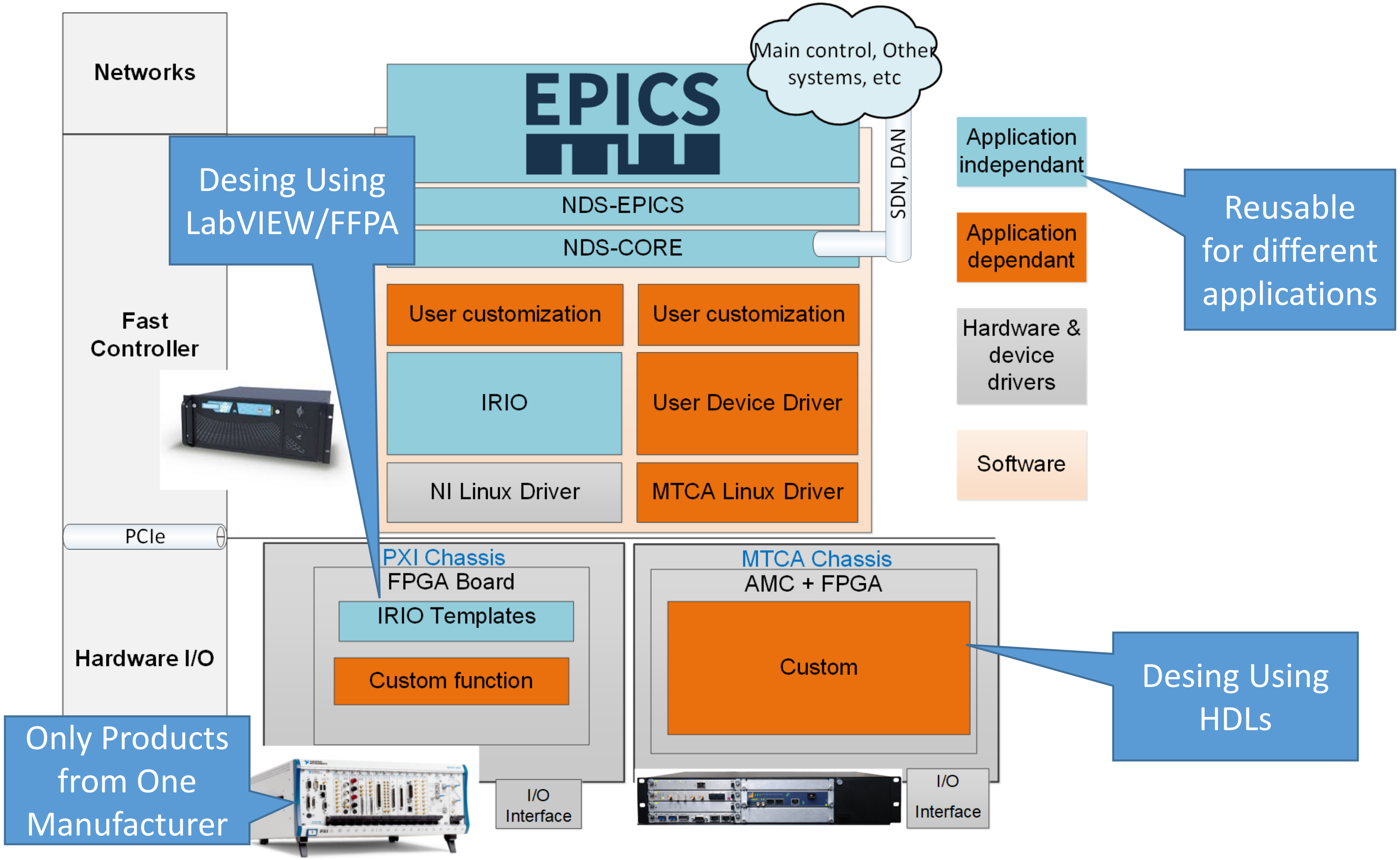
Our previous experience with DAQ-FPGA and EPICS





IRIO https://github.com/irio-i2a2

FPGA DAQ-based systems in ITER Fast Controllers



Enhancements

- Simplify the development of DAQ system using FPGAs
  - Can we avoid or minimize the use of Hardware Description Languages? Yes, we can use OpenCL as a high level synthesis language
  - Can we standardize the DAQ functionality to shift development effort only to the data processing? Yes, we can propose a model using high level description in OpenCL with solutions already given for Data Acquisition
  - Can we standardize the software to simplify the development/maintenance of applications?Yes, we can use NDSv3 modular design to separate and trace functionality changing its description in OpenCL. This functionality is even supported “in runtime” thanks to partial reconfiguration
- Let’s have a look to OpenCL

OpenCL: Overview

Implementers
Desktop/Mobile/Embedded/FPGA

AMD
IBM
intel
NVIDIA.
ARM
Imagination
QUALCOMM
VIVANTE
MEDIATEK
ST
SAMSUNG
ALTERA
XILINX.
TEXAS INSTRUMENTS
MARVELL

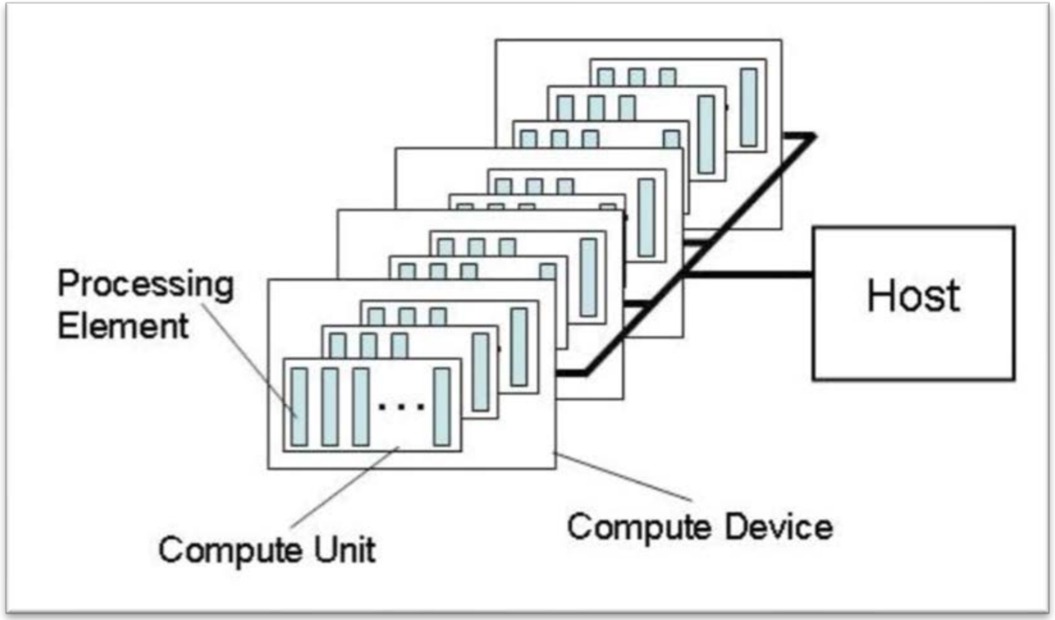
SYCL™
Single Source C++ Programming

OpenCL
Core API and Language Specs

SPIR™
Portable Kernel Intermediate Language

Working Group Members
Apps/Tools/Tests/Courseware

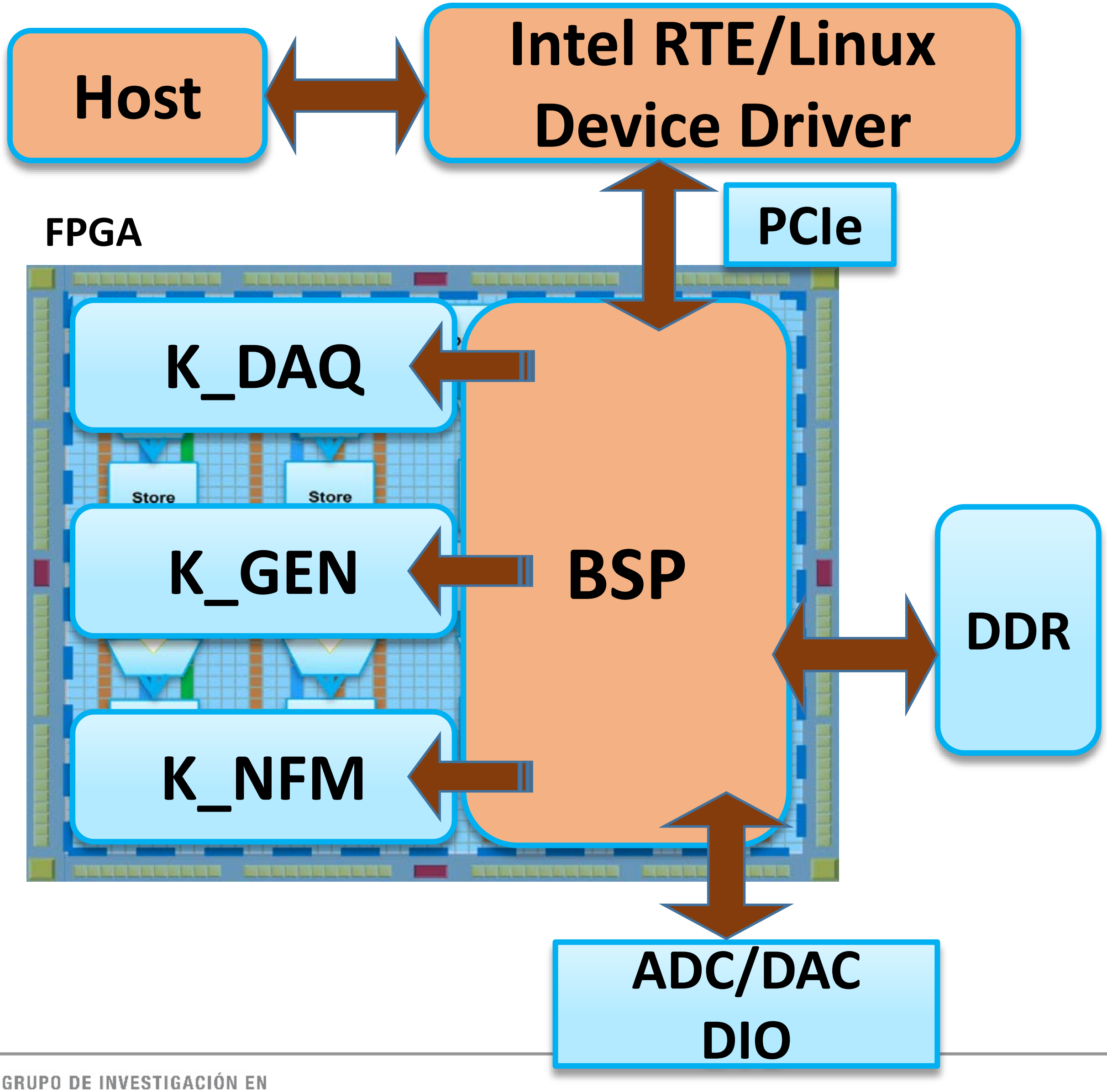
Adobe
codeplay
HUAWEI
mobica
freescale semiconductor
SONY
University of BRISTOL
University of Windsor
MULTICORE WARE
vmware
Los Alamos NATIONAL LABORATORY



HLS + COMPUTING MODEL

- A **host** and multiple **devices (CPU, GPU, FPGA)**
- Computation is divided into *tasks* called **Kernels**
- There is one or several **queues** that send the **Kernels** to execute concurrently
- Memory organized in **buffer/image**. Memory model

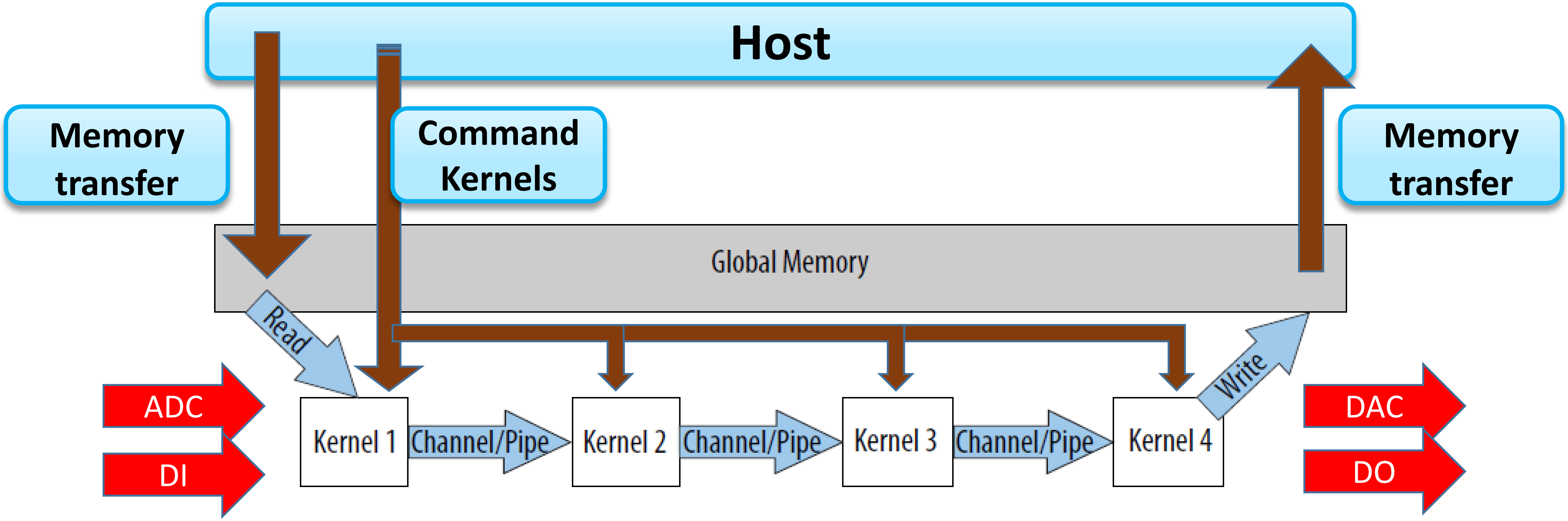
OpenCL FPGA Device: Kernel and BSP concept



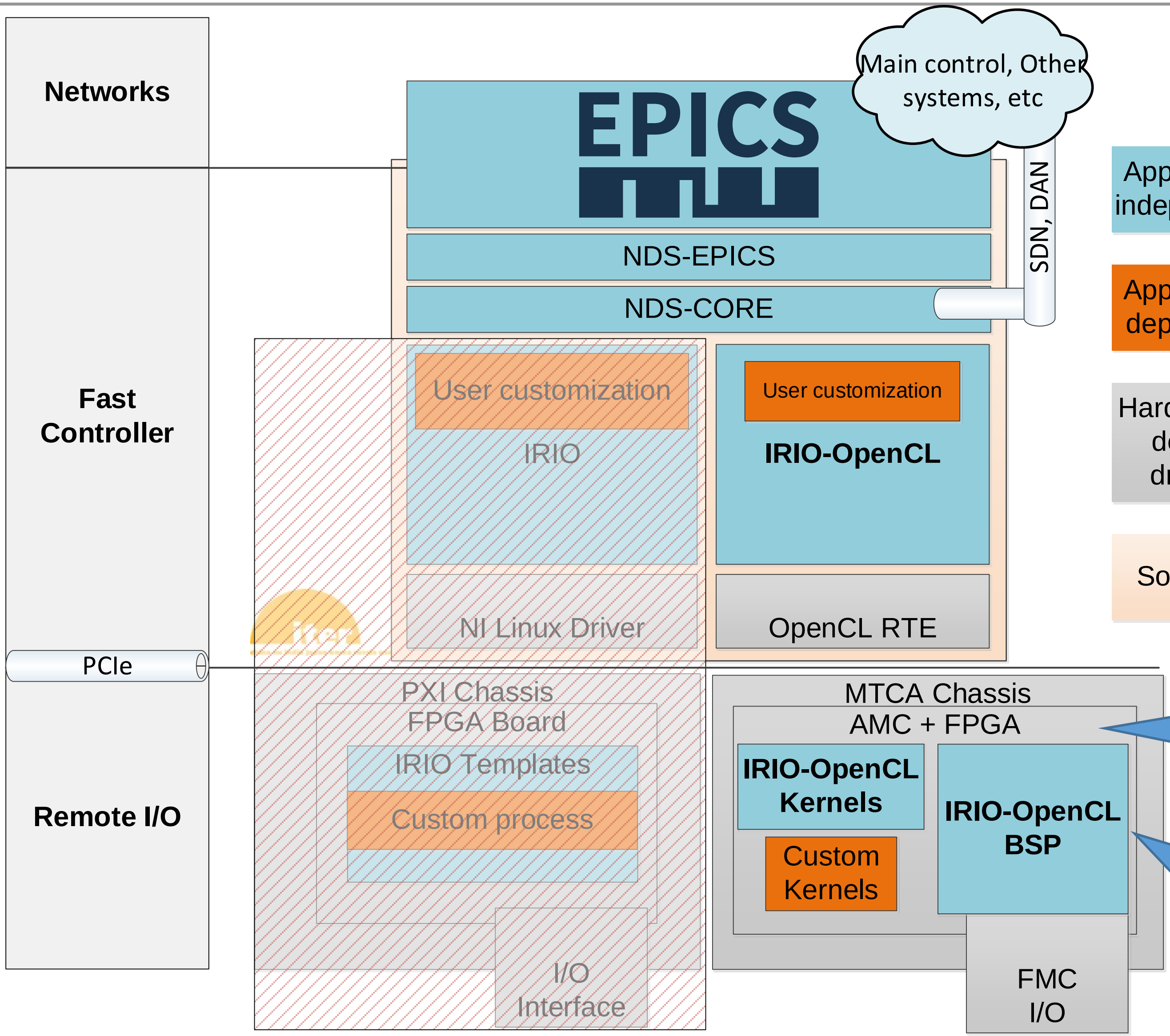
- **Kernels:** The processing pieces of OpenCL that go into the FPGA and are allocated in a specific FPGA partition to allow **partial reconfiguration**
- **Kernels are written in C (OpenCL C).**
- Kernels have access to all device memory layers
- Parallelization is achieved replicating kernels and using loop pipeline
- **BSP:** The part of OpenCL that goes into the FPGA and is FIXED
- Manages the access to hardware (DDR, PCIe, I/O)
- Manages the Queues to the kernels

OpenCL Host

- **Short: The part of OPENCL that goes into the C++ drivers (HOST)**
- Host sends **commands** and can read/write Global Memory (**DDR**) (SLOW!) and controls kernels execution, synchronization tasks, data-flow to and from device
- Critical processes can be organized with a chain of pipes (HDL=AXI ST)
- Data can be gathered from I/O, but kernels are launched with commands



IRIO-OpenCL



Application independant

Application dependant

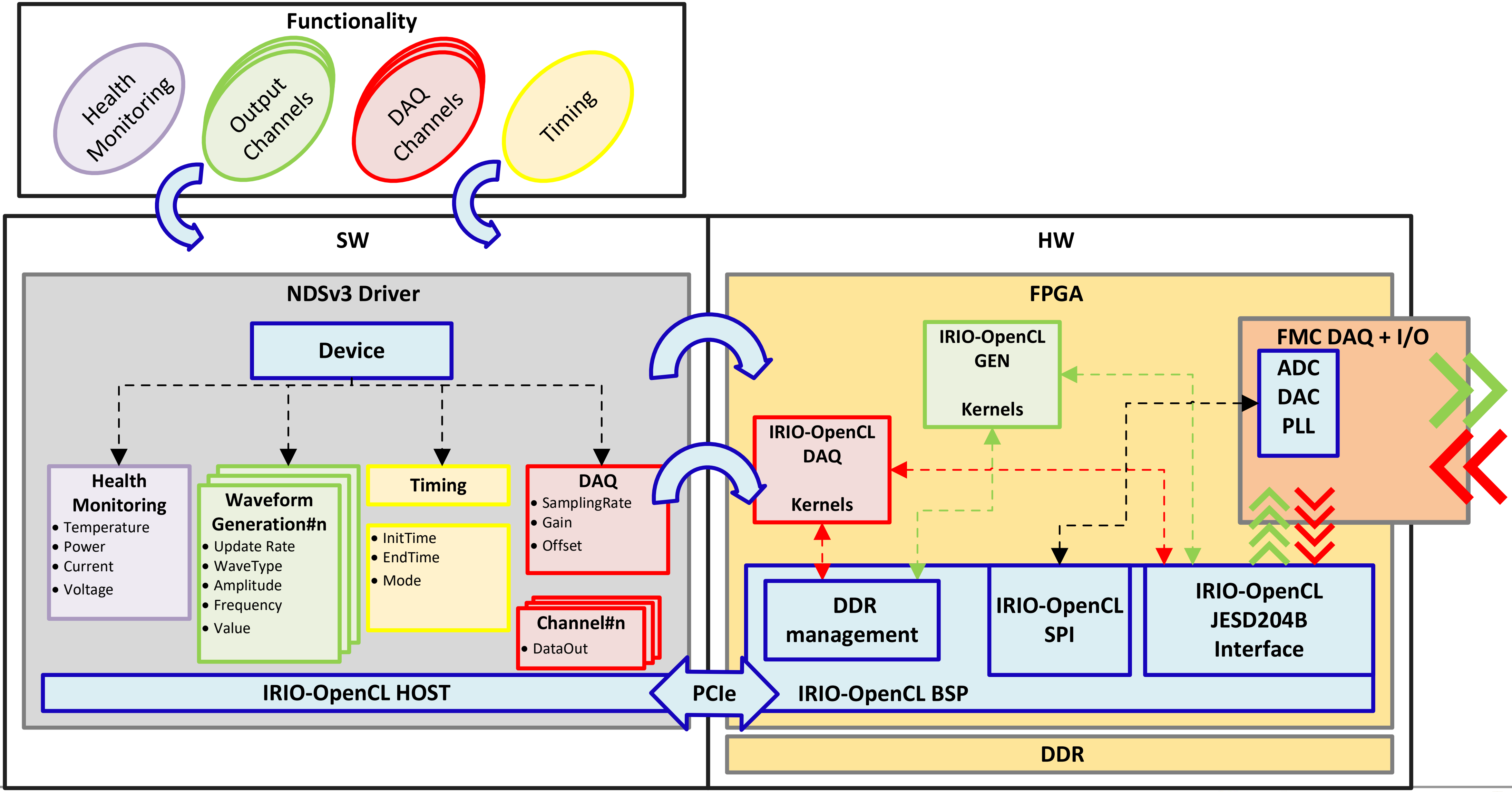
Hardware & device drivers

Software

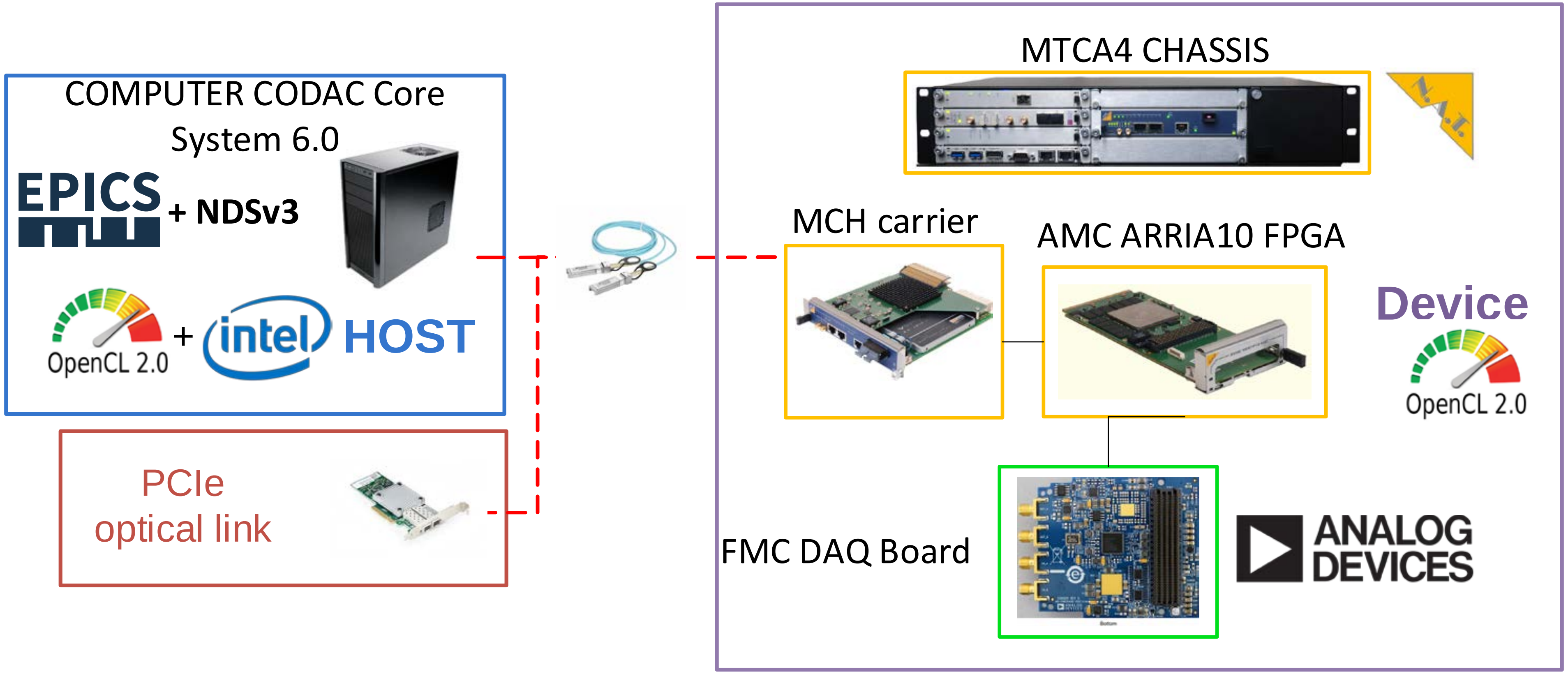
Desing Using OpenCL(80%) HDL(20%)

Products from multiple Manufacturers

Recap: NDSv3 + OpenCL



Results



COMPUTER CODAC Core System 6.0

EPICS + NDSv3

OpenCL 2.0 + intel HOST

PCIe optical link

MTCA4 CHASSIS

N.A.T.

MCH carrier

AMC ARRIA10 FPGA

Device

OpenCL 2.0

FMC DAQ Board

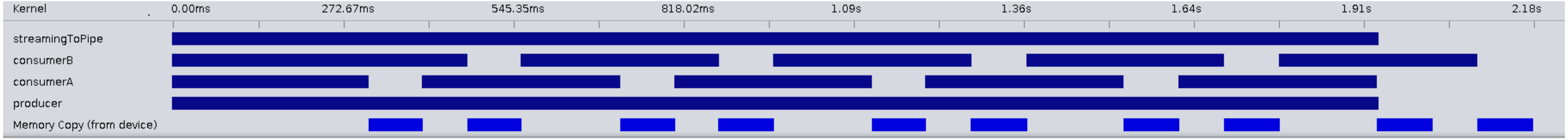
ANALOG DEVICES

Everything is COTS!!!

Results

- Solution integrated into ITER CODAC Core System 6.0
- Simple kernel programming
- Access to profiling tools to tune performance.
- No large overhead due to the BSP (~15%)
- Implemented solution on a Neutron Flux Measurement use case at 2 channels @ 1GS/s
- Integrated into EPICS thanks to NDSv3
- Kernel functional validation using emulation

```
__kernel void ADC_reader (
            uint size,
            __global uint8* restrict data)
{
        for (uint i = 0; i < size; i++){
            data[i] = read_channel_Intel(rx_link);
            mem_fence(CLK_CHANNEL_MEM_FENCE);
        }
}
```



Conclusions

- Standardization of the development of FPGA-based DAQ devices using OpenCL
- Tested in a MTCA platform using an ARRIA 10 FPGA
- Reuse of FPGA development using an OpenCL hardware description of a DAQ device
- OpenCL enables C-like development of FPGA with lots of OpenCL algorithms examples
- OpenCL handles data transfers and device interface, hardware abstraction
- Combined with NDSv3 a modular solution was developed, simplifying the interface with EPICS. **IRIO-OpenCL**
- **You only need to focus on "your specific algorithm"**

"One driver to rule them all…"

Future Work

- Implementation of Machine Learning Applications
- Expand IRIO-OpenCL functionalities
- **Implementation of use cases (possible collaborations)**

IRIO-OpenCL Team -> Contact us!!

UPM Big Science contact:
jm.cogollor@upm.es

I2A2-UPM contact:
mariano.ruiz@upm.es

Methodology to standardize the development of FPGA-based intelligent DAQ and processing systems on heterogeneous platforms using OpenCL

Thank You !

Questions?

M. Astrain[1], M. Ruiz[1], S. Esquembri[1], A. Carpeño[1], E. Barrera[1], J. Vega[2]

[1]Instrumentation and Applied Acoustic Research Group, Universidad Politécnica de Madrid, Madrid, Spain

[2]Laboratorio Nacional de Fusión, CIEMAT, Madrid, Spain

GRUPO DE INVESTIGACIÓN EN INSTRUMENTACIÓN Y ACÚSTICA APLICADA

miguel.astrain@i2a2.upm.es

UPM BIG SCIENCE

POLITÉCNICA

Acknowledgements

This work was supported in part by the Spanish Ministry of Economy and Competitiveness, **Projects Nº ENE2015-64914-C3-3-R** and Madrid regional government (YEI fund), **Grant Nº PEJD-2018-PRE/TIC-8571**

GOBIERNO DE ESPAÑA

MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD

Comunidad de Madrid

European Union
Investing in your Future
European Social Fund

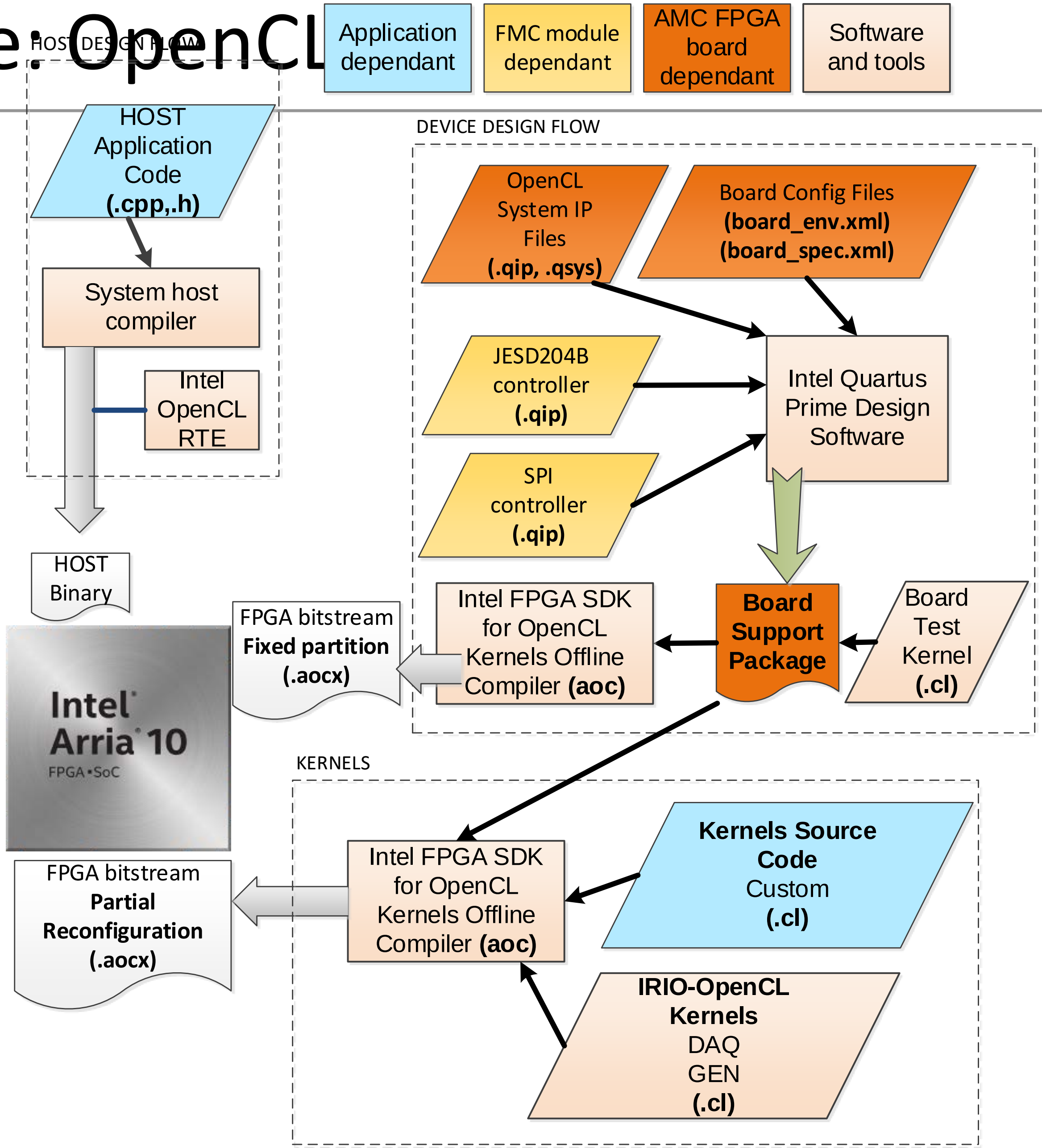
Development cycle: OpenCL

Three main scenarios for a new application:

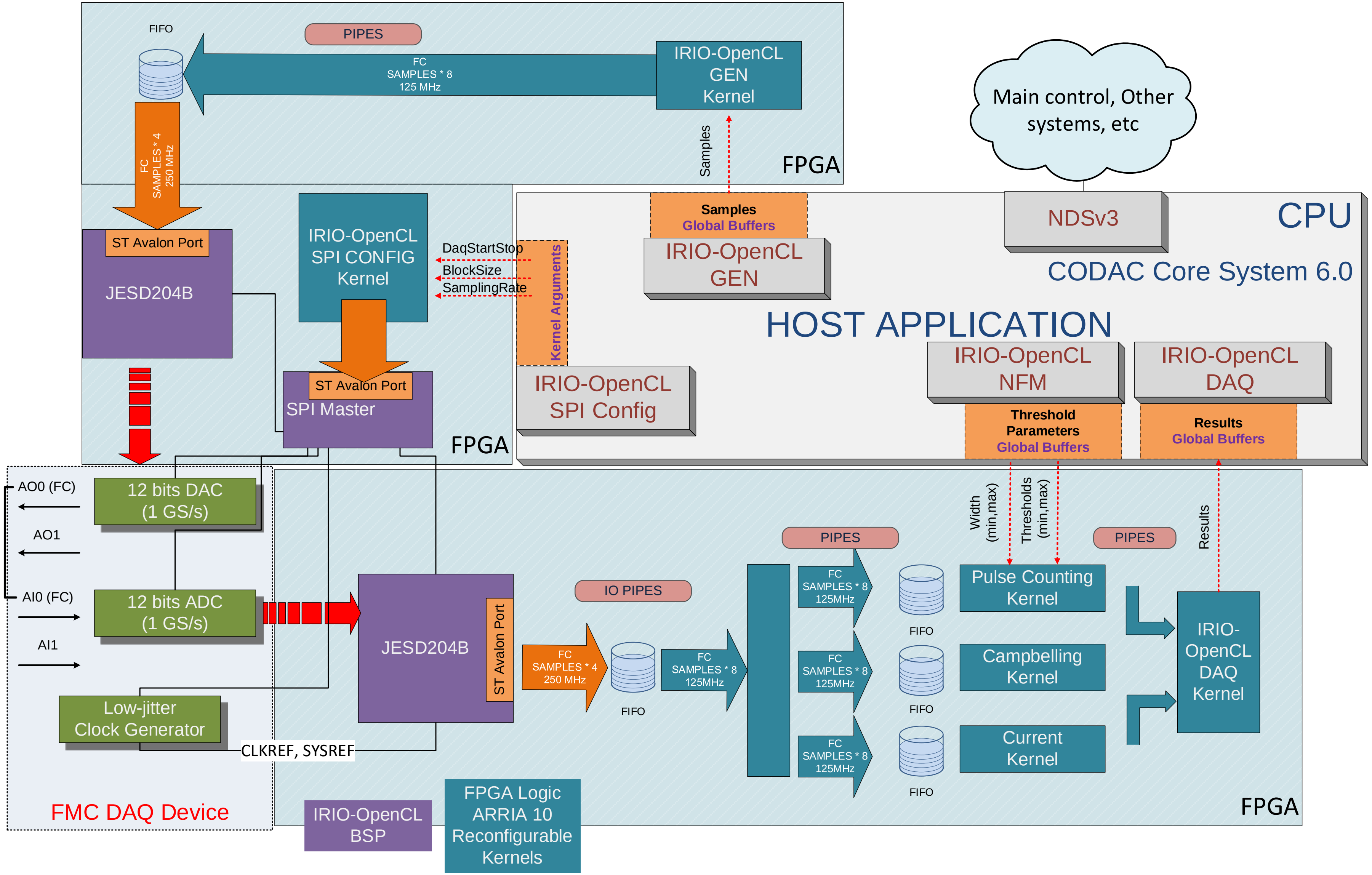
1- New algorithm

2- New algorithm + FMC module

3- New algorithm + FMC + AMC + FPGA


Application dependant
FMC module dependant
AMC FPGA board dependant
Software and tools
HOST DESIGN FLOW
HOST Application Code (.cpp,.h)
System host compiler
Intel OpenCL RTE
HOST Binary
Intel Arria 10
FPGA • SoC
FPGA bitstream Fixed partition (.aocx)
FPGA bitstream Partial Reconfiguration (.aocx)
DEVICE DESIGN FLOW
OpenCL System IP Files (.qip, .qsys)
Board Config Files (board_env.xml) (board_spec.xml)
JESD204B controller (.qip)
SPI controller (.qip)
Intel Quartus Prime Design Software
Intel FPGA SDK for OpenCL Kernels Offline Compiler (aoc)
Board Support Package
Board Test Kernel (.cl)
KERNELS
Intel FPGA SDK for OpenCL Kernels Offline Compiler (aoc)
Kernels Source Code Custom (.cl)
IRIO-OpenCL Kernels DAQ GEN (.cl)

Neutron Flux Measurement


FIFO
PIPES
FC
SAMPLES * 8
125 MHz
IRIO-OpenCL
GEN
Kernel
FC
SAMPLES * 4
250 MHz
Samples
FPGA
Main control, Other systems, etc
ST Avalon Port
JESD204B
IRIO-OpenCL
SPI CONFIG
Kernel
DaqStartStop
BlockSize
SamplingRate
Kernel Arguments
Samples
Global Buffers
IRIO-OpenCL
GEN
NDSv3
CPU
CODAC Core System 6.0
HOST APPLICATION
ST Avalon Port
SPI Master
FPGA
IRIO-OpenCL
SPI Config
IRIO-OpenCL
NFM
IRIO-OpenCL
DAQ
Threshold
Parameters
Global Buffers
Results
Global Buffers
AO0 (FC)
12 bits DAC
(1 GS/s)
AO1
AI0 (FC)
12 bits ADC
(1 GS/s)
AI1
Low-jitter
Clock Generator
CLKREF, SYSREF
FMC DAQ Device
JESD204B
ST Avalon Port
IO PIPES
FC
SAMPLES * 4
250 MHz
FIFO
FC
SAMPLES * 8
125MHz
PIPES
FC
SAMPLES * 8
125MHz
FIFO
FC
SAMPLES * 8
125MHz
FIFO
FC
SAMPLES * 8
125MHz
FIFO
Width
(min,max)
Thresholds
(min,max)
Pulse Counting
Kernel
Campbelling
Kernel
Current
Kernel
PIPES
Results
IRIO-
OpenCL
DAQ
Kernel
FPGA
IRIO-OpenCL
BSP
FPGA Logic
ARRIA 10
Reconfigurable
Kernels