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

M. Astrain\textsuperscript{1}, M. Ruiz\textsuperscript{1}, S. Esquembri\textsuperscript{1}, A. Carpeño\textsuperscript{1}, E. Barrera\textsuperscript{1}, J. Vega\textsuperscript{2}

\textsuperscript{1}Instrumentation and Applied Acoustic Research Group, Universidad Politécnica de Madrid, Madrid, Spain

\textsuperscript{2}Laboratorio Nacional de Fusión, CIEMAT, Madrid, Spain

miguel.astrain@i2a2.upm.es
• Context
• System Architecture Example
• OpenCL standard
• Development cycle
• Results
• Conclusions
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 ...)

miguel.astrain@i2a2.upm.es
Why is it expensive to develop for FPGAs

Desig Using LabVIEW/FFPA

Only Products from One Manufacturer

miguel.astrain@i2a2.upm.es
IRIO-OpenCL

Control System
- NDS-EPICS
- NDS-CORE

User customization
- IRIO
- IRIO-OpenCL

NI Linux Driver
- OpenCL RTE

Main control, Other systems, etc

Application independent
- Application dependant
- Hardware & device drivers
- Software

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

Products from multiple Manufacturers

miguel.astrain@i2a2.upm.es
• PC + PCIe + MTCA + AMC + (PROCESSING) + (I/O)
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.

### Diagram

- **Host** to **Intel RTE**
- **BSP**
  - **K_DAQ**
  - **K_GEN**
  - **K_TIMING**
  - **K_NFM**

**Contact**

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.

---

**Slide 11**

miguel.astrain@i2a2.upm.es
Three main scenarios for a new application:

1. New algorithm
2. FMC module
3. AMC + FPGA
Results

• Implementation: **more on ID 494 !!**

• Basic kernels FPGA resource utilization

<table>
<thead>
<tr>
<th>Kernel Name</th>
<th>ALUTs</th>
<th>FFS</th>
<th>RAMs</th>
<th>DSPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>consumer</td>
<td>1804</td>
<td>5113</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>manager</td>
<td>470</td>
<td>454</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>producer</td>
<td>1155</td>
<td>2872</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>streamingToPipe</td>
<td>635</td>
<td>400</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Kernel Subtotal</td>
<td>4134</td>
<td>8839</td>
<td>46</td>
<td>2</td>
</tr>
<tr>
<td>Channel Resources</td>
<td>164</td>
<td>648</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Global Interconnect</td>
<td>1244</td>
<td>2870</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>Board Interface</td>
<td>69809</td>
<td>133609</td>
<td>182</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>72402 (17%)</strong></td>
<td><strong>146024 (17%)</strong></td>
<td><strong>250 (13%)</strong></td>
<td><strong>2 (0%)</strong></td>
</tr>
<tr>
<td><strong>Available</strong></td>
<td>436560</td>
<td>873120</td>
<td>1049</td>
<td>1687</td>
</tr>
</tbody>
</table>

**Functionality:**
- ✓ DAQ
- ✓ WFG
- ✓ Processing
- ✓ Routing

**Under evaluation:**
- ○ Timing
- ○ Triggering
- ○ Routing

• Totally integrated in EPICS using NDS

• Tested in CODAC CORE SYSTEM 6.0
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.
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.

The Intel® FPGA SDK for OpenCL™ is based on a published Khronos Specification. Altera, Arria, Intel, the Intel logo, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission of the Khronos Group™. *Other names and brands may be claimed as the property of others.

miguel.astrain@i2a2.upm.es
Thank You! Questions?
asynArrayInt8
asynFloat64
<reason>DAQ.CH1
NDS-Core and NDS-EPICS Libraries
asynDriver ...
CONVERTERMUX D/A
CONVERTER
HW Function:
Health Monitoring
Temp Power
NDS-CORE
FlexRIO+NI5761
Results

• OpenCL tools profiler screen

• Graphs