PROGRAMMABLE HARDWARE DEVICES

Introduction

Introductions

- Andrea Triossi
  Dipartimento di Fisica ed Astronomia
  Via Marzolo 8 – Room 3-44, Lab. 3-24
  andrea.triossi@unipd.it

- Main interests
  - Digital Electronics for physics experiments
  - Reconfigurable computing

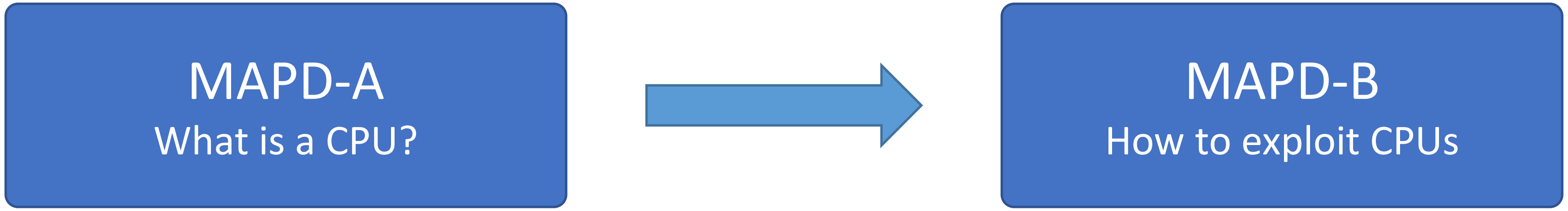
- What about you?

Physics

Computer science

Electronic Engineer

Background



Background

MAPD-A
What is a CPU?

MAPD-B
How to exploit CPUs

PHD
How to improve a CPU?

Is the CPU the best hardware architecture for computation?

+ Flexible (software programmable)
– Efficient (limited number of instructions)

Imagine a new device

- Design your own instruction (algorithm)
- Execute instructions in parallel
- Freely connect input and output of each instruction
- Many registers
- …

Hardware optimized for your needs

Imagine a new device

- Design your own instruction (algorithm)
- Execute instructions in parallel
- Freely connect input and output of each instruction
- Many registers
- …

Hardware optimized for your needs

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

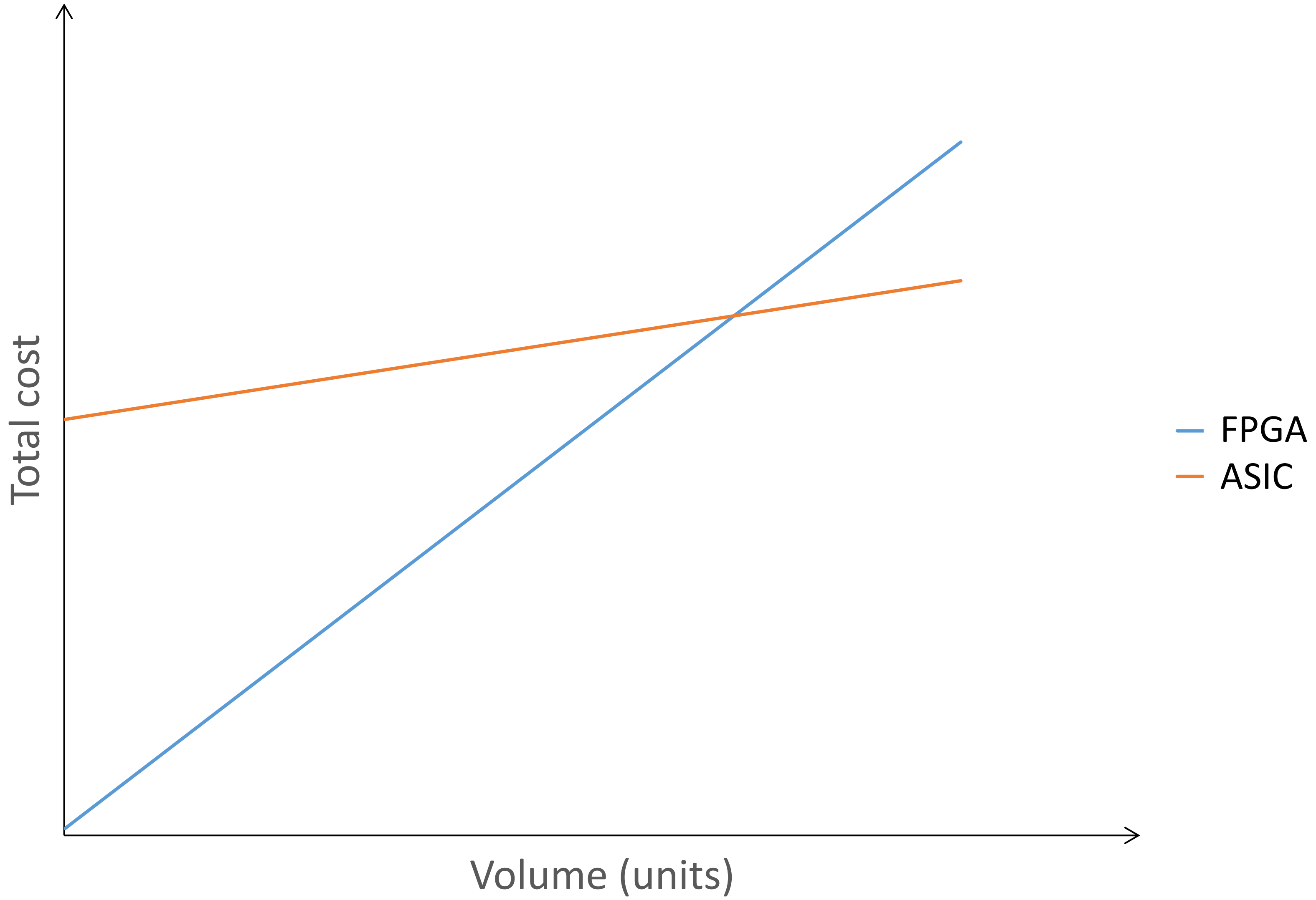
- Programming (software) is easier and faster than configuring (firmware)

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

Lab

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.

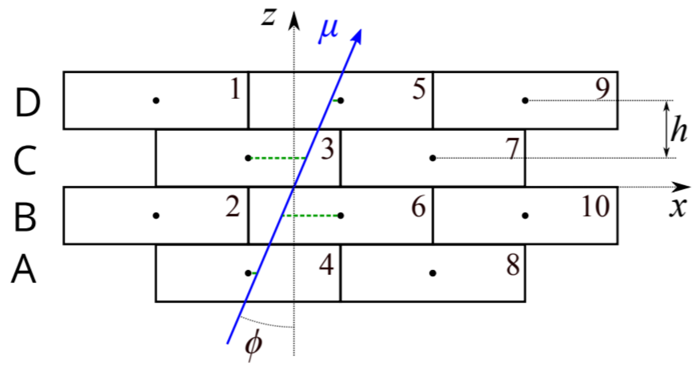
- Aerospace and defense
  - Avionics/DO-254
  - Communications
  - Missiles & munitions
  - Secure solutions
  - Space (i.e. with radiation hardening[40])
- Audio
  - Connectivity solutions
  - Digital-to-analog converter
  - Portable electronics
  - Software-defined radio
  - Digital signal processing (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

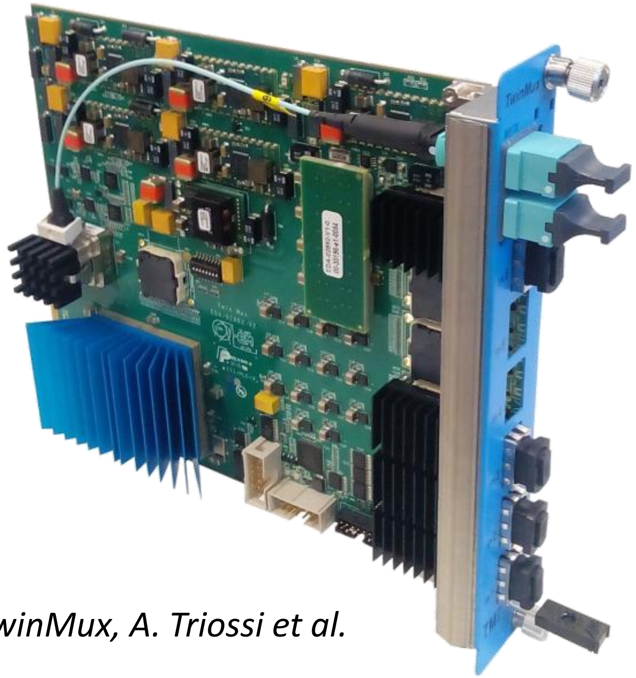
FPGA for trigger

- 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
- .....

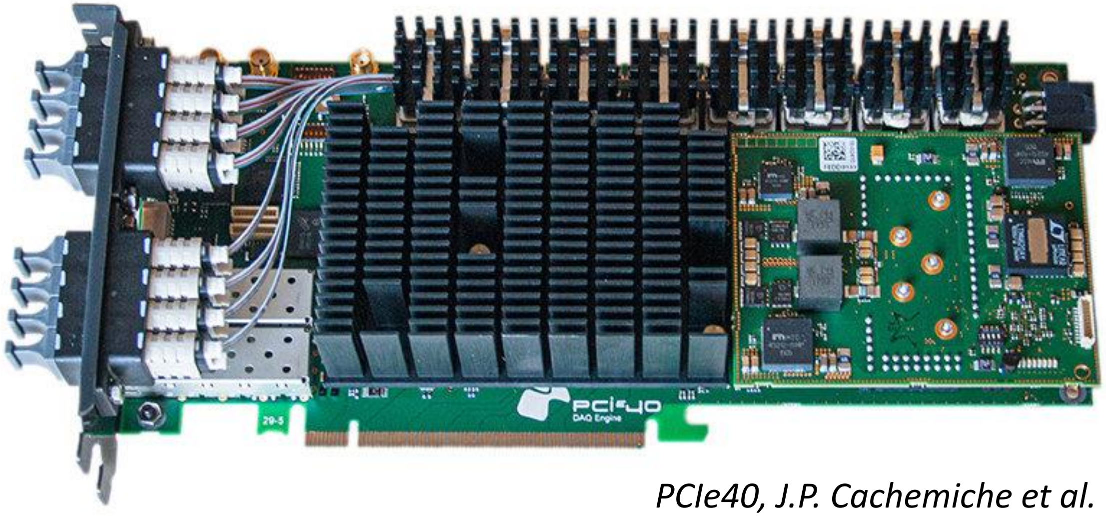




TwinMux, A. Triossi et al.

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.

Course info

Prerequisites

- Basic notions of digital electronics

  What you already learned in MAPD-A

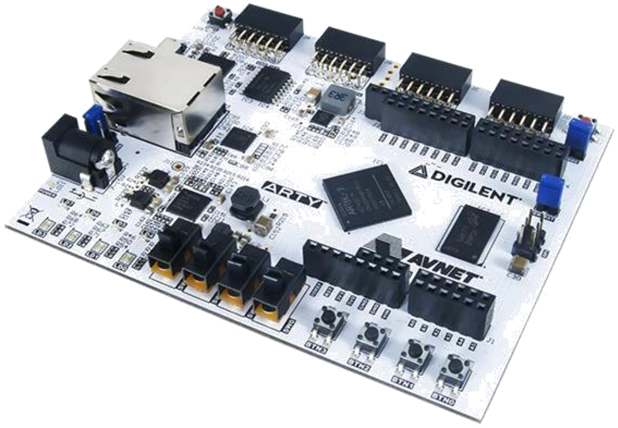
  - Combinatorial logic
  - Sequential logic
  - Arithmetic operations and Bit manipulation

- Basic programming elements (either Python or C/C++)

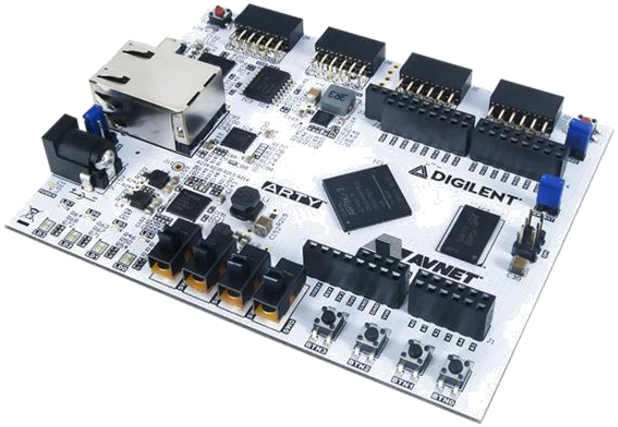
  What you already learned in LCP-A

Programmable Hardware Devices

- A laboratory
  - FPGA
  - Microcontroller for IoT (expert form CERN)
- 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
  - Network protocols for IoT (MQTT)
  - Bare-metal programming of a µC
  - Fundamentals of real-time operating systems for µC
  - Sensors and peripherals (audio/video)



FPGA part



- At the beginning
  - Simple but complete projects
  - Interactive tutorial
  - VHDL by example
- Later
  - A little bit more complex projects
  - You gain independence
- Build a UART (Universal Asynchronous Receiver/Transmitter) interface for data transfer with your laptop
- Develop your own project (it could be based on UART)

Microcontroller for IoT part

- Teacher: Prof. Christoph Schwick
  - Physics at University of Heidelberg
  - PHD at University of Hamburg
  - Since 1999 staff at CERN
    - Data Acquisition Project
    - Run Coordinator of the CMS experiment
  - Technology enthusiast

Practical info

- Room LabP036
  - In principle there are PCs
- Your laptop is need
  - In alternative we can provide a mini-PC
- Microcontroller for IoT (16 hours) will be squeeze in two weeks
  - Tentative period: weeks 49 and 50
- Software
  - Vivado (Version ≥ 2018.3)
  - Serial console
    - Tera Term for Windows, or Putty, or any other serial console
    - Screen, Picocom, Minicom, etc. for Linux

Practical info

- Book: Zwolinski M., “Digital System Design with VHDL”, Pearson Ed.
- Free books
  - VHDL Handbook
  - Free Range VHDL
- Web resources
  - EDAplayground
  - VHDLwhiz
  - Surf vhdl

Exam

- Single exam for the two parts
- Project on FPGA or Microcontroller for IoT
  - Presentation and discussion
  - General questions on the full course
- Level of the project depends on you
  - Algorithms
  - Control registers for I/O
  - Read/write memory
  - Soft CPU
  - Music effects
  - Neural networks
- What matters most is that you show that you can use properly VHDL and you have tackled the problem in a consistent way
- Having fun with hardware leads to a positive evaluation