Traineeships in Advanced Computing for High Energy Physics (TAC-HEP)

FPGA module training

Week-1: Introduction to FPGA and its architecture

Lecture-1: January 28th 2025

Varun Sharma & Sridhara Dasu

University of Wisconsin – Madison, USA

Welcome!

- Welcome to the first lecture of *FPGA training module*

- We will meet twice a week for an hour:
  - 1 hr = 45min lecture + 15min Q&A.
  - Tuesdays & Thursdays: 11:00-12:00 CT / 12:00-13:00 ET / 18:00-19:00 CET

- Feel free to interrupt during the lecture in case of any clarifications are needed

- Outside lecture hours:
  - Post your queries on slack channel (tac-hep-fpga2025), or;
  - Via email: varun.sharma@cern.ch

Goals

Training module for getting better at:

- Understanding of FPGAs, their architecture and usage in HEP context
- Overview of usage of programmable logic in gate arrays
- Improved interactions with electrical engineers regarding programmable logic
- Write your own firmware for a physics algorithm
- How to read/understand/debug some of the operational issues

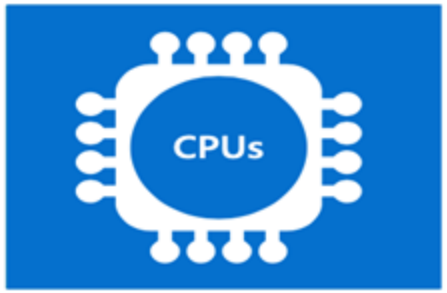
You may NOT:

- Become an electrical engineer, electronics is more than FPGAs...
- Being an FPGA "expert" it needs much more time..
- Improve your soldering skills, its just software/firmware ☺

Content

- Overview of FPGAs and comparison with other options
- HEP Motivation for using FPGAs
- FPGA architecture
- Parallelism in FPGA

CPUs



Sequential processing of instructions

Program counters keep incrementing and for each PC

Instruction and operands are loaded
Instruction is Executed
Result is stored in the output registers

Extremely flexible but driven by sequential execution

Stable Instruction set over decades!

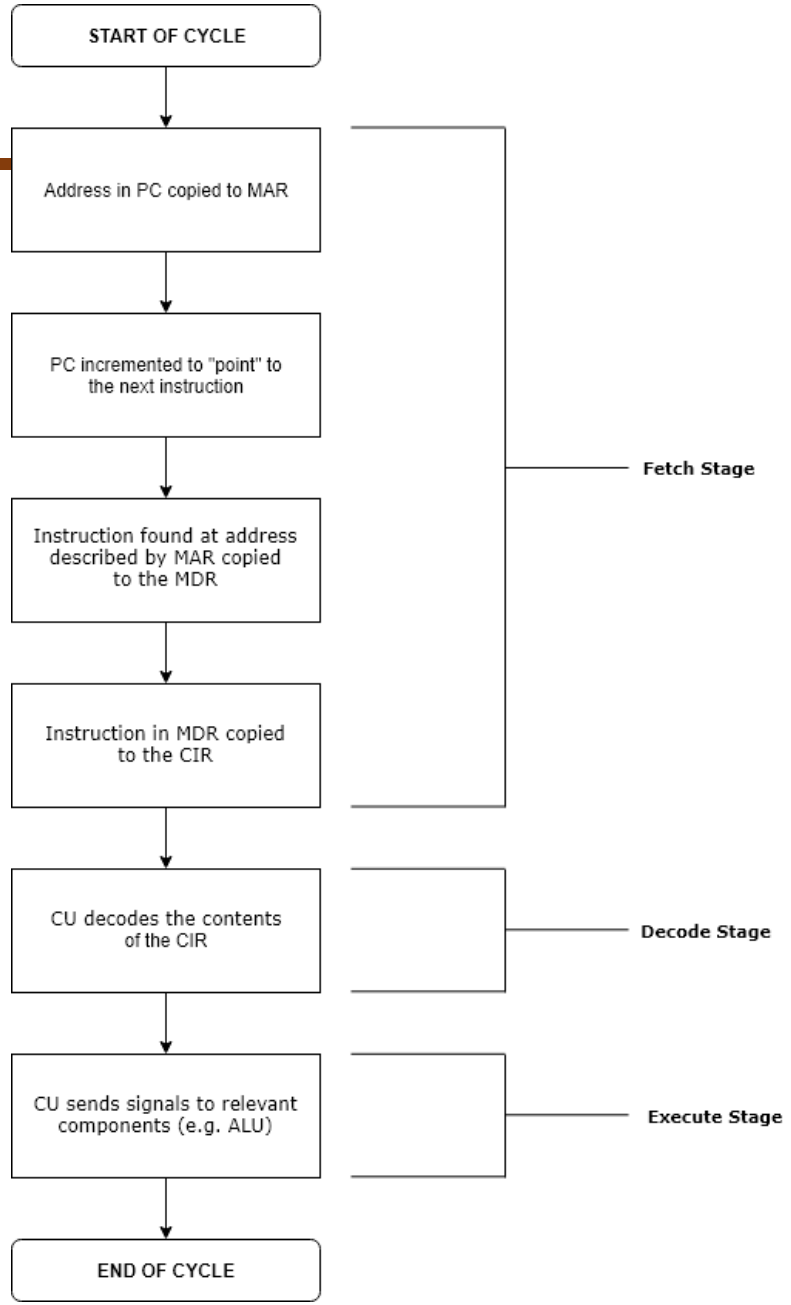
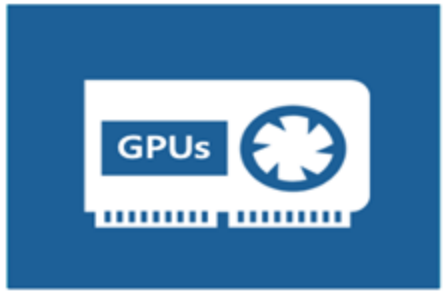


Fig. 6

GPUs



Multiple Sequential processing units – simpler set of instructions

Fast execution of identical operations on vectors

Excellent support for parallel processing

Fast evolving architectures with increasing parallelization

Driven by the AI/ML industry hunger for resources

NVIDIA Tech

Architectures | Enterprise & Developer

Blackwell Architecture (March 2024)

Fueling accelerated computing and generative AI with unparalleled performance, efficiency, and scal

Read More >

Hopper Architecture (March 2022)

Extraordinary performance, scalability, and security for every data center.

Read More >

Ada Lovelace Architecture (September 2022)

Performance and energy efficiency for endless possibilities.

Read More >

Previous Architectures:

Ampere Architecture (2020) >

Turing Architecture (2018) >

Volta Architecture (2017) >

Pascal Architecture (2016)

Maxwell Architecture (2014)

Kepler Architecture (2012)

Fermi Architecture (2010)

Tesla Architecture (2006)

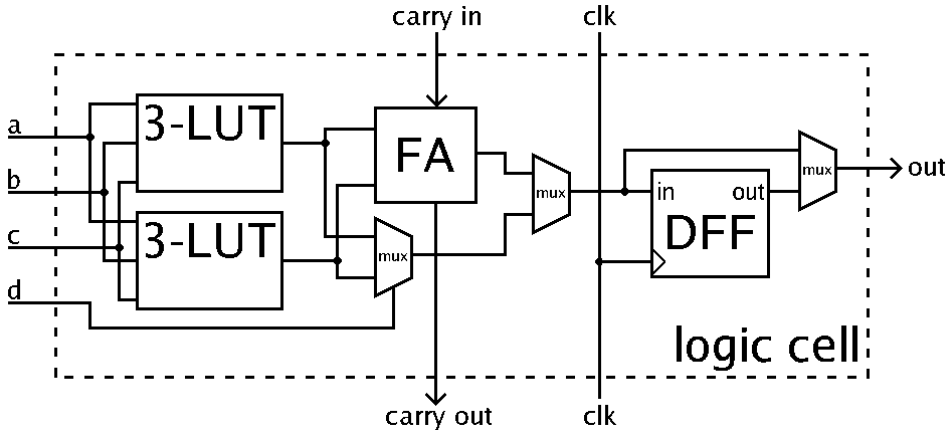
Curie Architecture 2004)

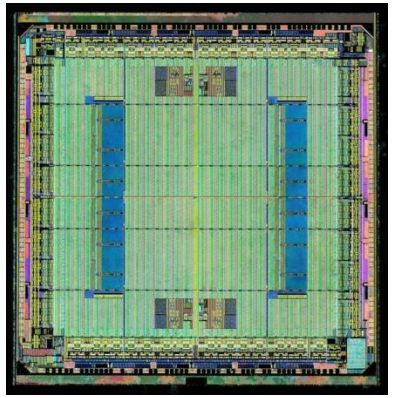
Rankine (2003)

Kelvin (2001)

Celsius (1999)

FPGAs



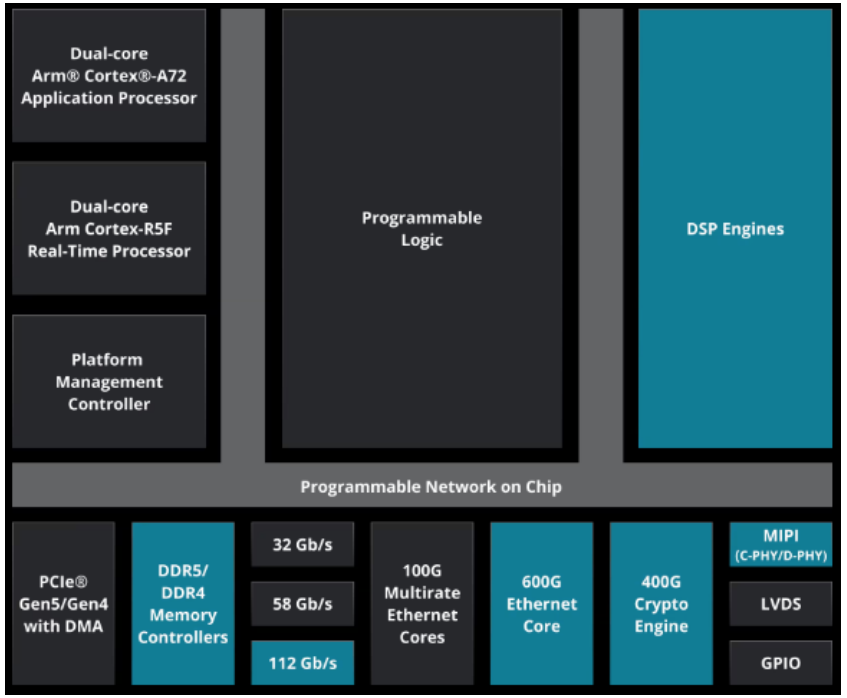


Basic Logic Cell Repeated million times!
Programmable interconnections
High-bandwidth data delivery to the chip

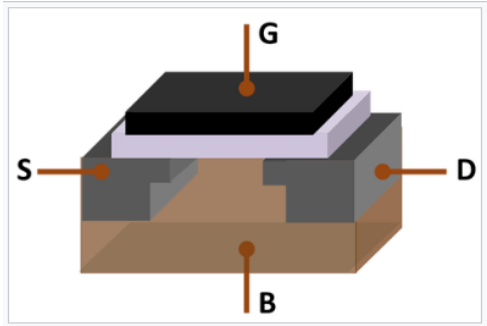
Flexible interconnects – low latency, optimal data flow

Excellent support for parallel processing

Addition of service blocks (multi-Gb serial IO, DSPs …)



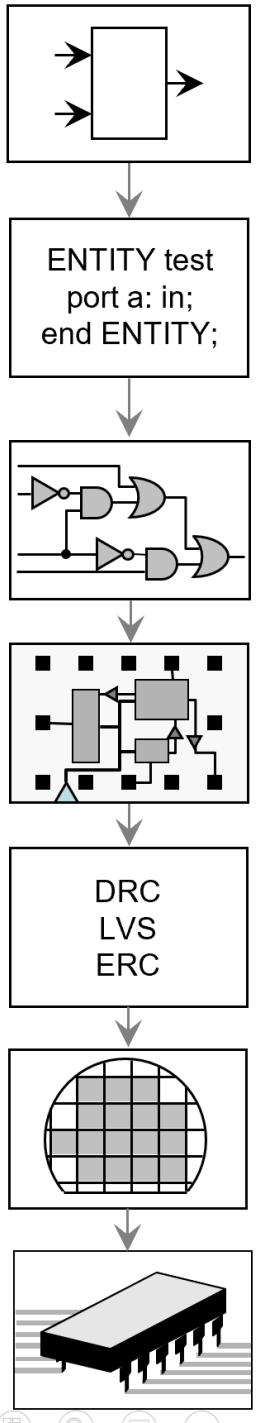
ASICs



MOSFET, showing gate (G), body (B), source (S), and drain (D) terminals. The gate is separated from the body by an insulating layer (pink).

Custom integrated chip for specific application
Enabled by modern chip design tools
Standardized MOS IC technology
Foundries with semiconductor fabs

They can make whatever you design for a price.
A chip development cycle could be years for a small team.
NRE charges are large!



System Specification
Architectural Design
Functional Design and Logic Design
Circuit Design
Physical Design
Physical Verification and Signoff
Layout Post Processing
Fabrication
Packaging and Testing

A comparison

Fig. 6

General-purpose processors, the performance of which isn't ideal for graphics and video processing

A popular choice for AI computations. GPUs offer parallel processing capabilities, making it faster at image rendering than CPUs.

FPGAs, such as those available on Azure, provide performance close to ASICs. They're also flexible and reconfigurable over time, to implement new logic.

Custom circuits, such as Google's Tensor Processor Units (TPU), provide the highest efficiency. They can't be reconfigured as your needs change.

This is from flexibility in programmability point of view

A comparison

Fig. 6

General-purpose processors, the performance of which isn't ideal for graphics and video processing

A popular choice for AI computations. GPUs offer parallel processing capabilities, making it faster at image rendering than CPUs.

FPGAs, such as those available on Azure, provide performance close to ASICs. They're also flexible and reconfigurable over time, to implement new logic.

Custom circuits, such as Google's Tensor Processor Units (TPU), provide the highest efficiency. They can't be reconfigured as your needs change.

Time to get the user algorithm executed given data availability at similar chip technology

A comparison

Fig. 6

General-purpose processors, the performance of which isn't ideal for graphics and video processing

A popular choice for AI computations. GPUs offer parallel processing capabilities, making it faster at image rendering than CPUs.

FPGAs, such as those available on Azure, provide performance close to ASICs. They're also flexible and reconfigurable over time, to implement new logic.

Custom circuits, such as Google's Tensor Processor Units (TPU), provide the highest efficiency. They can't be reconfigured as your needs change.

Flexibility in implementing streaming data interfaces to algorithm execution units

A comparison

Fig. 6

General-purpose processors, the performance of which isn't ideal for graphics and video processing

A popular choice for AI computations. GPUs offer parallel processing capabilities, making it faster at image rendering than CPUs.

FPGAs, such as those available on Azure, provide performance close to ASICs. They're also flexible and reconfigurable over time, to implement new logic.

Custom circuits, such as Google's Tensor Processor Units (TPU), provide the highest efficiency. They can't be reconfigured as your needs change.

Optimization of hardware resources (power savings) in executing a custom algorithm.

A comparison

Fig. 6

General-purpose processors, the performance of which isn't ideal for graphics and video processing

A popular choice for AI computations. GPUs offer parallel processing capabilities, making it faster at image rendering than CPUs.

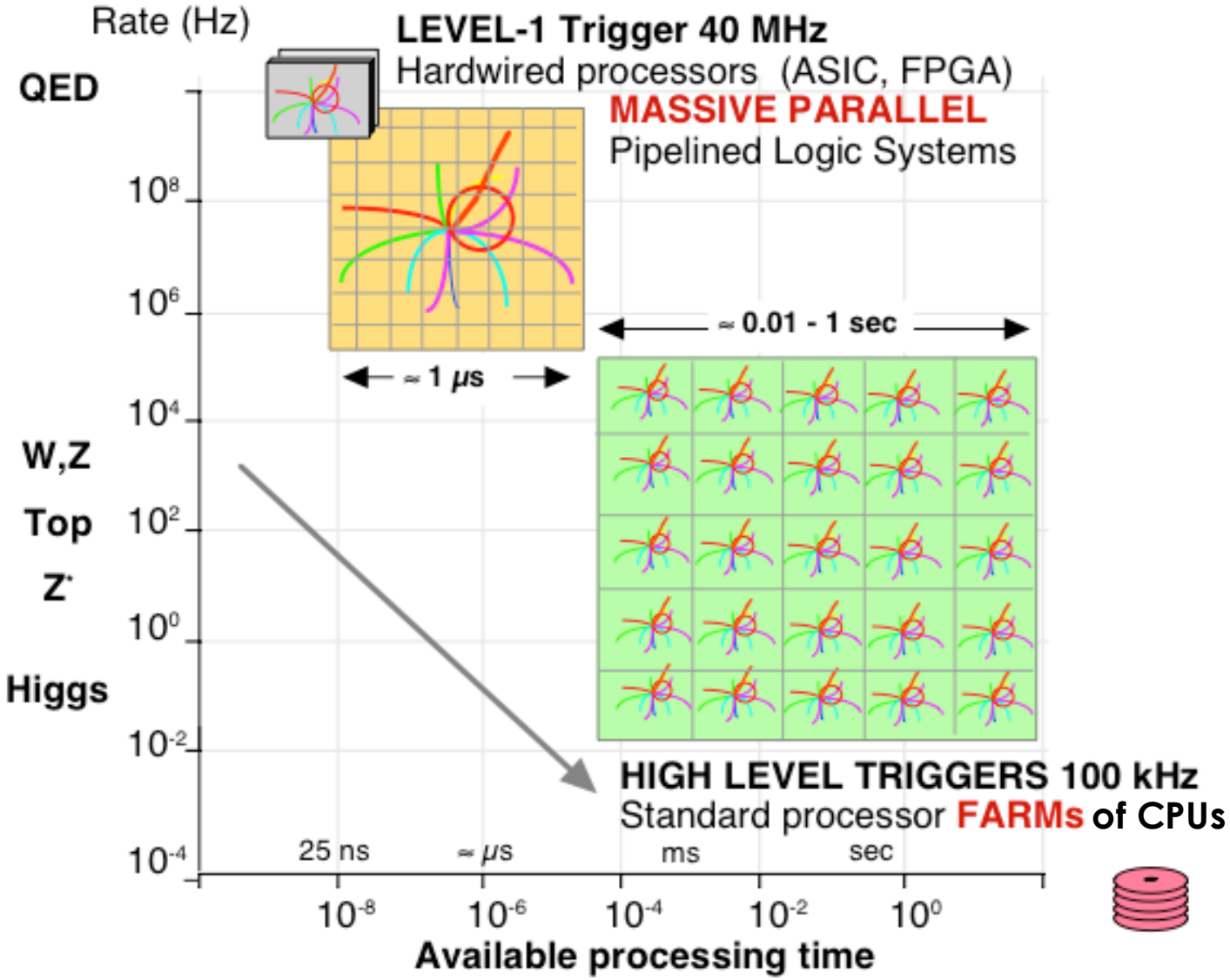
FPGAs, such as those available on Azure, provide performance close to ASICs. They're also flexible and reconfigurable over time, to implement new logic.

Custom circuits, such as Google's Tensor Processor Units (TPU), provide the highest efficiency. They can't be reconfigured as your needs change.

Milliseconds → microseconds → nanoseconds → nanoseconds

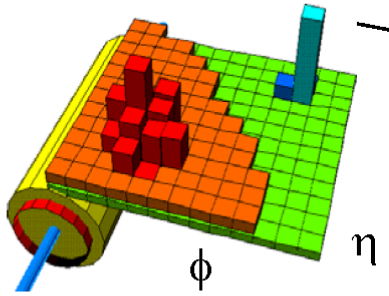
Although clock may be GHz (ns) **user control of algorithm** execution times varies.

LHC Data Processing

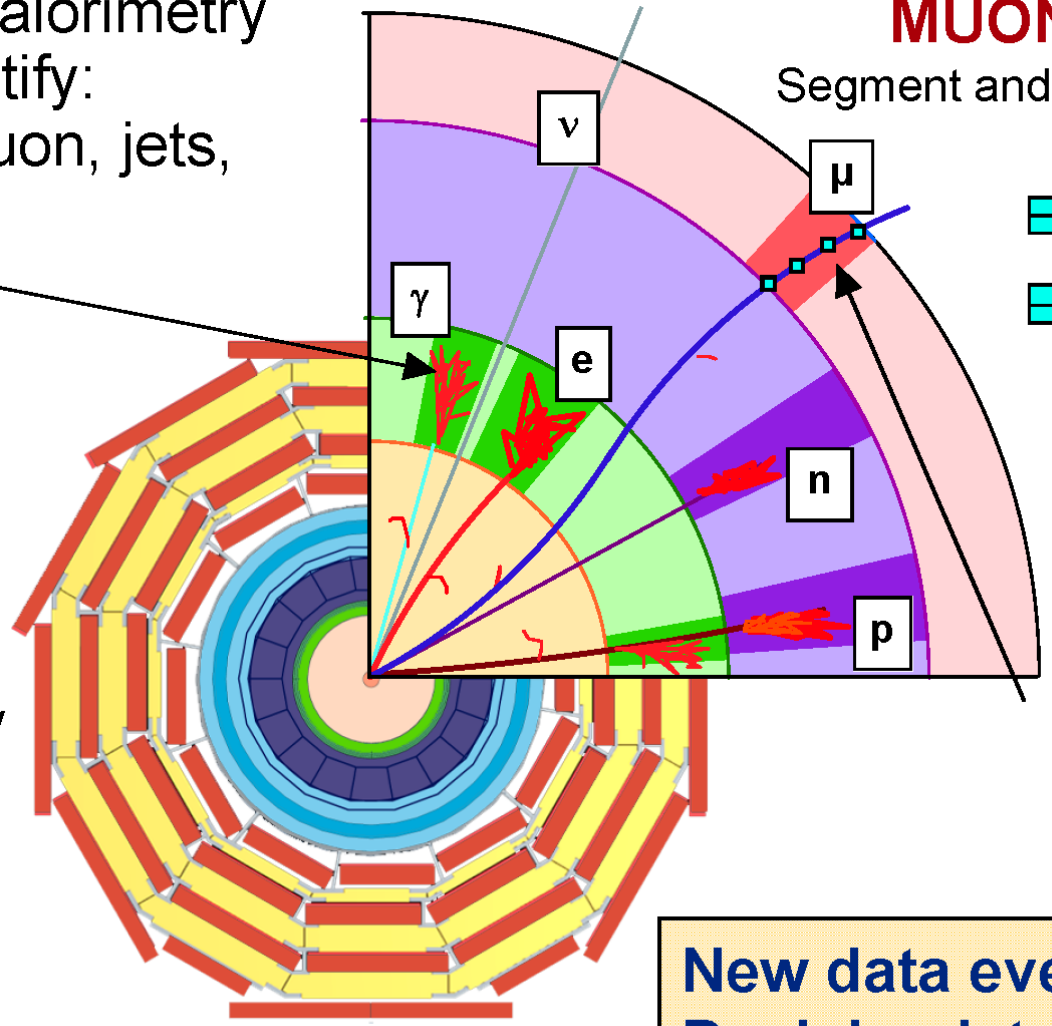


Level-1 Trigger Data Processing

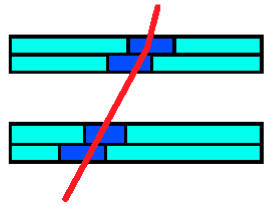
Use prompt data (calorimetry and muons) to identify:
High p_t electron, muon, jets, missing E_T



CALORIMETERs
Cluster finding and energy deposition evaluation



MUON System
Segment and track finding



Input data is streamed in over thousands of Gbps fibers

Makes most sense to process local patterns in parallel

New data every 25 ns
Decision latency ~ μs

Why do we need to learn about FPGA?

- Most experiments use FPGAs for some trigger/DAQ tasks (CMS, ATLAS, Neutrino, etc.)
- All experiments collect physics data via optical/electrical links
  - Initial Readout and processing in almost all cases is based on FPGAs
- In general, distribution of work in experiments:
  - Physicist:
    - Algorithm, Firmware, tests, commissioning, etc…
  - Engineers:
    - Design, layouts, production, etc…
- To better understand and troubleshooting relevant components of HEP experiments
  - Physicists need FPGA knowledge
  - Understand the processing happening inside the FPGA
  - To talk to engineers and explain the needs

Beyond HEP?

- Progammable Logic is state of the art:
  - Most high-tech electronics product designers start with FPGAs
  - Designing prototype electronics
  - Allow for easy reconfiguration of ideas and testing
  - Plenty of opportunities beyond HEP

- FPGAs vs ASICs:
  - Producing ASICs is expensive (>$100K) and slow vis-à-vis PCBs
  - FPGA-based PCBs with standardized multi-Tbps IO cost < $100K
  - FPGA boards can reduce cost & allow many applications
  - Opportunities for FPGA-experienced people in product development
    - Within HEP and beyond

TAC-HEP 2025

Questions?