A BREADBOARD FOR REAL-TIME IMAGE PROCESSING ON BOARD GAIA

ESA's cornerstone mission Gaia is dedicated to build a homogeneous magnitude-limited survey (from V=6 to 20). The resulting phase map (6D) of the galaxy will allow scientists to tackle questions relative to its formation, evolution and structure.
Launch is scheduled for December 2011 from Kourou in French Guiana with a Soyuz-FREGATE launcher. The minimal operation duration is five years, plus one extended time. The final catalog is expected for 2020.
Gaia's observing principle is inherited from the HiPParCoS mission: two distant fields of view gradually scanning the celestial sphere but has to deal with the absence of catalog. Indeed, a large amount of stars Gaia will observe during its mission are unknown. Thus, a couple of CCDs (SM1 and SM2, one for each field of view) are foreseen onto the focal plane to detect the objects passing through the FoVs and to sort the objects of interest (see picture 1).

This survey is realized by a satellite rotation along its main axis. This induces a scanning motion of the sky and making the stars passing through the CCDs. Normally, this may produce a trajectory parallel to the CCD lines. Thus, the CCDs are read out in a TDI (Time Delay Integration) mode. The pixel charges are shifted from one pixel to the next one, following the star motion onto the CCD and performing an integration during the transit duration over the larger axis of the CCD (see picture 1).

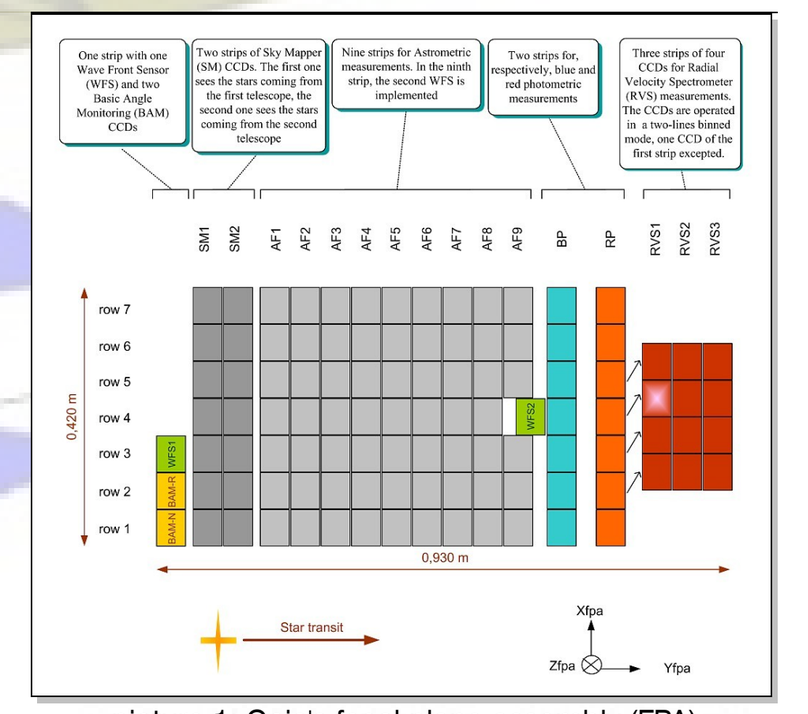
The number of stars that these CCDs will observe ranges from a typical 150 000 stars per square degree in the disk up to 3 000 000 stars per square degree when the completeness in the magnitude domain is reached. Picture 2 shows how a 600 000 stars-per-square degree density is imaged in Gaia's CCDs. One can see here the large volume of data to be processed and the pixel size.
As Gaia will operate on a small amplitude Lissajous orbit around L2, 1.5 x 10^6 km away from the Earth, the available bandwidth for transmission to ground will be reduced compared to a closer orbit such as polar or geostationary ones. This implies to reduce the volume of the transmitted data by sending the pixels related to an object. This assumes to estimate the position of the objects in the CCDs by an on-board processing.

The main tasks devoted to the on-board processing are:

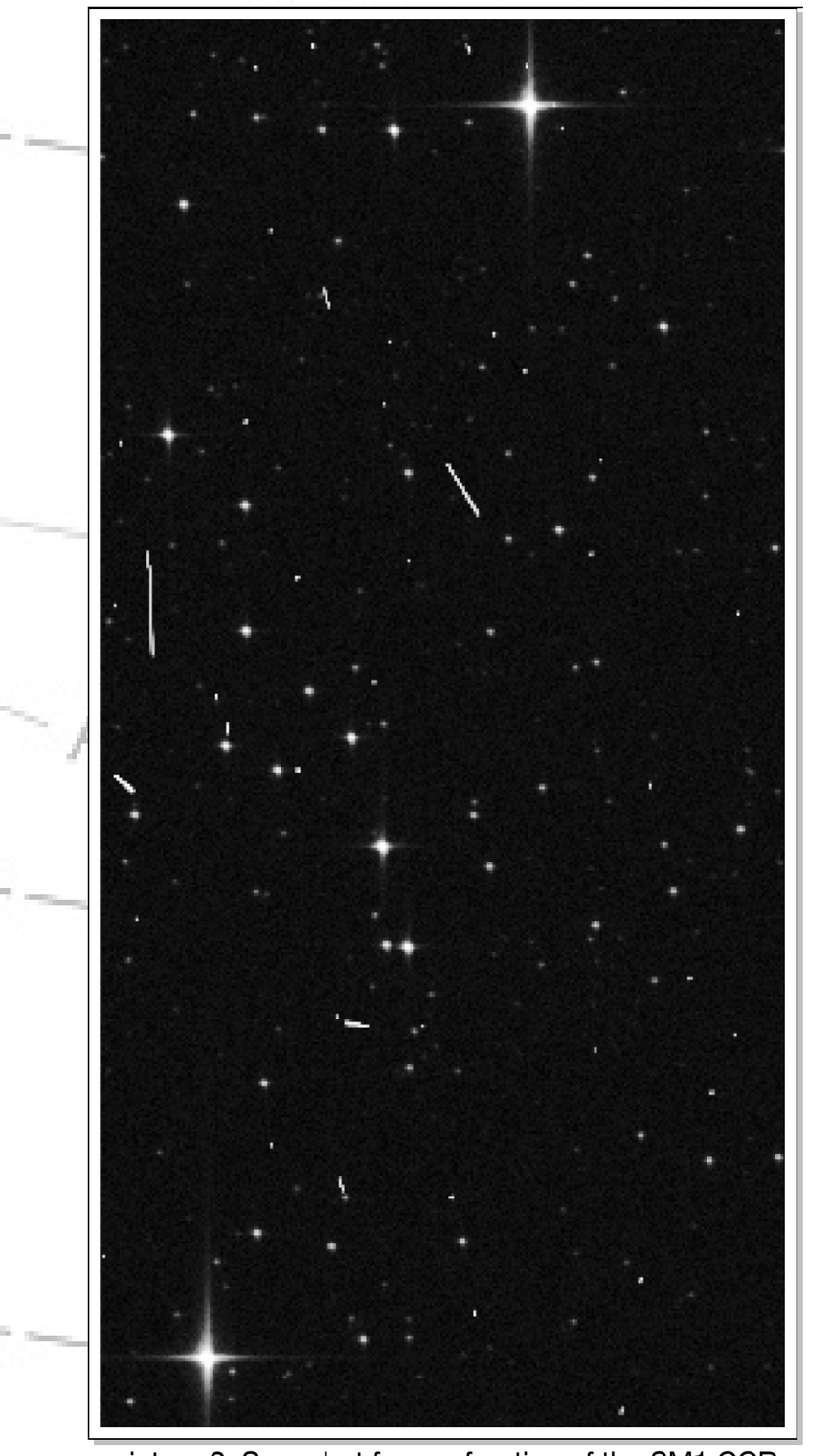
- **(1)** detect the objects in SM1 and SM2 and sort them as relevant (stars, asteroids...) and undesirable (transients, particles...),
- **(2)** to predict object positions through the FPA (Focal Plane Array),
- **(3)** configure and command the read-out of the CCDs.

In this poster, we chose to address the task 1. The sequence of processes needed is explained to the right.
For each pixel, the main features are:

- pre-calibration: calibration, white and black samples, non homogeneous sensitivity, etc.
- background estimation,
- selection of the current pixel if a signal to noise ratio condition is fulfilled (the sample belongs to an object),
- gathering the samples pinpointed by the previous process in two dimensional objects (connected components),
- estimation of the position of the object in the CCDs,
- confirmation that the object is not an artifact.



picture 1: Gaia's focal plane assembly (FPA) (courtesy of ASTRIUM SAS)



picture 2: Snapshot from a fraction of the SM1 CCD at the design density (courtesy of DPAC)

Pixels can be considered to arrive every 1 μs. This timescale is calculated by considering (1) the number of pixels in one column (983) and (2) the spin speed leading to a read-out rate of approximatively 1 MHz. It is a very demanding requirement and tests performed by ASTRIUM GmbH showed that a full software solution will not be available with the foreseen technology in space. An over-tasking of about 100% of the processor would occur due to the star density.

Fortunately, this sequence can be divided in two groups. The first four processes dealing with pixels can be seen as a pipeline (pre-calibration, background and selection) well suited for a logic implementation onto a FPGA because it requires only simple instructions. The second group of processes (estimation and confirmation) is more complex and should be devoted to the software. In fact, the observed object could be single or multiple stars, solar system or extended ones. The on-board processing will have to deal with this bestiary to recognize the object to provide to the ground the right information.

In this poster, we present the electronic unit we have imagined to fulfill the hardware part of the on-board processing, devoted to the processing of the data flow coming from the SM1 and SM2 CCDs, from two point of view: the logical one (VHDL programming) and the analogical one.

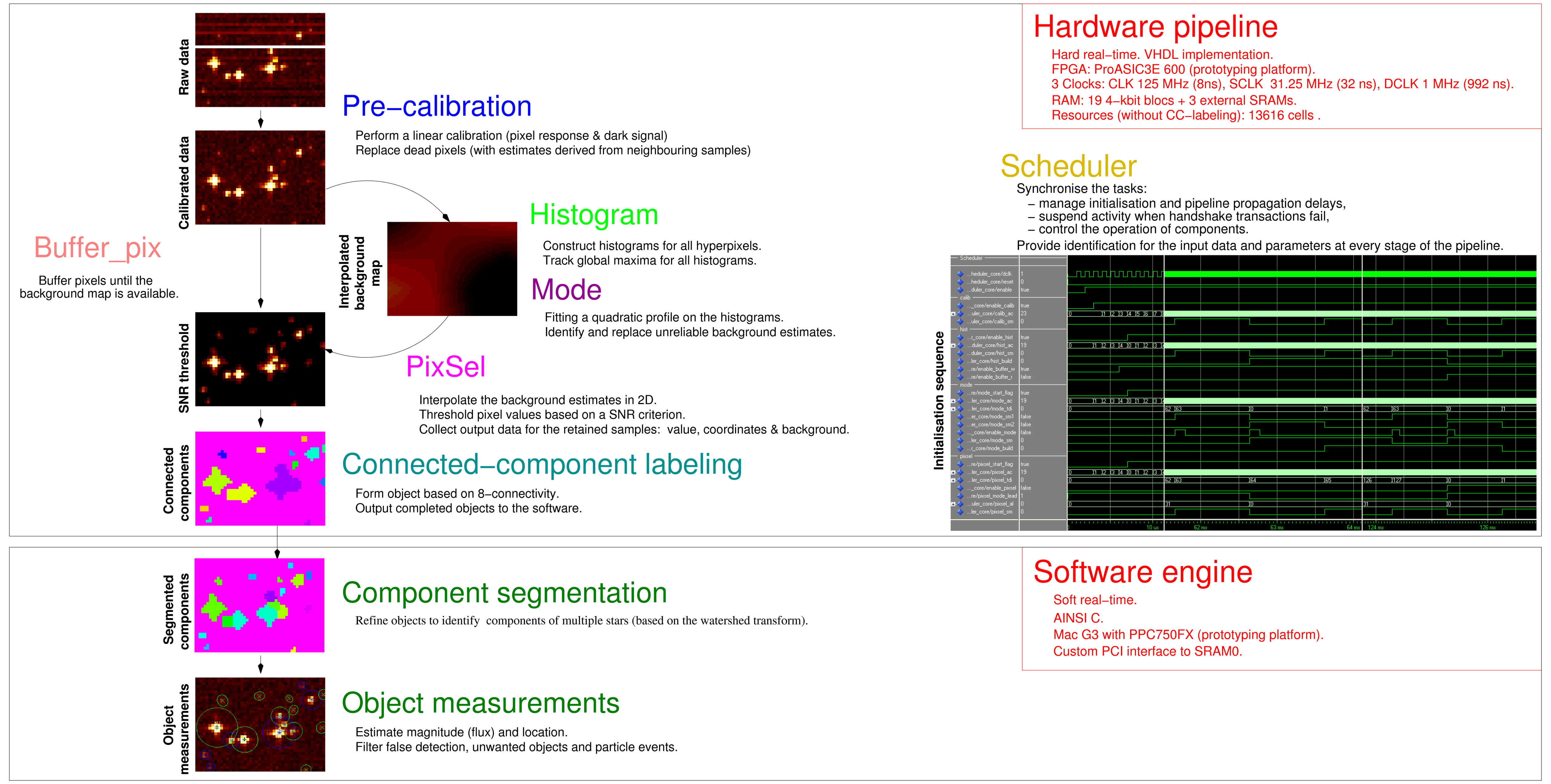
THE TECHNICAL CONSTRAINTS

It is necessary to fulfill certain conditions so that the demonstrator is as representative of the space technology as possible.

We use components of the commercial grade having spatial equivalents in terms of electrical behavior, tension of use, command signals, etc.
Some examples:

- the chosen FPGA is an ACTEL ProASIC3E (Flash), used as a model for a RTAX-S (Anti-fuse),
- the operational frequencies are comparable to spatial ones (1 to 32 MHz),
- the memories are static and asynchronous.

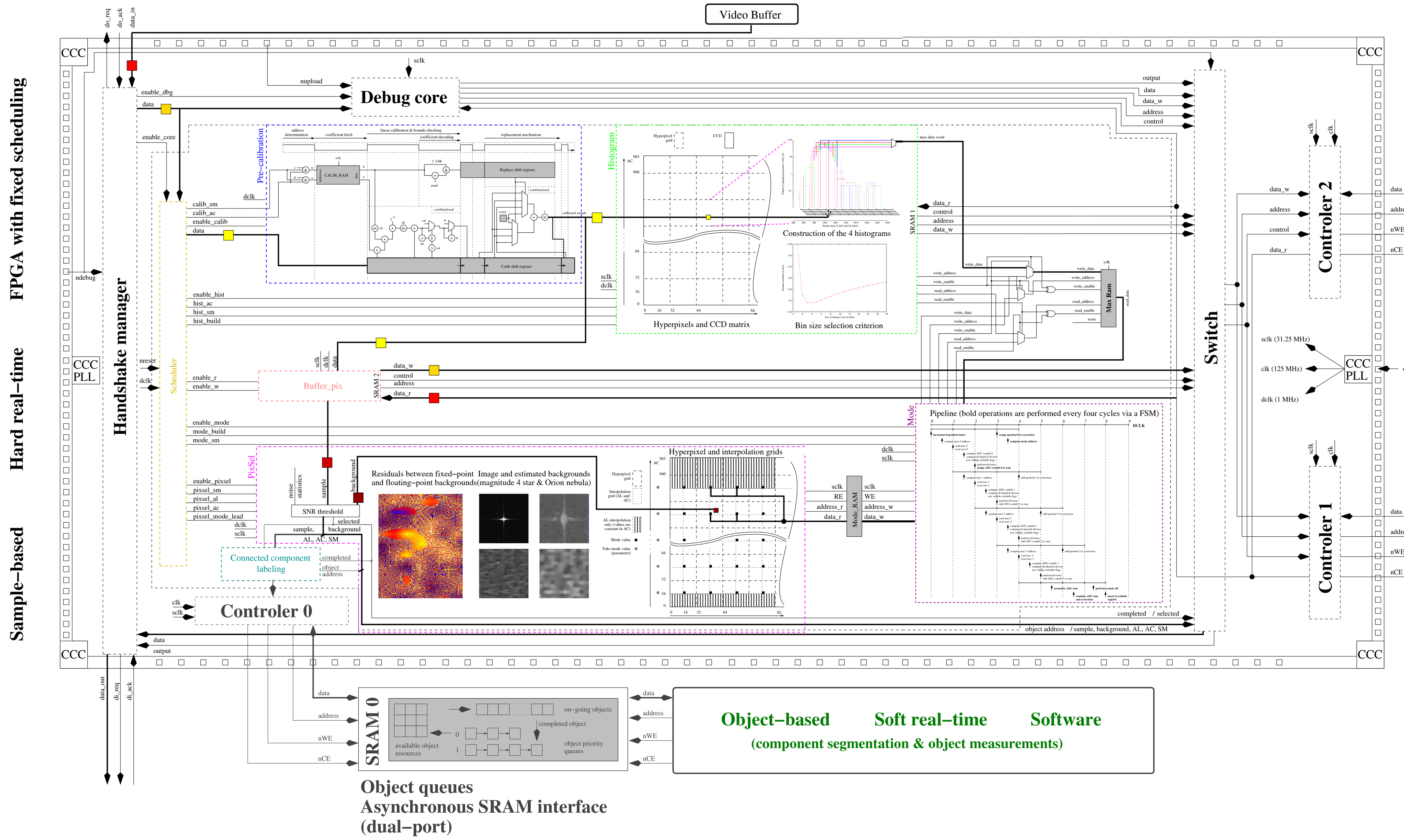
We use an ACTEL development kit to program the FPGA.
Only PQ208 packages are available, so, due to the numerous signals we had to connect to the FPGA, only single-ended signaling is used.



Criterion	Status
General	
VHDL-93 standard only	✓
No tools' specific types	✓
Consistent English	✓
Emphasise readability	✓
Indent with space character	✓
No generated VHDL	Smartgen
Floating point portability	✓
Implementation limitations	specified ranges
Preferred predefined operators	✓
Documentation	
Fully documented	✓
Descriptive comments	✓
Headers	incomplete
API descriptions	incomplete
Document configuration files	Srecords
Version control	subversion
Names	
No extended identifiers	✓
Recognisable active low signals	n- or $\bar{s}$
Name according to purpose	✓
Traceable signals via names	✓
Unique identifiers	component-wise
No encapsulation for renaming	✓
Signals	
MSB at left position	✓
Consistent index ordering	✓
No use of default value	✓
No number of values in type clause	symbolic
No use of enumeration order	✓
Use standardised types	✓
No global signals	✓
Minimum duplicate signals	✓
Commutative resolution	none
Associative resolution	none
No shared variables	✓
No guarded constructs	✓

Criterion	Status
Files	
`std.textio` for file access	✓
No `std.textio.input`	✓
No `std.textio.output`	✓
No file-based communication	✓
Ports	
No top-level `buffer` ports	✓
Ports in logical order	function-wise
Named-association port mapping	✓
`std_logic`-derived ports only	✓
No linkage port mode	✓
Entities & architectures	
Only `port` and `generic` in entity	✓
Assertion severity nomenclature	×
Sensitivity lists for processes	✓
Labels for processes	✓
Labels repeated at end	✓
Identical `component` and `entity`	✓
No configuration clauses	✓
Packages & Libraries	
Preferred IEEE	✓
Packages by functionality	layer-wise
Consistent ordering	✓
Full documentation	✓
Individual test benches	×
Distinct libraries for models	×
No reference to `Work` library	×
Test library for test benches	×
Minimum `library` & `use`	debugging
Verification	
Independent verification	×
Comparison to other model	bit-wise vs. GD
Test bench verification only	✓
Test of every executable line	×
Test of boundary conditions	×
Test of singularities	×
Fast board-level model	component-level
Pull-up & pull-down modelling	×

ESA VHDL coding guidelines and status of the demonstrator.



THE PRINCIPLES OF BREADBOARDING

As input, to simulate the data stream (16 bits every microsecond) coming from the CCDs, we use a computer equipped with a 2x16 digital I/O card and a simple asynchronous communication protocol (handshake). The latches of the I/Os are IDT74FCT373 and the connector is a SCSI 100 pins.

Due to the need for a real-time processor to retrieve data from the output SRAM, two architectures are foreseen (see figure 1).
The first corresponds to a simplified architecture sending the output data back to the computer (64 bits every microsecond for the selected pixels).
This simplification has several goals:

- enable testing the modules before the system is complete,
- avoid too many interfaces in the early development,
- reduce the volume of logic.

The second represents the final architecture. The FPGA passes the data on towards the real-time software engine, in charge of characterizing the objects, through an asynchronous dual-port memory.

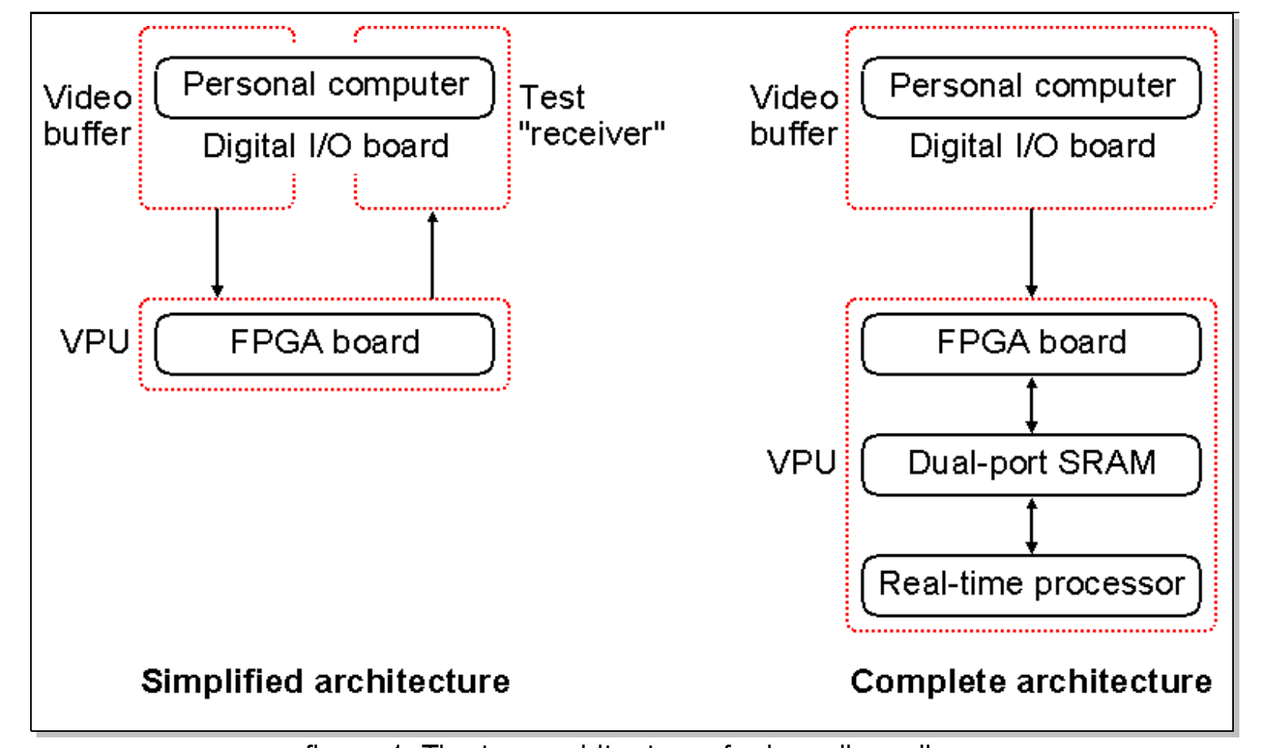


figure 1: The two architectures for breadboarding

FIRST VERSION OF THE DEMONSTRATOR

At the PCB level, besides the evaluation board, four boards were used. This solution aimed at providing flexibility in the routing of signals and the pin allocation which are easier with flat ribbon cables than with traces. All the boards are FR4 with double sided 35 μm thick copper.

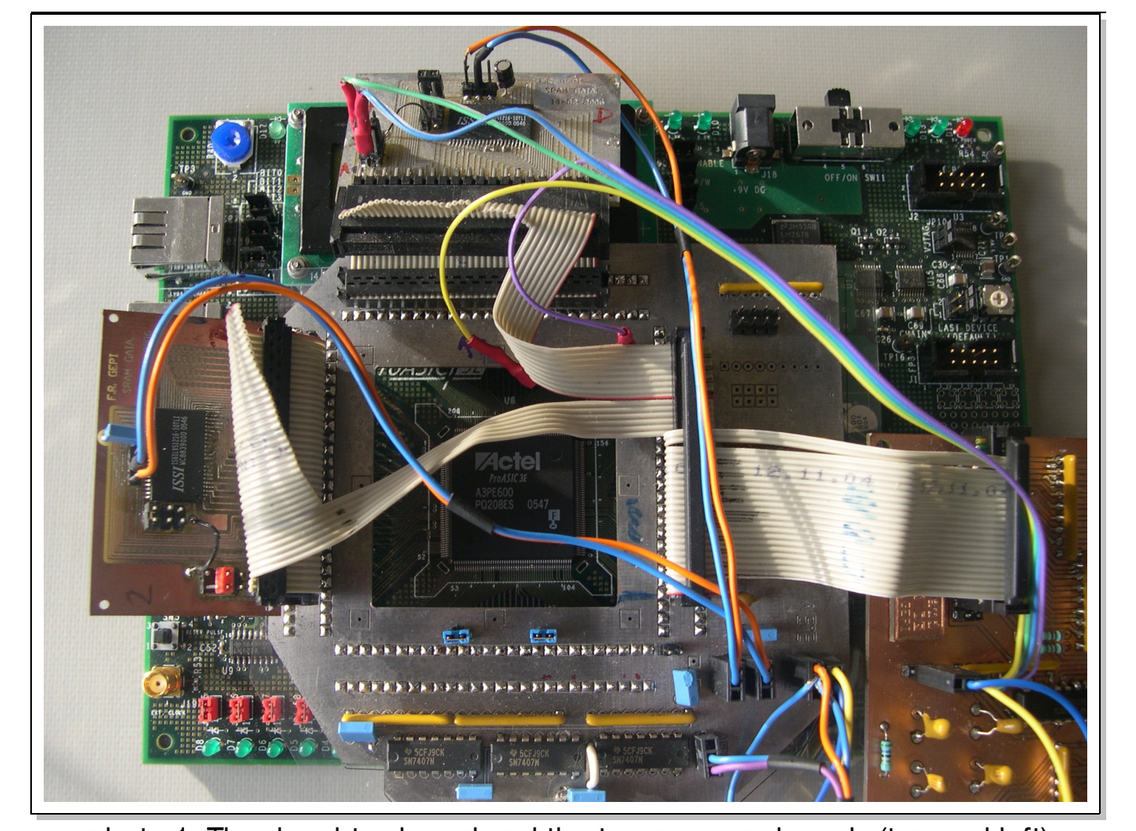


photo 1: The daughter-board and the two memory boards (top and left)

The first one (photo 1) is the daughter-board of the evaluation board, and is directly connected to it using the four connectors (4 x 52 pins) located around the FPGA on ACTEL's evaluation board.
Similarly, the memories are mounted on two additional boards which are connected to the daughter-board with flat ribbon cable (photo 1).
The fourth one (bottom right in photo 1, and photo 2) is an intermediate board connecting the daughter-board (with flat ribbon cables) with the computer's I/O board (with a shielded SCSI cable).
In this first version, the output data is sent back to the computer through this interface.

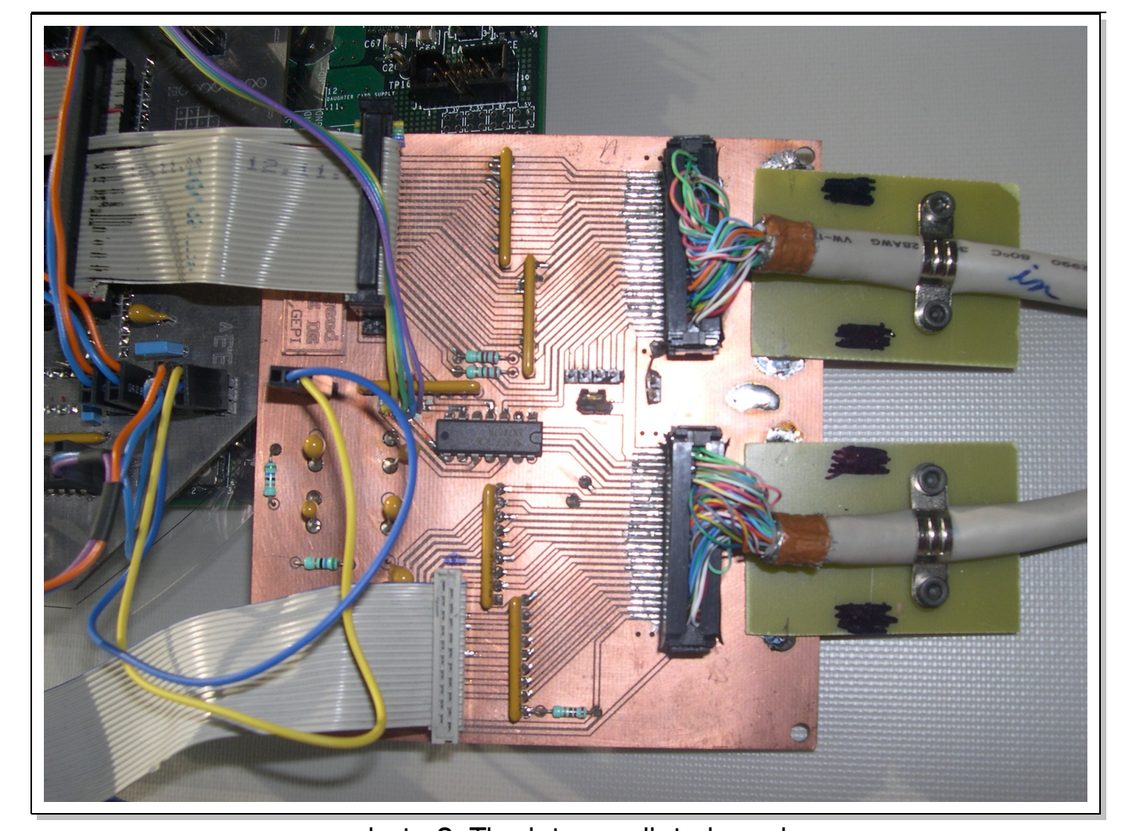


photo 2: The intermediate board

A CASE STUDY

We have chosen to test this experimental version in an extensive way to identify problems and secure the development of a second version.
The first tests permitted us to observe that the signals were very disrupted, presenting ringings and glitches.
These electric problems were correlated with the number of bits switching simultaneously (Simultaneous Switching Outputs, SSO) and the simultaneous in and out data transmission.
The system functioned satisfactorily, roughly, up to 60 % of the load (100 % of the load being defined by 16 bits switching identically and simultaneously).

The identified causes were:

- fast transitions (~1 ns), on the digital I/Os side as well as on the FPGA side,
- many common impedances,
- poor quality ground impedance, leading to ground bounce,
- the absence of adaptation of impedance and line terminations,
- a superfluous active level-adaptation,
- long parallel traces, leading to cross-talk.

To remedy these problems, two directions were investigated.

- the boards were modified:
  - diversion of the current paths to minimize the common impedances,
  - reduction of the ground impedances by doubling cables,
  - improvement of the decoupling of the power supplies by adding capacitors,
  - adaptation of impedance, where possible, by adding components,
  - replacement of the active level-adaptation with passive components (resistors or Zener diode and resistor),
  - remove of the 2.5 V and 5 V power supplies (leaving only the 3.3 V),
  - addition of ferrite beads around the flat ribbon cables and around the ground wires.
- the characteristics of the I/Os of the FPGA were modified:
  - the drivers' strength was decreased (from 12 mA to 4 mA),
  - the drivers' capacitors were modified (no significant effect),
  - Schmitt triggers were added to the inputs,
  - insertion of strong '0' (24mA, high slew) forced outputs between sensitive signals,
  - introduction of delay buffers to desynchronize the SSO.

These developments permitted to obtain a functioning satisfying for more than 90 % of the load. Only a few defects were noticeable at maximum load.

TOWARDS A VERSION 2.0

It was thus possible to raise a list of points requiring a particular attention during the design of a second demonstrator:

- adapt the impedances,
- add series or AC terminations,
- adapt the rise times and the drivers strength,
- use planes for ground and power supply,
- use multi-layers PCBs,
- make sure of a good filtering of the power supply in the used frequencies,
- minimize the lengths of cables and traces,
- minimize the parallel lengths of the traces,
- minimize the number of neighboring signals switching simultaneously,
- insert strong '0' forced outputs to isolate sensitive signals,
- desynchronize SSO with delay buffers or different trace lengths.

The second version will feature:

- dual-port memory data transfer to the computer handling the objects,
- loading of the constants and parameters from an EEPROM (hard-coded in the first version).

CONCLUSION

With this first functional prototype, we have validated some logical and hardware topics.

- Logical:
  - advantage of three clocks, for power consumption, timing and optimization (resource sharing and synthesis optimized versus resources) point of view,
  - conditional instantiation: validation of each process,
  - synchronous design: generate VHDL portable to RTAX, flexible timing parameters (synchronization of clocks and delays imposed by the PCB).
- Hardware, in spite of the slow design:
  - need to take into account the signal integrity problems (fast transitions, crosstalk...),
  - need for a carefully designed ground plane,
  - need to consider ground equivalents to space technologies.

This prototype was also serve as a reference during the development phase (phase A) and now compares (phase B) with the model created by ASTRIUM SAS.
A further work is obviously a second version, designed with the experience of this version. It will allows the implementation of the all processes in order to provide a full operational system. We will also tackle the software part to propose a complete prototype of the whole sequence of operation.

Shan Mignot, Philippe Laporte and Gilles Fasola

shan.mignot@obspm.fr — **Observatoire de Paris — GEPI** — +33 1 45 07 79 07