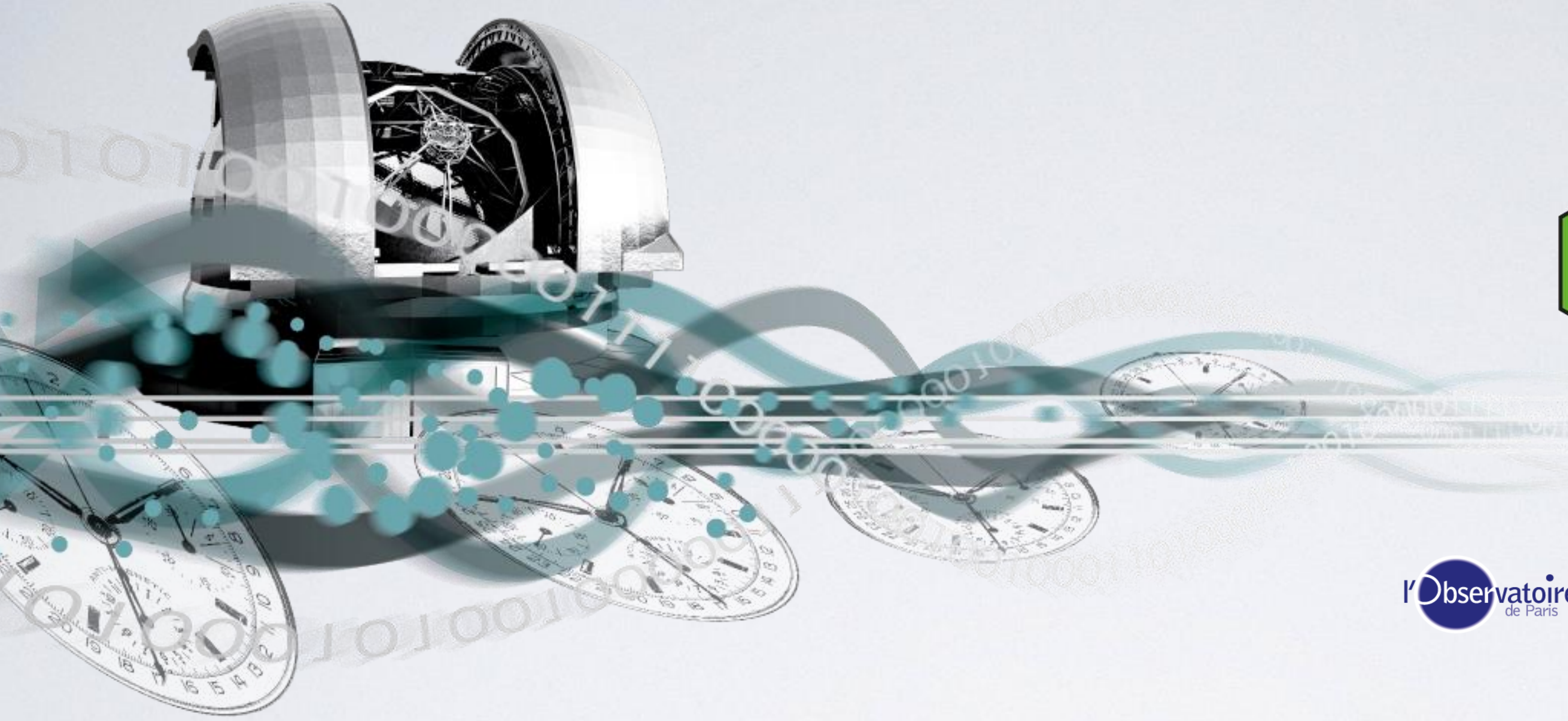
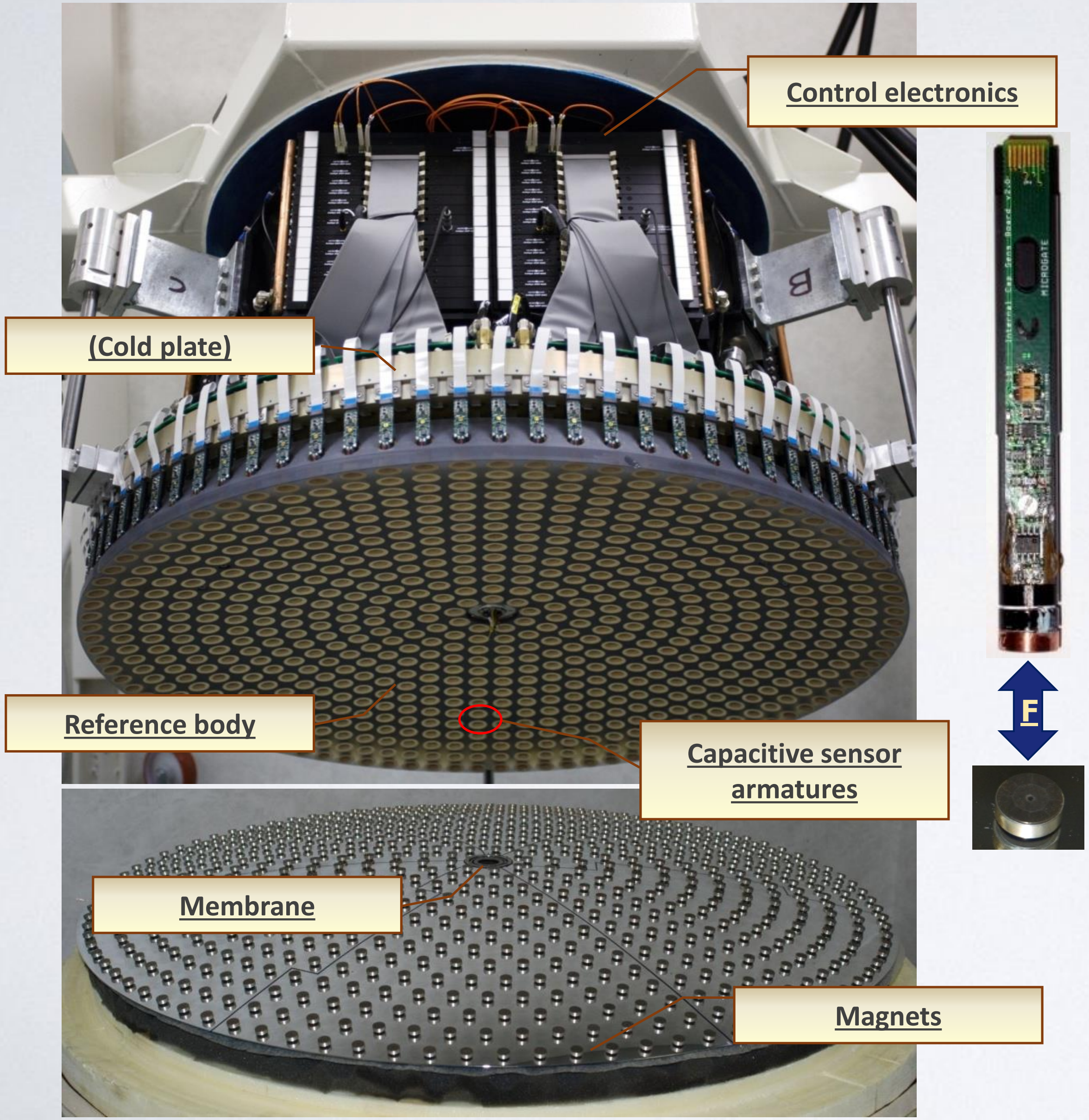


FPGA solutions for AO RTC

R.Biasi - Microgate

RTC4AO4, Paris 21 December 2016

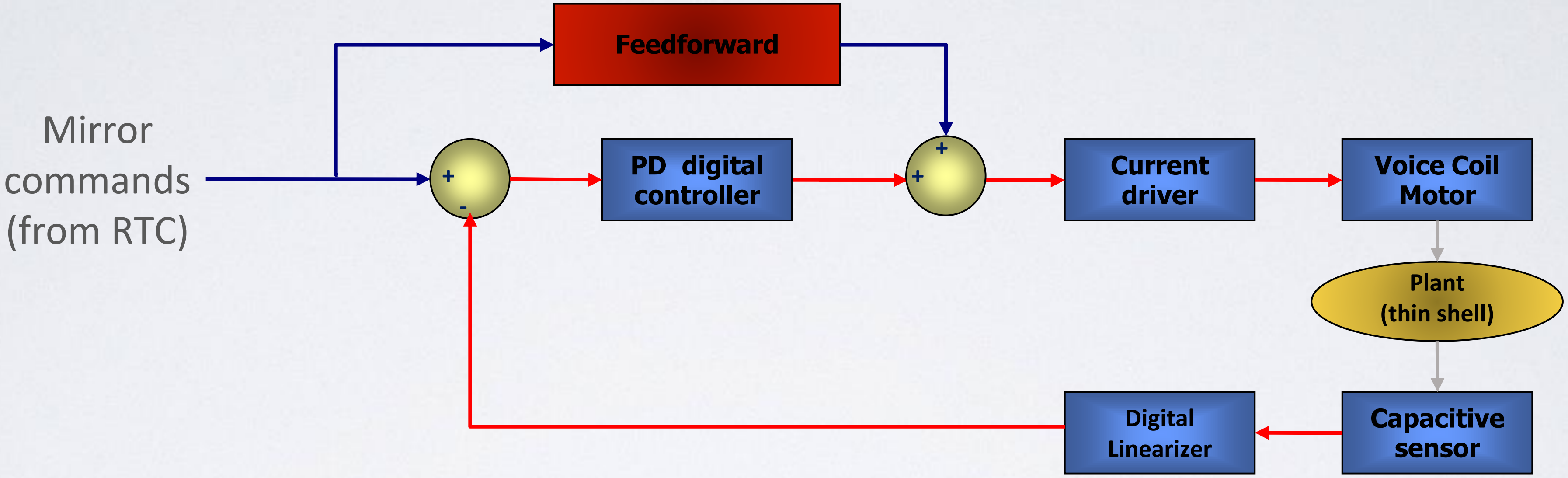
How did we start: large, contactless adaptive mirrors



- Concept by P.Salinari (1993) still adopted by <u>all</u> units
- Contactless operation
  - Force actuators + capacitive sensors, position loop
- Membrane(s) to counteract in-plane forces
- Reference body providing the 'fixed' reference
- <u>**On-board control electronics**</u>

Large, contactless adaptive mirror concept - control

How to make a *position* actuator act as a *force* actuator?



Mirror commands (from RTC)

Feedforward

+

-

PD digital controller

+

+

Current driver

Voice Coil Motor

Plant (thin shell)

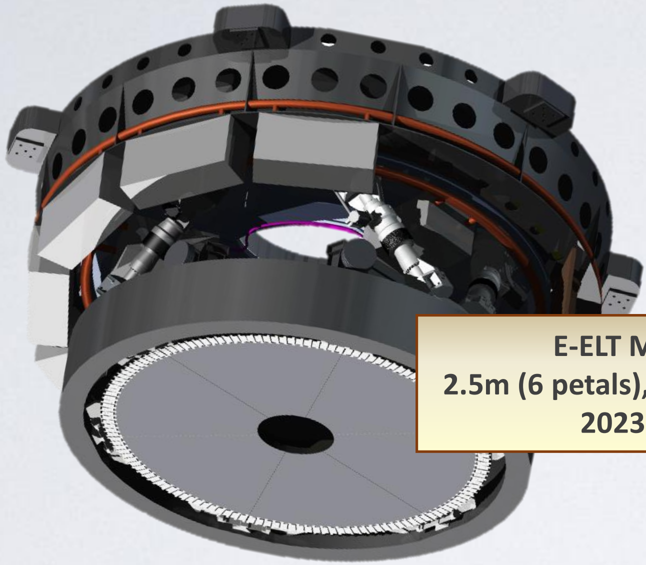
Capacitive sensor

Digital Linearizer

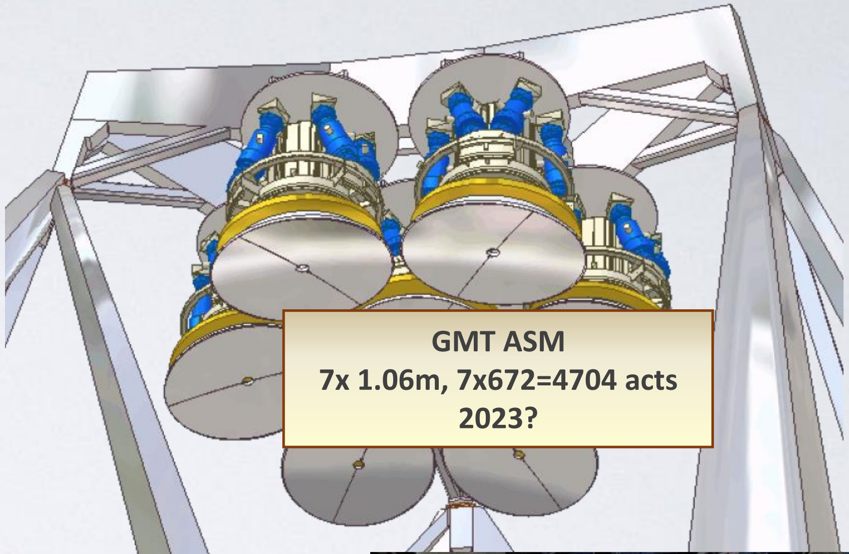
→ **WFS loop frequency (~ 1kHz)**

→ **co-located loop frequency (~ 80kHz)**

Evolution through systems

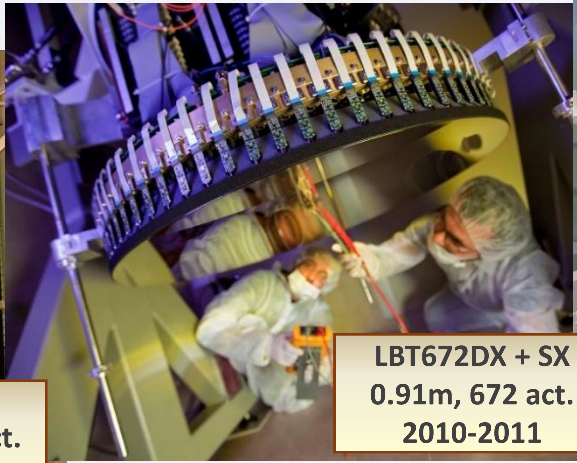


E-ELT M4
2.5m (6 petals), 5316 acts
2023

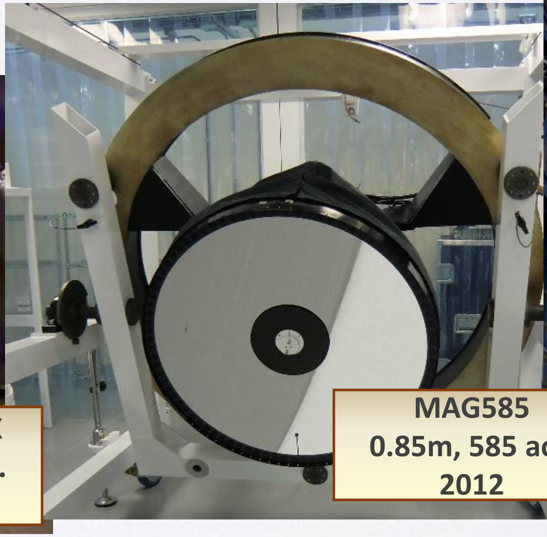


GMT ASM
7x 1.06m, 7x672=4704 acts
2023?

MMT336
0.64m, 336 act.
2002



LBT672DX + SX
0.91m, 672 act.
2010-2011



MAG585
0.85m, 585 act.
2012

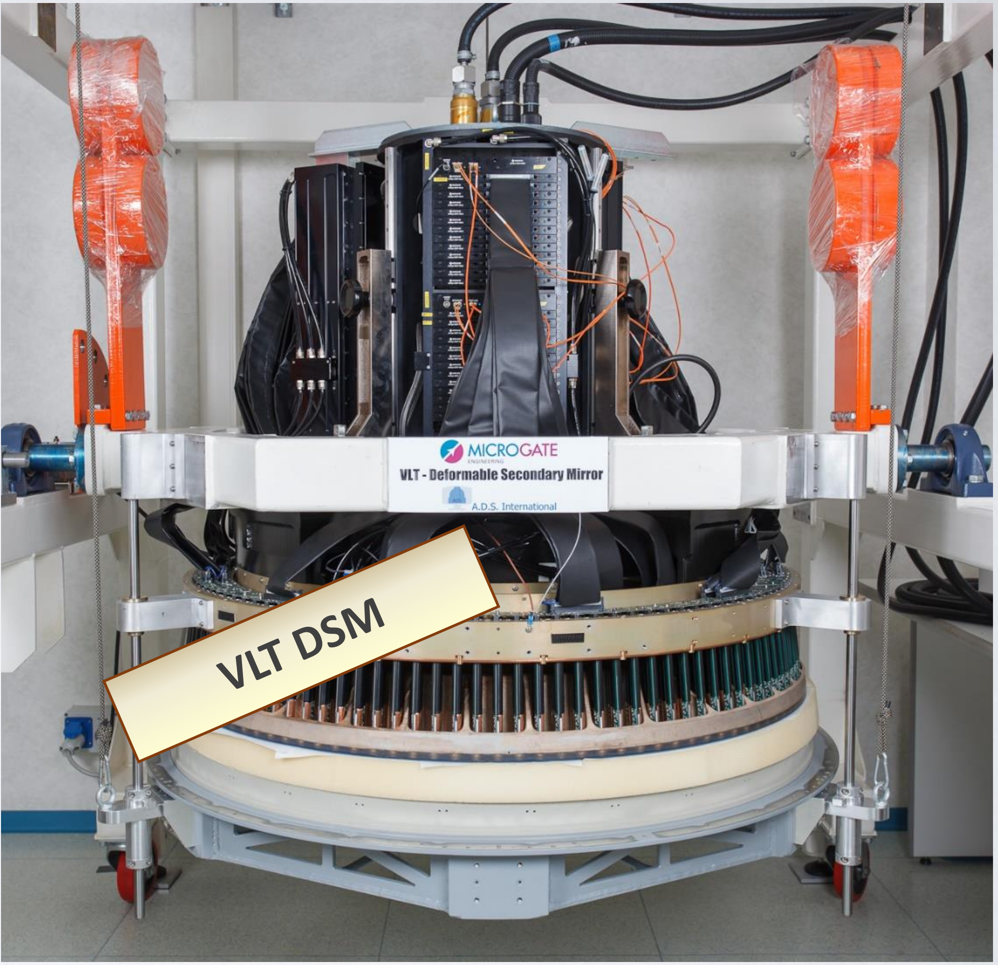
VLT DSM
1.12m, 1170 act.
2016

MICROGATE

Common custom HW/SW

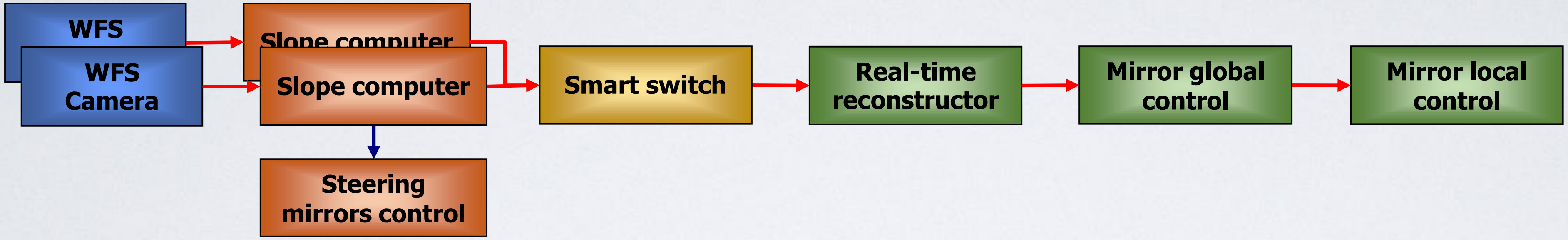
Main building blocks

- BCU (Basic Computational Unit) board: backplane arbiter, communication interface, slope computation, smart switch
    - 1x ADSP-TS101S
    - 1x first generation Altera Stratix
- DSP board: real-time control, analog interface (capacitive sensors, current drivers for VCMs)
    - 2x ADSP-TS101S (ADSP-TS201S on VLT)
    - 1x first generation Altera Stratix (Stratix 2 on VLT)
    - Exist in various flavors, e.g. 6ch HV piezo control with strain gage feedback, 8ch high speed ADC for direct CCD acquisition, 'digital only', …
- Relatively complex systems: e.g. VLT DSM has 156 ADSP-TS201S, ns-level synchronization, fully distributed computation



Common custom HW/SW

Covering the whole RT pipeline (+ analog control/acquisition)



Firmware/Software considerations

- Brute force vs. elegance (Caroline…): hard-real time coded only in ASM → theoretical computational power achieved, e.g. VLT DSM performs 150 GMAC/s FP sustained
- Very flexible, configurable firmware (code once, use for many configurations): all parameters/configurations loaded at run time and user configurable
- FPGA used mainly for real-time communication, routing gluing logic, and some data (pre)processing

Long lasting performance

All these systems are still **maintained and upgraded**

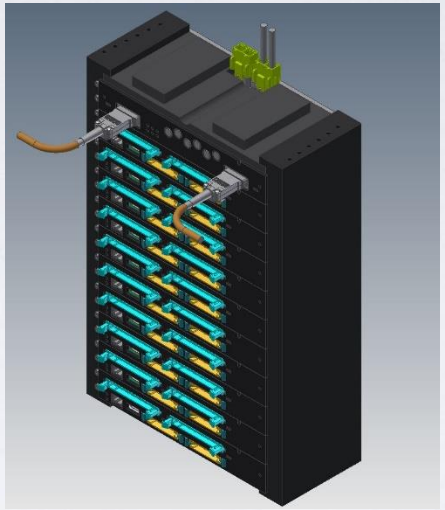
- Keck NIR-TT: multiple ROI Near Infrared Tip-Tilt (NGS) control
- MagAO 2k: upgrade to run at 2kHz and use CCID-75 camera for WFS
- LBT SOUL: upgrade to run at 2kHz and use First Light OCAM 2 camera for WFS

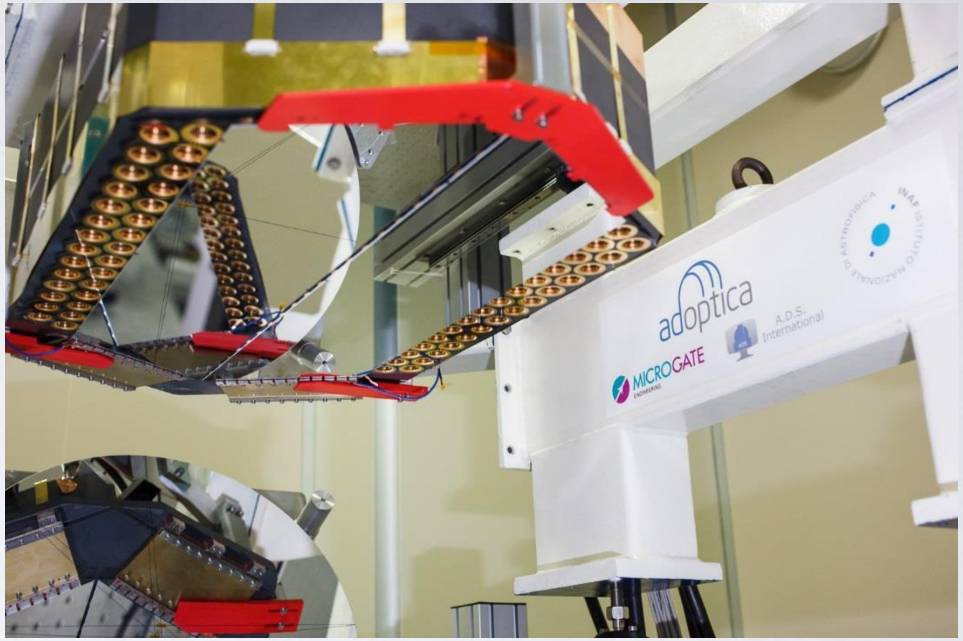
Still respectable performances

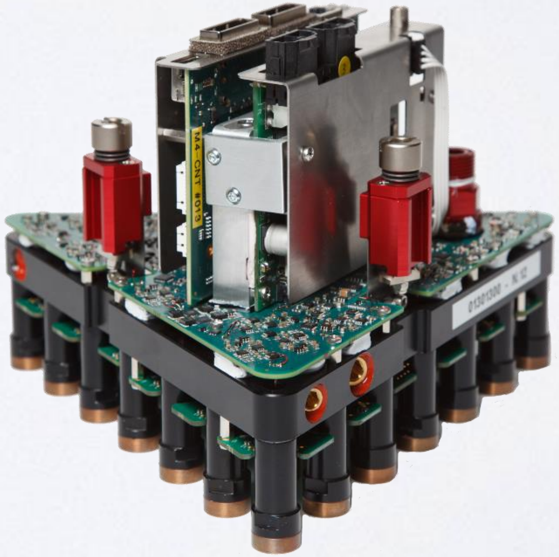
- Slope computer processing 1600 slopes with pyramid sensor (or SH, user configurable) fully pipelined → substantially zero (few µs) delay after last pixel acquired
- Real time reconstructor for 672 modes with multi-tap MIMO capability (beyond pure integrator) + mirror global control: total latency ~300µs
- Jitter in the few µs level ptv (mainly due to end-of-parallel processing check)

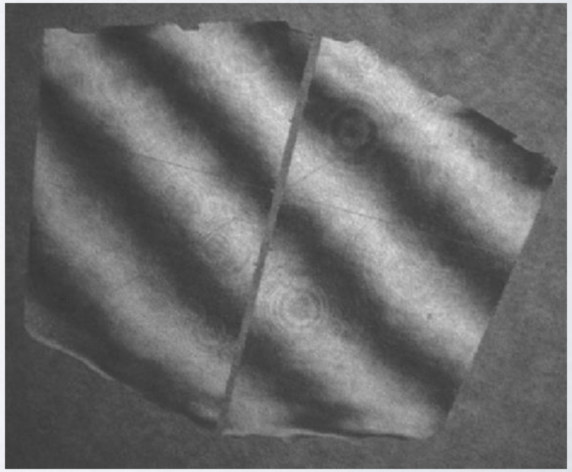
New generation: E-ELT M4, GMT

- Embedded control based on FPGA only (Arria V). Motivation:
  - Development roadmap: DSPs are becoming obsolete
  - Computational throughput vs. space
  - Computational throughput vs. power
  - FPGA needed anyway for HW interfacing
  - SoC concept (with NIOS soft-core)
- Performance:
  - Local control (quite complex!) latency ~300ns
  - Up to 36ch controlled by one FPGA @~80kHz, including redundant deterministic communication, housekeeping, safety, …
- Global control part moved away from embedded control system
  - Minimize on-board power consumption and complexity
  - System modularity









What is the *global control* of these mirrors?

- Interface to the telescope deterministic network
- Pre-processing of deterministic input from the telescope to generate mirror commands
  - Conversion from *modal* commands to *zonal* commands (actuator space)
  - Saturation control by adjusting the number of controlled modes (force-limited rather than stroke limited)
  - Safety checks
  - Computation of feedforward forces
- Interface to the telescope timing network (typically PTP)
  - Timestamping of all events
- Acquisition and formatting of real-time telemetry

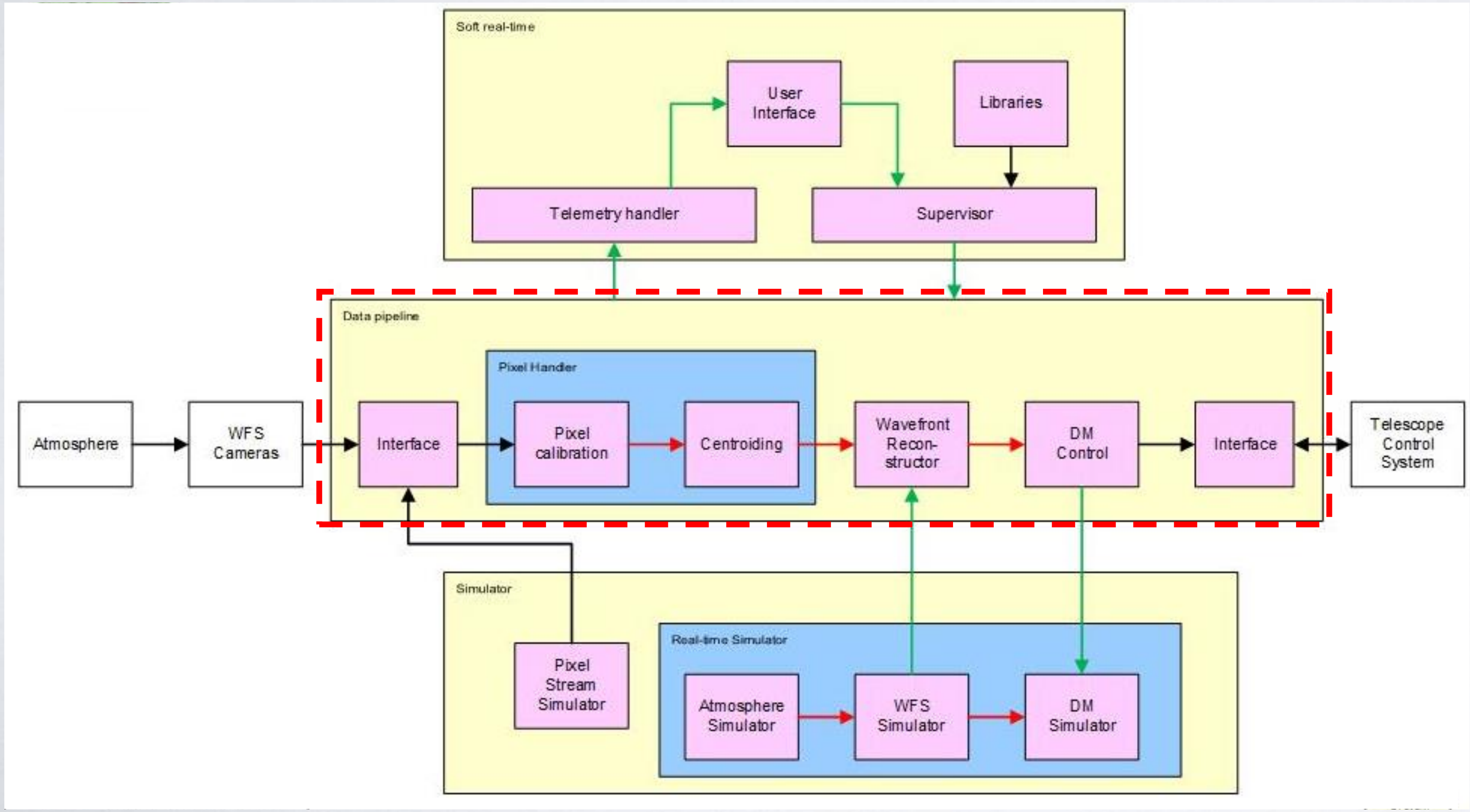
Quite demanding task: typical ELT case ~180 GMAC/s with ~10µs jitter

Technical solution

- Let’s be less exotic than we used to be before… PCIe boards
- But.. build our own ones!
- Motivations – why not to simply buy COTS:
  - Taylor the design to the specific AO needs
  - Own/manage the entire HW/FW/SW design
  - Support customers on long term - fundamental for Microgate! See past experience
  - Extend the application to other fields/systems in AO (RTC, WFSs, …) and possibly also in other markets: these units will become COTS!
  - Gain further expertise in the field – ok, it still makes fun!
  - (Greenflash: additional stimulus + guideline for development + interaction with the group partners + QuickPlay)

The GreenFlash ecosystem: MicroServer concept

Within GreenFlash, Microgate will develop a solution for the hard real-time data pipeline with FPGA based boards for realizing a **stackable, energy efficient microserver** for data-intensive applications.

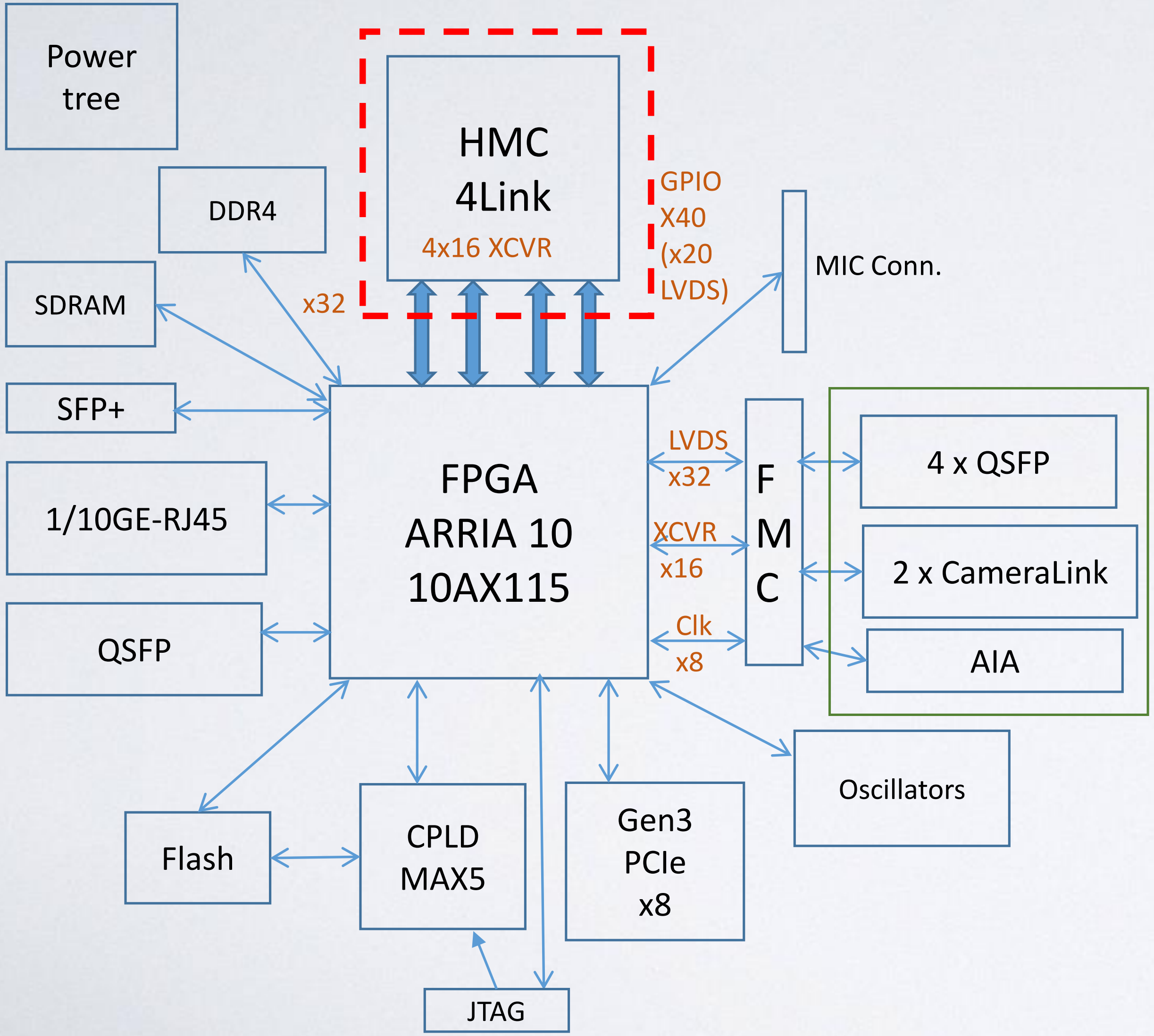


Green Flash

l'Observatoire de Paris
Durham University
MICROGATE
PLDA

MICROGATE

µXComp board

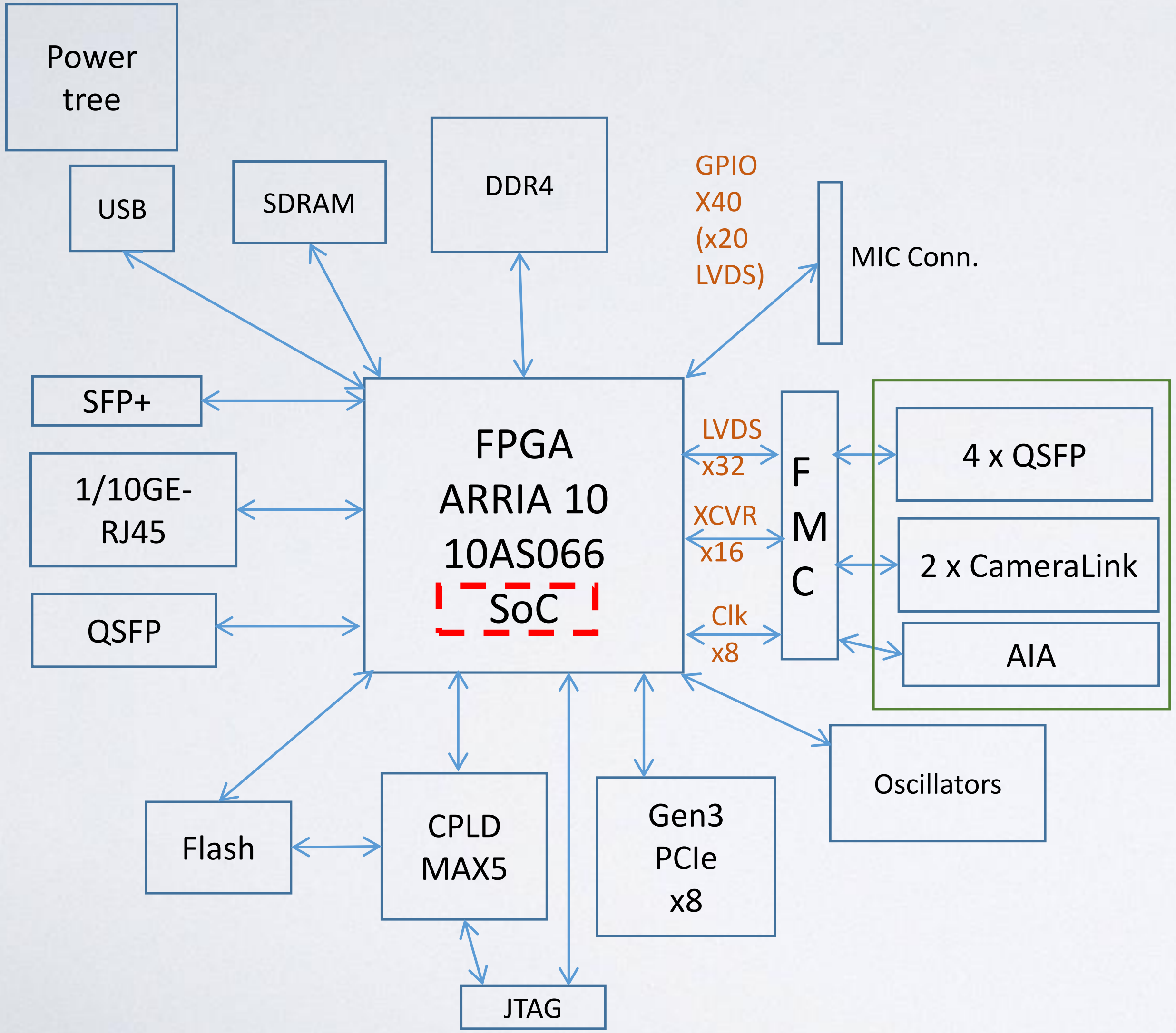


Computational-tailored board

- Based on ARRIA 10AX115:
  - 1518 DSP blocks
  - 6.6MB int. RAM
  - 96 XCVR
- Board features:
  - Optimized for heavy deterministic computation in floating-point
  - Large Bandwidth between HMC and FPGA - 4 links 16 lanes/link up to 15Gbps/lane = 120GB/s bidirectional
  - Extremely low jitter
  - More power efficient compared to GPUs
  - Offers a lot of different interfaces on board or via the FMC connector and extension cards e.g. CameraLink, S-FPDP

MICROGATE

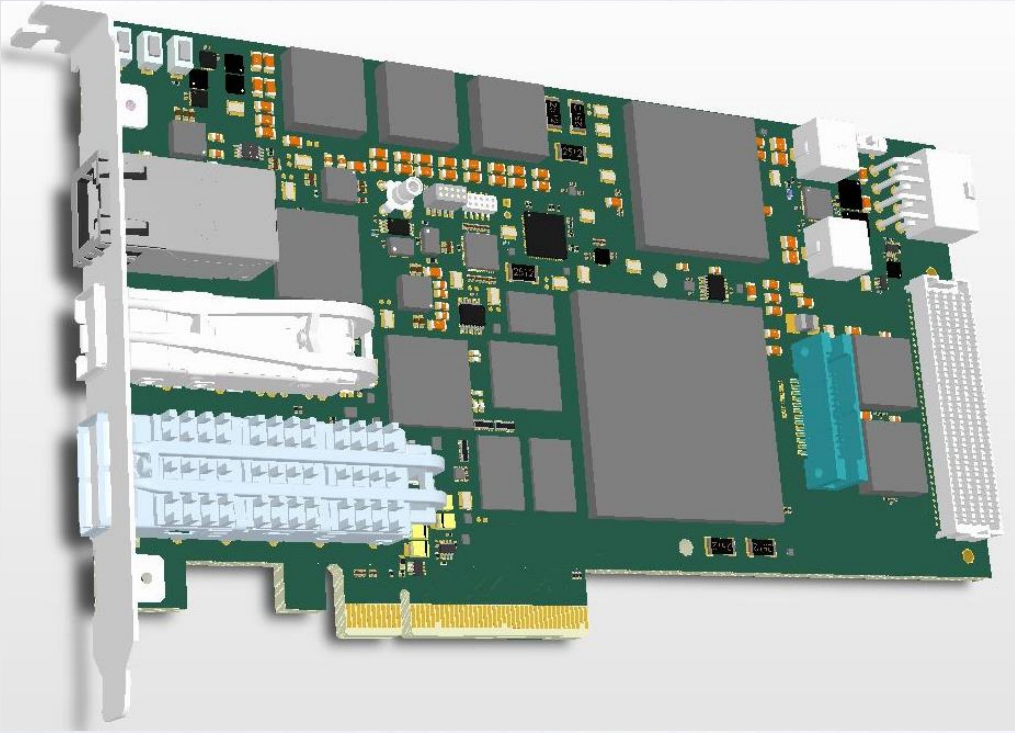
µXLink board



Microserver/interface board

- Based on ARRIA 10AS066 SoC:
  - 1.5GHz ARM dual-core
  - Cortex-A9 on-chip processor
  - 1855 DSP blocks
  - 5.2MB int. RAM
  - max. 48 XCVR
- Board features:
  - ARM embedded processor for stand-alone real-time box
  - Powerful PCIe root port because of ARM and OS
  - Management of accelerator cards on the PCIe interface
  - Running control software using a full OS (e.g. Linux)
  - 10/40 GbE + Infiniband
  - Easy implementation of different communication protocols
  - Offers a lot of different interfaces on board or via the FMC connector and extension cards

µXComp features

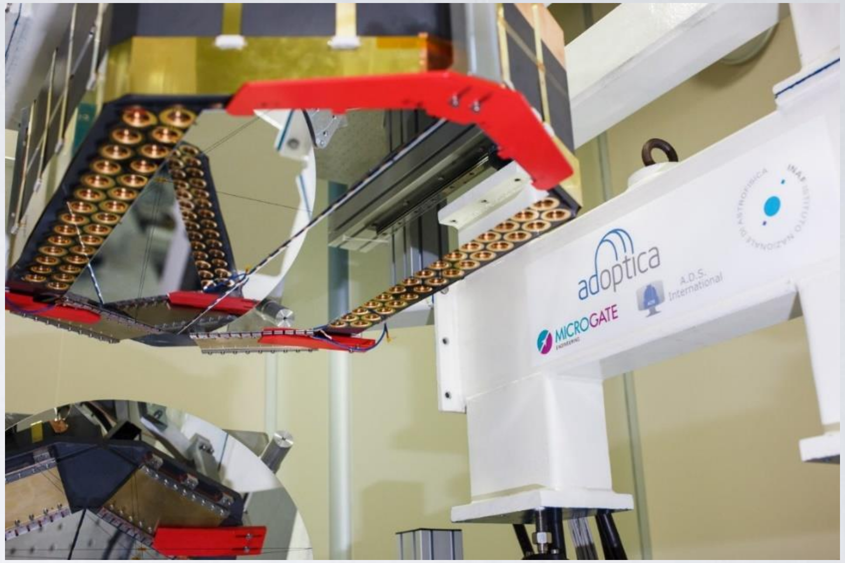


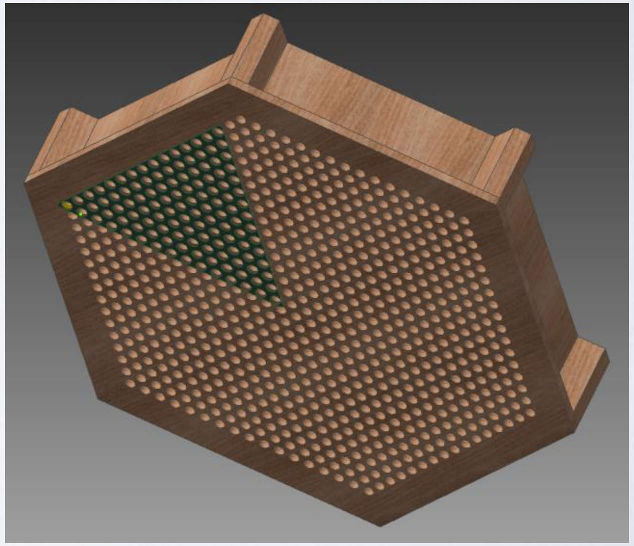
Features

- Board size (111x200 mm) compliant with PCIe standard single slots full height and >= ¾ length PCIe up to x8 Gen 3 (64Gb/s each direction)
- Front-panel interfaces: 1/10Gb Ethernet via fiber (SFP+) and via copper (RJ45), 40Gb Ethernet or Infiniband (QSFP)
- FMC standard extension boards attachable on the back (up to length 130mm to fit in a PCIe full length slot (320mm)
- **Sustained** performance 30 GMAC/s (floating-point) considering the memory transfer rate of 120GByte/s each direction
- Peak performance in the TMAC/s range
- **Compatible with QuickPlay** – systematic interaction with PLDA to add this board to the supported portfolio

µXComp status

- Several delays due to Arria 10 availability, power tree, PCB manufacturing accident…
- Prototype manufacturing ongoing – **ready before end 2016**
- Power tree already tested on a partially assembled prototype
- FW/SW already developed and partially tested on Altera dev board:
  - PCIe + Linux device driver – tested!
  - 10GbE – tested!
  - PTP – tested!
  - S-FPDP-like interface to the mirror – tested!
  - Our typical ‘global control’ configuration with~5000 input modes and 222 outputs with telescope interface – tested!
    full pipeline including data transfer executes in < 200µs
  - HMC interface – implemented and simulated
- Complete HW/FW testing in very **early 2017**
- Plan to test with a full M4 petal emulator in Q3/2017

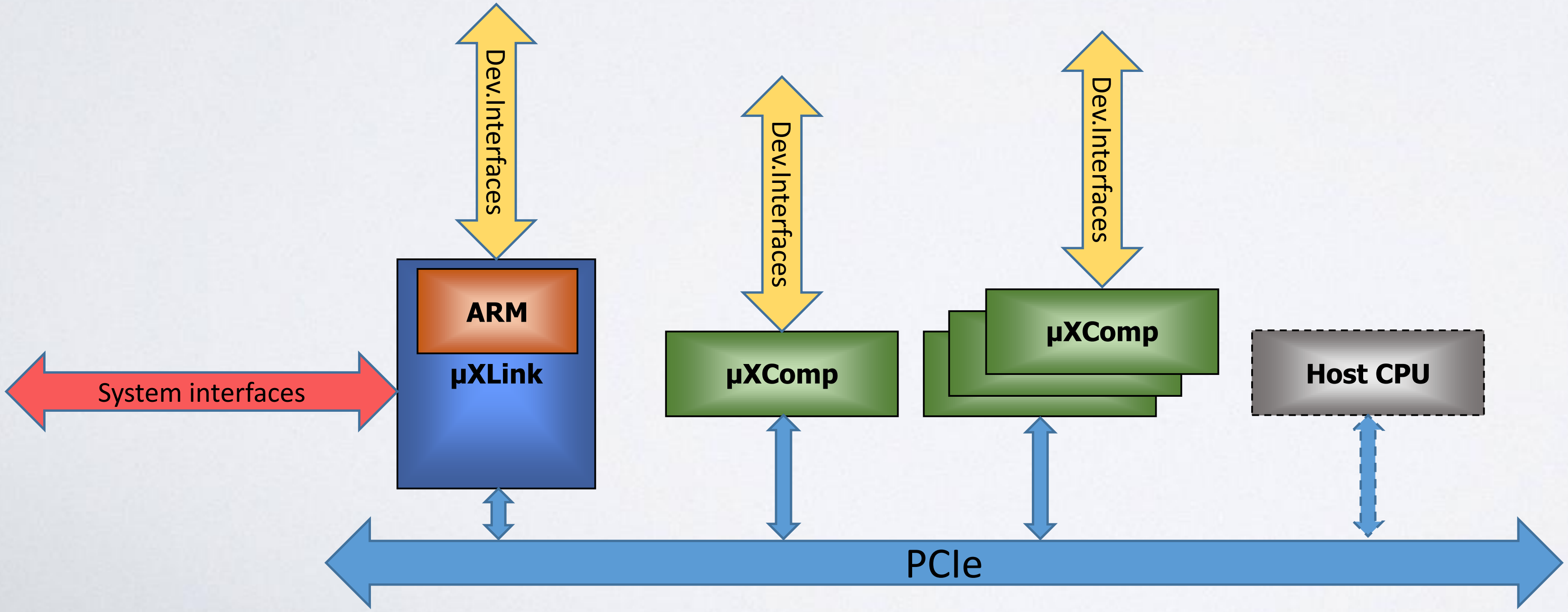




Microserver development plan

µXLink + Microserver system development plan

- µXLink development starts early 2017 – strong leverage of existing experience
- µXLink prototyping and testing Q2/beg. Q3 2017
- Proceed afterwards with Microserver system integration: µXLink (acting as PCIe root complex) + several µXComp boards. Host CPU not needed!
- Application (and market) ready in 2018



Limits of the FPGAs and some hint

- Development time
  - Need an FPGA expert (HDL needs HW comprehension, is not pure SW, timing constrains, simulation, ...)
  - All this may be abstracted but not completely solved by high level tools like QuickPlay. **Long compile/fitting time still there! Try and correct philosophy not very efficient**
- Key aspect (as before with DSPs...): design a **very flexible FPGA configuration**, capable of covering a variety of problems around the main one; configurations and parameters loaded at runtime

Summary and future goals

- The FPGAs are still a valuable solution to target medium computational complexity, deterministic timing, flexible interconnect
- Powerful not only for MVM-like heavy parallel computation but also to implement other algorithms thanks to their flexibility
- High level tools (QuickPlay) are very promising – still to be fully evaluated and validated by implementing real, complex cases
- Development time anyway not to be underestimated

Future goals

- MicroServer system compatibility with other accelerators, in particular GPUs - still more powerful in terms of brute computational power. Is GPUDirect possible? (GreenFlash goal)
- Stratix 10 is a very appealing device for a next generation
  - Very powerful SoC (Quad-core ARM Cortex-A53)
  - Embedded HMC – outstanding memory transfer rate
  - Cost?

MICROGATE