

OpenCL implementation of an adaptive disruption predictor based on a probabilistic Venn classifier

Disruption prediction in fusion devices (JET)

- Use of an standalone system with a System On Chip (ARM+FPGA)
- Implemented using OpenCL

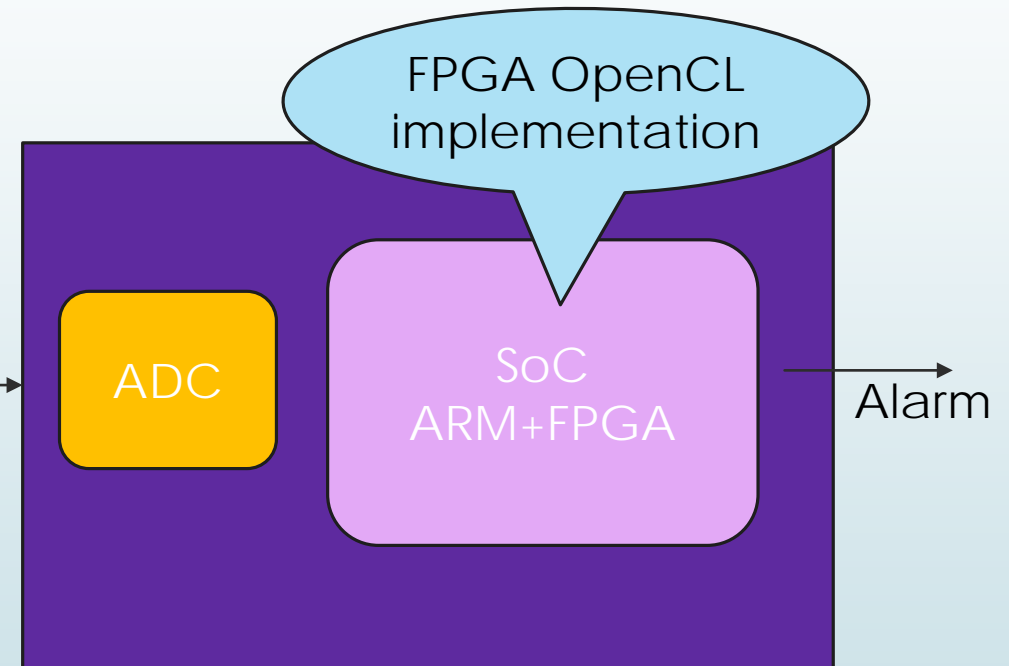
The challenge

- Response time < 1ms
- Implementation using OpenCL for the ARM processor and for the FPGA
- Implementation of OpenCL Board Support Package including an ADC

The solution

- Embedded FPGA-based architecture using a SoC (Cyclone V)
- Deterministic response time. Prediction of the disruption in 400us.
- Implementation of the prediction model in a GPU to shorten the execution time using OpenCL

Ip, LM Li



Poster #492

Abstract







OpenCL implementation of an adaptive disruption predictor based on a probabilistic Venn classifier

C. González¹, A. Carpeño¹, M. Ruiz², J. Vega³, E. Domínguez-Canto⁴, **E. Bernal**¹, J. Lee¹, and JET-EFDA Contributors¹
 JET-EFDA, Culham Science Centre, Abingdon, OX14 3DB, UK
¹Universidad de Almería, Almería Research Group, Technological University of Almería, Almería, Spain, Email: EBERNAL@UNIVERSIDAD-ALMERIA.ES
²Universidad Nacional de Educación a Distancia, Madrid, Spain
³Centro Informático y Automático - INRSO, Almería, Spain
⁴Real Instituto Elctrico, Madrid, Spain

ABSTRACT

The ability and flexibility of the Open Computing Language (OpenCL) for task parallelism in heterogeneous computing platforms (FPGA, CPU, GPU) represent a remarkable advantage when designing advanced data acquisition and processing systems. This work shows a specific implementation of an adaptive probabilistic disruption predictor for a fusion device, based on signals obtained from JET database. This implementation uses OpenCL as base technology for the design cycle. The system was realized using an FPGA-based architecture that comprises a Cyclone V and a GPU-based architecture that contains an AMD FirePro V390 inserted into a computer running Scientific Linux as Operating System. This contribution presents the methodology, the hardware-software system architecture, and the implementation results in both hardware platforms. The work is focused on the critical aspects involved in the design of these intelligent data acquisition and processing systems with OpenCL. When dealing with this technology, it is essential to be aware of aspects such as the significant differences in the design flow concept between FPGA and GPU implementations, or how to select the part of the algorithm that is better to be executed in each platform, which is not an easy task. The best results show that it is possible to achieve predictor times shorter than 500 ns.

ADAPTIVE DISRUPTION PREDICTOR

DESCRIPTION

1. Implementation of an adaptive predictor to predict the next event in Venn predictor using an iterative probabilistic Venn classifier for each event.
2. Performance evaluation of the implementation in a specific FPGA and GPU-based architectures.

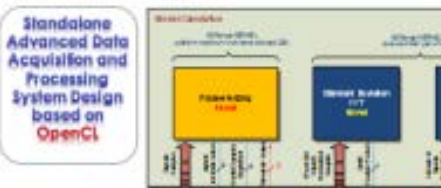
KEYWORDS

- OpenCL
- Probabilistic Venn classifier
- Adaptive disruption predictor
- Real-time processing

FPGA CONTROLLER DESIGN

Design of the hardware for controlling the FPGA device, responsible for forwarding the signals coming to the FPGA-based processing hardware.

[VHDL/Verilog/Quartus II design cycle]



OPENCL APPLICATION DEVELOPMENT

Development of the OpenCL kernel code and the C++ host code.

- a) Analysis of the processing algorithm and the strategy for the optimal task parallelization.
- b) Development of kernel code in OpenCL language. Compilation is a slow process and it is worth to debug in simulator before generating the final program for the FPGA.
- c) Development of the host application code in C++ language.

[OpenCL and C++ design cycle, Intel OpenCL offline compiler and gcc compiler]



OPENCL BOARD SUPPORT PACKAGE GENERATION

Porting your specific FPGA device reference design to an OpenCL compliant MP to include your new acquisition hardware.

- a) Modify the reference design that includes OpenCL-specific components, host-to-FPGA communication IP and memory IP, to incorporate the existing changes to your specific hardware and the new acquisition hardware.
- b) Perform the timing closure and location of the logic block regions of the design partition for the specific kernel logic.
- c) Create the JTAG file that informs the OpenCL compiler about your custom hardware.

[Quartus II and Platform Designer design cycle]

RESULTS

Platform	Implementation	Hardware	Software	Configuration
C14E-1	OpenCL	1.5G	1.5G	1.5G
GPU	OpenCL	1.5G	1.5G	1.5G

CONCLUSIONS

- Probabilistic Venn classifier implemented in OpenCL on a Cyclone V FPGA, with a logic architecture detector (for real-time detection) of the major components (10 ns per prediction) showed no significant overhead.
- Real-time implementation of the adaptive predictor on a Cyclone V FPGA (1.5G) and a GPU (1.5G) showed no significant overhead.
- The implementation of the adaptive predictor on a Cyclone V FPGA (1.5G) and a GPU (1.5G) showed no significant overhead.

ACKNOWLEDGMENTS

This work was partially funded by the Spanish Ministry of Economic and Competitiveness under the DPI2014-58054-PP (DPI2014-58054-PP) and DPI2014-58054-PP (DPI2014-58054-PP) grants. The authors would like to thank the European Union for its financial support under the Horizon Europe program.

Initial MATLAB algorithm with the Venn Probabilistic Classifier

System architecture solution

- Using a SoC (Cyclone V)
- A Linux host

Conclusions

Implementation of the Board Support Package for OpenCL

Results: Execution Time

Enrique Bernal
 Technical University of Madrid