Porting MADGRAPH to FPGA

Héctor Gutiérrez[1], Luca Fiorini, Alberto Valero, Arantza Oyanguren, Francisco Hervas, Carlos Vico, Javier Fernandez, Santiago Folgueras, Pelayo Leguina

[1]Instituto de Física Corpuscular (CSIC-UV)

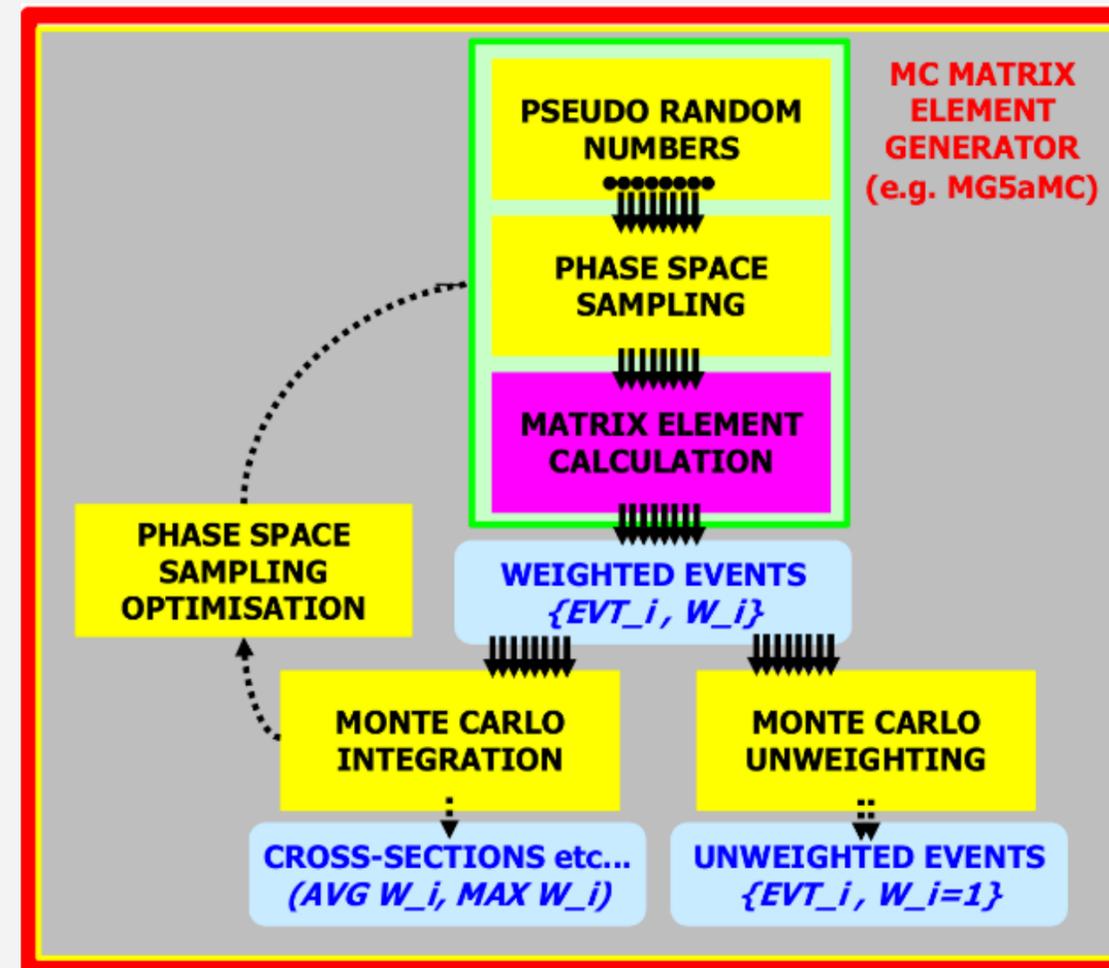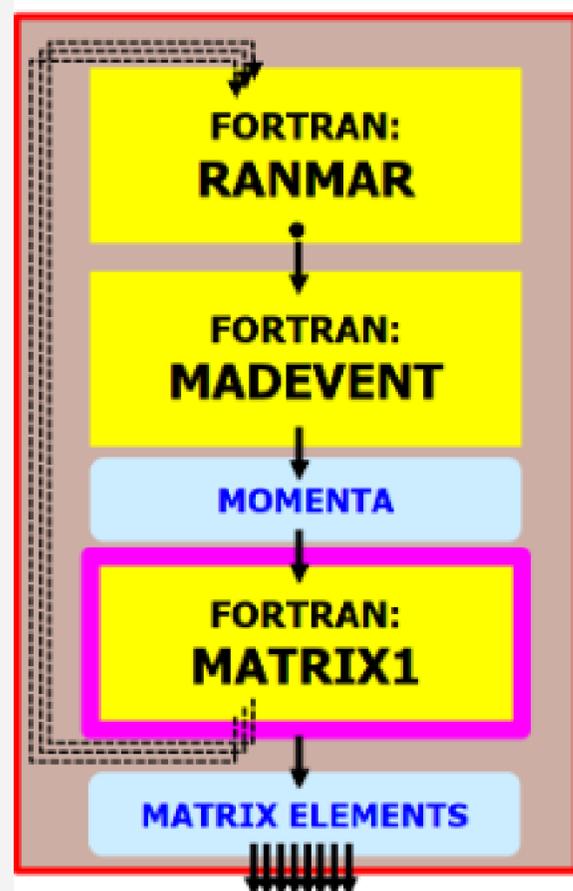Jornadas Tecnológicas IFIC

16 de Septiembre
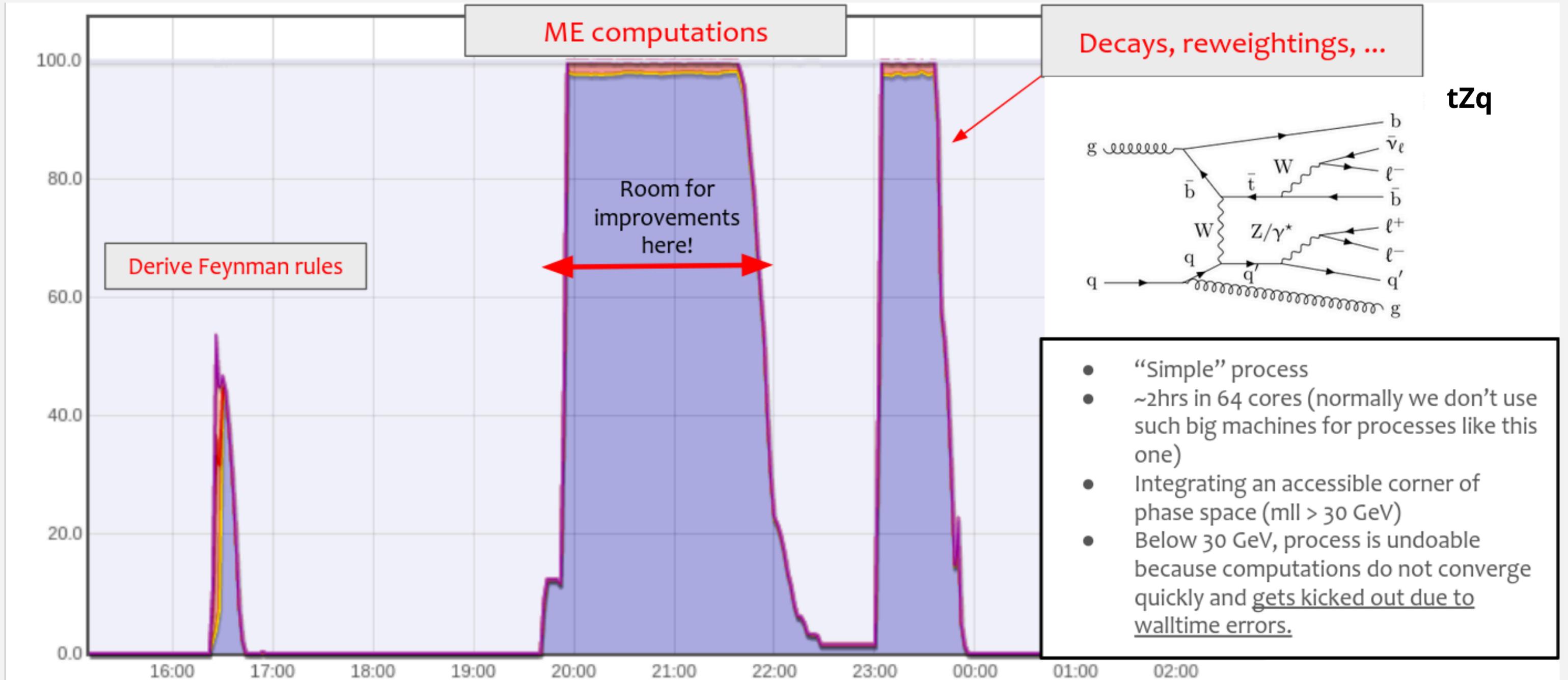
Contact:

Hector.Gutierrez@ific.uv.es

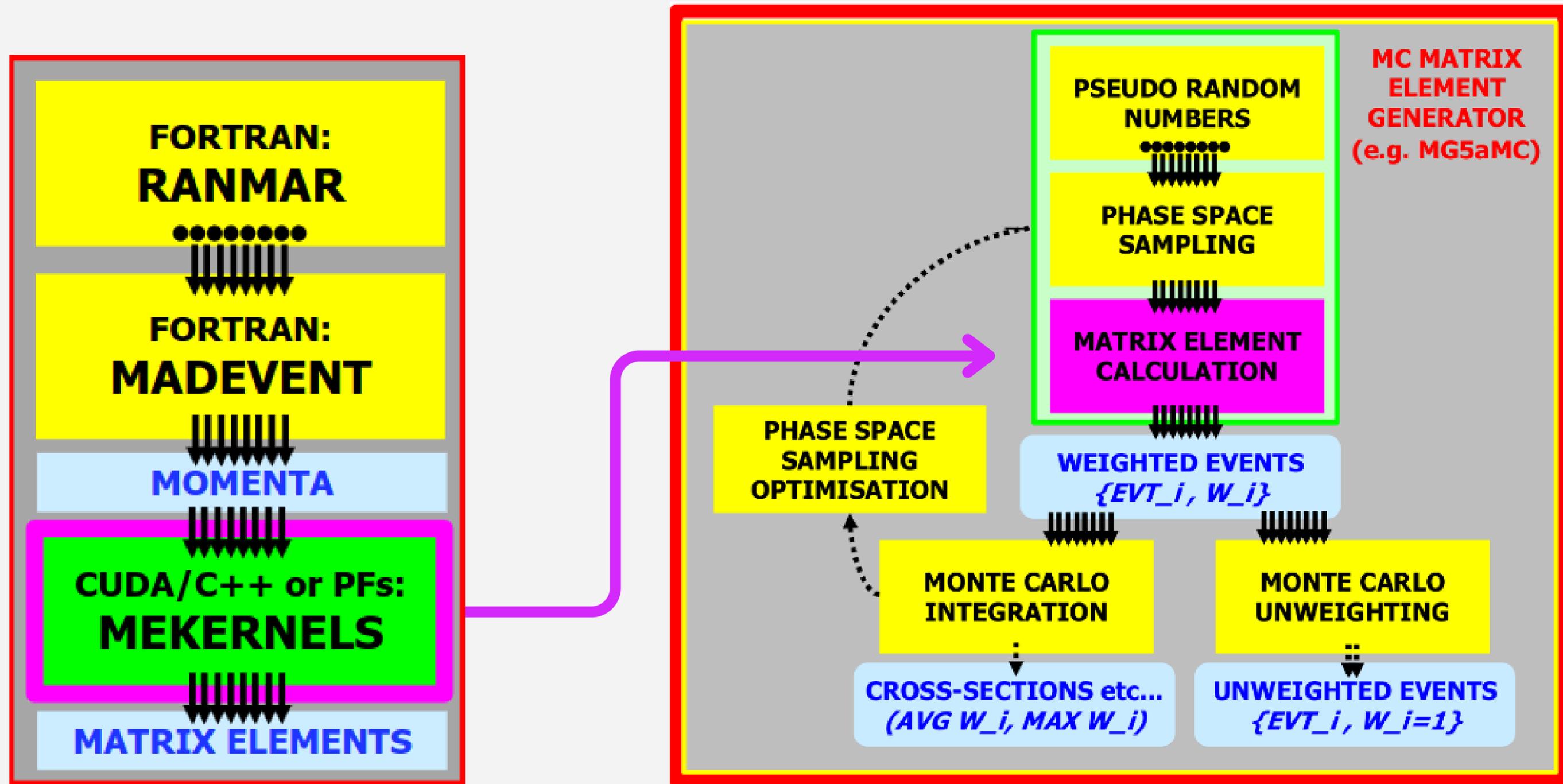# MADGRAPH_aMC@NLO CPU

What is MADGRAPH?:

"_MadGraph5_aMC@NLO_ 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. "
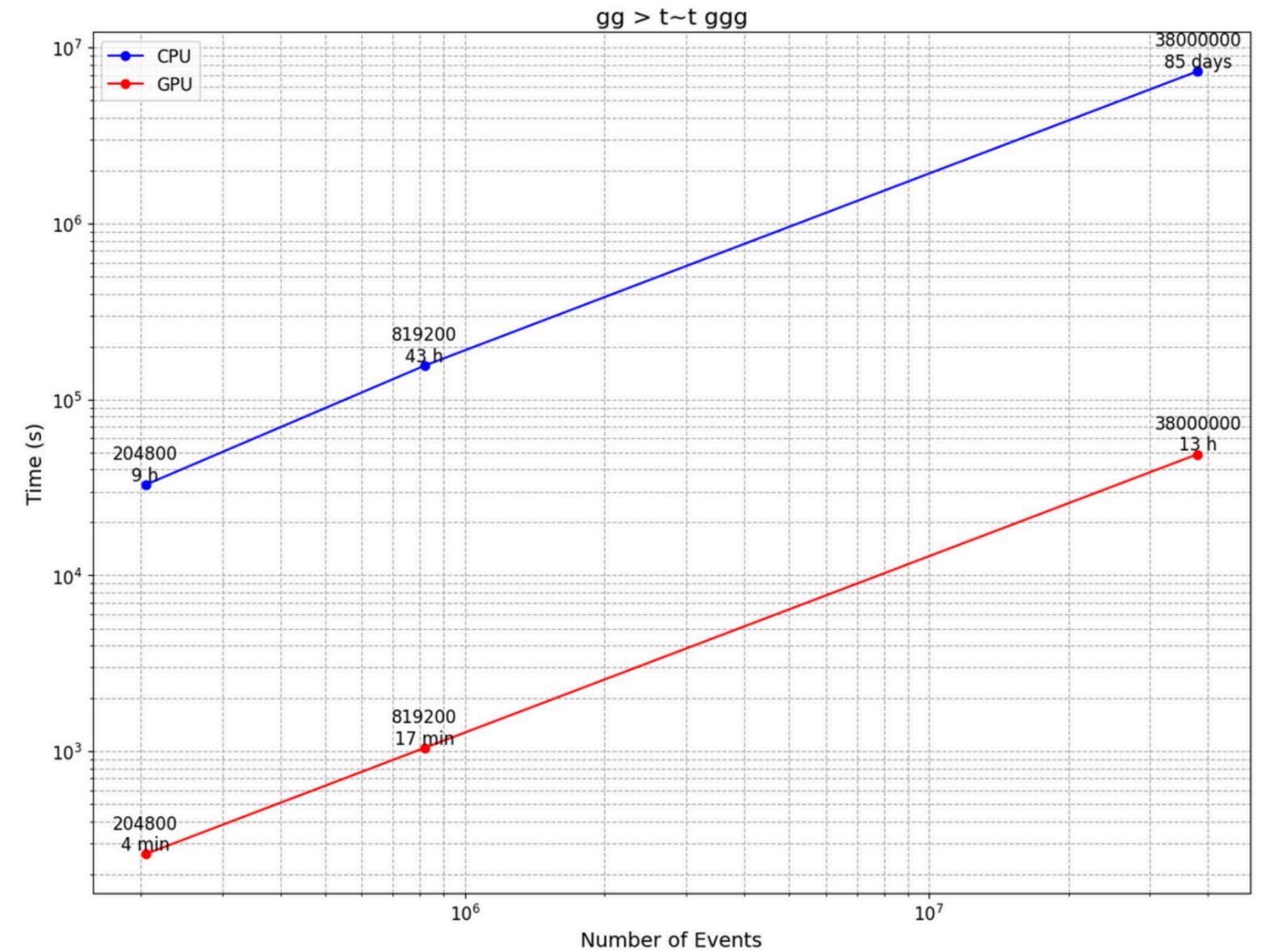
MADGRAPH_aMC@NLO CPU

ME computations

Decays, reweightings, ...

tZq

Room for improvements here!

Derive Feynman rules

- "Simple" process
- ~2hrs in 64 cores (normally we don't use such big machines for processes like this one)
- Integrating an accessible corner of phase space (mll > 30 GeV)
- Below 30 GeV, process is undoable because computations do not converge quickly and gets kicked out due to walltime errors.

MADGRAPH4GPU

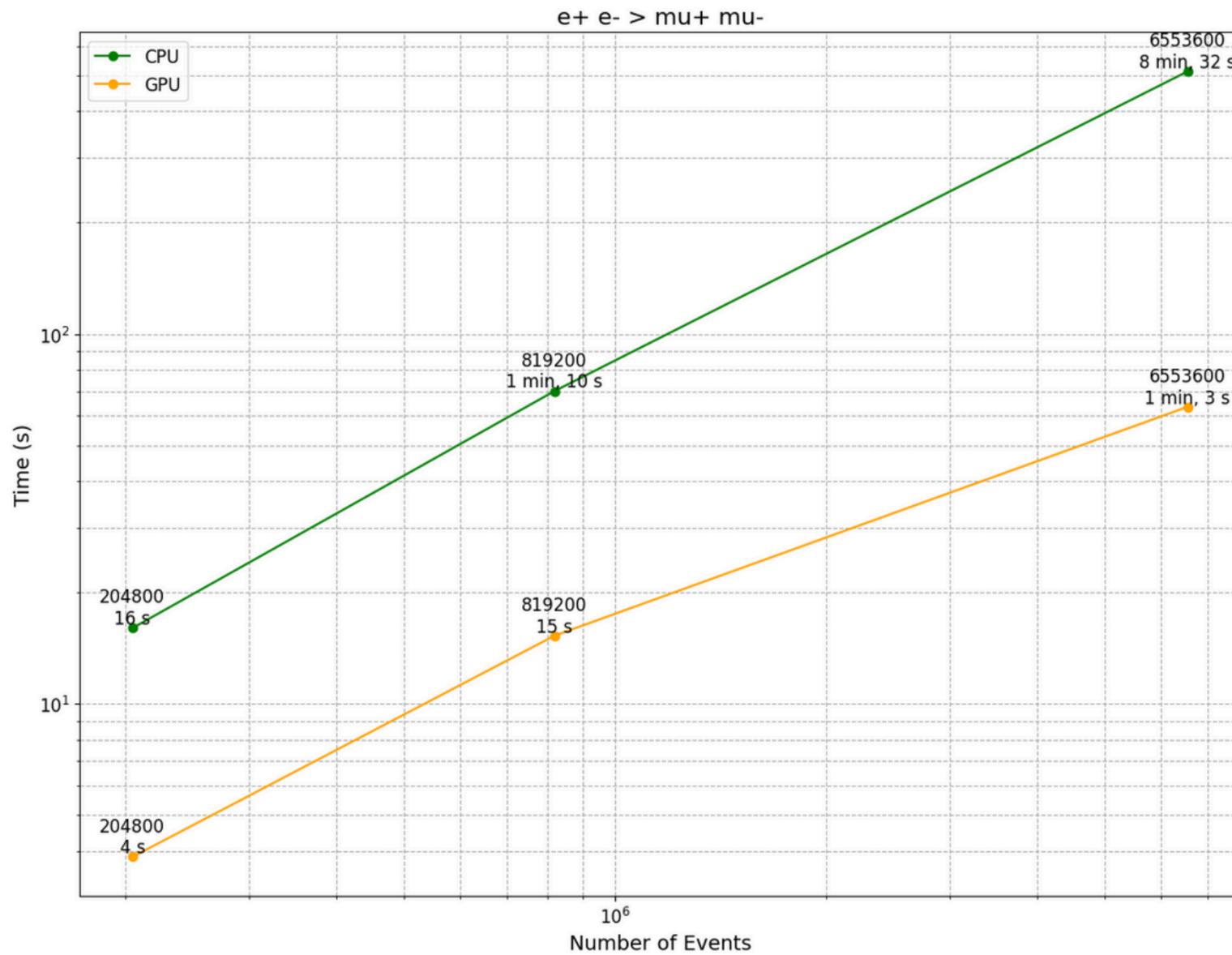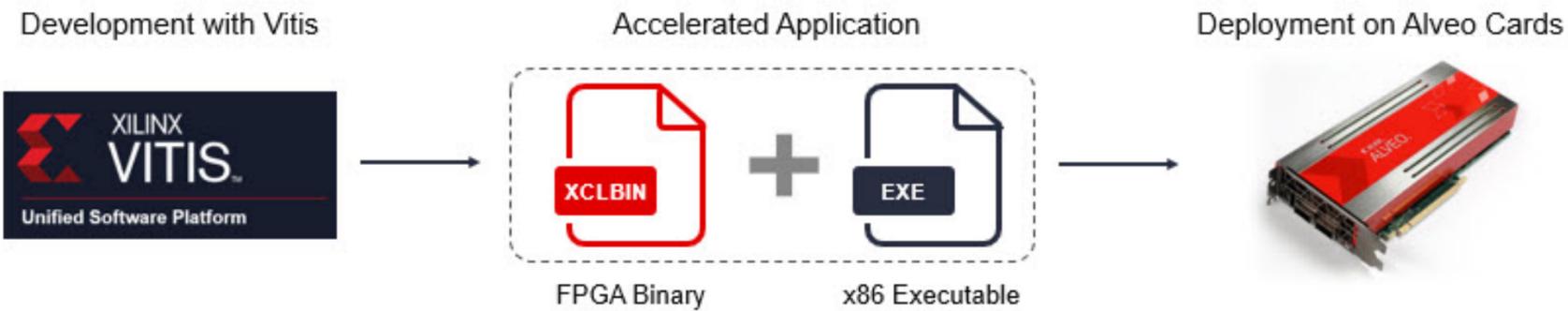**Credits to: Madgraph5_aMC@NLO for GPUs group**

Results CPU vs GPU

CPU : 13th Gen Intel(R) Core(TM) i7-13700

GPU : NVIDIA GeForce RTX 3080 8Gb

HOW TO PROGRAM IN HLS

- FPGA application development is more complex, often using low-level languages like Verilog or VHDL.
- The Vitis environment allows using C/C++/OpenCL C to design functions (kernels).
- Kernels are automatically converted into RTL using High-Level Synthesis (HLS).
- Once RTL is generated, Vitis manages:
  - Synthesis
  - Mapping
  - Creation of the bitstream (packaged in an xclbin file) to program the FPGA.

- Developing applications for Alveo involves two parts:
  - Programming the host (runs on x86 processors).
  - Programming the FPGA (accelerates specific functions).
- Host development is similar to regular software development.
- C/C++ and the OpenCL API are used to:
  - Manage tasks on the FPGA
  - Transfer Data
  - Program the FPGA in real-time, optimizing its resources

$$e^+ e^- \rightarrow \mu^+ \mu^-$$
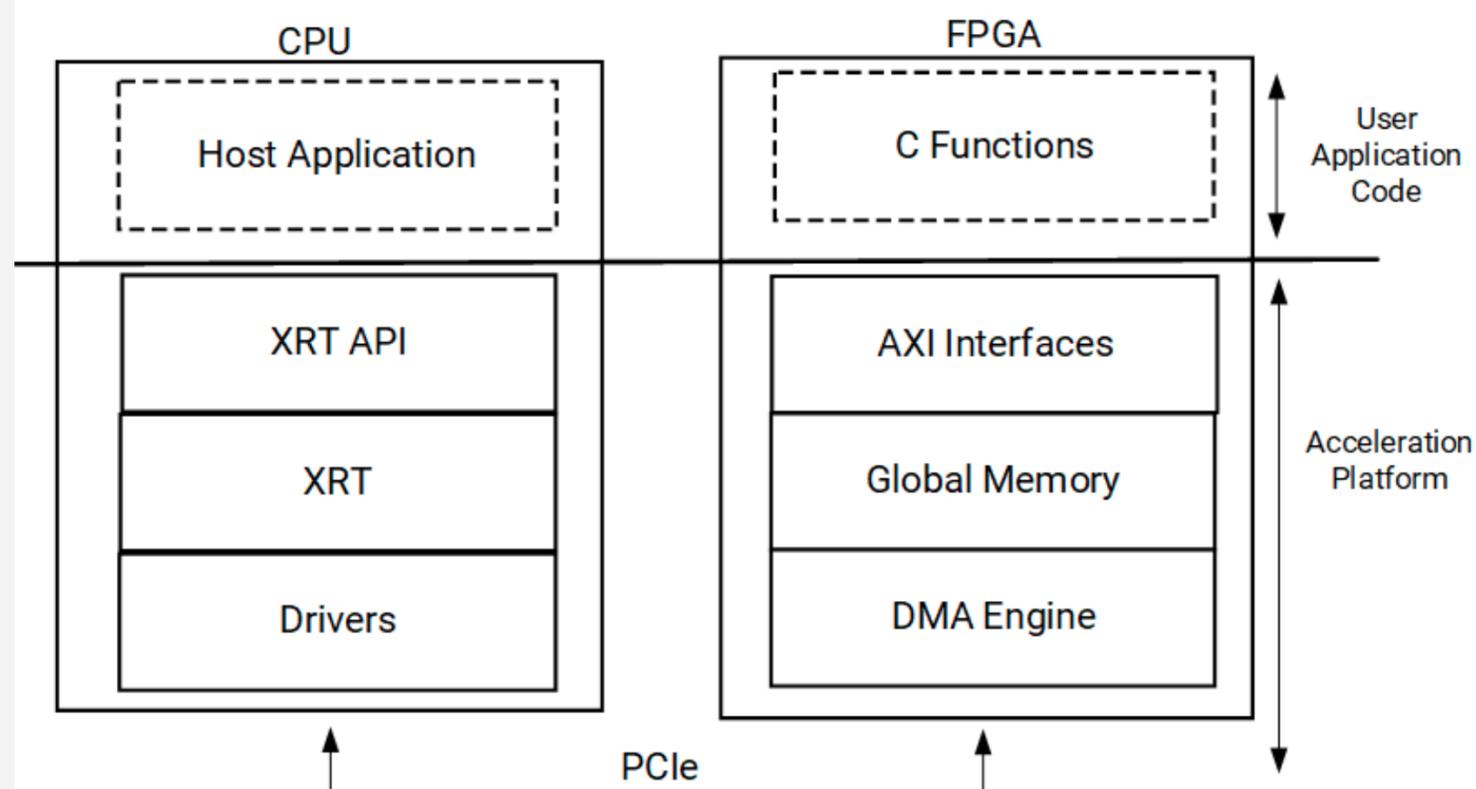
- **top -> check_sa.cpp (IN:NUMBER_OF_EVENTS, IN:RANDOM_EVENTS, OUT: MOMENTA, OUT:MATRIX_ELEMENT)**
  - **Rambo -> Momenta**
  - **SigmaKin -> Matrix Element**
- **Rambo(Energy, masses, weight, masses_size, random inputs)**
  - **Obtain Momenta**
- **SigmaKin**
  - **InitProc() -> SetIndependentCouplings**
  - **SetParameters & SetDependentcouplings**
  - **Calculate waveforms**
  - **Calculate matrix of the process**
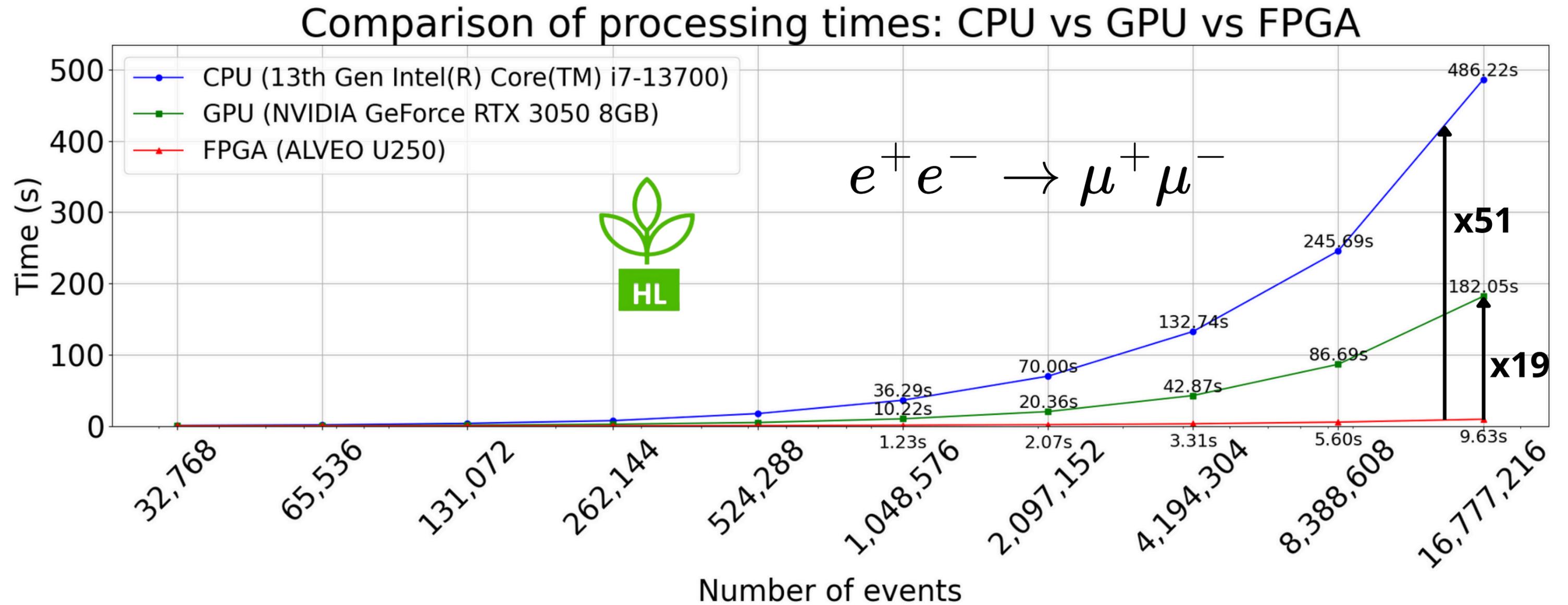   - **Obtain Matrix Element**

7
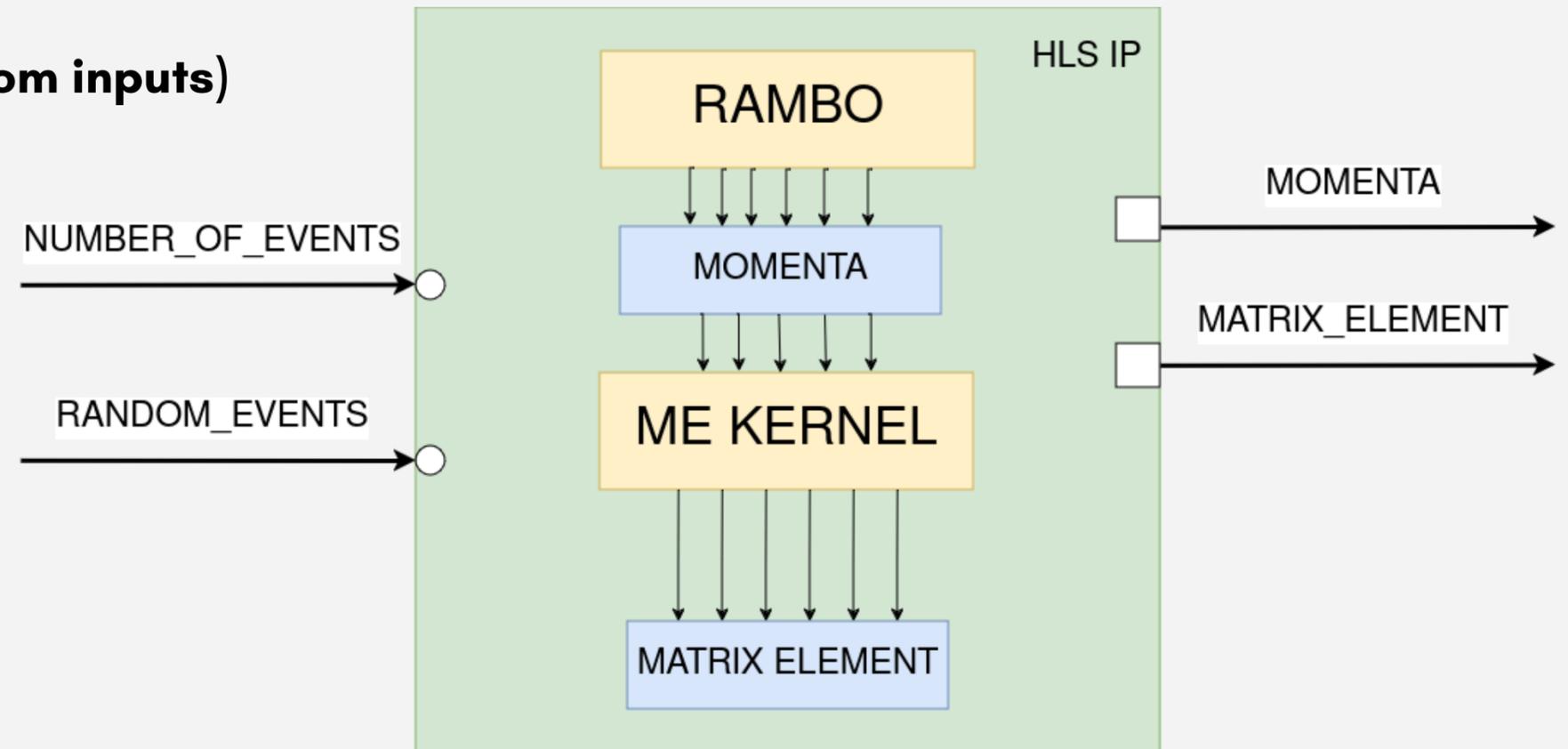
FPGA Resources

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

Frecuency : 121.95 MHz

Comparison of processing times: CPU vs GPU vs FPGA

Resource Optimization

$$e^+ e^- \rightarrow \mu^+ \mu^-$$

Resource	Utilization	Avalible	Utilization(%)
LUT	404715	1759631	23
FF	549021	3660140	15
DSP	5218	12424	42
BRAM	22	5442	0.6

Reduce precision and Rewrite the HLS code:
double -> float or fixed precision
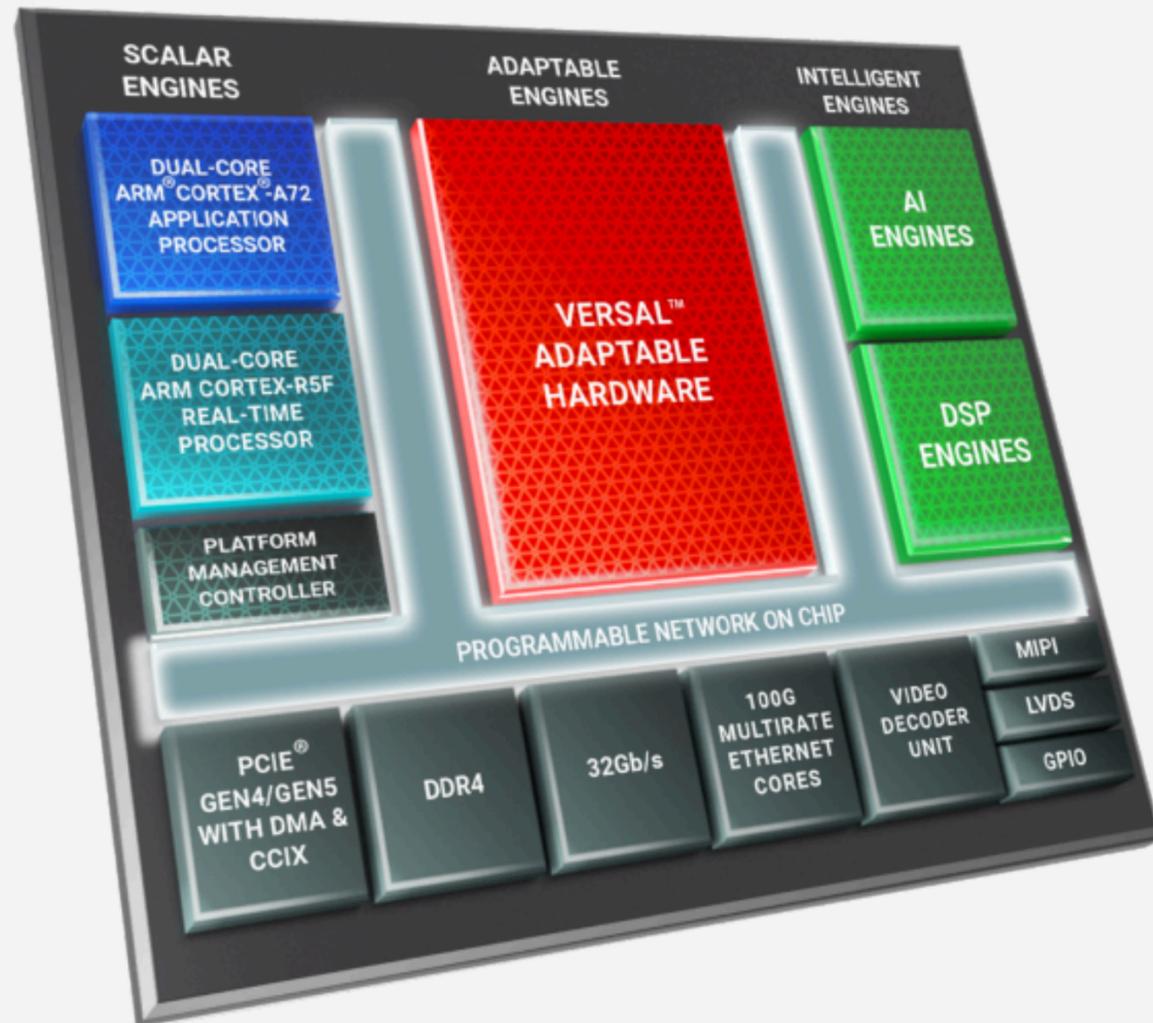
Write a specific part of the bottleneck

?

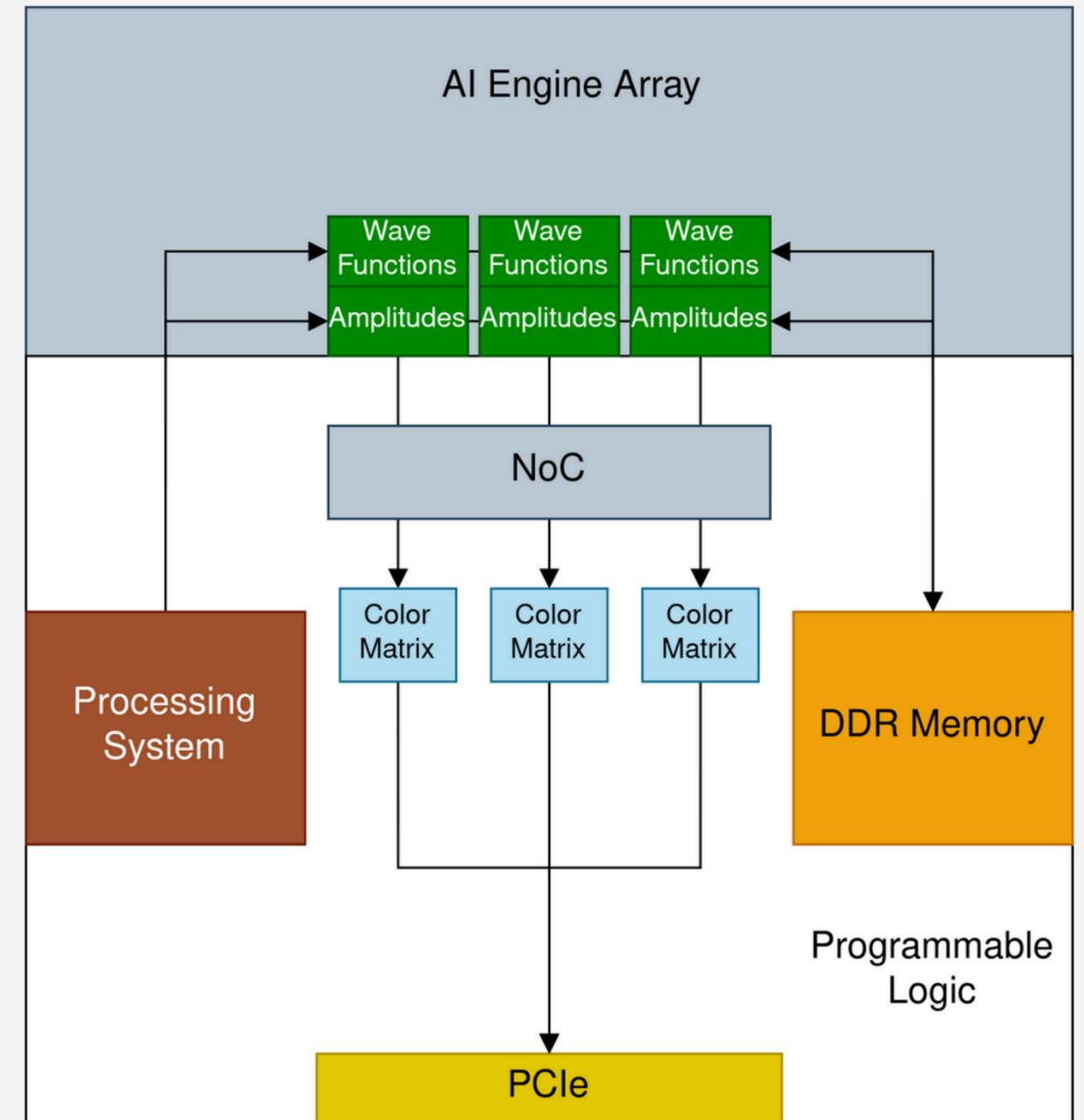|	$gg \to t\bar{t}$	$gg \to t\bar{t}gg$	$gg \to t\bar{t}ggg$
madevent	13G	470G	11T
matrix1	3.1G (23%)	450G (96%)	11T (>99%)

color
amplitude
int/propagator
external
not ME
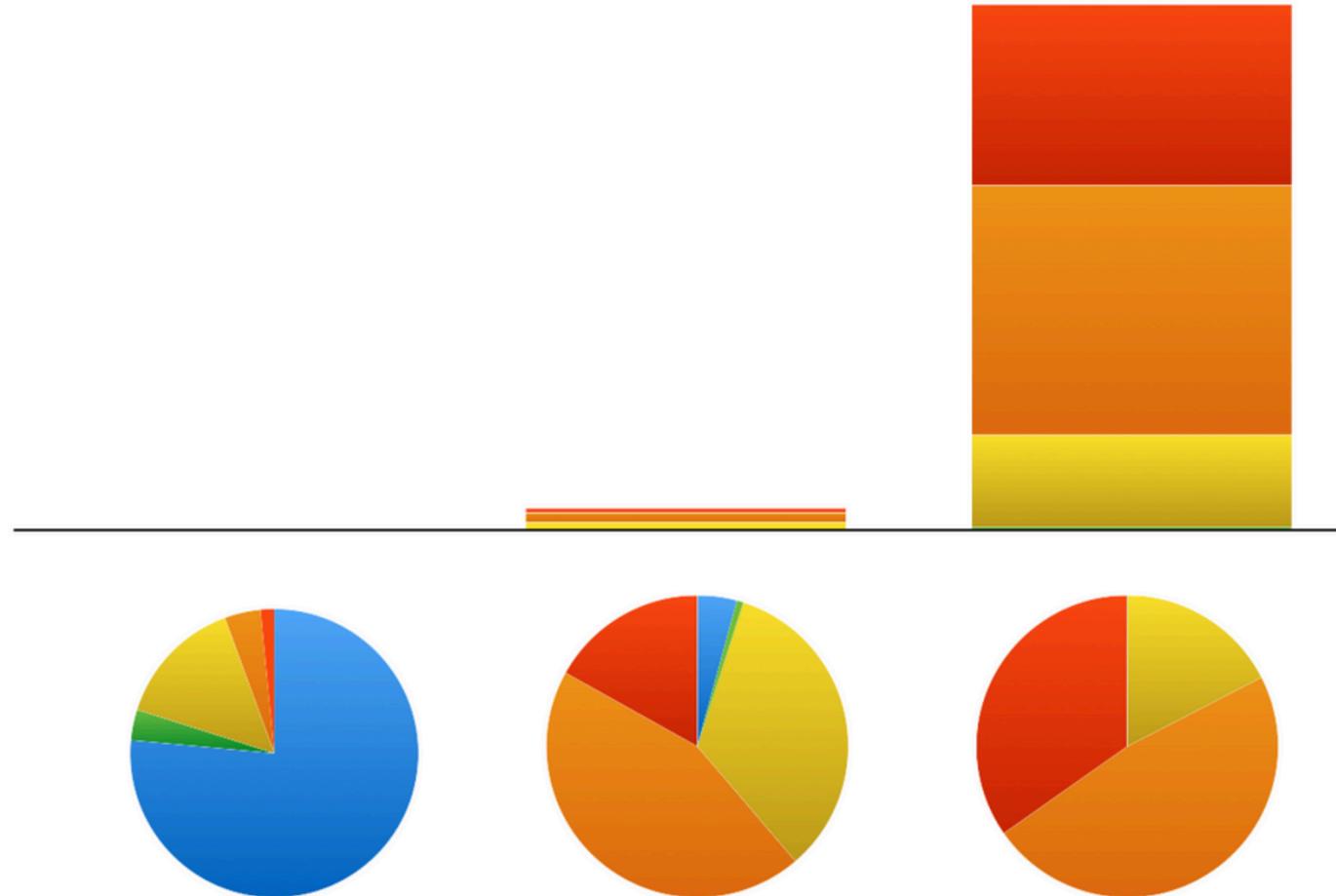
COLOR IMPLEMENTATION

VERSAL VCK190

FUTURE IMPLEMENTATIONS

- Create the code for a complex process

- Create the new version for all LO processes

- Study the implementation of BSM processes and NLO

- Create a version that combines all three implementations (CPU + GPU + FPGA)

Porting MADGRAPH to FPGA

Héctor Gutiérrez[1], Luca Fiorini, Alberto Valero, Arantza Oyanguren, Francisco Hervas, Carlos Vico, Javier Fernandez, Santiago Folgueras, Pelayo Leguina

[1]Instituto de Física Corpuscular (CSIC-UV)

Jornadas Tecnológicas IFIC

16 de Septiembre

Contact:

Hector.Gutierrez@ific.uv.es