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

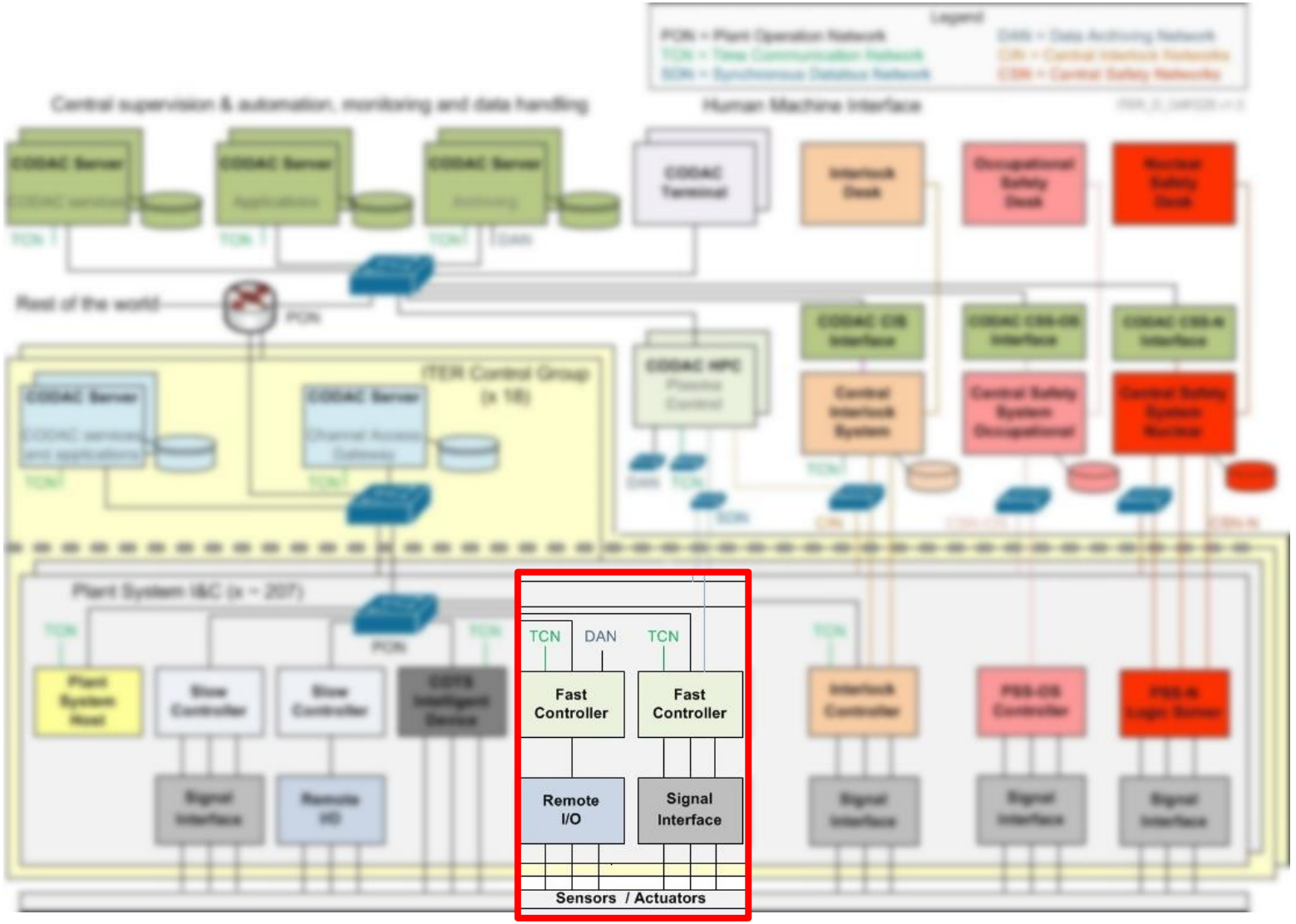
[2]Laboratorio Nacional de Fusión, CIEMAT, Madrid, Spain

OUTLINE

- Context
- System Architecture Example
- OpenCL standard
- Development cycle
- Results
- Conclusions

miguel.astrain@i2a2.upm.es

Context



miguel.astrain@i2a2.upm.es

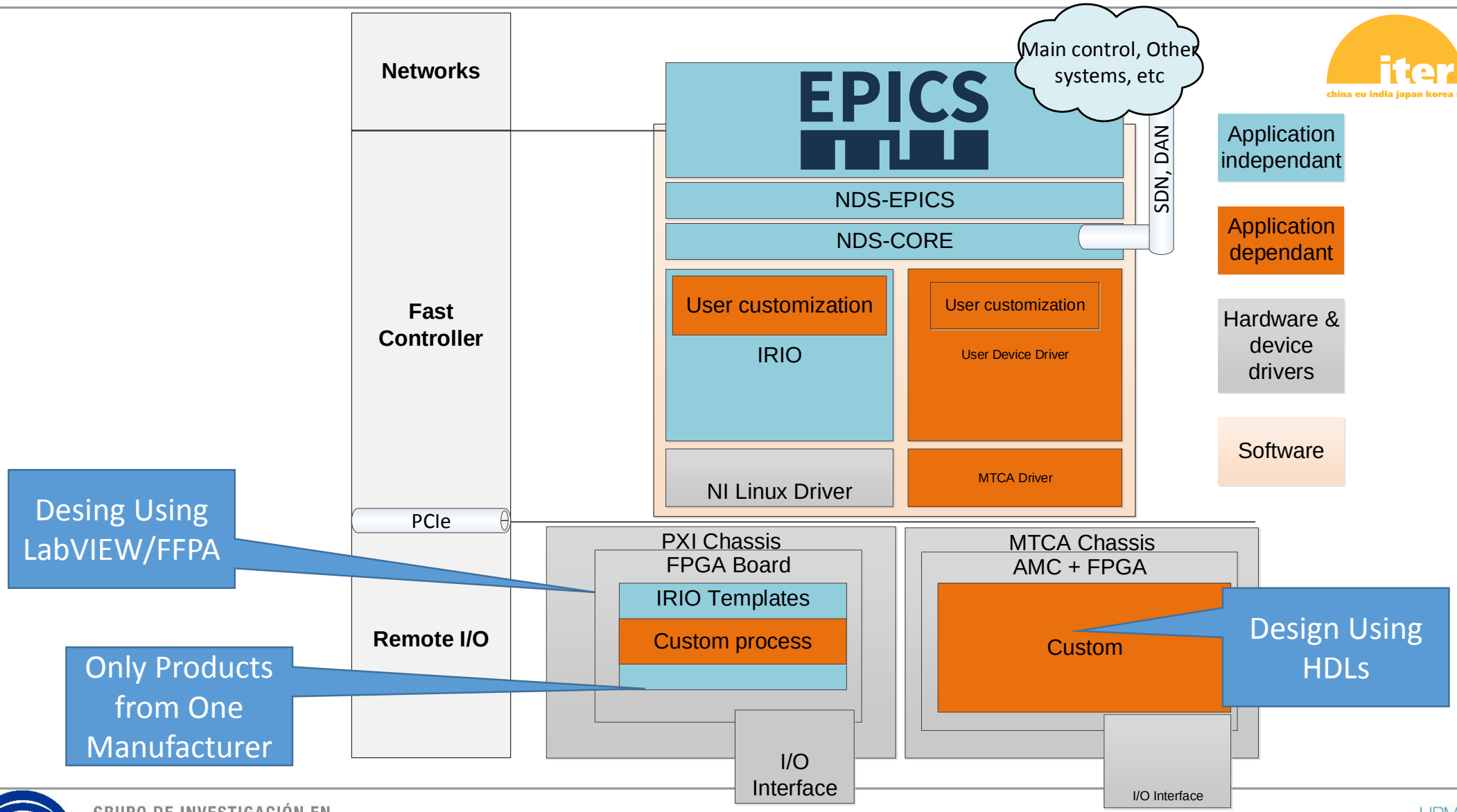
Context

Advantages of Data Acquisition (DAQ) systems based on FPGAs

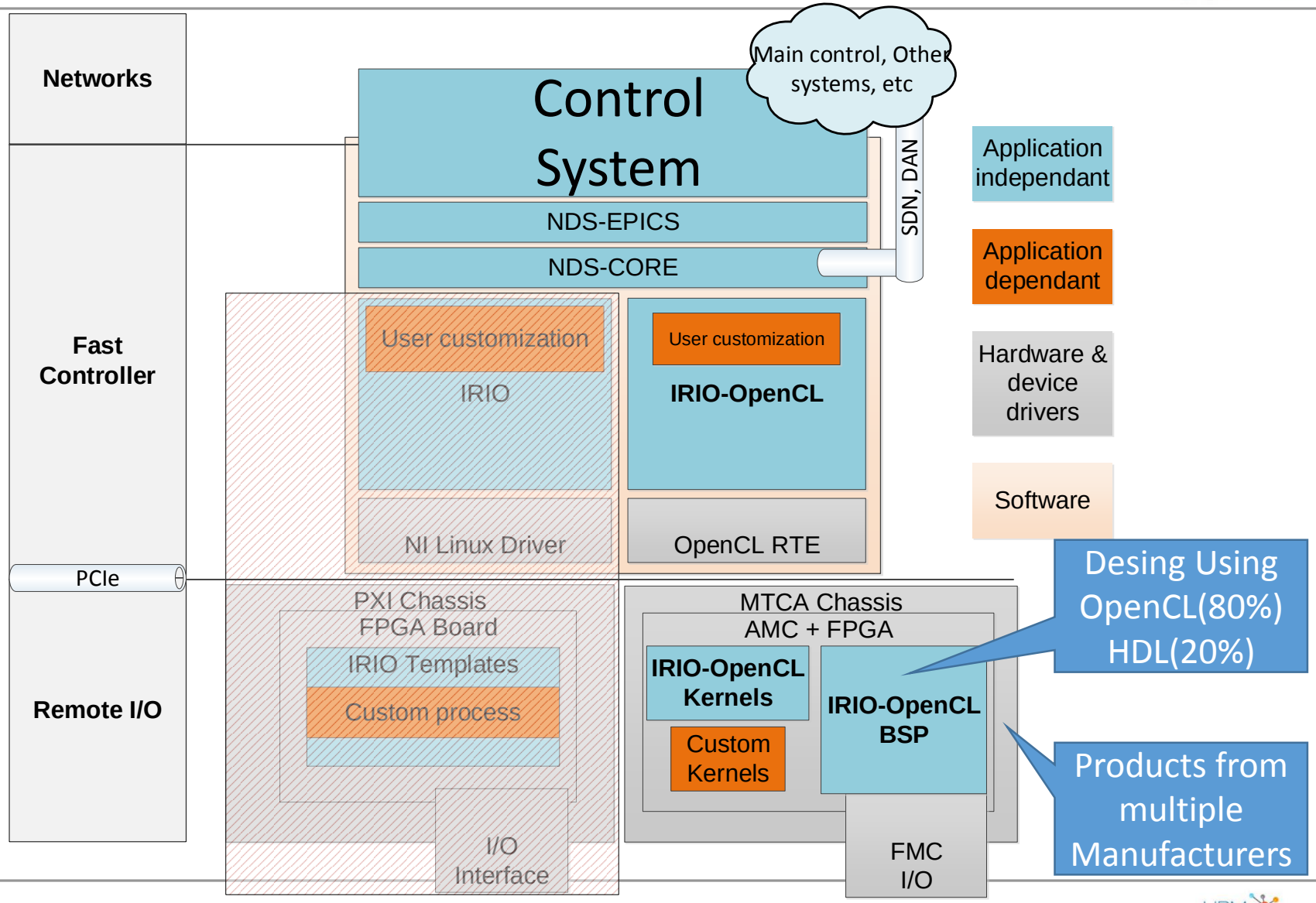
+ Flexibility to modify the design
+ Low latency
+ Deterministic behavior
+ **High-performance processing**
+ **Parallelization**

But developing for FPGA is complex and **time consuming (expensive …)**

Why is it expensive to develop for FPGAs

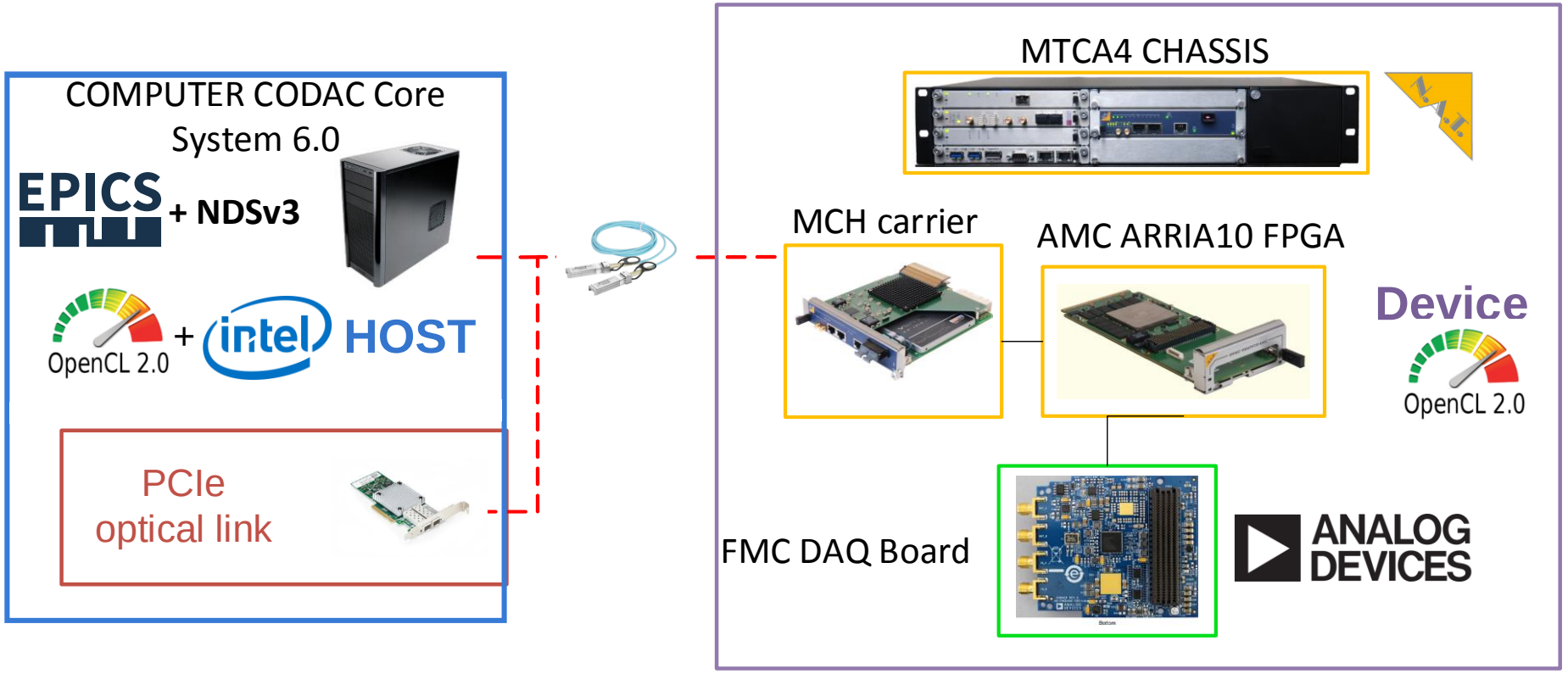


IRIO-OpenCL



miguel.astrain@i2a2.upm.es

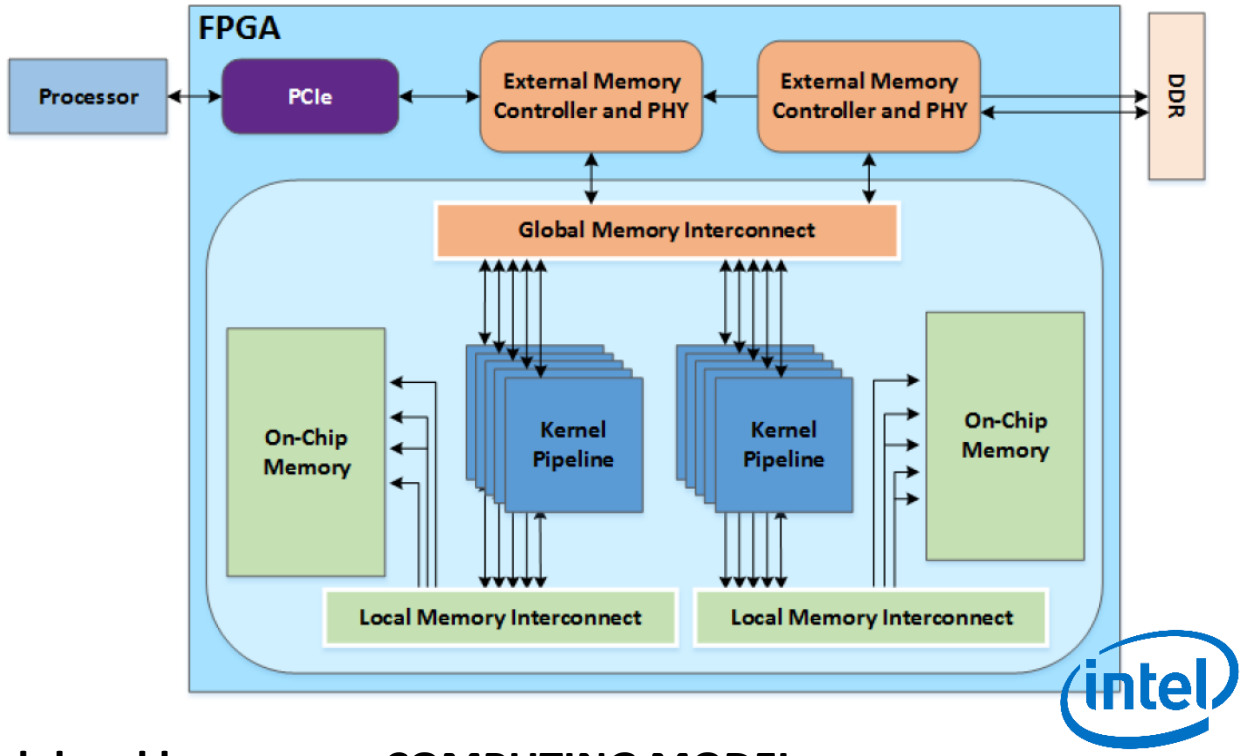
System Architecture Example



- PC + PCIe + MTCA + AMC + (PROCESSING) + (I/O)

miguel.astrain@i2a2.upm.es

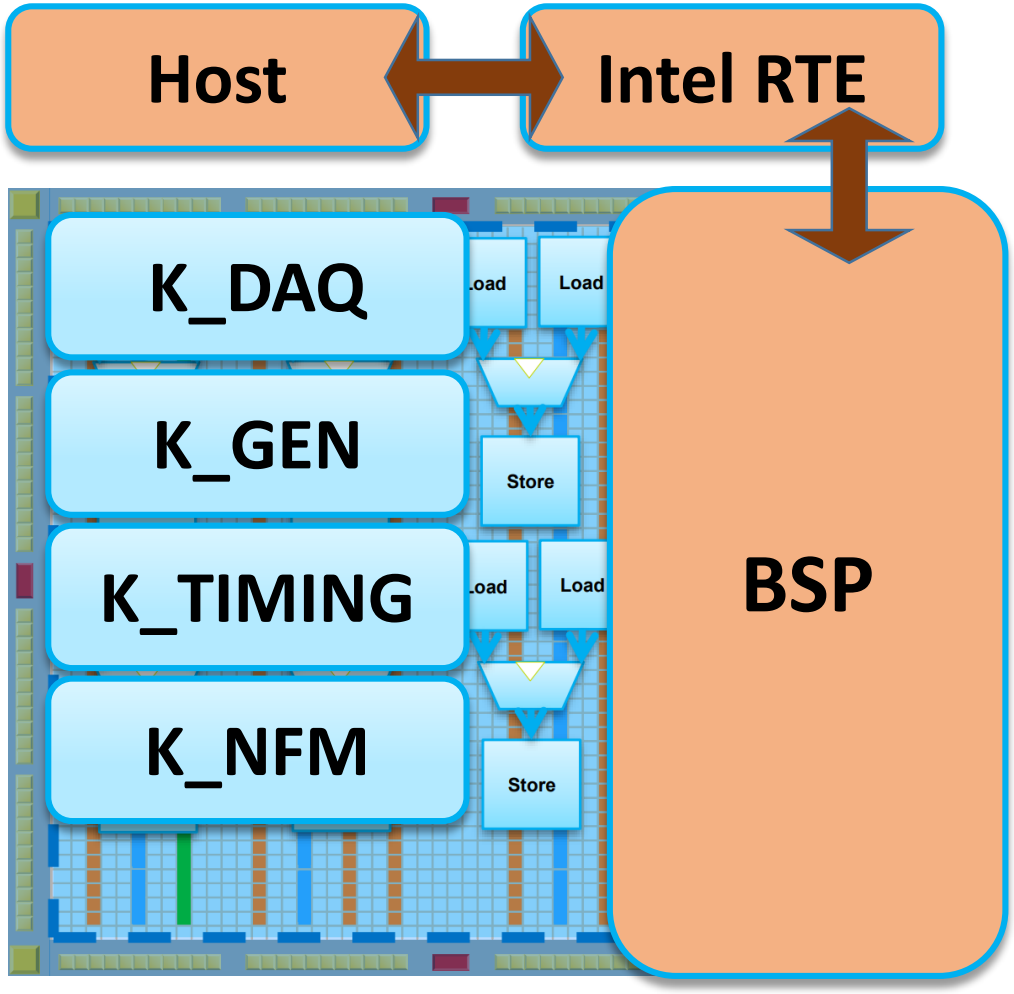
OpenCL: Basic overview



High level language + COMPUTING MODEL

- A **host** and multiple **devices (CPU, GPU, FPGA)**.
- Computation is divided into *functions* called **Kernels**.
- There is one or several **queues** that send the **Kernels** to execute concurrently.
- Memory organized in **buffers/images** and transfers are explicit.
- Parallelization is a big focus.

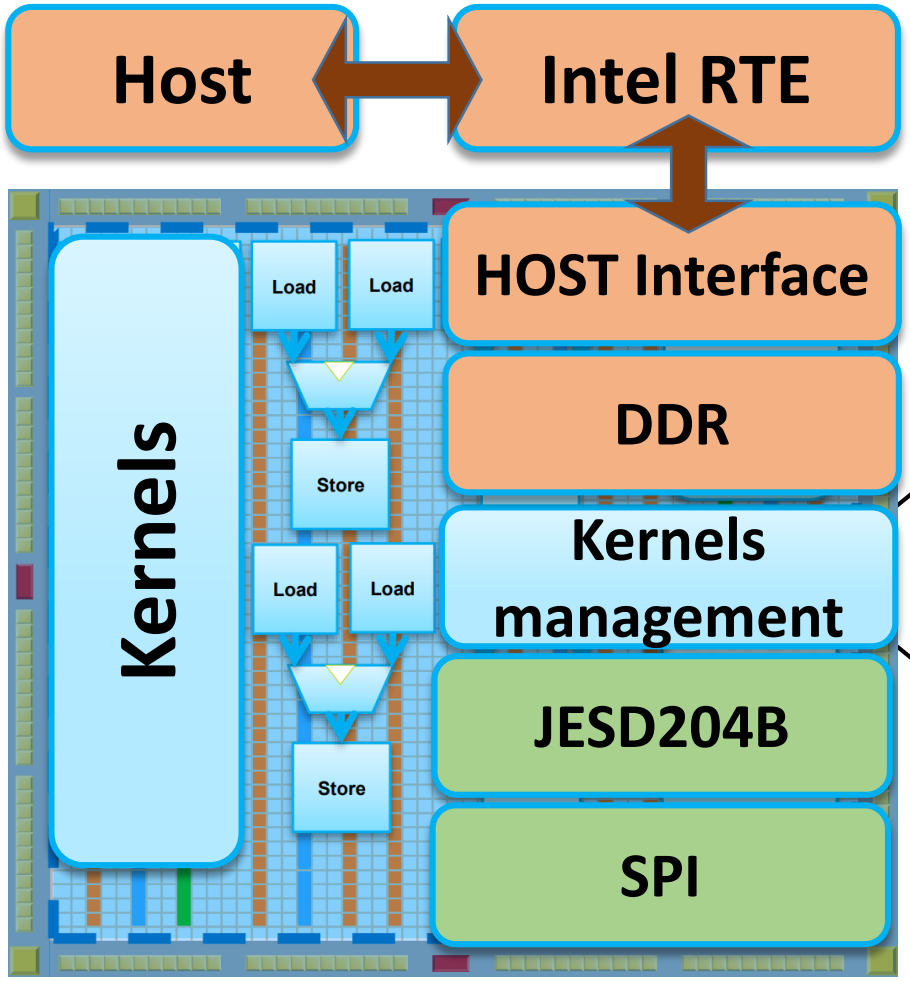
OpenCL: Kernels

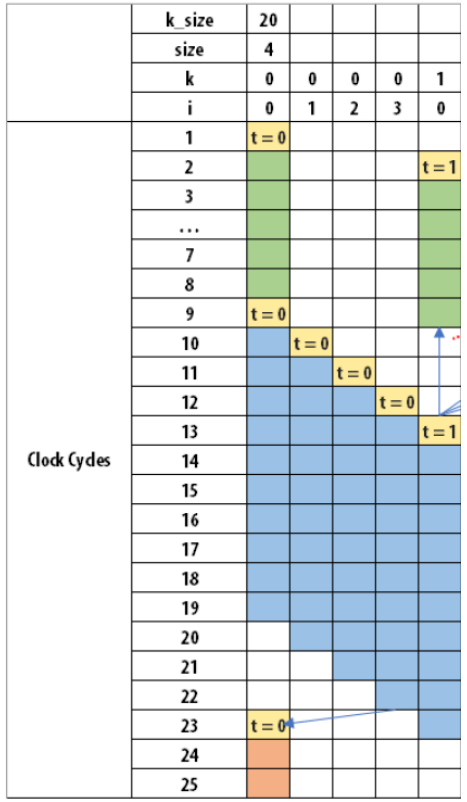


- **Short:** The part of OPENCL that goes into the FPGA and is allocated in the FPGA in **partial reconfigurable partition**.
- Kernels have access to all device memory layers.
- Key to performance is optimizing this memory usage.
- Parallelization is achieved in the form of a pipeline for FPGA.

miguel.astrain@i2a2.upm.es

OpenCL: BSP

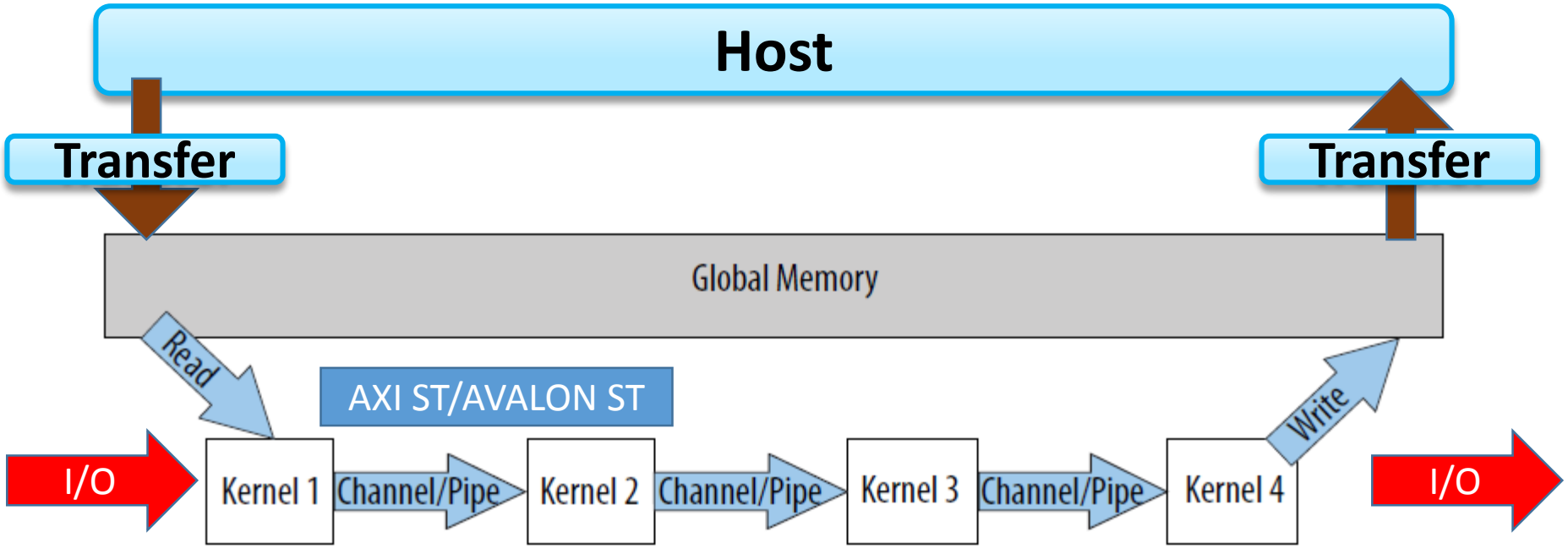




- **Short:** The part of OPENCL that goes into the FPGA and is **FIXED**.
- Manages DDR memory.
- It interfaces with the host
- JESD204B interface
- Requires HDL to modify (usually given)

OpenCL: Host

- **Short: The part of OPENCL that goes into the C++ drivers.**
- Host sends **commands** and can set Global Memory (**DDR**) (SLOW!)
- Critical processes can be organized with a chain of pipes (HDL=AXI ST)
- **Data can be gathered from I/O pins**, but kernels are **queued** from host.



miguel.astrain@i2a2.upm.es

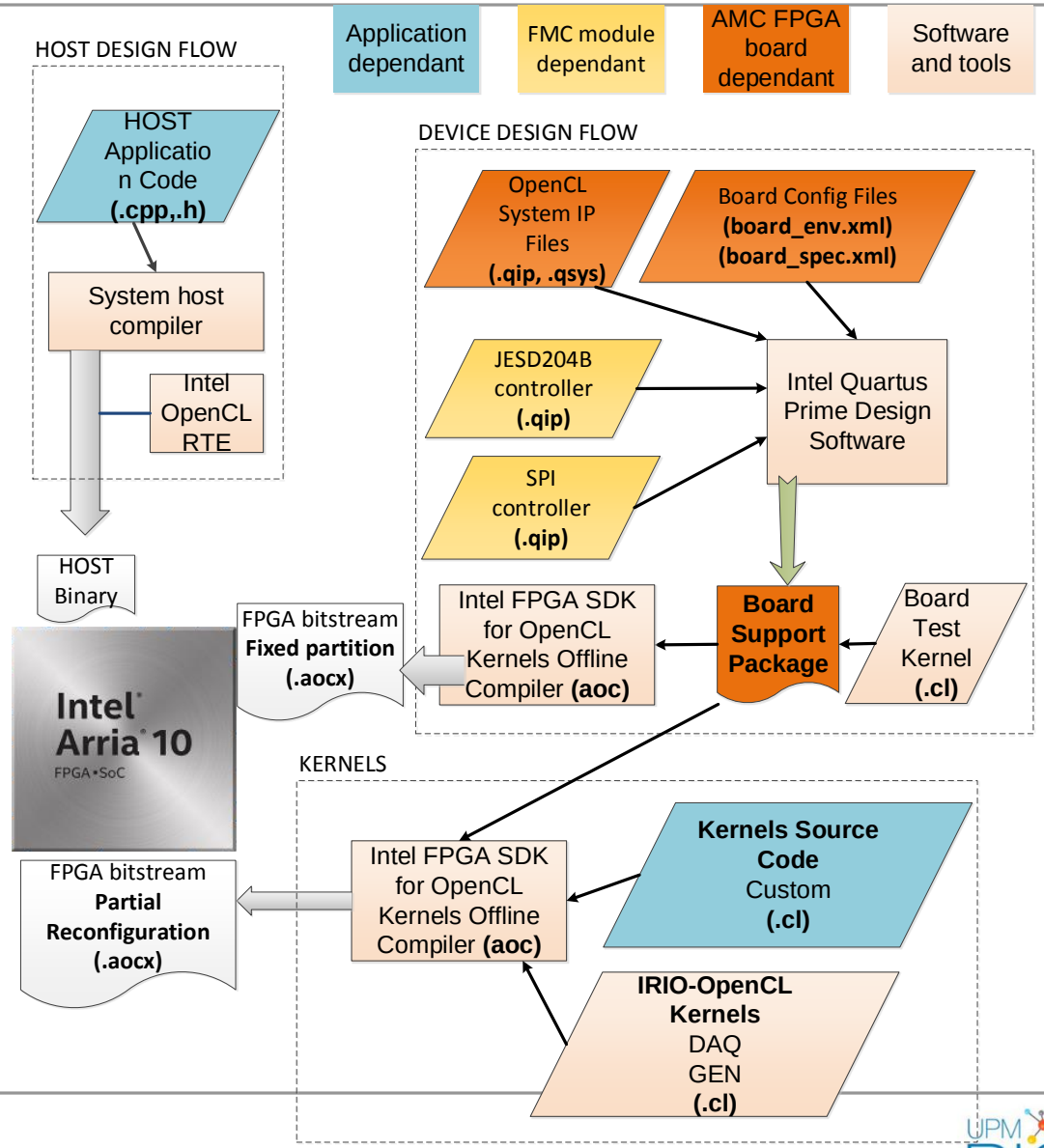
Development cycle: OpenCL

Three main scenarios for a new application:

1- New algorithm

2- FMC module

3- AMC + FPGA



miguel.astrain@i2a2.upm.es

Results

- Implementation: **more on ID 494** !!
- Basic kernels FPGA resource utilization
- Totally integrated in EPICS using NDS
- Tested in CODAC CORE SYSTEM 6.0

Kernel Name	ALUTs	FFs	RAMs	DSPs
consumer	1894	5113	30	0
manager	470	454	0	2
producer	1155	2872	16	0
streamingToPipe	635	400	0	0
Kernel Subtotal	**4154**	**8839**	**46**	**2**
Channel Resources	**164**	**648**	**2**	**0**
Global Interconnect	**1344**	**2870**	**18**	**0**
Board Interface	**66800**	**133600**	**182**	**0**
Total	**72462 (17%)**	**146024 (17%)**	**250 (13%)**	**2 (0%)**
Available	**436560**	**873120**	**1949**	**1687**

Functionality:

- ✓DAQ
- ✓WFG
- ✓Processing
- ✓Routing

Under evaluation:

- Timing
- Triggering
- Routing

miguel.astrain@i2a2.upm.es

Results

- Working example: fission chamber neutron flux measurement.
- Why this example:
  - The measurements benefit from high sampling rates.
  - High data rate but simple operations (floating point).
  - Logic inside the FPGA can classify different pulses
  - Parallelization enables on-line comparison of different algorithms making it and **intelligent** system.
  - Alternatively other algorithms based in machine learning techniques can be executed in parallel.

Conclusions

- DAQ combined with OpenCL reduces development of high-performance processing.
- The DAQ and processing systems developed with OpenCL are less manufacturer dependent.
- 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, abstracted from the control system. **IRIO-OpenCL**
- **You only need to do the algorithm.**

miguel.astrain@i2a2.upm.es

Future Work

- **Future work:**
  - Implementation of Machine Learning Applications using machine learning and Deep Learning Tools*.
  - Expand IRIO-OpenCL functionalities
  - **Looking for more use cases (possible collaborations) using IRIO-OpenCL -> Contact us!**

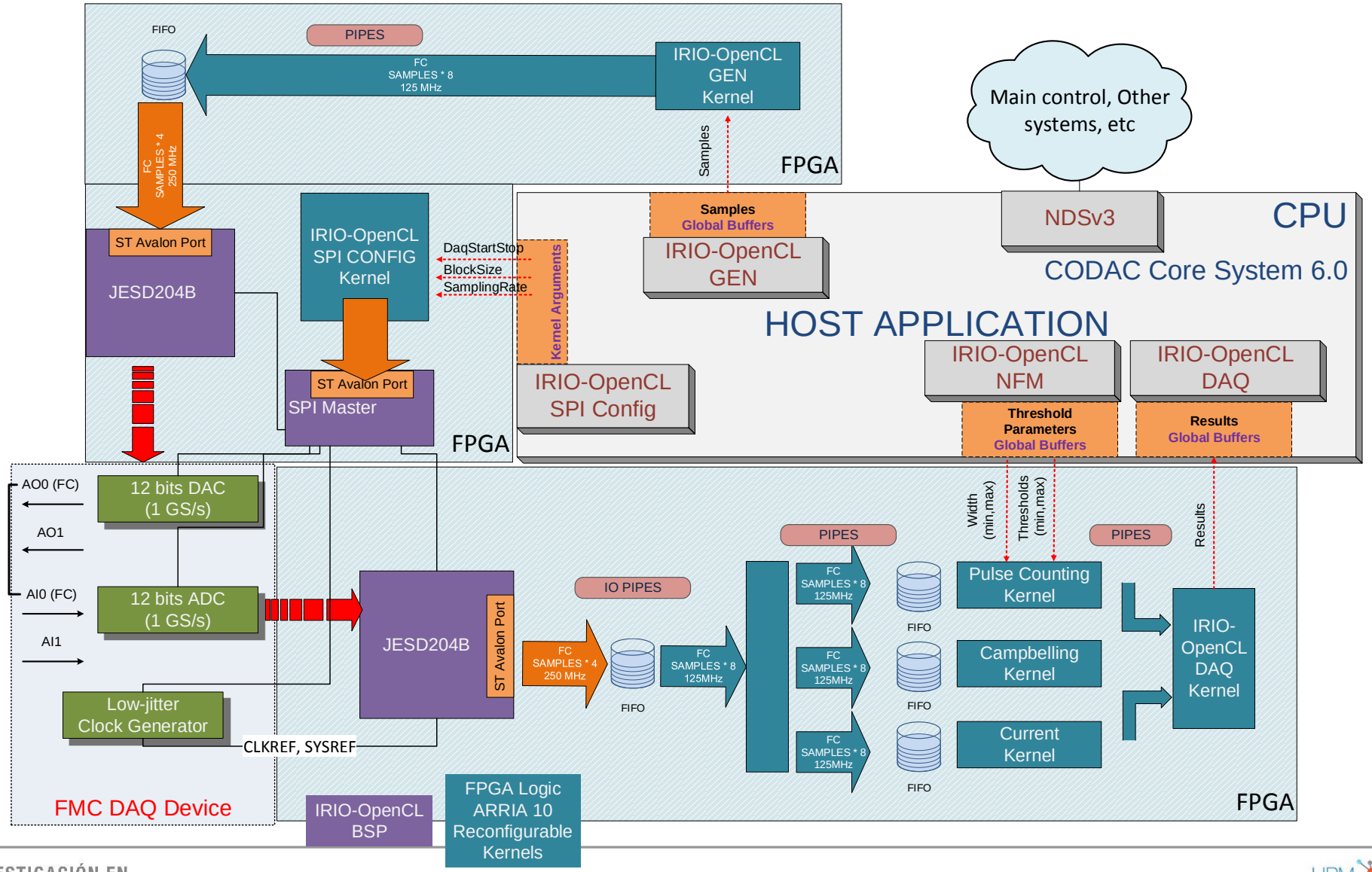
* Dr Jesus Vega (**484. Automatic recognition of plasma relevant events: implications for ITER**) **14 may. 2019 9:20**

Acknowledgements

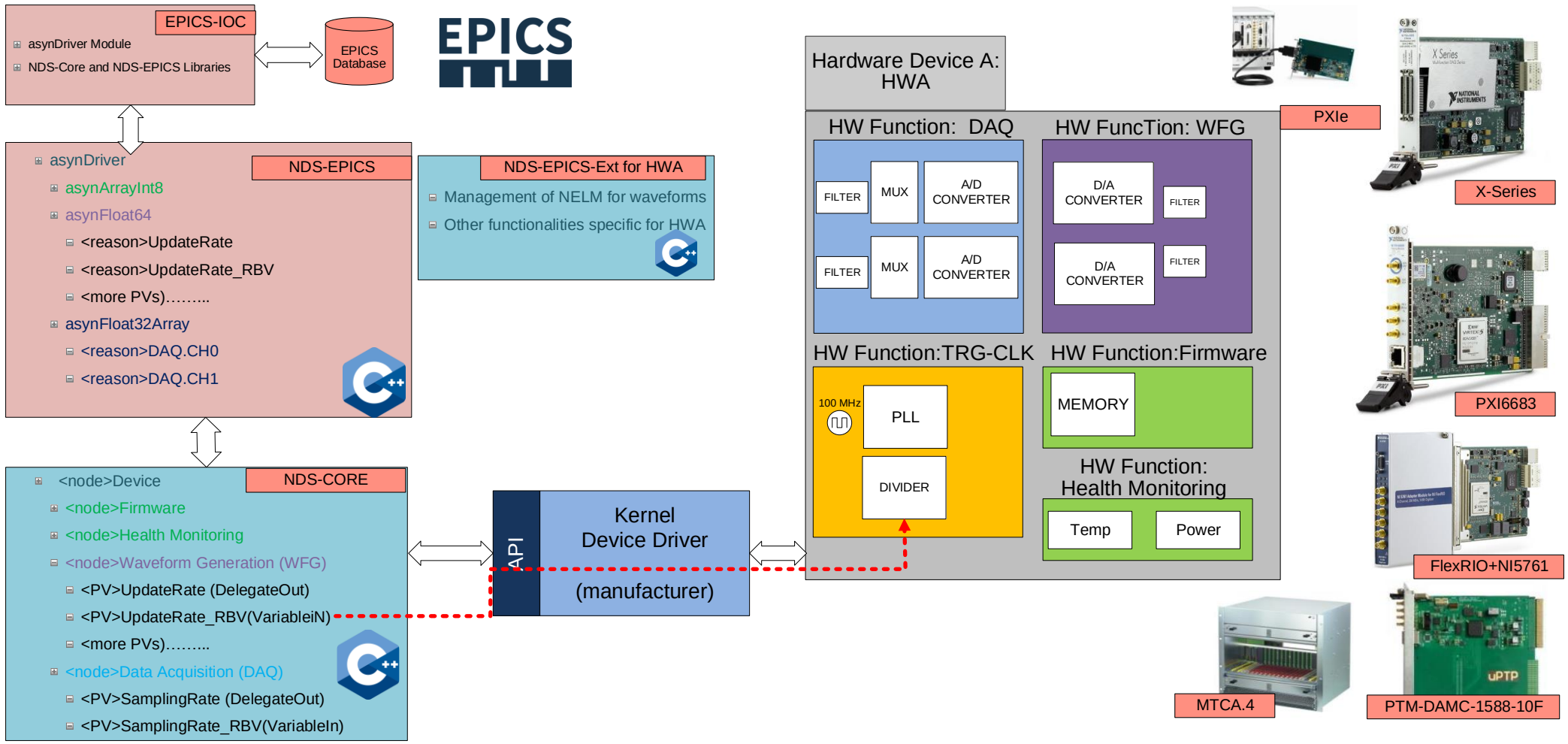
This work was supported in part by the Spanish Ministry of Economy and Competitiveness, **Projects Nº ENE2015-64914-C3-3-R** and Madrid regional government (YEI fund), **Grant Nº PEJD-2018-PRE/TIC-8571**.

Thank You! Questions?

miguel.astrain@i2a2.upm.es


FIFO
PIPES
FC
SAMPLES * 8
125 MHz
IRIO-OpenCL
GEN
Kernel
Samples
FPGA
FC
SAMPLES * 4
250 MHz
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
IRIO-OpenCL
NFM
IRIO-OpenCL
DAQ
Threshold
Parameters
Global Buffers
Results
Global Buffers
ST Avalon Port
SPI Master
IRIO-OpenCL
SPI Config
FPGA
AO0 (FC)
AO1
AI0 (FC)
AI1
12 bits DAC
(1 GS/s)
12 bits ADC
(1 GS/s)
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
FC
SAMPLES * 8
125MHz
FC
SAMPLES * 8
125MHz
FIFO
FIFO
FIFO
Width
(min,max)
Thresholds
(min,max)
PIPES
Results
Pulse Counting
Kernel
Campbelling
Kernel
Current
Kernel
IRIO-
OpenCL
DAQ
Kernel
FPGA
IRIO-OpenCL
BSP
FPGA Logic
ARRIA 10
Reconfigurable
Kernels

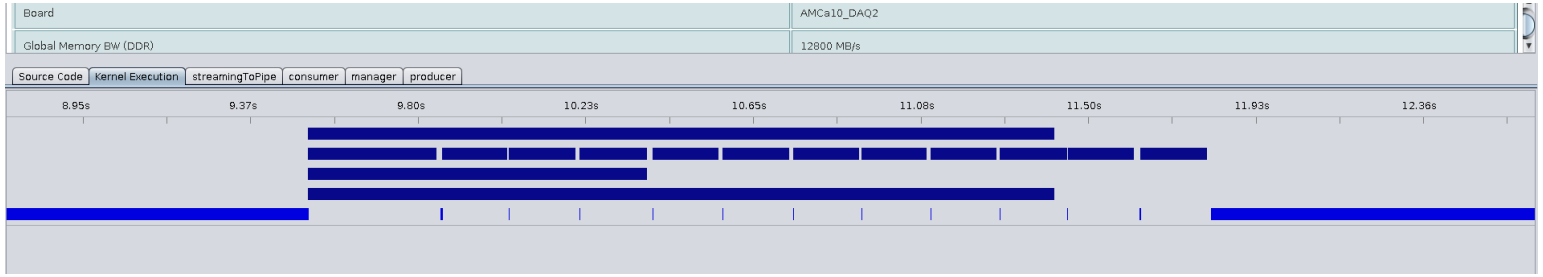
Backup


EPICS-IOC
asynDriver Module
NDS-Core and NDS-EPICS Libraries
EPICS Database
EPICS
asynDriver
NDS-EPICS
asynArrayInt8
asynFloat64
<reason>UpdateRate
<reason>UpdateRate_RBV
<more PVs)……..
asynFloat32Array
<reason>DAQ.CH0
<reason>DAQ.CH1
NDS-EPICS-Ext for HWA
Management of NELM for waveforms
Other functionalities specific for HWA
<node>Device
NDS-CORE
<node>Firmware
<node>Health Monitoring
<node>Waveform Generation (WFG)
<PV>UpdateRate (DelegateOut)
<PV>UpdateRate_RBV(VariableiN)
<more PVs)……..
<node>Data Acquisition (DAQ)
<PV>SamplingRate (DelegateOut)
<PV>SamplingRate_RBV(VariableIn)
API
Kernel Device Driver
(manufacturer)
Hardware Device A: HWA
HW Function: DAQ
FILTER
MUX
A/D CONVERTER
HW FuncTion: WFG
D/A CONVERTER
HW Function:TRG-CLK
100 MHz
PLL
DIVIDER
HW Function:Firmware
MEMORY
HW Function: Health Monitoring
Temp
Power
PXIe
X-Series
PXI6683
FlexRIO+NI5761
MTCA.4
PTM-DAMC-1588-10F

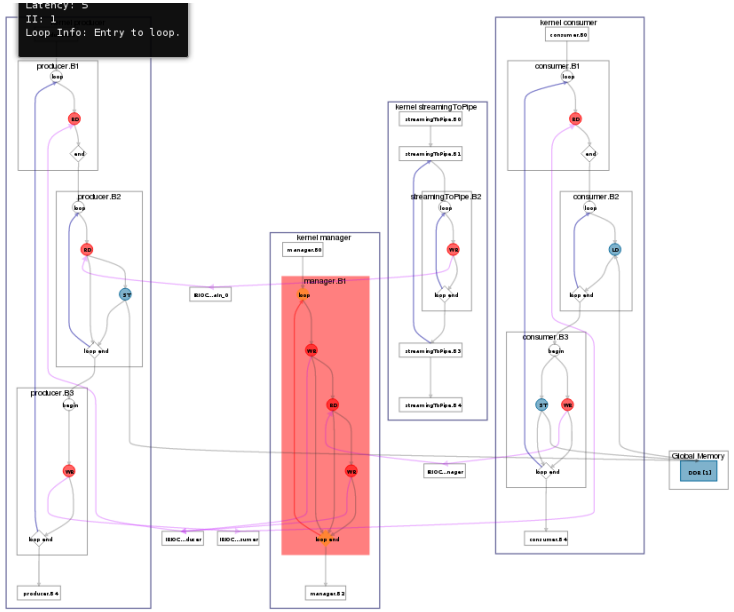
Backup

Results

- OpenCL tools profiler screen



- Graphs



Backup

miguel.astrain@i2a2.upm.es