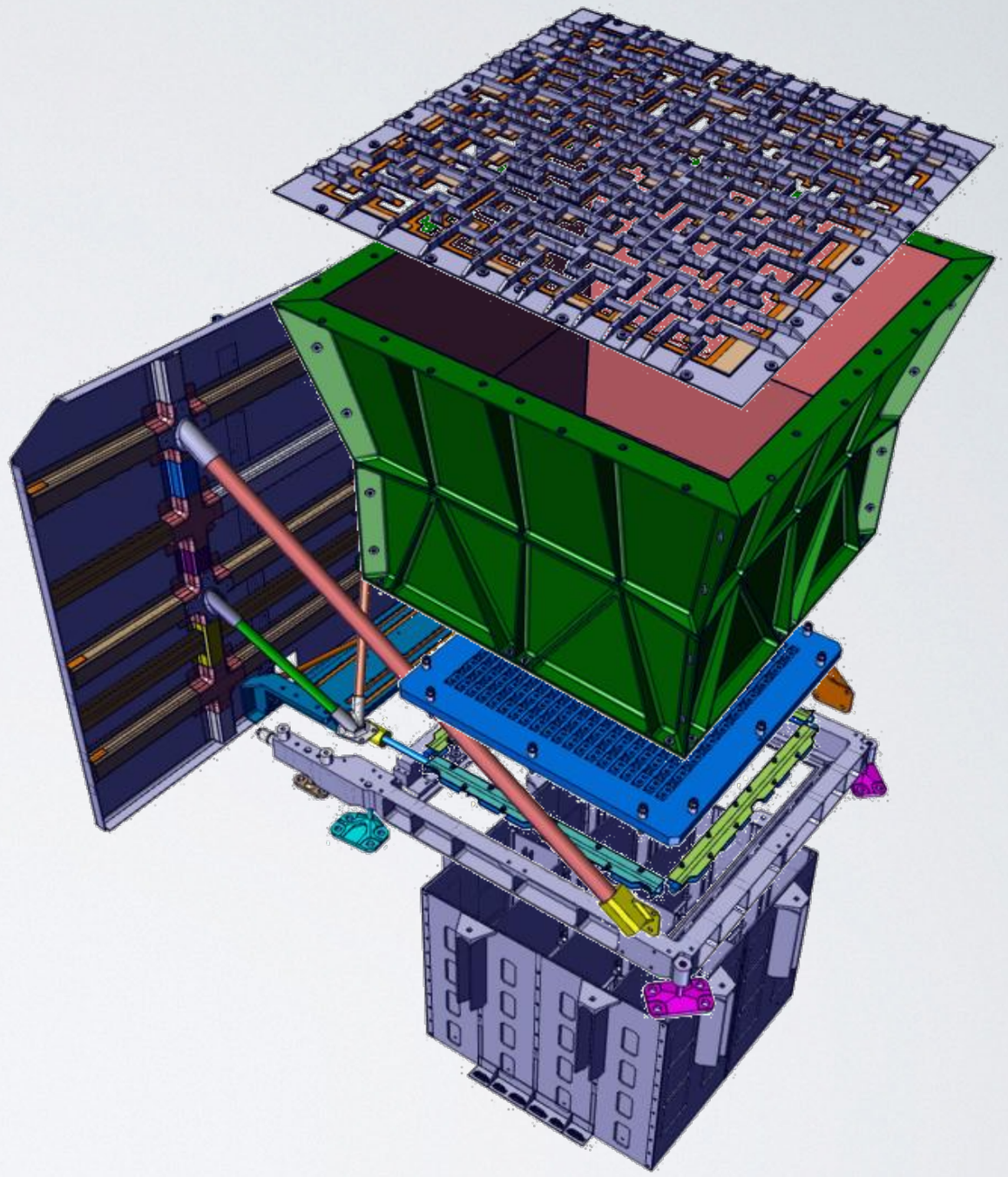
Space FPGA users Workshop
April 9th-11th 2018, ESA Noordwijk

USE OF FPGAS IN A SCIENTIFIC INSTRUMENT DEVELOPMENT

Damien Rambaud
IRAP CNRS

OUTLINE

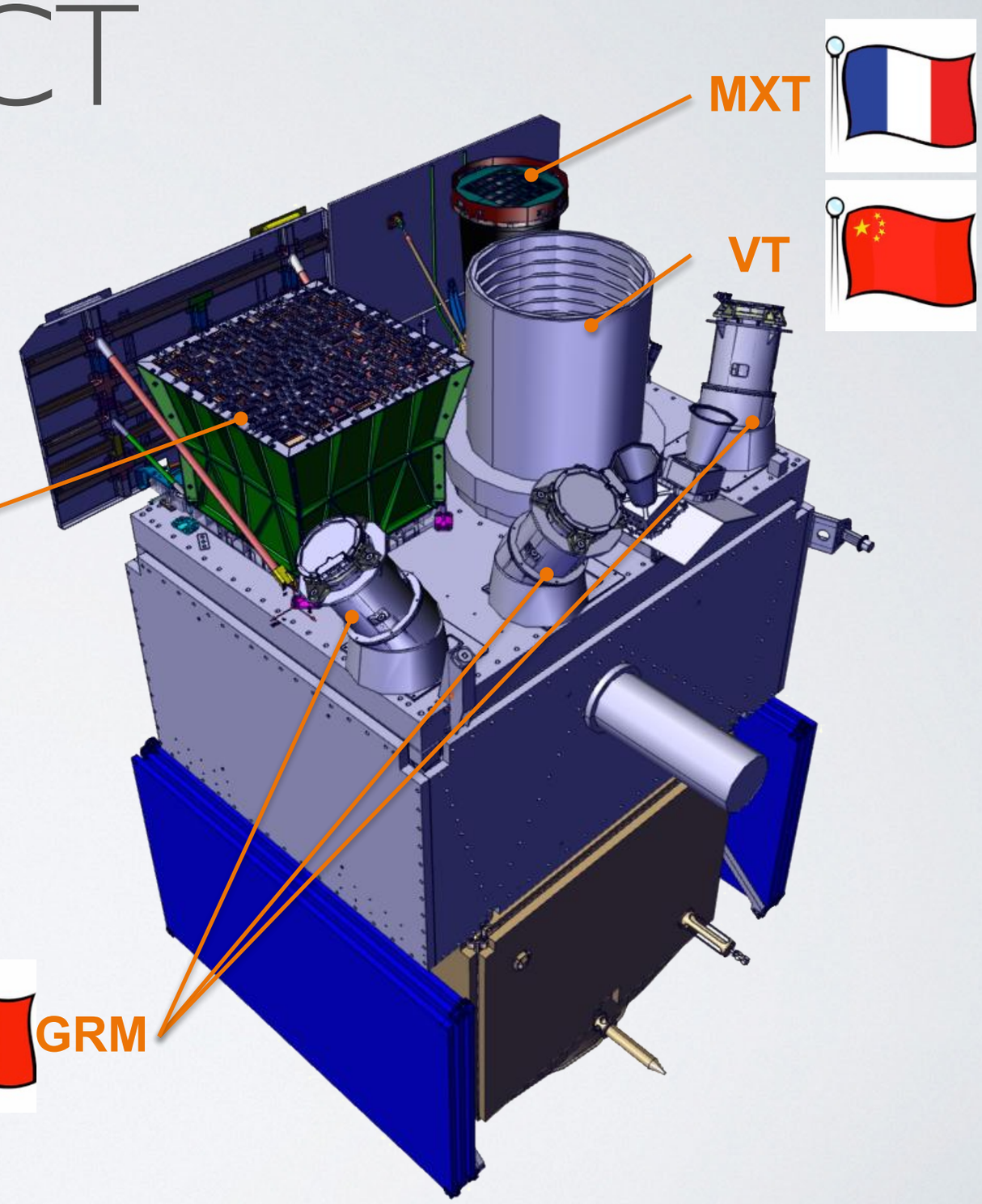
- SVOM / Eclair project
- FPGA needs in each parts of the project
- NX-Medium experience
- Athena X-IFU example
- Conclusion



SVOM ECLAIRS

SVOM PROJECT

SVOM is a French-Chinese astronomy mission to detect gamma-ray bursts generated by the explosion of massive stars or the merger of neutron stars or black holes.

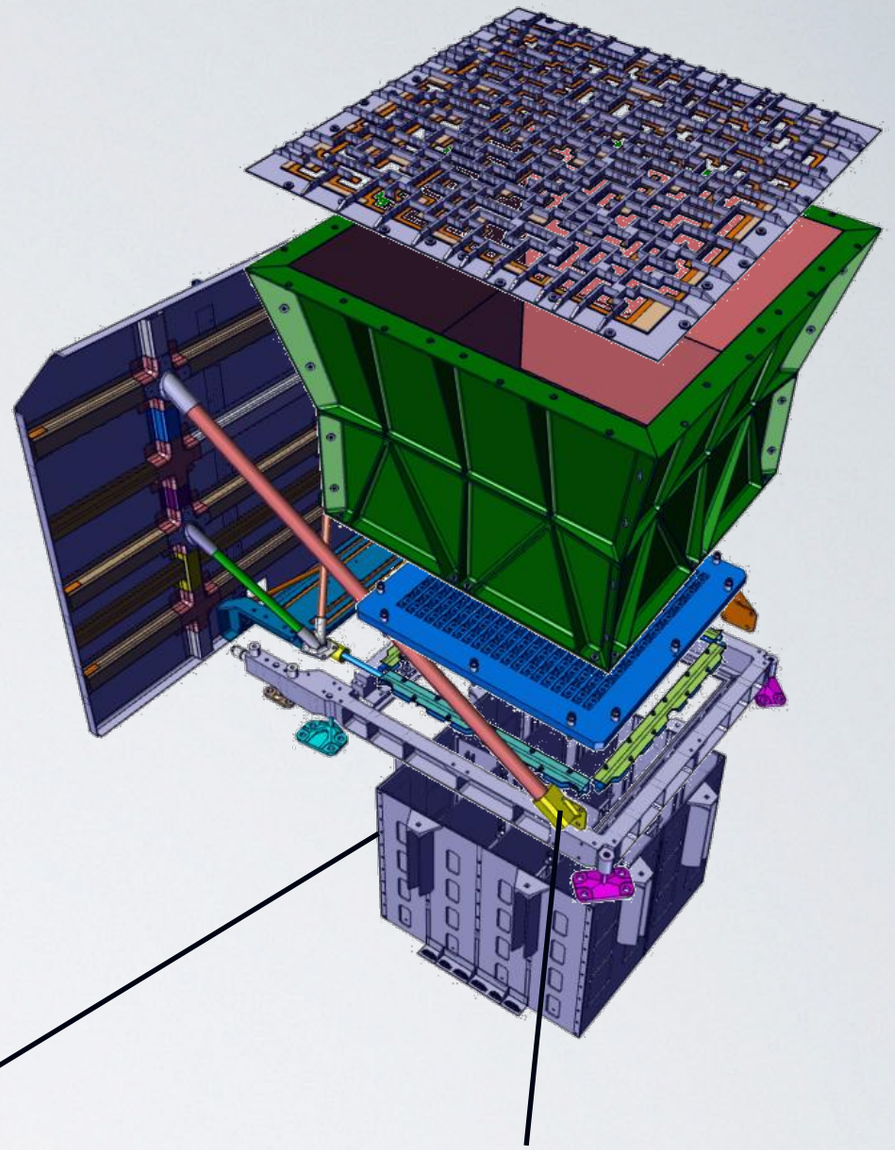


ECLAIRs

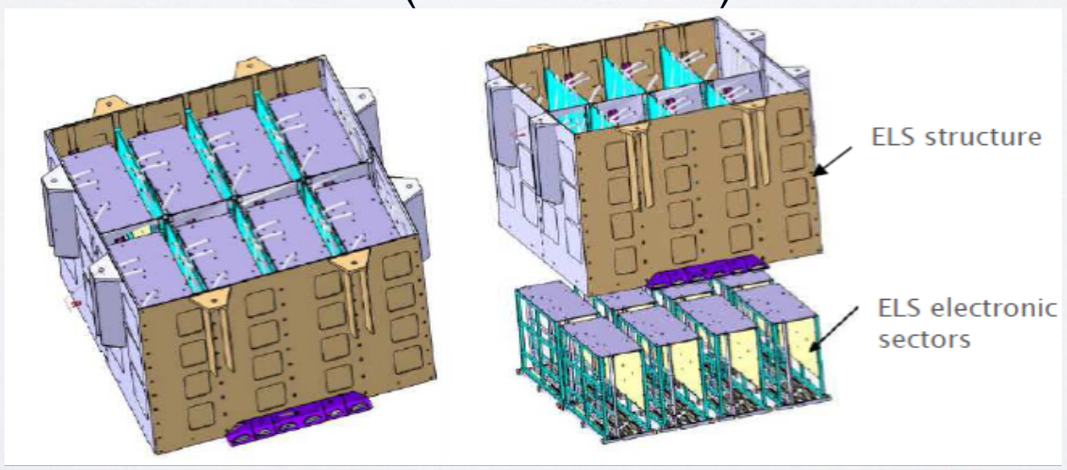
- One of the 4 instruments of the SVOM Mission
- Developed and provided by French teams
- Dedicated to GRBs detection and early alert

SVOM / ECLAIRS

- IRAP is in charge of the detection plane and Front-end electronic of the Eclairs instrument
- The instrument will measure energy between 4 and 150KeV.
- 1000 cm2 detection plane
- 6400 CdTe detectors
- 200 Asics managed by 8 FPGA

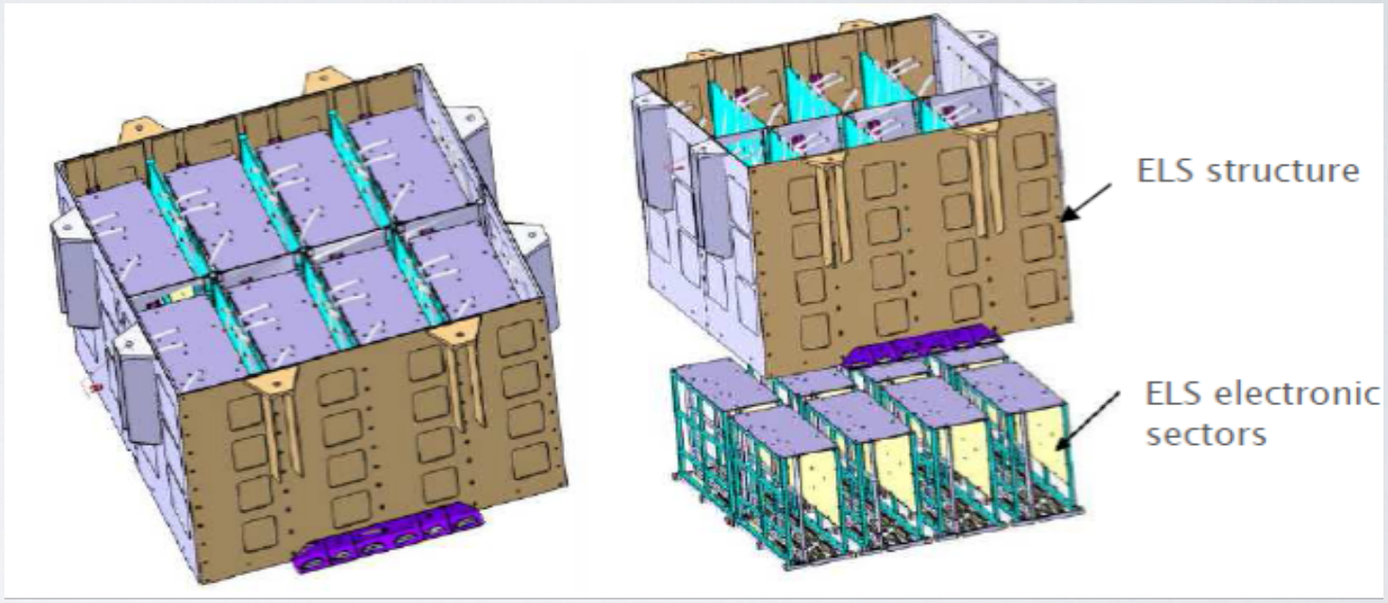


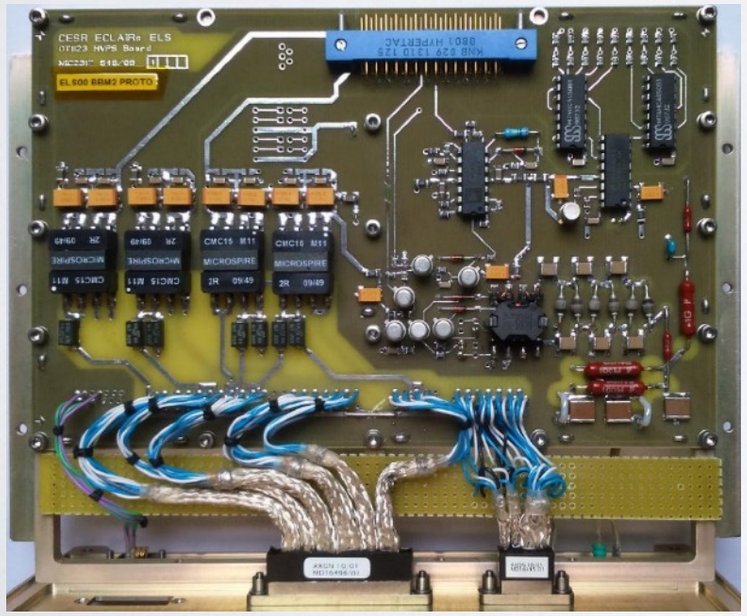
8 Front-end electronic boxes : ELS
(FPGA inside !)

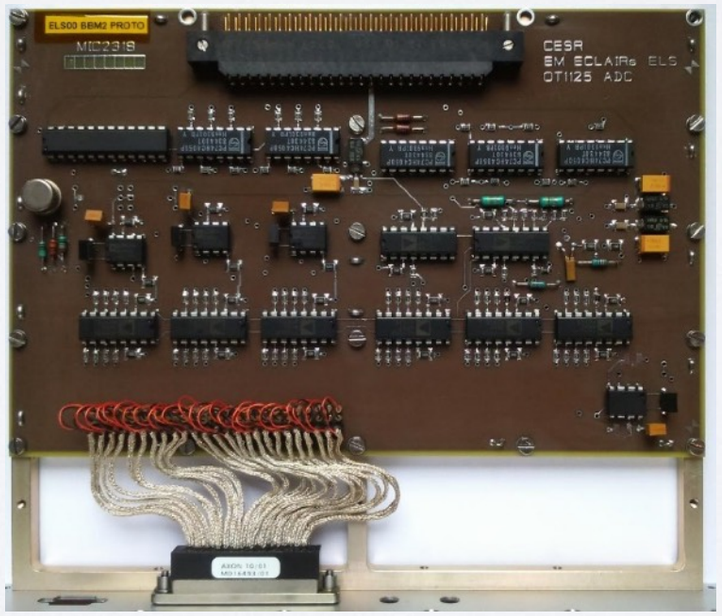


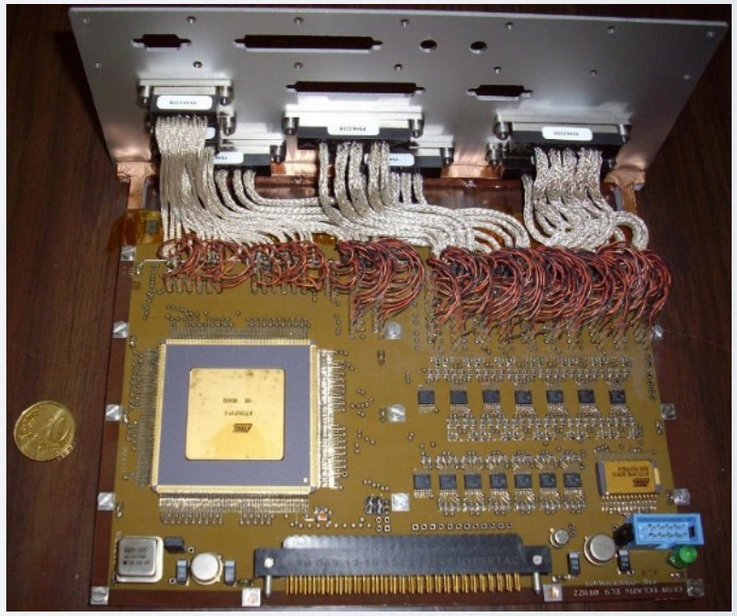
ELS BOX

- Detection plan read-out and command / Control
- Divided into 8 sectors
- For each sector :
  - LVPS/HVPS board (0 à -500V)
  - MUX/ADC (12 bits) board
  - Processing board (FPGA)
  - Backplane board









PEOPLE INVOLVED IN DEVELOPMENT

- Scientists
  - Need to be able to simulate the instrument as precise as possible
  - Need to control the instrument for calibration

- Software engineers
  - Need simulators to validate software (OBSW or EGSE)
  - Need interface board to control EM QM, FM

- Electronic engineers
  - Use flight FPGA to build the instrument
  - Need breadboards with engineering models
  - Need interface board to control flight instrument and validate flight model

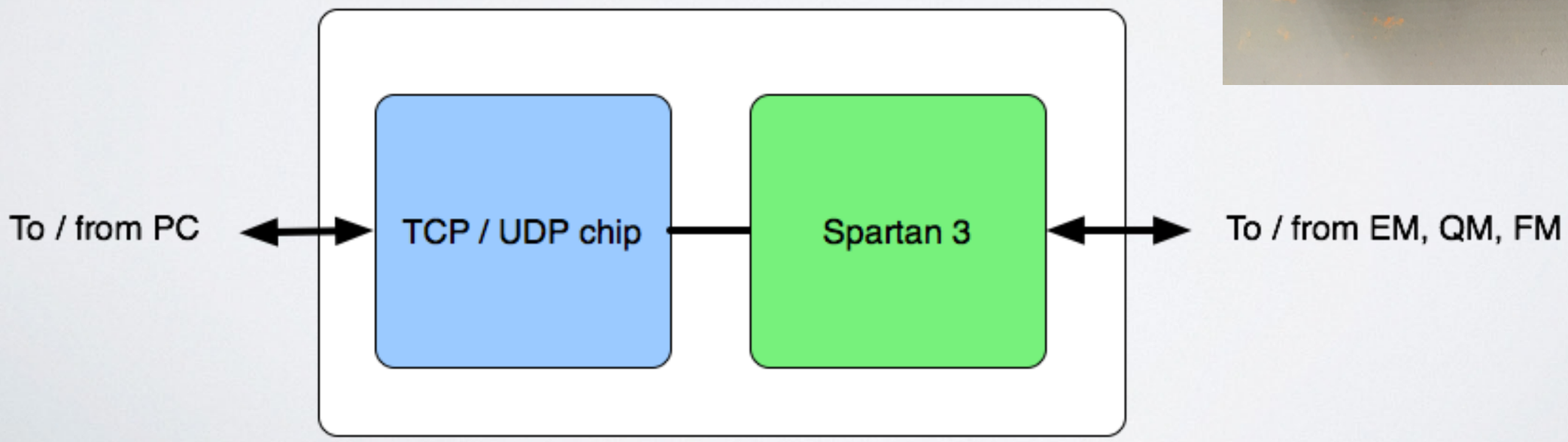
- Instrumentalist engineers
  - Need interface board to control EM QM, FM

FPGA NEEDS IN A PROJECT

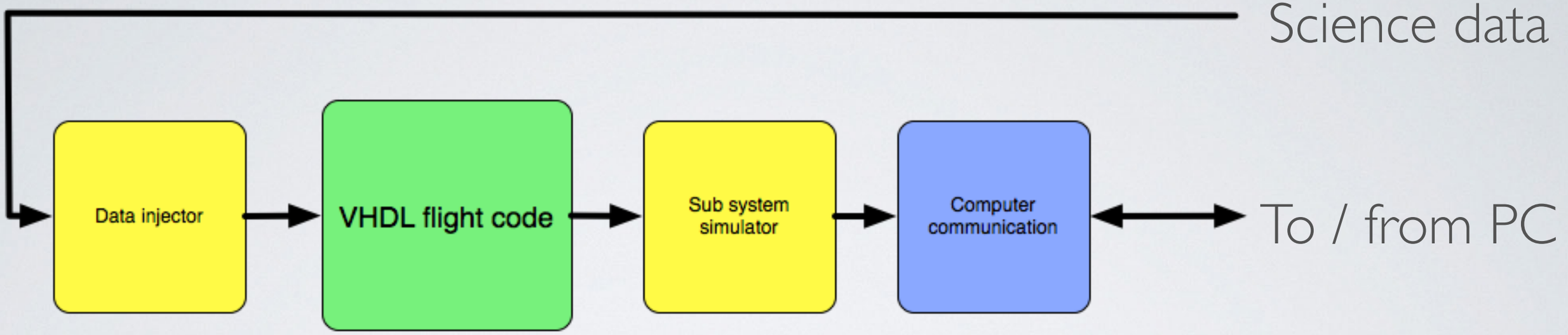
Where :	Instrument interfacing	Simulator	EM, QM, FM development
Scientists	×	×	
Electronic engineers	×		×
Software engineers	×	×	
Instrumentalist engienners	×		

INSTRUMENT INTERFACING

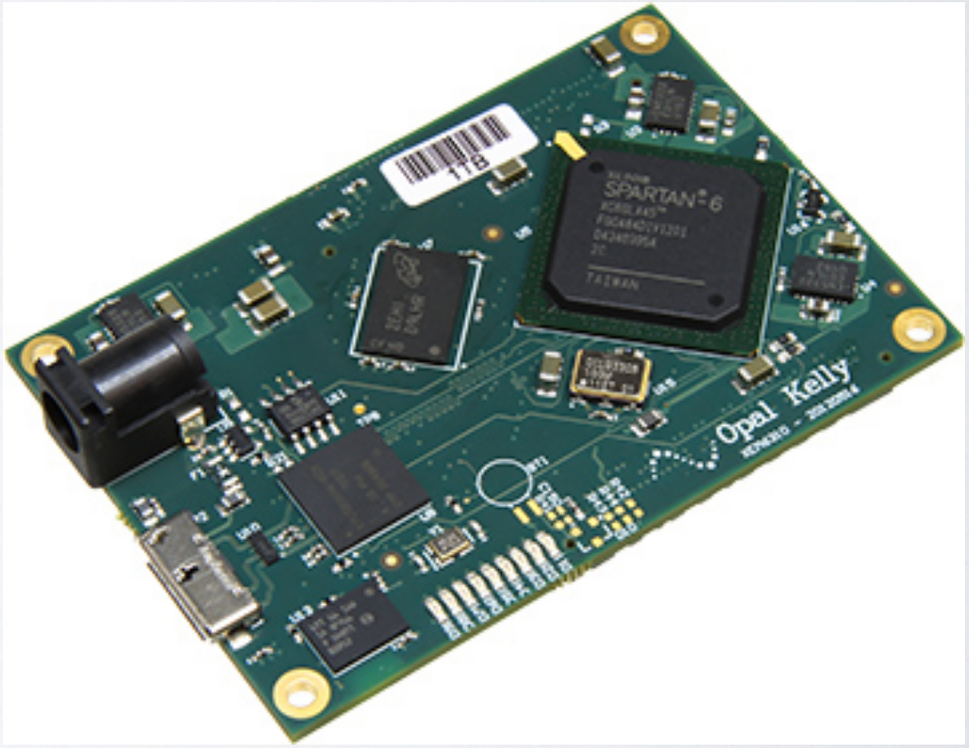
- Useful to control EM, QM, FM
- Not too expensive
- Must be easily duplicable



SIMULATOR

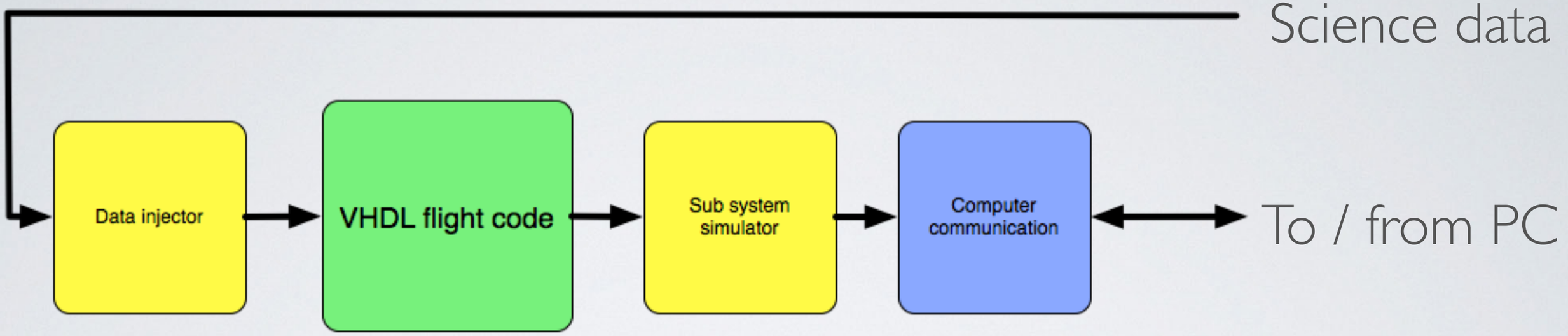


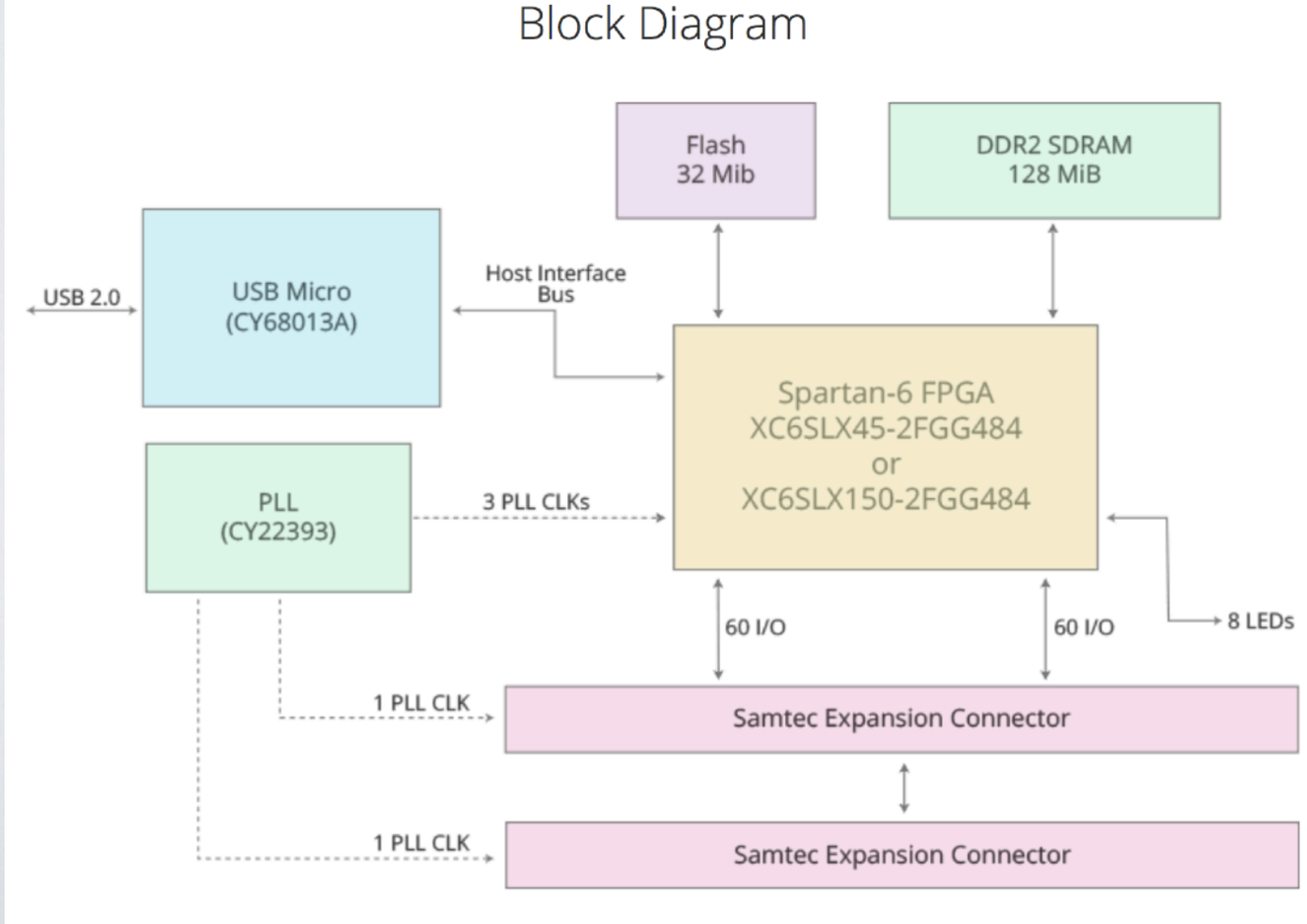
- Help in EGSE development by providing instrument simulation
- Provides a convenient way to simulate the instrument
- Be as representative as possible
- If possible, not too expensive
- This allows scientist to check the performances of the digital part very easily
- Easy to use

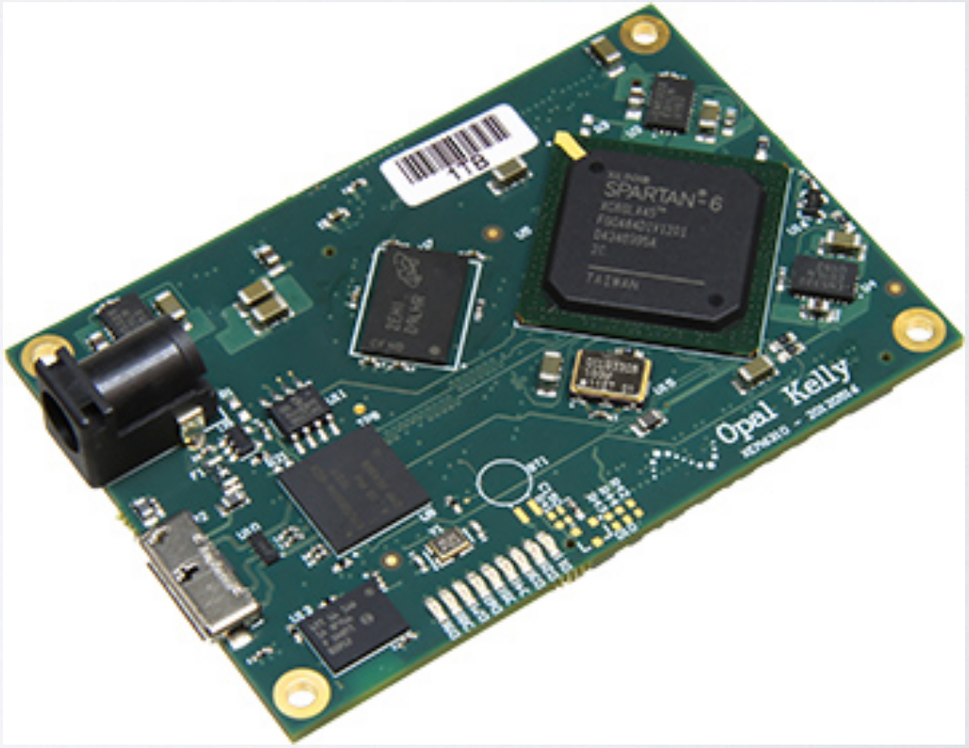


Opal Kelly XEM6010

SIMULATOR



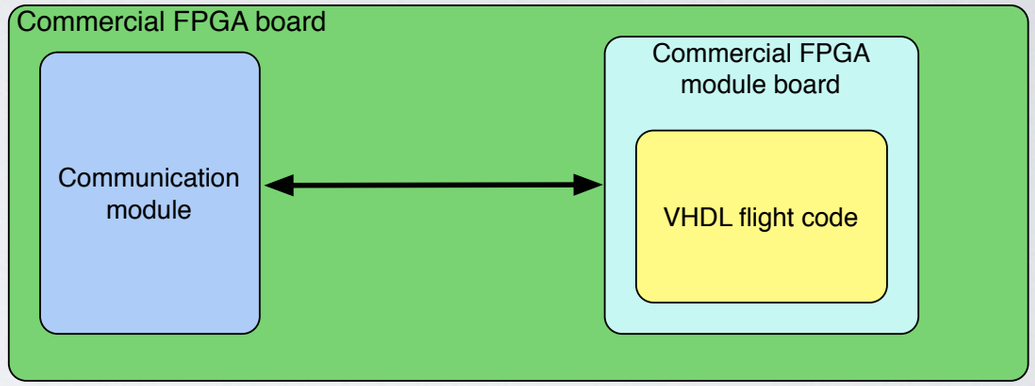


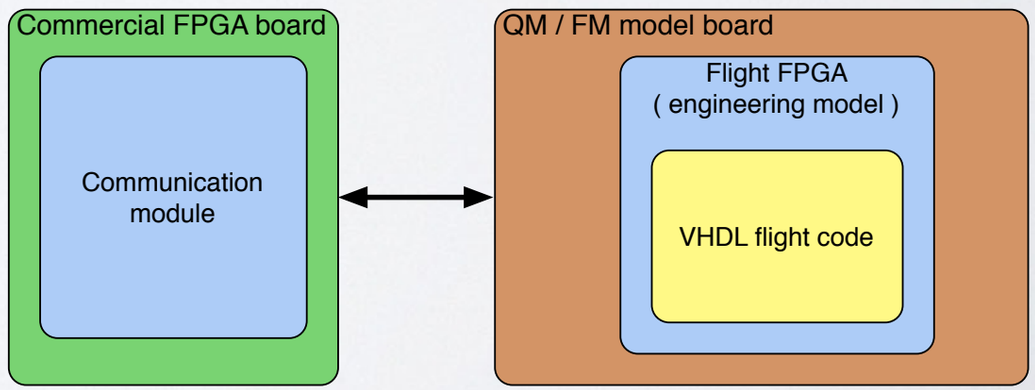


Opal Kelly XEM6010

FLIGHT INSTRUMENT DEVELOPMENT

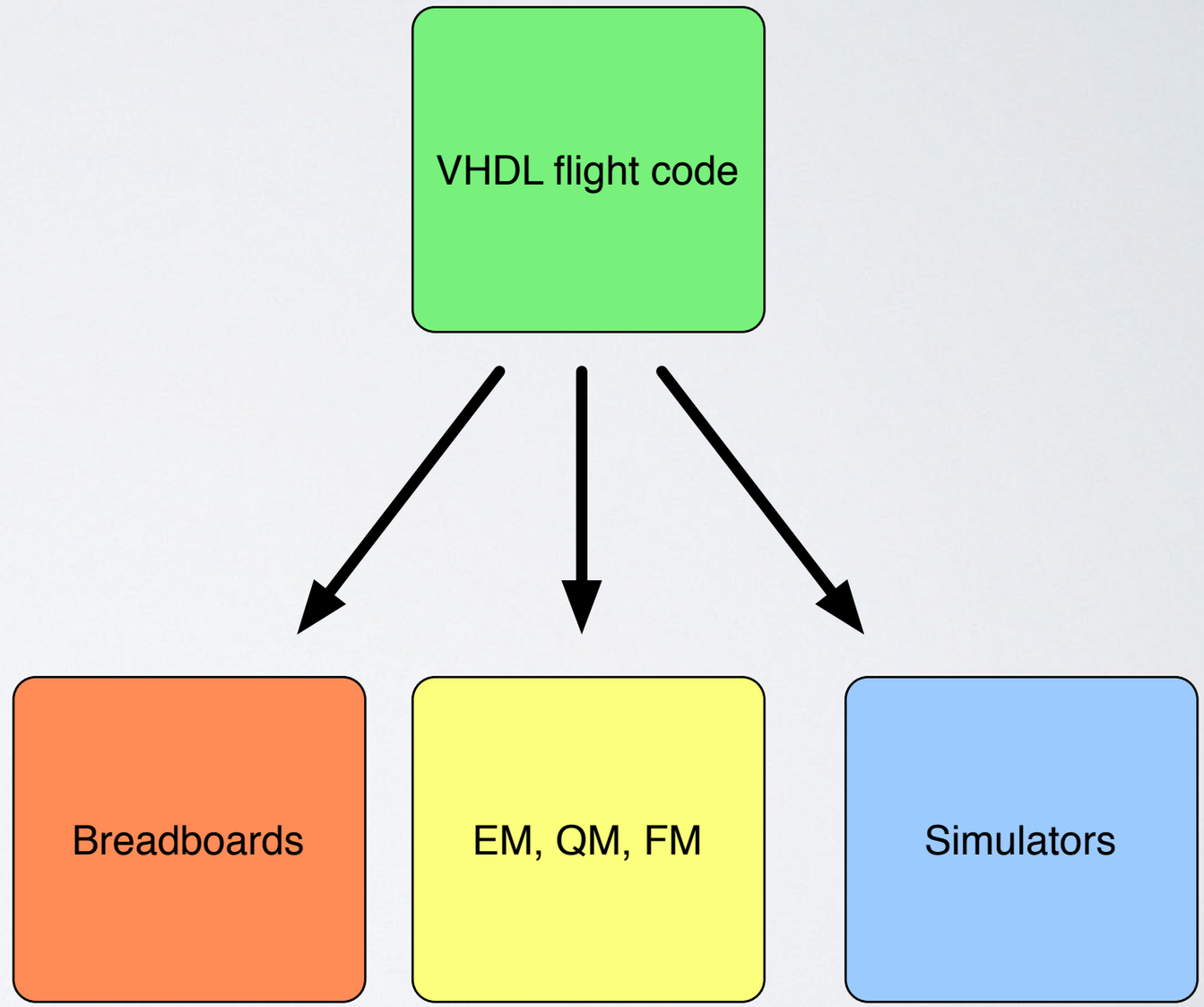
- Breadboard, EM : use of cheap commercial devices or FPGA engineering models
- QM, FM : use of in different grades of final target
- Anyway this implies the use of different types of FPGA…



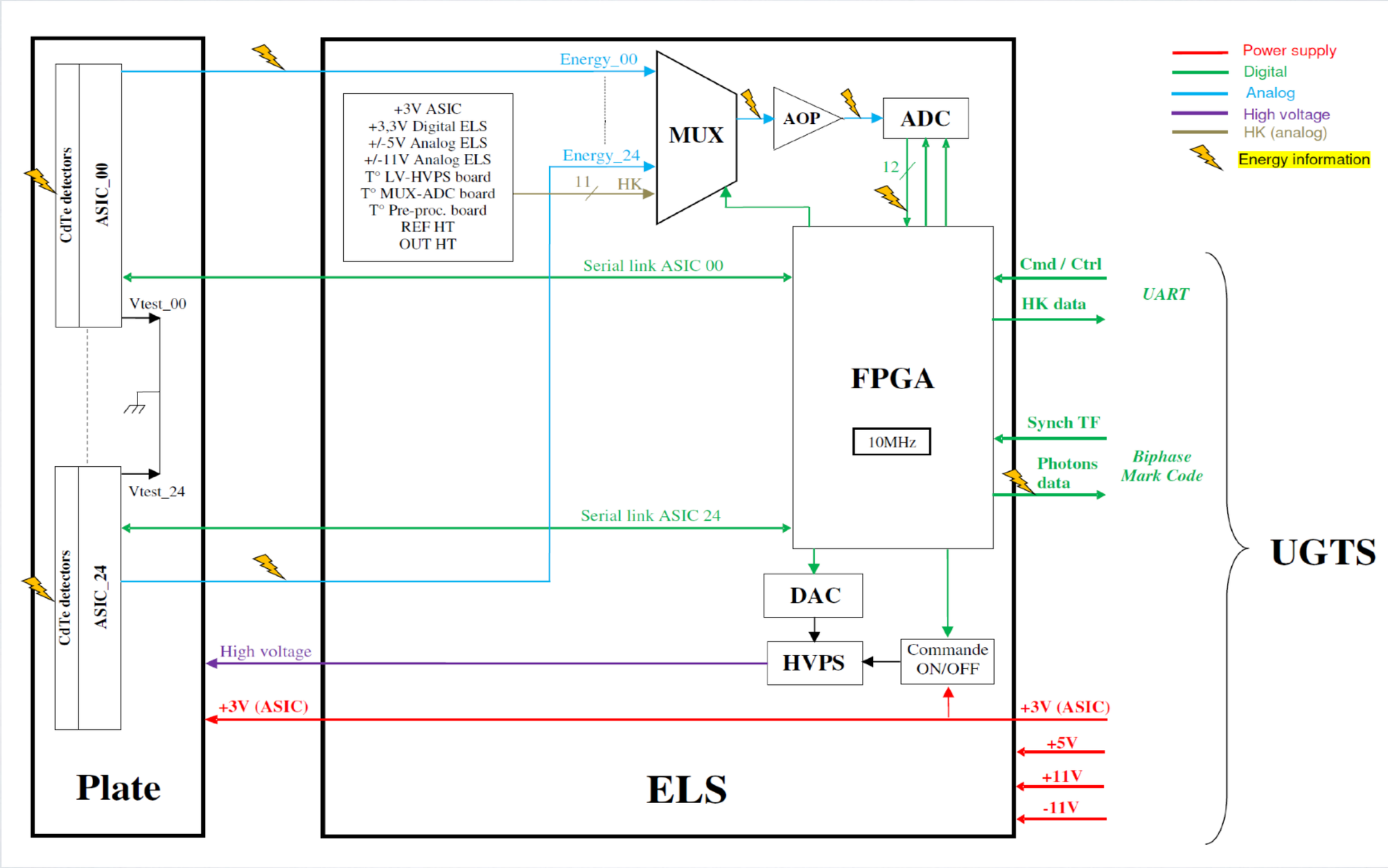


CONSTRAINTS

- All these developments need more than one type of FPGA (Xilinx, Nanoxplore, Atmel…)
- VHDL flight code needs to be synthesized on different targets
- When possible, use coding rules that ease portability
- If possible, build breadboards based on the flight chip

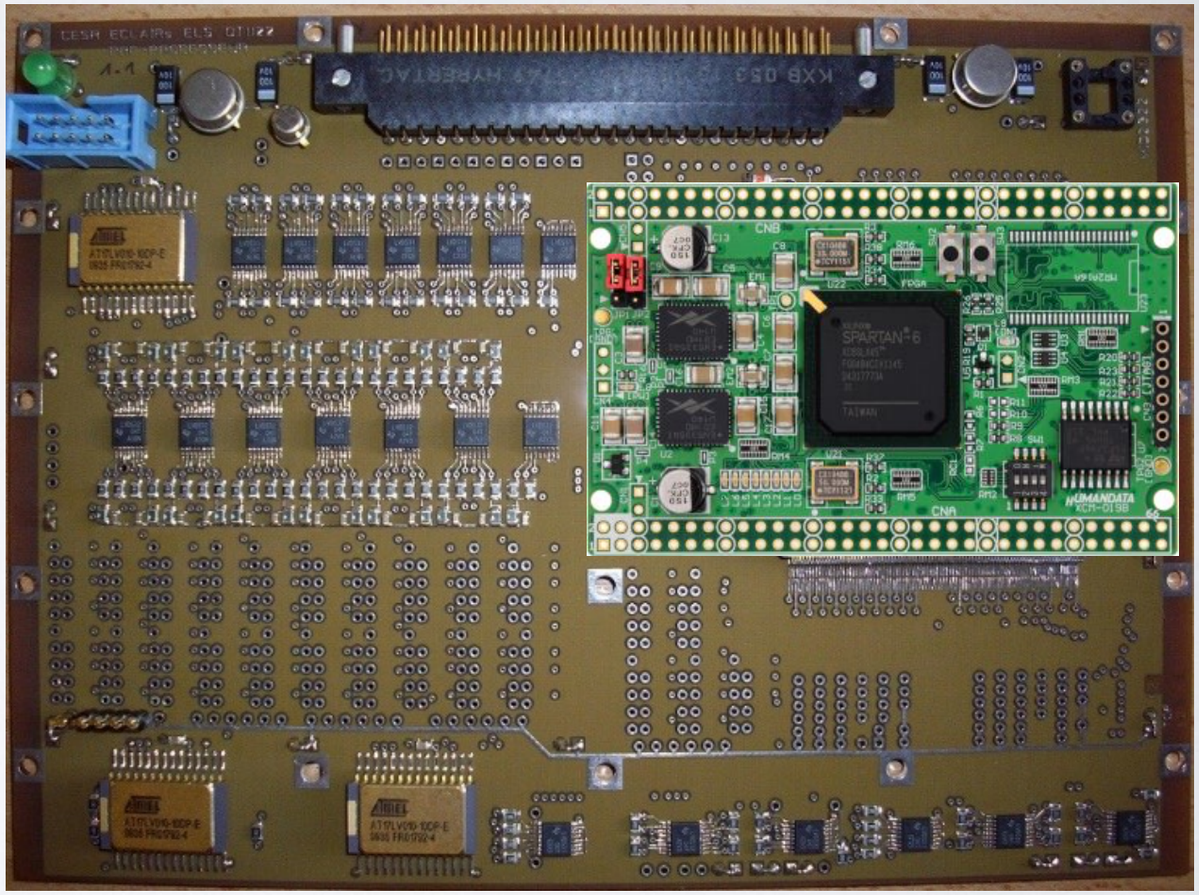


ELS ARCHITECTURE



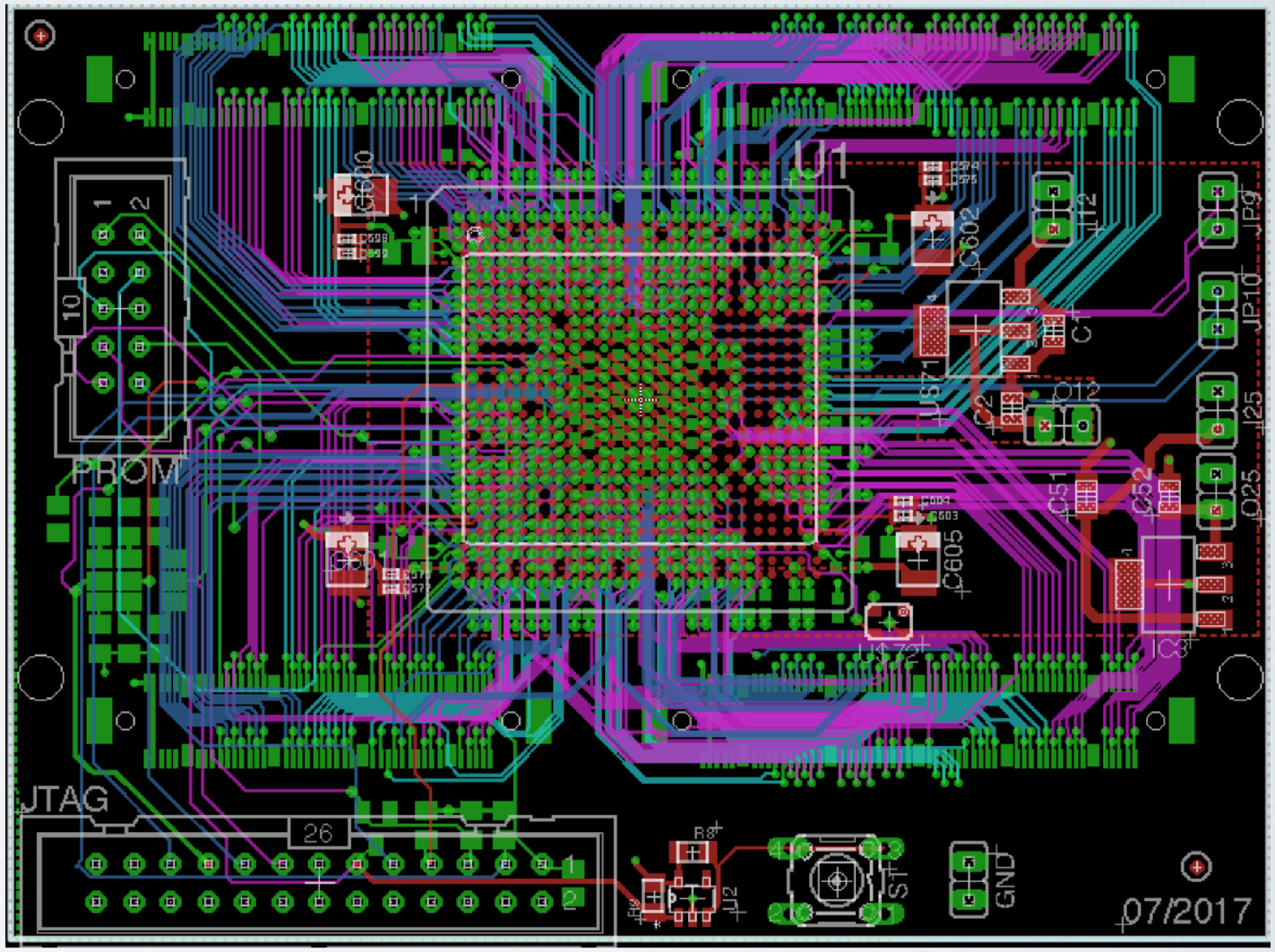
SVOM ECLAIRS CONSTRAINTS

- We can't us any US device
- Until last year, FPGA was ATF280F
- Unfortunately, implementation of our design as not been possible due to the synthesizer performances
- We tried several changes without success
- To be able to continue to work on other subsytems, we replace the ATF280 by commercial FPGA module from Humandata (spartan-6 based)
- In January 2017 we start thinking about digital ASIC
- In April 2017 we choose to switch to NX-MEDIUM

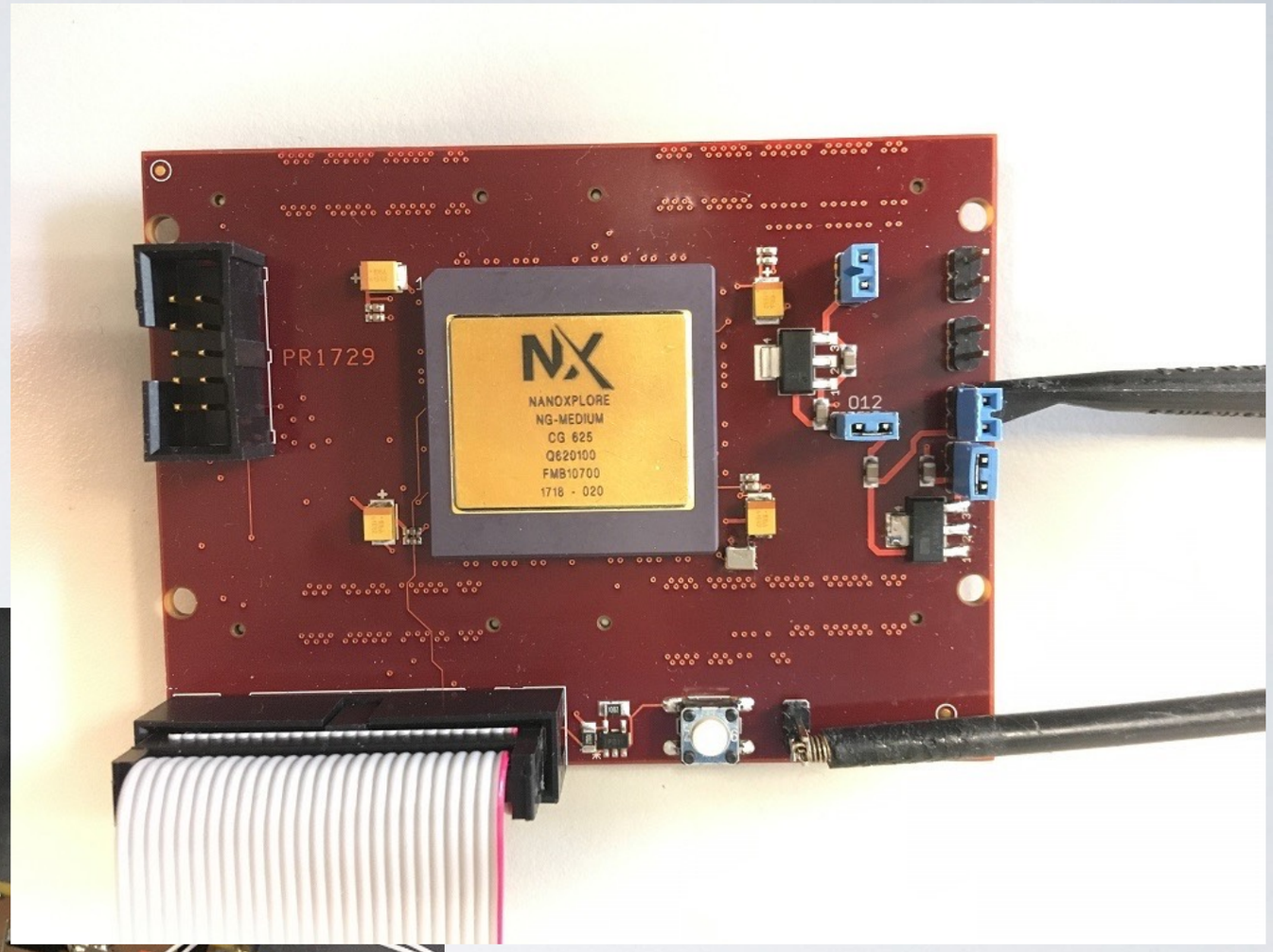


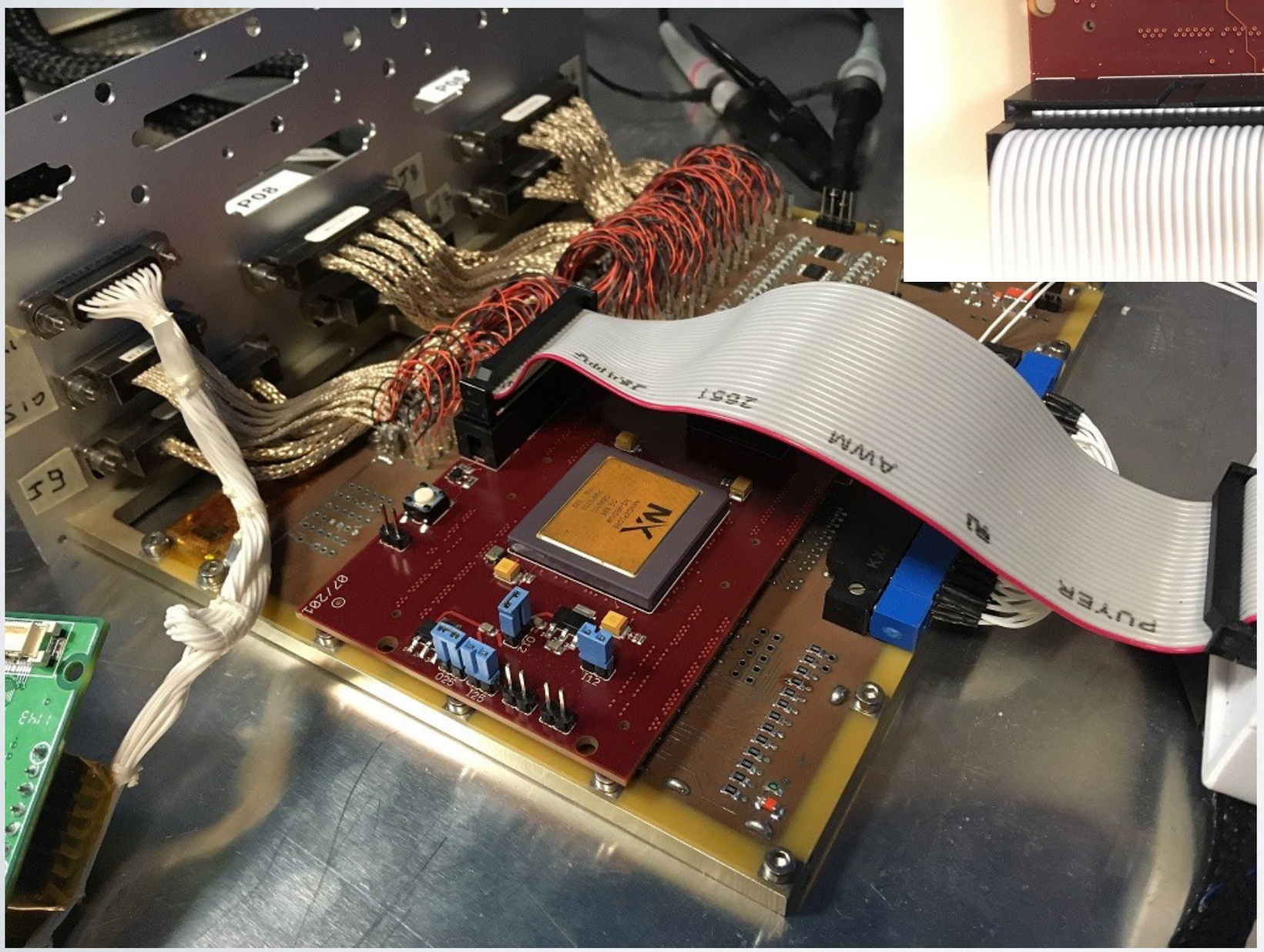
NX BREADBOARD

- Due to the problems with ATF280, we have a board that uses spartan-6 FPGA
- This board was designed as a replacement of the spartan-6 existing board
- It was designed with active support of NanoXplore



NG-MEDIUM BREADBOARD





SYNTHESIS

	Xilinx (ProtoDpix) Spartan 6	ATMEL ATF280	NanoXplore brave NG-MEDIUM
Occupation rate	6%	70% with 70% of features	38%
Bitstream generation time	20 min	4 h	7 min
Real life test	yes	yes	yes
Power supplies	-	3.3V, 1.8V	3.3V, 2.5V, 1.2V
			3.3V 0,75W 2,5V 0,4W 1,2V 0,12W **Total : 1,27W**
Packaging	-	QPF352	QPF352

NG-MEDIUM EXPERIENCE

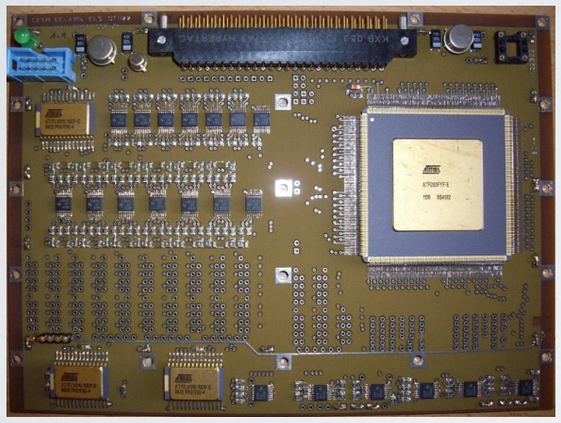
- Our configuration was : Ubuntu 14.04.2 LTS, nanoXmap 2.7.3 for NG-Medium
- Bitstream generation takes **7 minutes** for our design. It takes 38% of the FPGA.
- Python script is very convenient
- NX bitstream has been converted to be used with the Atmel space programmer
- NX-Medium is now loaded by the Atmel flash PROM
- **Our design has been successfully ported on the NX-MEDIUM**

DEVELOPMENTS

NetPix Board : Is able to communicate with the instrument

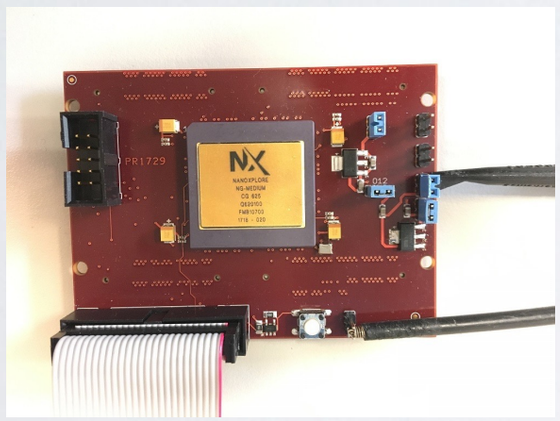
Base on custom design (works with ethernet)

Used by Electronic engineers, instrumentalists engineers and scientists

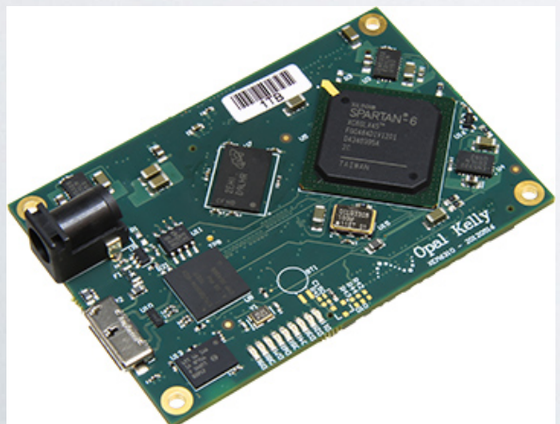


Flight board : Initialy built with ATF280

Has been modified to use NX-MEDIUM (see next)



NX Breadboard : use on the electrical model to replace a Spartan-6 commercial board

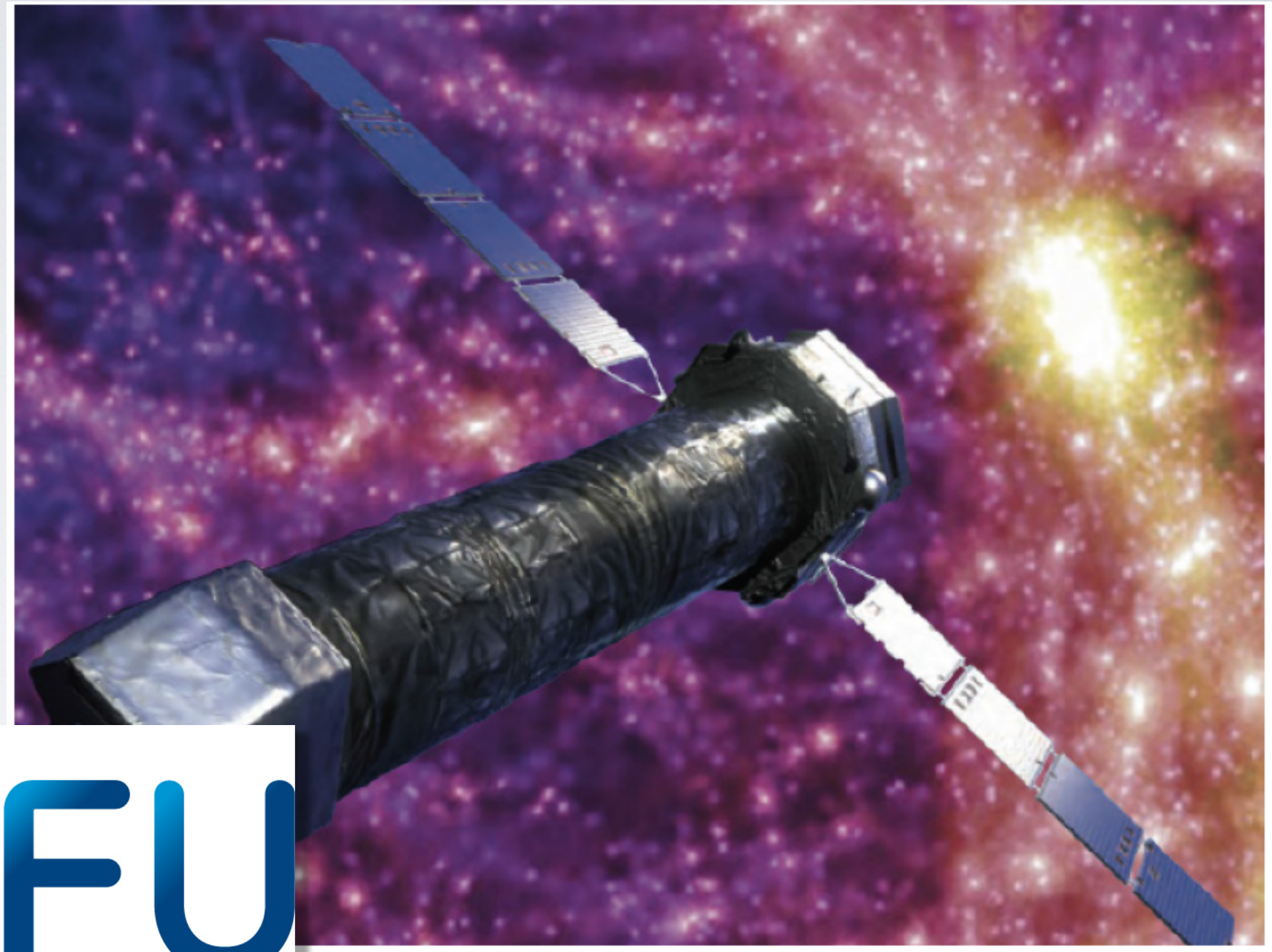


EL Simulator : is able to simulate one ELS

Base on Opal Kelly XEM6010 (spartan 6)

Used by Software engineers and scientists to perform software validation and simulations

ANOTHER (VERY BRIEF) EXAMPLE



IFU

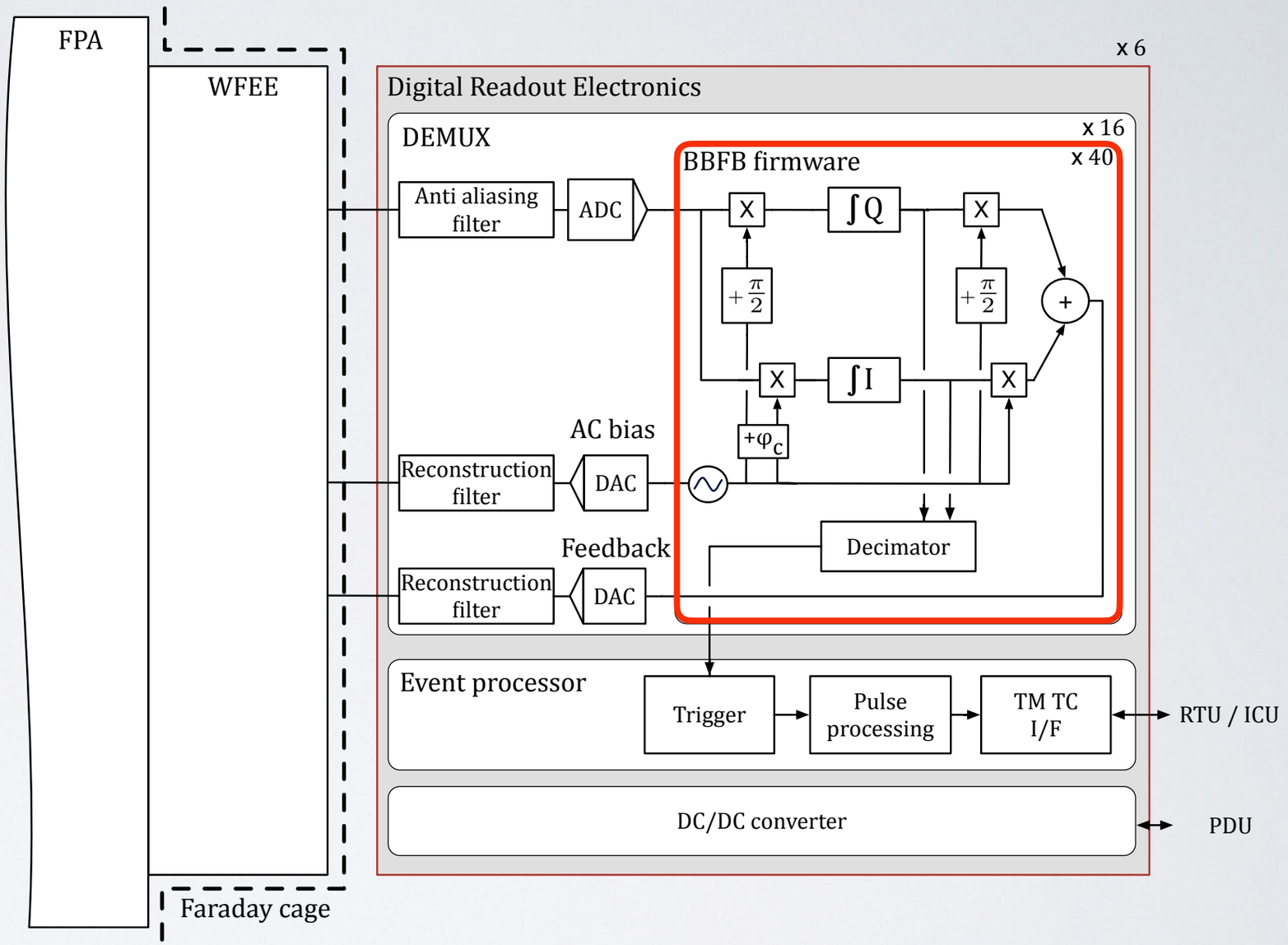
ATHENA X-ray Integral Field Unit

The Digital Readout Electronics (DRE) of Athena X-IFU

- Cryogenic detection array made of 4000 TES-type micro-calorimeters

- TES: Transition Edge Sensors (operated at 90mK, cooled at 50mK)

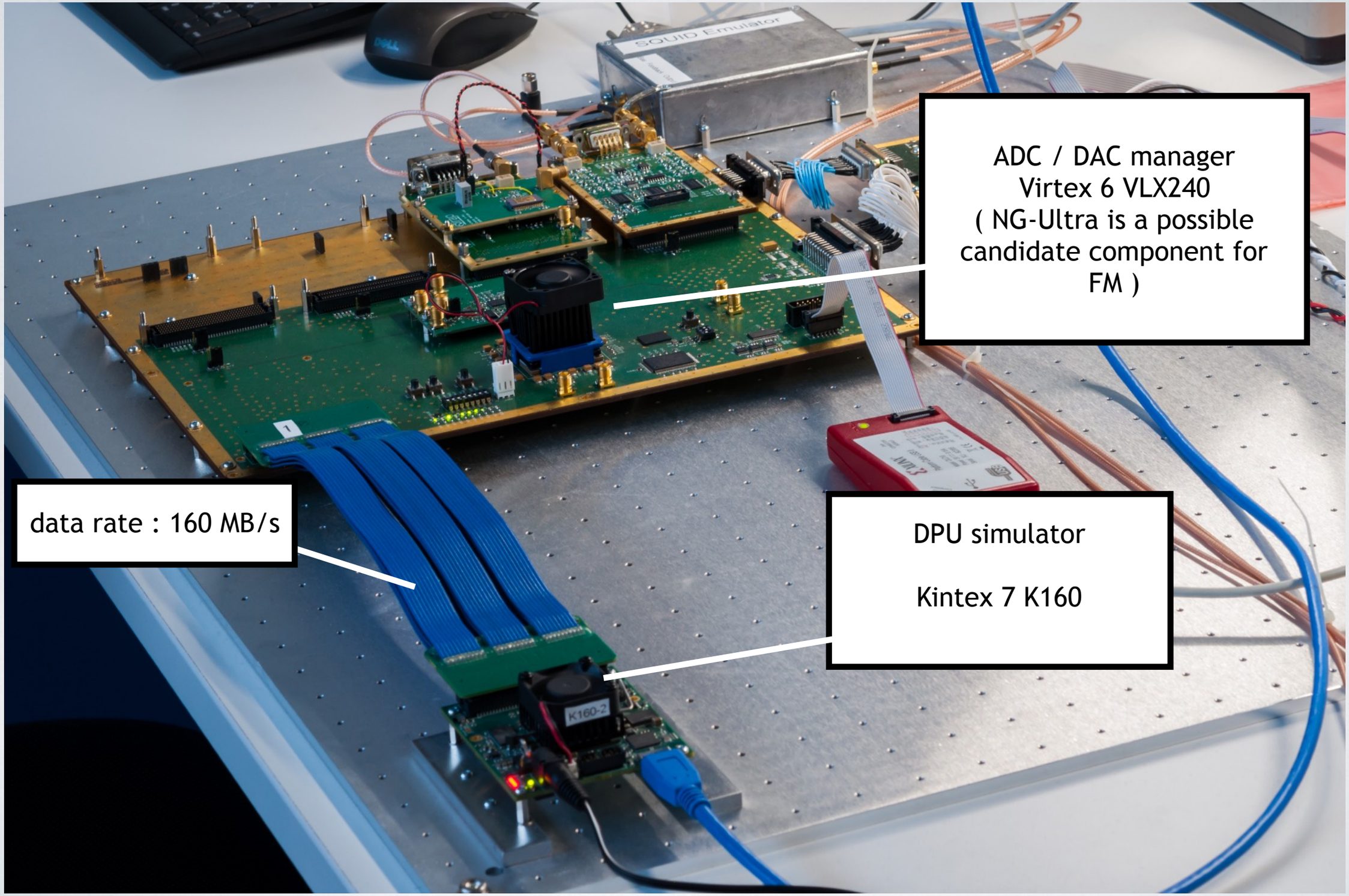
- Frequency Domain Multiplexing (96 readout chains of 40 pixels each)



DSP	656
Clock Speed	80 MHz
Memory	84 banks of 128 words of 32-bits
User I/Os	~300

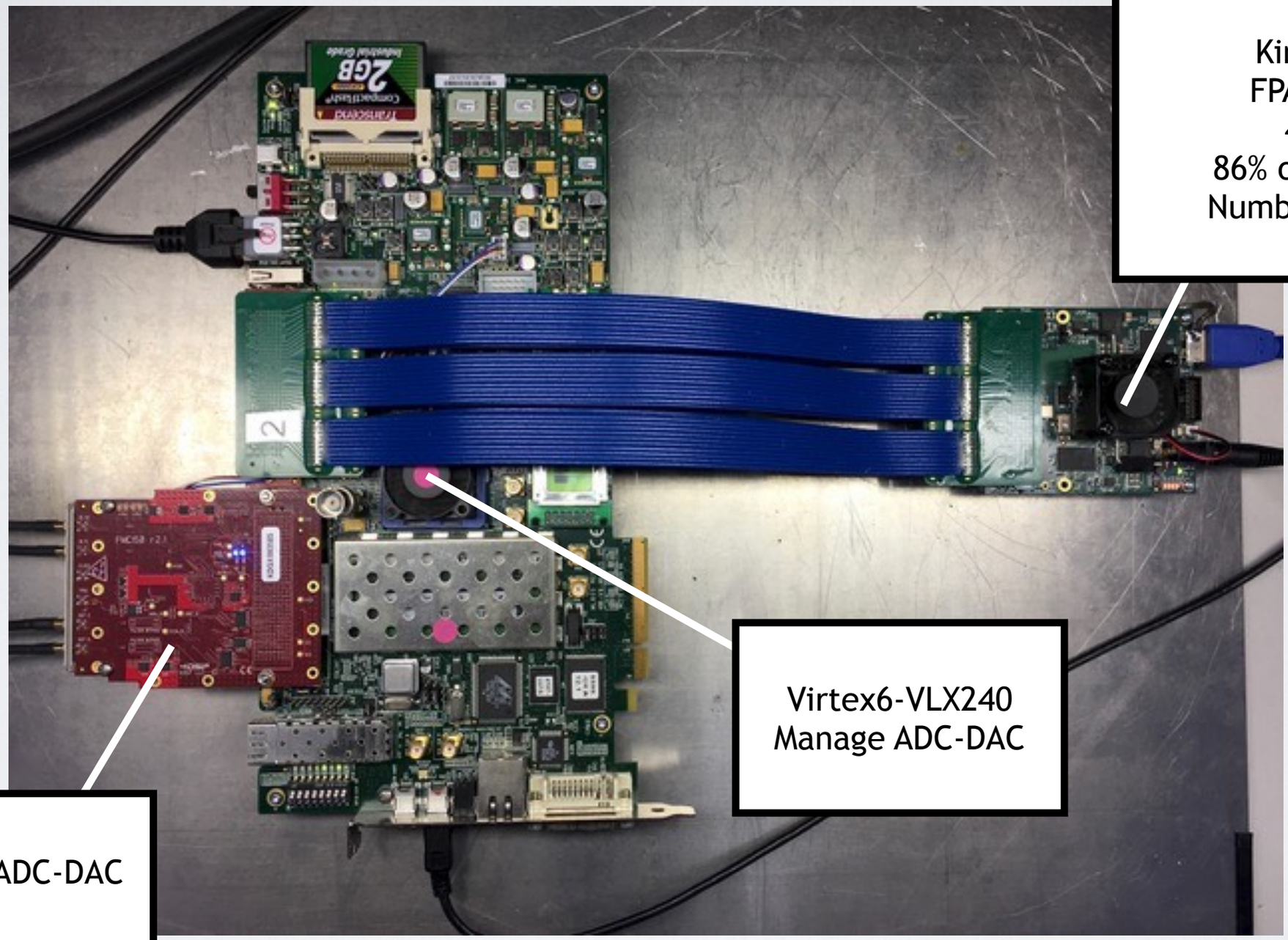
48 FPGAs like this one are needed in the DRE to process the 96 readout chains

The Digital Readout Electronics (DRE) of Athena X-IFU



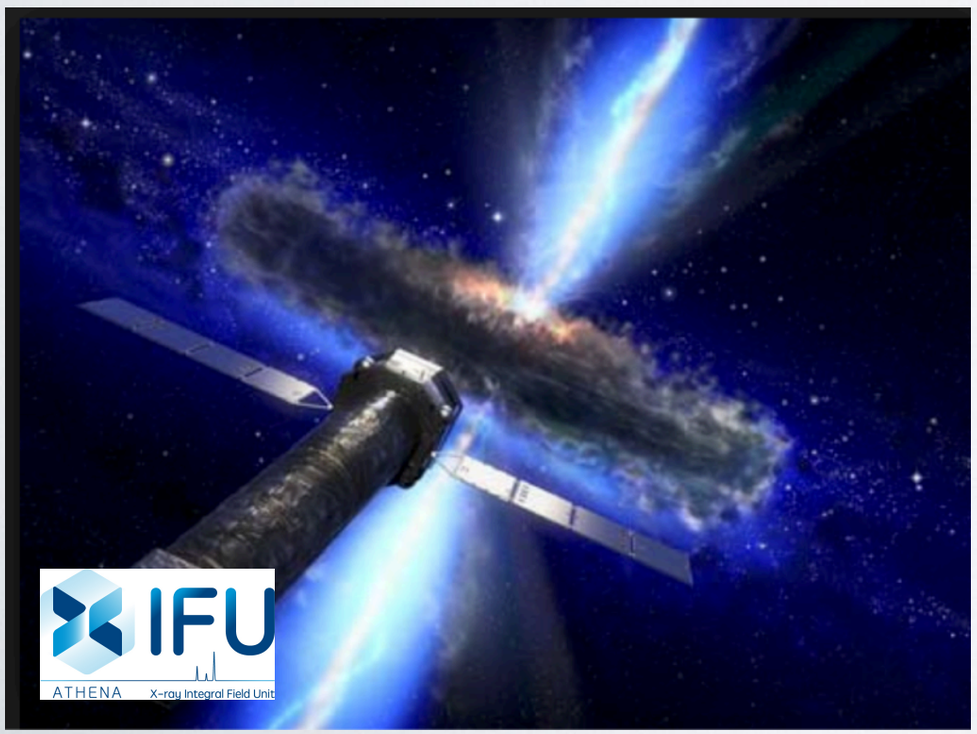
Designed and developed by Bernard Bertrand & Antoine Clénet.

FOCAL PLANE SIMULATOR



Designed and developed by Bernard Bertrand & Antoine Clénet.

CONCLUSION



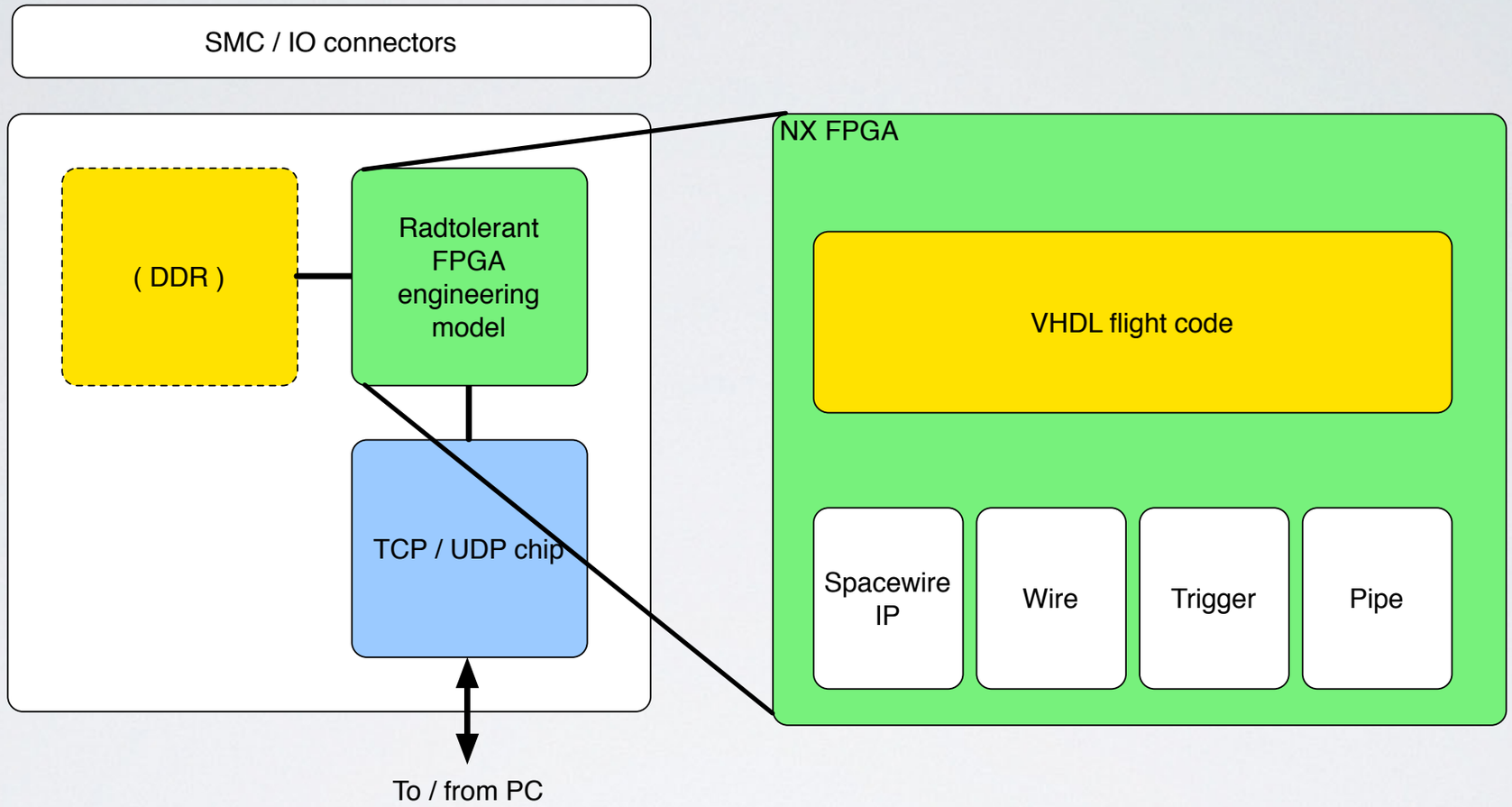
CONCLUSION

- We use a (too ?) wide variety of FPGA : Spartan 3, Spartan 6, Kintex 7, ATF-280, NG-Medium…
- Pro : We can easily handle different needs
- Cons : need to handle multiple development kits
- Cons : need to take care when writing VHDL to ensure portability
- Switching from a target to another takes time (coding, validation…)

Do you have this kind of problematic ?

How do you solve this situation ?

NEXT STEP ?



- It would be a good idea (?) to try to use the same chip everywhere
- It also be good to have only one EGSE communication protocol (and not dealing with USB, Ethernet…etc…)
- Will be the main part of a distributed development environment
- Could be used in simulators and breadboards
- Board will be a little more expensive but we will gain time and design costs
- Work in progress…