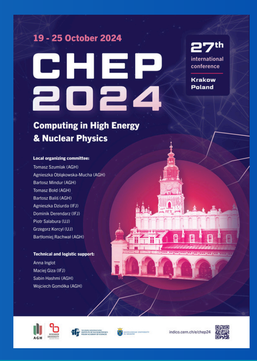
Porting MADGRAPH to FPGA using High-Level Synthesis (HLS)

Héctor Gutiérrez, Luca Fiorini, Alberto Valero, Francisco Hervas
IFIC (Universitat de València - CSIC)
HIGH-LOW TED2021-130852B-I00
27th CHEP - Krakow
Poster Session - 21th October 2024



ABSTRACT

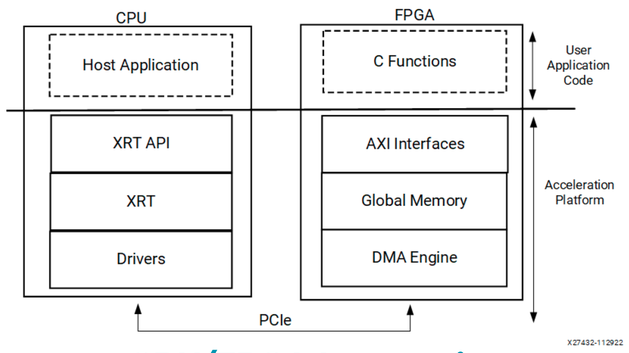
The escalating demand for data processing in **particle physics research** has spurred the exploration of novel technologies to enhance efficiency and speed of calculations. This study presents the development of a **porting of MADGRAPH,** a widely used tool in particle collision simulations, to **FPGA** using **High-Level Synthesis (HLS).**

Experimental evaluation is ongoing, but preliminary assessments suggest a promising enhancement in **calculation speed** compared to traditional **CPU** implementations. This potential improvement could enable the execution of more **complex simulations** within shorter time frames.

This study describes the complex process of adapting MADGRAPH to FPGA using HLS, focusing on optimizing algorithms for parallel processing. A key aspect of the FPGA implementation of the MADGRAPH software is reduction of the **power consumption**, which important implications for the scalability of computer centers and for the environment. These advancements could enable **faster execution** of complex simulations, highlighting FPGA's crucial role in advancing particle physics research and its environmental impact.

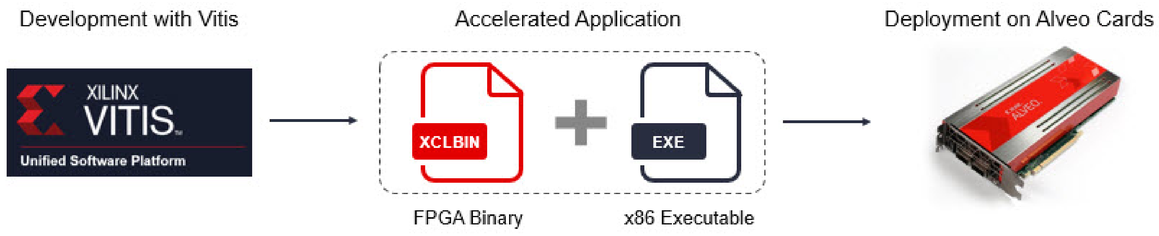
FPGA IMPLEMENTATION

Developing applications for **Alveo** [2] involves two parts: programming the host, which runs on x86 processors, and programming the **FPGA**, which accelerates specific functions. **Host** development is similar to regular software development, using C/C++ and the OpenCL API to **manage tasks** on the FPGA, **transfer data**, and **program the FPGA** in real-time, optimizing its resources.



CPU/FPGA Interaction

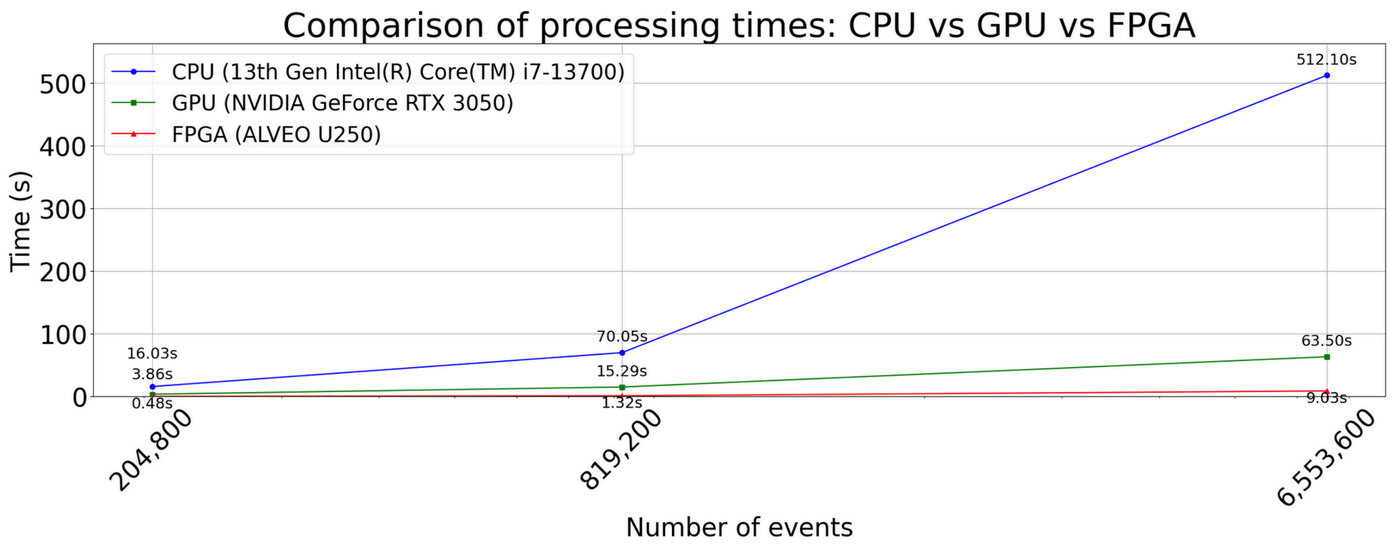
FPGA application development is more complex, typically using **low-level languages** like **Verilog** or **VHDL**. However, the **Vitis** environment allows using **C/C++/OpenCL** to design functions, called kernels, which are **automatically transformed into RTL** using High-Level Synthesis (HLS). Once the RTL is generated, Vitis handles the synthesis, mapping, and creation of the bitstream, packaged in an xclbin file, to program the FPGA.



Vitis Development Flow

RESULTS

Process : e+ e− > μ+ μ−



CPU vs GPU vs FPGA

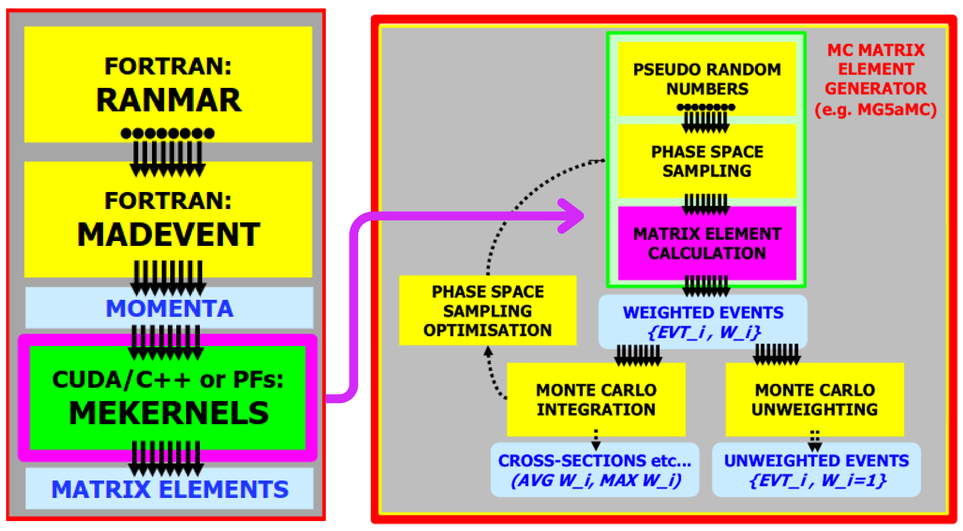
This graph compares the processing times for running the MADGRAPH process e+e−→μ+μ− on three different platforms: CPU, GPU, and FPGA. These results highlight the efficiency of FPGA for accelerating this process, outperforming both CPU and GPU, particularly for large-scale computations.

REFERENCES

[1] MadGraph. (n.d.). MadGraph: A program for event generation in high-energy physics. Retrieved September 19, 2024, from http://madgraph.phys.ucl.ac.be/
[2] Xilinx. (n.d.). Vitis development flow. Xilinx. https://xilinx.github.io/graphanalytics/vitis-dev-flow.html

MADGRAPH

MadGraph5_aMC@NLO [1] is a **framework** that aims at providing all the elements necessary for **SM and BSM** phenomenology, such as the computations of cross sections, the generation of hard events and their matching with **event generators**. Processes can be simulated to **LO** accuracy for any user-defined Lagrangian, and the **NLO** accuracy in the case of **QCD** (Quantum Chromo Dynamics) corrections to SM processes. **Matrix elements** at the tree- and one-loop-level can also be obtained.

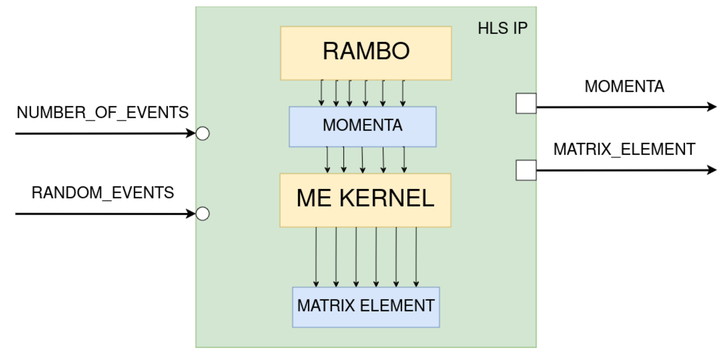


MADGRAPH's functioning and GPU port

Credits to: Madgraph5_aMC@NLO for GPUs group

HLS AND FPGA RESOURCES

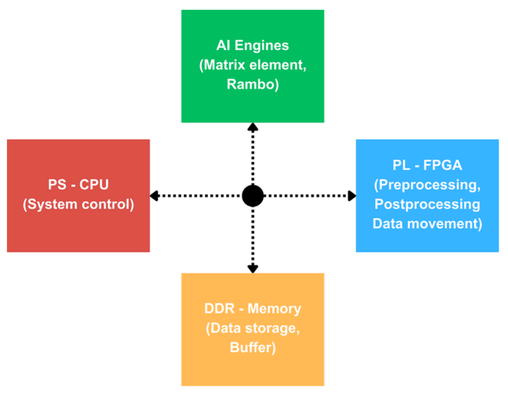
The CPU sends the number of events and the random events data to the FPGA, where this calculates the Momenta and the Matrix Element. Both results are sent back to the CPU for further evaluation.



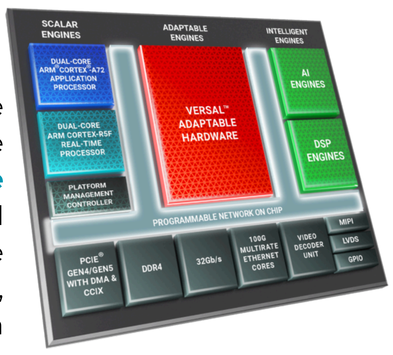
HLS IP (ALVEO U250)

Resource	Utilization	Avalible	Utilization(%)
LUT	404715	1759631	23
FF	549021	3660140	15
DSP	5218	12424	42
BRAM	22	3280	0.6

FUTURE IMPLEMENTATIONS



The idea is to have a **hybrid system,** where, in addition to using the CPU and FPGA for the most computationally intensive tasks, **AI Engines** are introduced to further accelerate calculations, allowing the FPGA to be used for other tasks.



The AI Engines in Xilinx's **Versal ACAP** are specialized units designed to accelerate artificial intelligence tasks and **intensive data processing**. They offer high parallel computing power and flexibility to handle applications such as computer vision, neural networks, and real-time data analytics.