Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Fondazione ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Introductory course to VHDL

Andrea Triossi – University of Padova – INFN Padova

Overview

- FPGA
  - Quick intro
  - Design flow
  - Architecture
- VHDL
  - Code examples
  - Programming tool
  - Full project


```
entity comb_function is
 port (a, b, c : in BIT; z: out BIT);
end entity comb_function;
architecture netlist of comb_function is
 signal p, q, r : BIT;
begin
 g1: entity WORK.Not1(ex1) port map (a, p);
 g2: entity WORK.And2(ex1) port map (p, b, q);
 g3: entity WORK.And2(ex1) port map (a, c, r);
 g4: entity WORK.Or2(ex1) port map (q, r, z);
end architecture netlist;
```

Field Programmable Gate Array

- Field-Programmable: reconfigurable by the user by means of programming languages
- Gate-Array: programmable logic gates (but also many other hardware blocks) and configurable interconnections

FPGA vs CPU

- Low (and deterministic) latency
- Easier connectivity (higher bandwidth)
- Higher degree of parallelism

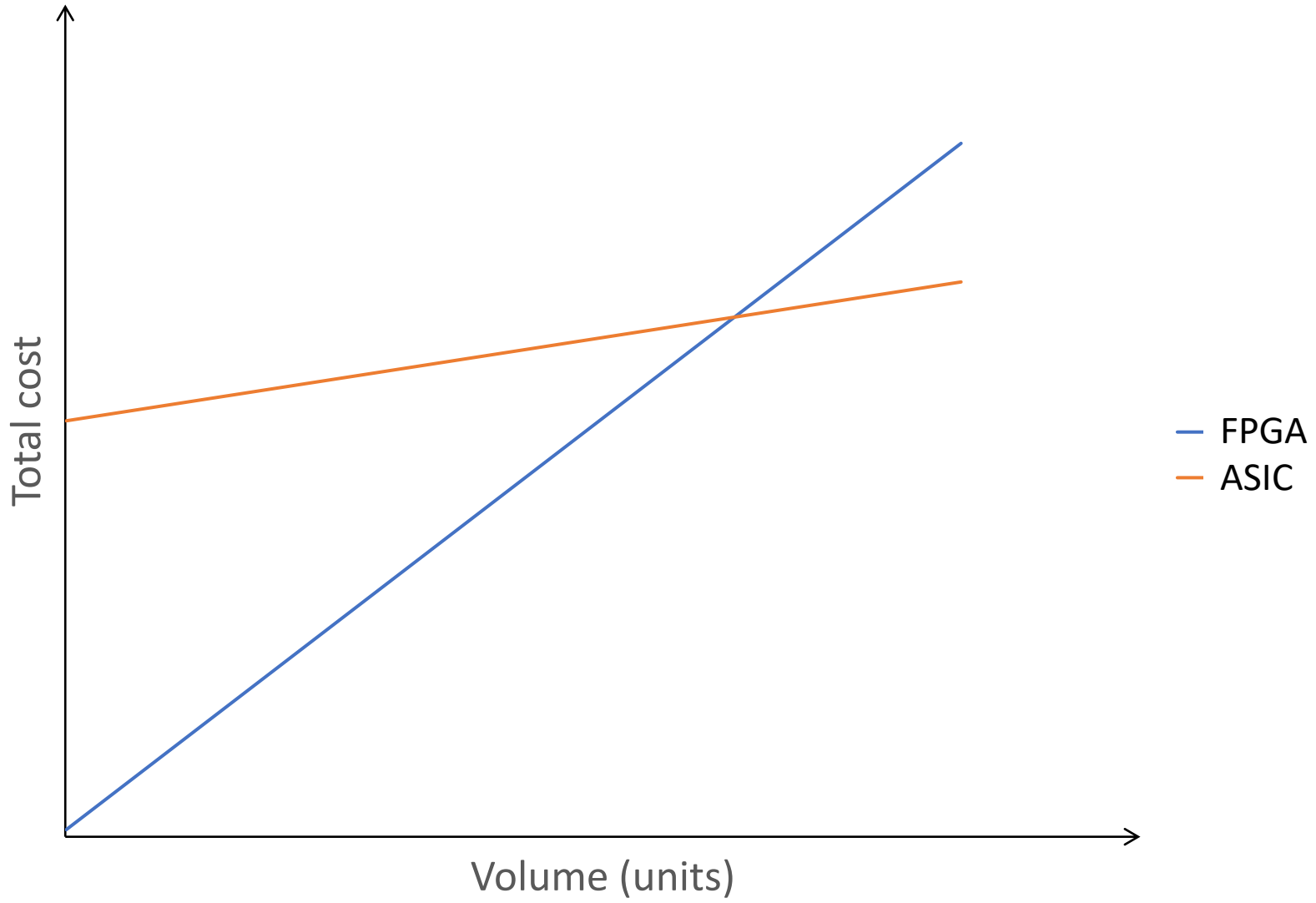
- Programming (software) is easier and faster than configuring (firmware)

FPGA vs ASIC

- Reconfigurable with different design (even partially)
- Design is specified by hardware description languages (HDL) like VHDL or Verilog
- Low entry-barrier (affordable price for a single chip)
- Easy and quick design flow. Usually, designer doesn’t have to care about reset and clock tree, physical or manufacturing details, routing etc…

- Power demanding
- Not recommended for high-volume
- Limited in operating frequency
- Analog design not possible (only few programmable blocks are available)

FPGA vs ASIC



FPGA programming

FPGA programming is about designing digital logic circuits to define the behaviour of FPGAs while software programming is about the execution of a sequence of sequential instructions to perform a specific behaviour in software

FPGA programming

FPGA programming is about designing digital logic circuits to define the behaviour of FPGAs while software programming is about the execution of a sequence of sequential instructions to perform a specific behaviour in software

- FPGA programming flow
- FPGA architecture
- Hardware description language
- Simulation
- Synthesis & Implementation
- Debugging

Lab →

- Vivado installed locally (or in VM)
- Simulation local
- Synthesis & Implementation local
- FW deployment from remote

Thanks to Giulio Bianchini e Mirko Mariotti!

FPGA applications

From Wikipedia…

Detectors for Physics

⇩

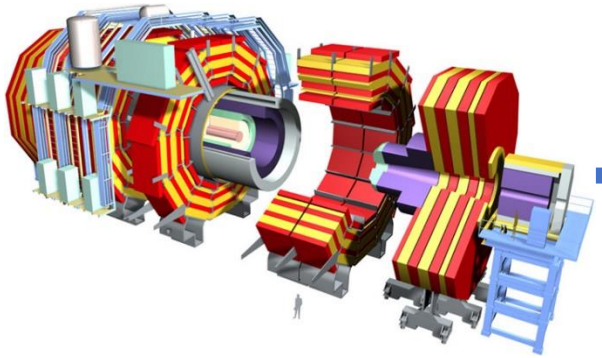
Trigger and DAQ systems

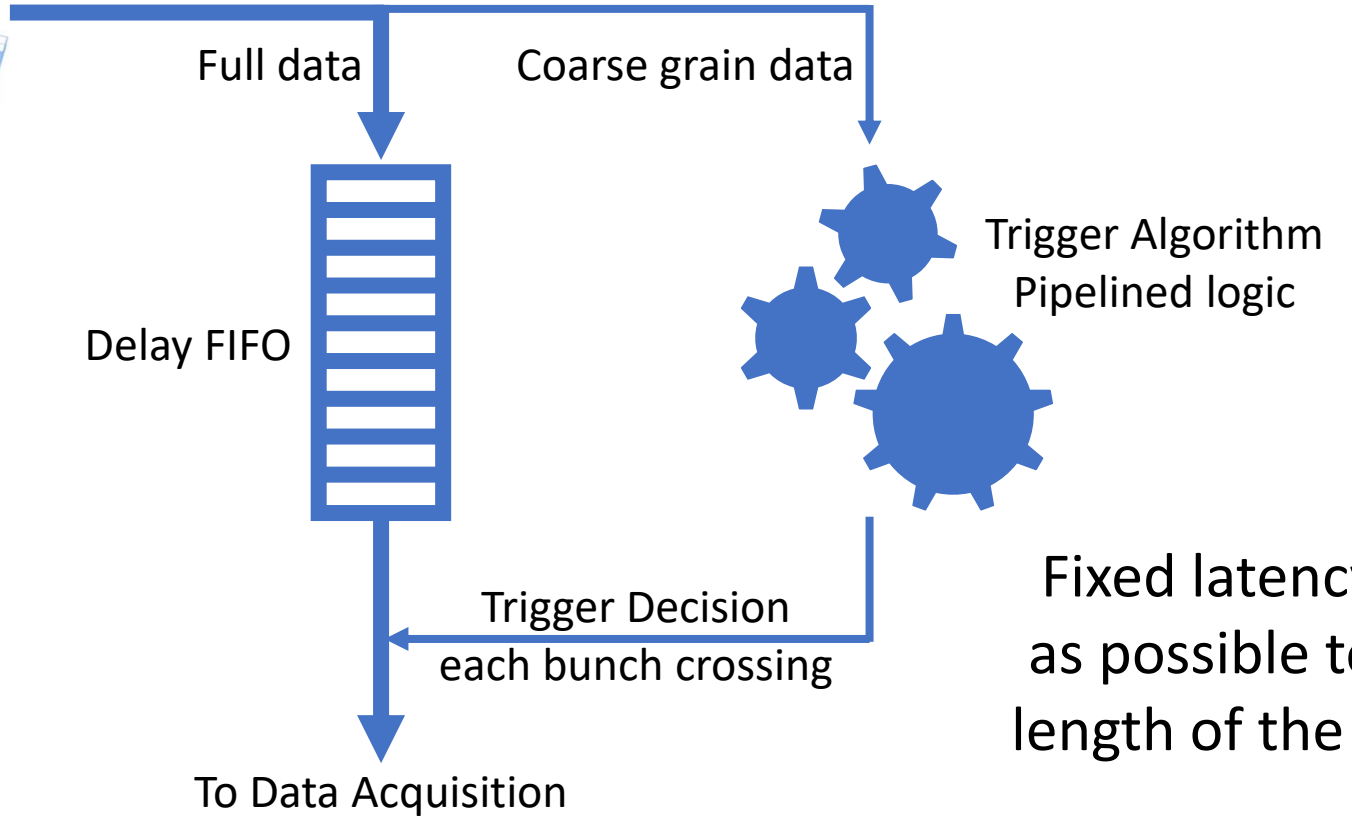
Common applications [edit]

This is a dynamic list and may never be able to satisfy particular standards for completeness. You can help by adding missing items with reliable sources.

- Aerospace and defense
  - Avionics/DO-254
  - Communications
  - Missiles & munitions
  - Secure solutions
  - Space (i.e. with radiation hardening[40])
- Audio
  - Connectivity solutions
  - Digital-to-analog converter
  - Portable electronics
  - Software-defined radio
  - Digital signal processing (DSP)
  - Speech recognition
  - Synthesizers
- Automotive
  - High resolution video
  - Image processing
  - Vehicle networking and connectivity
  - Automotive infotainment
- Artificial neural networks
- Bioinformatics
- Broadcast
  - Color grading
  - Real-time video engine
  - EdgeQAM
  - Encoders
  - Displays
  - Switches and routers
- Consumer electronics
  - Digital displays
  - Digital cameras
  - Multi-function printers
  - Portable electronics
  - Set-top boxes
  - Flash cartridges
- Data center
  - Servers
  - Security
  - Hardware security module[41]
  - Routers
  - Switches
  - Gateways
  - Load balancing
- High performance computing
  - Servers
  - Super computers
  - Signals intelligence systems
  - High-end radars
  - High-end beam forming systems
  - Data mining systems
- Industrial
  - Industrial imaging
  - Industrial networking
  - Motor control
- Integrated circuit design
  - ASIC prototyping
  - Computer hardware emulation
- Financial
  - Crypto mining
  - High-frequency trading
- Medical
  - Ultrasound
  - CT scanning
  - MRI
  - X-ray
  - PET
  - Surgical systems
- Scientific instruments
  - Lock-in amplifiers
  - Boxcar averagers
  - Phase-locked loops
  - Radio astronomy
- Security
  - Industrial imaging
  - Secure solutions
  - Hardware security module[41]
  - Password cracking
  - Image processing
- Test and measurement equipment
  - Oscilloscopes
  - Spectrum analysers
  - Vector network analyzers
  - Signal generators
  - Data acquisition (DAQ) and logging
  - Multiplexers and switching arrays
- Video & image processing
  - High resolution video
  - Video over IP gateway
  - Digital displays
  - Industrial imaging
  - Computer vision
  - Thermal imaging
- Wired communications
  - Optical transport networks
  - Network processing
  - Connectivity interfaces
- Wireless communications
  - Baseband
  - Connectivity interfaces
  - Mobile backhaul
  - Radio

First-Level trigger





Data are moved synchronously with the accelerator clock:
25 ns bunch crossing

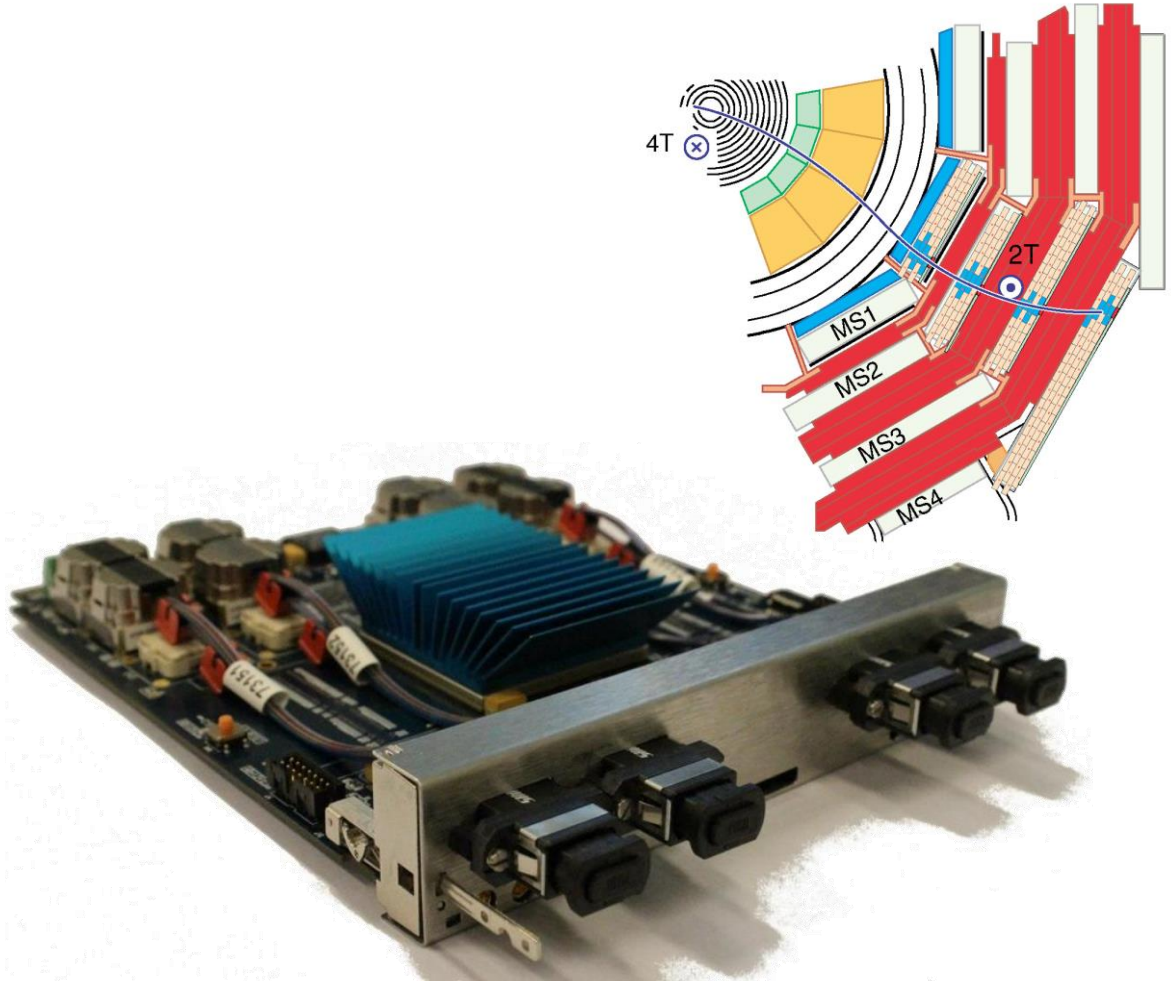
Fixed latency as short as possible to limit the length of the delay FIFO

FPGA for trigger

- Low latency execution comparing with discrete electronics (all connections are internal)
- Many inputs that can collect and combine data from many parts of the detector
- High degree of parallelization very useful for pipelined logic
- Re-programming plays a key role in optimization of trigger algorithms

Examples of trigger algorithms

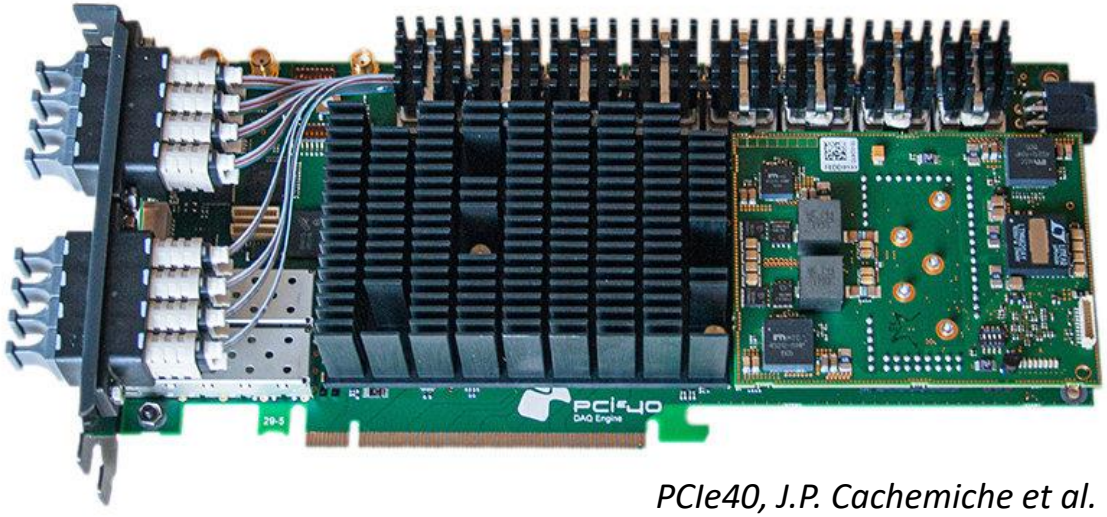
- Peak finding
- Pattern recognition
- Track finding
- Clustering
- Energy summing
- Sorting
- Topological algorithms
- Data merging
- Machine learning inference
- …..



MP7, A. Rose et al.

Data acquisition

- Front end
  - Pedestal subtraction
  - Zero suppression
  - Compression
  - .....
- Custom data links
  - E-LINK (up to 1.28 Gb/s) on copper
  - LpGBT (10.24/2.56 Gb/s) on optical
  - .....
- Interfaces from custom to commercial
  - PCIe Gen4
  - 10/40/100 Gb/s Ethernet
  - .....



PCIe40, J.P. Cachemiche et al.

What you will learn

- Hardware Description Language
  - SW programming -> execution of sequential instructions
  - HW programming -> design of concurrent digital logic
- FPGA programming workflow
  - Simulation
  - Synthesis & Implementation
  - Debugging

- At the beginning
  - Simple but complete projects
  - Interactive tutorial
  - VHDL by example
- Later
  - A little bit more complex projects
  - You gain independence