An end-to-end approach to the EUCLID NISP on-board pre-processing operations: tests and latest results

Carlotta Bonoli, Favio Bortoletto, Maurizio D'Alessandro, Leonardo Corcione, Sebastiano Ligori, et al.

Carlotta Bonoli, Favio Bortoletto, Maurizio D'Alessandro, Leonardo Corcione, Sebastiano Ligori, Luciano Nicastro, Massimo Trifoglio, Luca Valenziano, Filippo M. Zerbi, Pierre-Elie Crouzet, Andreas Jung, "An end-to-end approach to the EUCLID NISP on-board pre-processing operations: tests and latest results," Proc. SPIE 8442, Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave, 844231 (21 September 2012); doi: 10.1117/12.925707

SPIE.

Event: SPIE Astronomical Telescopes + Instrumentation, 2012, Amsterdam, Netherlands

An end-to-end approach to the EUCLID NISP on-board pre-processing operations. Tests and latest results

Carlotta Bonoli*[a], Favio Bortoletto[a], Maurizio D'Alessandro[a],
Leonardo Corcione[b], Sebastiano Ligori[b],
Luciano Nicastro[c], Massimo Trifoglio[c], Luca Valenziano[c],
Filippo M. Zerbi[d],
Pierre-Elie Crouzet[e], Andreas Jung[e]
[a]INAF, Osservatorio Astronomico di Padova, vicolo Osservatorio 5, 35122 Padova, Italy; [b]INAF, Osservatorio Astronomico di Torino, strada Osservatorio 20, 10025 Pino Torinese, Italy, [c]INAF, Istituto di Astrofisica Spaziale e Fisica Cosmica, via Piero Gobetti, 101,40129 Bologna, Italy; [d]INAF, Osservatorio Astronomico di Brera, via Brera 28, 20121 Milano; [e] ESTEC, Keplerlaan 1 2201 AZ Noordwijk, The Netherlands

ABSTRACT

NISP is the near IR spectrophotometer instrument part of the Cosmic Vision Euclid mission. In this paper we describe an end-to-end simulation scheme developed in the framework of the NISP design study to cover the expected focal-plane on-board pre-processing operations. Non-destructive detector readouts are simulated for a number of different readout strategies, taking into account scientific and calibration observations; resulting frames are passed through a series of steps emulating the foreseen on-board pipeline, then compressed to give the final result. In order to verify final frame quality and resulting computational and memory load, we tested this architecture on a number of hardware platforms similar to those possible for the final NISP computing unit. Here we give the results of the latest tests. This paper mainly reports the technical status at the end of the Definition Phase and it is presented on behalf of the Euclid Consortium.

Keywords: space, simulations, on-board processing.

1. INTRODUCTION

Euclid is a wide-field space mission concept dedicated to the high-precision probing of dark energy and dark matter. It will carry out an imaging and spectroscopic wide survey of the entire extra-galactic sky (20000 deg^2) along with a deep survey covering 10-100 deg^2. To achieve these science objectives the current Euclid reference design consists of a wide field telescope to be placed in L2 orbit by a Soyuz launch in 2019 with a 7 year mission lifetime. The payload consists of a 1.2m diameter 3-mirror telescope light feeding two instruments: a VISible imager instrument (VIS) and a Near Infrared Spectrometer/Photometer instrument (NISP). Both instruments observe simultaneously the same Field of View on the sky and system design is optimized for a sky survey in a step-and-stare tiling mode[1].
Differently from the visual channel the NISP instrument, based on a 4x4 IR detectors focal plane implements on board pre-processing on partial non-destructive readouts to obtain the final science frame.
In order to study the performance of the spectrograph and to drive the requirements on the various instrument parameters, we developed an end-to-end scheme for the on-board pre-processing operations that allows detailed comparison of different readout strategies based on realistic characteristics of the detectors and having in mind the achievement of the expected mission scientific goals. The present NISP instrument pre-processing concept and simulations results obtained during the Definition Phase study are here presented.

*carlotta.bonoli@oapd.inaf.it; phone +39 049 8293481; fax +39 049 8759840

Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave, edited by Mark C. Clampin, Giovanni G. Fazio, Howard A. MacEwen, Jacobus M. Oschmann, Jr., Proc. of SPIE Vol. 8442, 844231

2. THE NISP FOCAL PLANE

The reference detector of the NISP instrument focal-plane is the 2048x2048 pixels Hawaii-2RG produced by Teledyne. This ternary HgCdTe detector, abuttable on all four edges, will fill a 4x4 elements mosaic, as reported in Figure 1, providing a total frame of 8192x8192 pixels equivalent to 45.8x43.3 min^2 in sky and to 134 Mbytes of data per single readout.
Detectors wavelength cut-off (I.E. 50% roll-off wavelength) is assumed presently to be placed at 2.4 μm, thanks to the possibility to trim wavelength cut-off in ternary photosensitive semiconductors[2]. The exact final wavelength is still part of a global performance optimization.

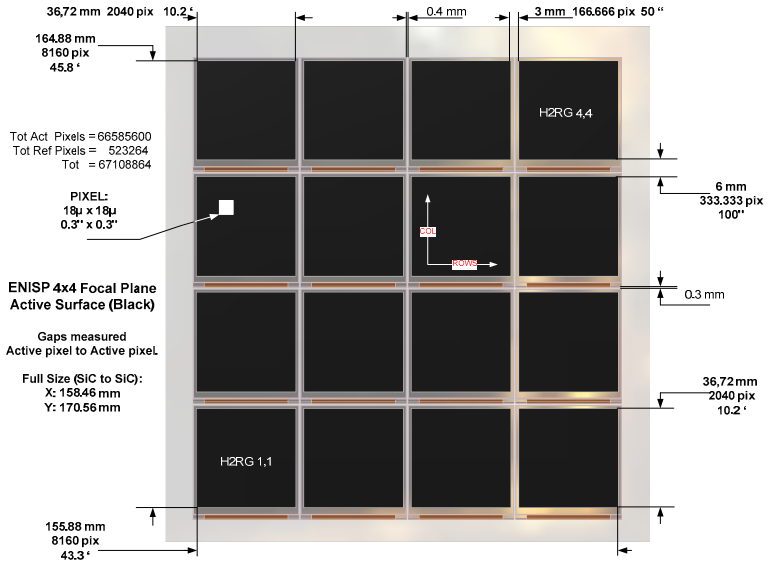

Figure 1 – NISP focal plane detector mechanical distribution.

Detector readout is based on the H2RG silicon multiplexer carrying up to 32 parallel channels of video information and giving a frame readout time of about 1.3 Seconds when driven by a Sidecar ASIC operating at 100 KHz parallel analog to digital conversion.
Detector focal plane is expected to work at the stabilized temperature of 100 K° with a maximum allowed dark current of 0.1 e^-/Pix/Sec while the Sidecar ASIC electronics, again composed by 4x4 devices, is expected to stay at a thermally decoupled temperature of about 140 K°.

3. READOUT STRATEGY

A number of requirements influence the choice of the readout mode for the NISP focal plane:

- Equivalent noise reduction (current specs are between 7 and 9 e^- RMS) even after the longest elementary exposure time (ET, ~ 570 Sec for the spectrograph case)
- The total telemetry budget (850 Gbit /day for the whole Euclid, 240 Gbit/day dedicated to NISP), requiring on board raw frames processing.
- The not negligible number of cosmic rays hitting on the detectors on longest exposures. JWST mission assumptions gave rates between 5 and 30 events/Sec/Cm^2, I.E. from $2x10^{-5}$ to $10x10^{-5}$ events/Sec/Pixel for the single H2RG detector[3], imposing on-board real-time correction.

In consideration of the high level of operational stability required the detector mosaic is under continuous synchronous readout during each overall ET with frames to be pre-processed uniformly extracted in a Multi-Accumulation scheme [4].

Multi-Accumulation (MACC) readout, represented in Figure 2, has some advantages :

- Provides interleaved time for real-time processing (groups least-square fit, cosmic glitch finding and correction).
- Provides final Signal/Noise control by number of inserted groups and number of averaged readouts per group.
- Minimizes the amount of needed processing core memory.
- Allows cosmic glitch effect mitigation.
- Allows needed margins with the down-stream allocated bandwidth.

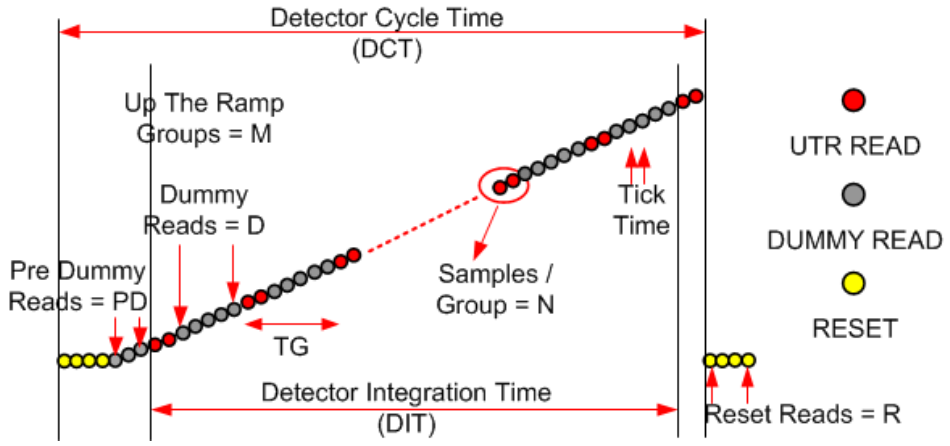

Figure 2 - Sampling in continuous time-uniform non-destructive readout (NDRO) mode illustrated for a single H2RG pixel. Different readout strategies (Fowler, Up the Ramp and Multi-accumulation) can be accommodated by dummy reads interposition (grey dots).

Within this generalized readout scheme the required ET will be defined in terms of ticks of time equal to the detector single readout time (i.e. 1.31 Sec). Shortest exposures (about 100 Sec for the photometer case, as in Figure 3 operational cycle) are made in Fowler mode, which can be considered as a peculiar MACC case with just 2 groups of samples. Consequently frames averaging is common both to the spectrometer and photometer mode and is efficiently implemented at hardware level.

Further processing is implemented on sequential processors; the photometer mode (Fowler) requires only a simple subtraction between final averaged group and starting averaged group and the spectrometer mode requires real-time least-square fitting of progressively produced data in order to recover final signal and to discard glitches by local threshold comparison.

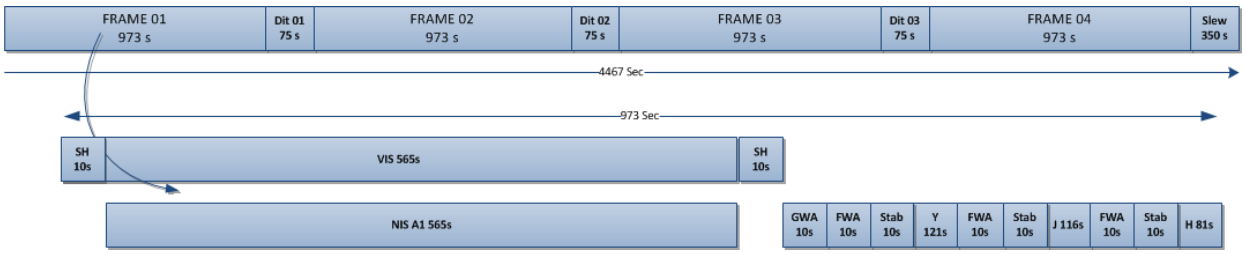

Figure 3 – NISP operational cycle. Each slew is covered by 4 spectro-photometer exposure cycles interleaved by three dithers. Mission telemetry allocation imposes one single, compressed, frame per exposure.

4. END TO END SIMULATION: DETECTOR FRAME SIMULATOR

A MACC frame simulator was developed to provide realistic input frames for the real-time pre-processing code and to analyze the behavior of some parameters crucial for the achievable observation sensitivity.

To deal with realistic data, we used slitless spectroscopic images produced with aXesim, a software tool developed to simulate HST WFC3 data[5]. aXeSIM creates simulated spectra further used by the frame simulator as scientific elementary exposure templates for a 'pure object' signal.
The NISP frame simulator operates on the template on the base of selectable parameters like: readout sequence, the elementary exposure time and parameters referring both to the simulated frame (background, PSF, cosmic rate, etc.) and to the detector behavior (read-out-noise, dark current, etc.).
During the selected, simulated, observation the program generates a sequence of up the ramp frames with increasing signal and affected by the appropriate uncorrelated/correlated noise (I.E. readout, Poisson noise); cosmic rays are also added to the data. In such a way a realistic complete 'data-cube' reporting all the intermediate MACC steps is produced and stored on disk ready for further processing. A detailed description of the simulator, the generation of the ramp frames and the insertion of noise and cosmic rays is given in a previous paper[4].
Statistical results provided by the IDL based MACC simulator on delivered frames (basically correlated and uncorrelated noise) are compared at each run with the same quantities obtained by analytical formulas[6]; the behavior of the simulator has been demonstrated to be extremely close, at the level of a fraction of electron, to the analytical case.

Two basic operational modes (See Figure 4) are allowed:
1. Frame data-cube creation.
2. Data-cube pre-processing with glitches removal in true MACC mode, simple Fowler averaging and differentiation.

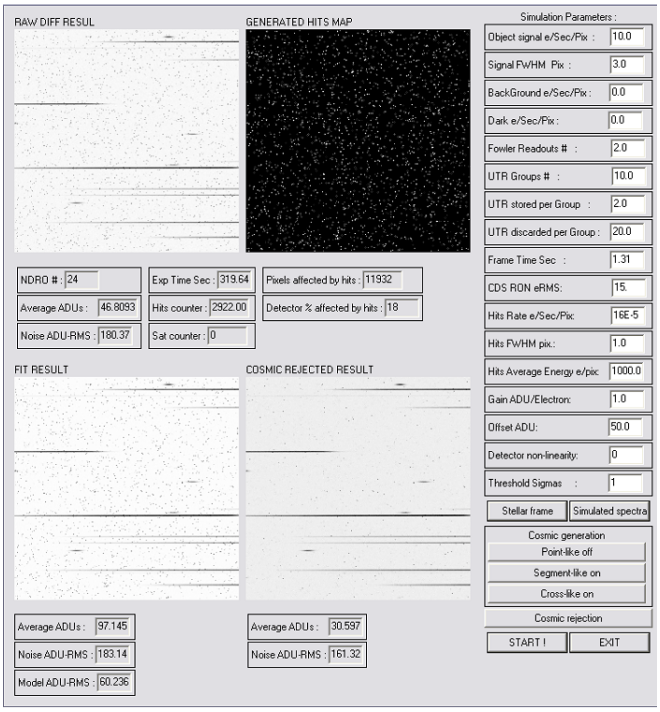

Figure 4. The detector frame simulator IDL interface. Right side: input parameters describing the simulation. On windows: raw frames while generated (upper left), map of inserted hits positions (upper right), result after MACC fit (bottom left) and result after the cosmic ray rejection (bottom right).

5. PRE-PROCESSING PIPELINE

Figure 5 shows the on-board pre-processing pipe-line structure. Assumptions are:

- Spectrograph frames are obtained in $MACC_{M,N,D}$ mode, where M is the number of recorded groups, N is the number of raw frames per group and D the interleaved dummy readouts extended to spectroscopic and photometric modes.
- Cosmic hits deglitching is assumed only for the spectroscopic mode.

The current implementation assumes science data be processed at two different stages based on different hardware, following Figure 5 :

- Stage I Operations : group averaging and operations, fast, synchronous operations, common to the Spectroscopic and Photometric mode performed on dedicated FPGA hardware.
- Stage II Operations: the set of operations needed for MACC mode plus data compression to be performed on sequential processors. This scheme is the one used to develop the code here described and was adopted in ESTEC for parallel tests.

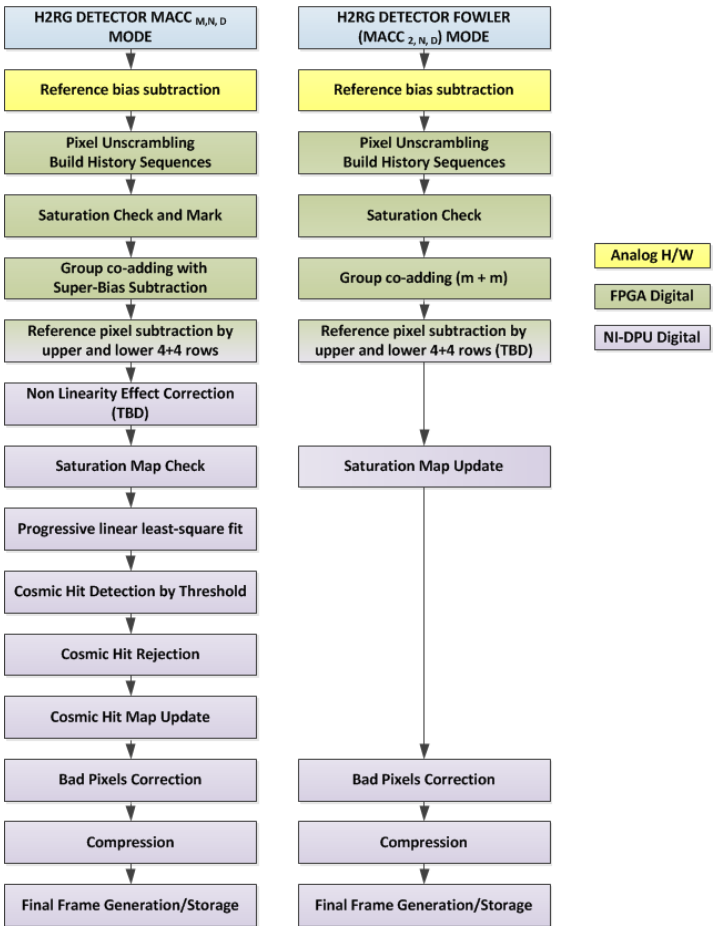

Figure 5. Pre-processing steps to be performed on scientific data before final storage on spacecraft mass-memory. Left thread: Spectroscopic mode (Multi Accumulation), right side: Photometric mode (Fowler).

6. ALGORITHM IMPLEMENTATION

The core of the Stage II operations is the real-time Least Square Fit. Assuming averaged groups equi-spaced in time and using a 'running' non-weighted fit, standard formulas[6] to derive slope and intercept can be simplified minimizing both the amount of computations and data to be stored in memory. The two crucial quantities to be refreshed, pixel by pixel, in dedicated memory-buffers at every new co-added group arrival are signal variance and covariance, here simplified as:

$$SY = \sum_{i=1}^{M} Y_i \qquad SXY = \sum_{i=1}^{M} i \times T_G Y_i = T_G \times \sum_{i=1}^{M} i \times Y_i = T_G \times XY$$

Where: $XY = \sum_{i=1}^{M} i \times Y_i$

Defining:

$$ST1 = T_G \times \sum_{i=1}^{M} i = T_G \times \frac{M \times (M+1)}{2} \quad \text{and} \quad ST2 = T_G{}^2 \times \sum_{i=1}^{M} i^2 = T_G{}^2 \times \frac{M \times (M+1) \times (2M+1)}{6}$$

with Y = Signal ADUs, X = Discrete Time and T_G = Group to Group Time, the fit slope B coefficient after *M* co-added groups is :

$$B_M = \frac{COV(X,Y)}{VAR(X)} = \frac{M}{M} \times \frac{\sum_{i=1}^{M} (X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^{M} (X_i - \overline{X})^2} = \frac{M \times SXY - ST1 \times SY}{M \times ST2 - (ST1)^2} = \frac{1}{T_G} \times \frac{6 \times (2 \times SXY - (M+1) \times SY)}{M \times (M^2 - 1)}$$

while the corresponding intercept *A* is :

$$A_M = \overline{Y} - B_M \times T_G \times \overline{i} = \frac{SY}{M} - \frac{B_M \times (M-1)}{2} \times T_G = \frac{2 \times SY \times (2 \times M^2 - 3 \times M + 1) - 6 \times TS1 \times (M-1)}{M \times (M^2 - 1)}$$

Coefficients *A* and *B* can be progressively evaluated, at the arrival time of each co-added or single non-destructive readout, from running estimates of incremental quantities SXY and SY.
In this situation the amount of CPU core memory needed for data processing, for a single H2RG detector producing 2048x2048 = 8.4 Mbytes of data, is reported on Table 1.

Table 1 – Minimization of processing memory needed for real-time interpolation and deglitching. The Delta buffer is needed to track cosmic glitches signal jumps and to store the final result.

Buffers	**Depth Bytes Normal Case**	**Data Mbytes**	**Depth Bytes Compressed Case**	**Data Mbytes**
SY	4	16.8	5	21
SXY	4	16.8		
Delta	2	8.4	2	8.4
Input	2	8.4	2	8.4
Total		50.4		37.8

Under the above assumptions a typical CPCI bus DPU equipped with 256 Mbytes of core memory can process 4x H2RG (50.4x4= 201.6 Mbytes) detectors in parallel with margins for extra operations.

A further minimization of the allocated core memory involves a better analysis of the dynamics needed for buffers SY and SXY avoiding under/over-flow losses, as described hereafter.

Defining: $NTS_M = E \times \frac{SXY_M}{M \times (M+1)}$ $NSS_M = E \times \frac{SYY_M}{M}$ $DIFF_M = 2 \times NTS_M - NSS_M$

slope and intercept can be rewritten as: $B_M = \frac{6 \times DIFF_M}{(M-1)}$ and $A_M = NSS_M - \left(\frac{b_M \times (M-1)}{2}\right)$

Buffered quantities at each group *M* are now NSS and DIFF. The dynamic expansion factor *E* provides a way to match the minimal number of bits in NSS and DIFF to avoid under/over flow losses. Monte Carlo simulations shows that *E*=1, I.E. 16 bits dynamics in NSS and DIFF is enough to avoid overflow losses with signals reaching saturation but can give a remarkable uncertainty in case of very low signal, where underflows have the largest impact. The multiplication of NSS and NTS by a factor 16 maintains the desired precision even for low signals, keeping throughout the whole computation the otherwise truncated quantities.
Given the different buffer dynamics NSS and DIFF buffers (expanded by *E*=16) are memory packed as shown in Figure 6. A single 32-bit word pack the first 16 bits of both buffers and one byte pack 4 extra bits for NSS and 2 extra bits for DIFF.

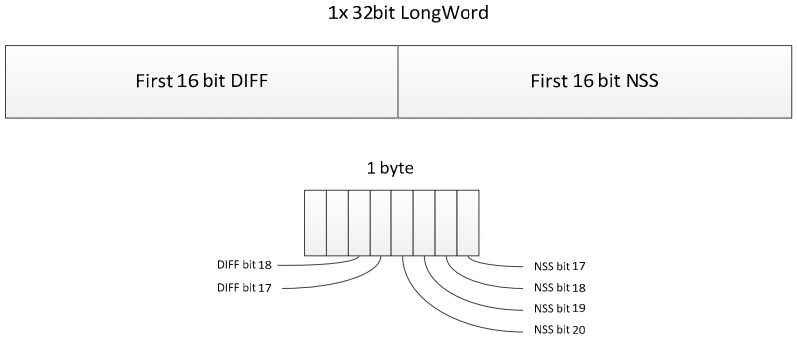

Figure 6. Compressing the NSS and DIFF buffers in 1 LongWord plus 1 byte (2x20 bit).

The recursive updating of the above defined quantities is then done in every step *M* as:

$$NSS_{M=} \frac{NSS_{M-1} \times (M-1) + S_M * 16}{M}$$

$$NTS_{M=} \frac{(DIFF_{M-1} + NSS_{M-1}) \times (M-1) + 2 \times S_M * 16}{2 \times (M+1)} \qquad DIFF_M = 2 \times NTS_M - NSS_M$$

Tests have been done to check the precision of the implemented algorithm. Table 2 reports the results compared with data from the analytical model. Simulations have been made on the so far illustrated end to end simulator and also on practical DPU boards (CPCI and VME bus based) mounting Power PC class processors to verify the processing speed; results are provided in Table 4. The 32-bit + 1 byte configuration speed could be tested on only one PPC board: the burden due to packing and unpacking bits results in a time increment of about 10% with respect to the 16-bit case.

Table 2. Results of the least-square fit algorithm compared with data from model (first rows)

Algorithm Mode	**Input Signal value**	**Noise RMS**
From Model	0.1 e^-/s	8.5
	100 e^-/s	252.6
Integer 32-bit buffers	0.1 e^-/s	8.7
	100 e^-/s	255.0
Integer 16-bit buffers	0.1 e^-/s	11.7
	100 e^-/s	252.9
Integer 32-bit + 1 byte buffers	0.1 e^-/s	8.7
	100 e^-/s	252.7

7. DATA COMPRESSION

A constraining condition is the daily telemetry rate for the science data that are collected by the Euclid experiments: the maximum daily downlink telemetry rate is 850 Gbits/day for a daily telemetry communication period of 4 hours. This data rate provides an upper limit for the maximum number of independent exposures (e.g. by using filters or by dithering), which can be collected during one day. As shown in Figure 3, the Euclid operational cycle time is of 4467 s, leading on a total 17.2 Gbit/cycle, which, added to the 39 Gbit/cycle produced by the VIS instrument, leads to a very large data quantity. On board data compression thus plays an important role in enabling more and better science to be carried out.

Having assumed as viable only loss-less compression to not degrade the quality of the data, preliminary tests have been made with the well known Rice algorithm[7]. The Rice algorithm is an excellent multi-purpose compression algorithm and we tested it on simulated and real H2RG images to understand the required CPU time and the compression factor reachable.

Table 3 - Compression factor and time estimate for some typical NISP images with DPU time projection to the Maxwell PPC-750 case. Base: 1x H2RG detector.

Frame	**Compression Factor**	**Measured Time (s)**	**Estimated Time on a Maxwell PPC-750 (s)**
NIP Background with bad pixels	2.6	0.16	1.12
NIP Background (bad pixels corrected)	2.9	0.15	1.05
NIP Objects + Background with bad pixels	**2.4**	**0.12**	**0.84**
NIP Objects + Back (bad pixels corrected)	**2.6**	**0.11**	**0.77**
NIS Background with bad pixels	2.2	0.19	1.33
NIS Background (bad bixels corrected)	2.3	0.19	1.33
NIS Spectra + Background with bad pixels	**2.1**	**0.18**	**1.26**
NIS Spectra + Background (bad pixels corrected)	**2.2**	**0.12**	**0.84**

To have test images as realistic as possible we combined flat fields and dark frames taken from real H2RG detectors with simulated objects and slitless spectra. The used background frames are taken from a 2.5 μm cut-off, science grade device. Simulated objects and spectra were obtained by the NISP Simulation Group. Examples of the results are shown in Figure 7 showing a typical NIS background, with an enlarged area with unusable pixels in evidence, and a simulated slitless spectra frame superimposed on the background. Possible cosmic ray residuals after the processing operations are consistent with the bad pixel amount assumed in the frames.

To understand the impact of bad pixels and cosmic rays on the result of the compression operation we tested the algorithm both on raw and clean frames. Results are shown in Table 3 and refer to 1 single 2048x2048 H2RG detector, CPU time is for a modern Pentium Quad processor with an estimate projected for a Maxwell PPC-750 board case.

From Table 3 it is important to note that, in the presence of a low number of bad pixels (we can expect a few percent from a scientific grade detector like in this case), there is no huge benefit, neither in terms of compression factor nor in terms of time spent by the algorithm, in working on the raw image or on a bad pixel corrected image version. In such a case, it will be possible to keep the bad pixel correction entirely within the on-ground processing pipeline with no overburden on the on-board DPU.

Nevertheless it is worth to consider the possibility of an on-board bad pixels simple correction based on an upgradable logical look-up table, also in consideration of the possible time degradation of detectors.

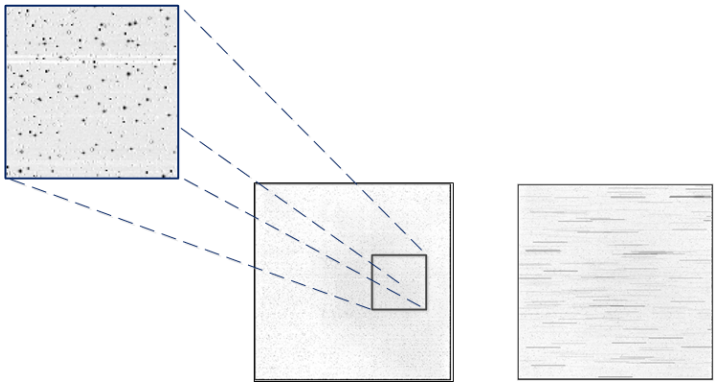

Figure 7. Images used for the compression tests. Left: NIS background with a mean level of 700 e^-/pixel (expected Zodiacal background); upper panel shows an enlarged portion of the frame with unusable pixels in evidence. Right: NIS simulated slitless frame superimposed on background.

Different is the case of the cosmic ray glitches where, in the NIS case, we can expect up to 25% of the detector impacted by hits at the end of the exposure time or more than 10% of bad pixels.

8. HARDWARE IMPLEMENTATION

The current implementation baseline is based on a hardware architecture with 4 CPCI-CPUs controlling 4 detectors each[8], as schematically shown in Figure 8.

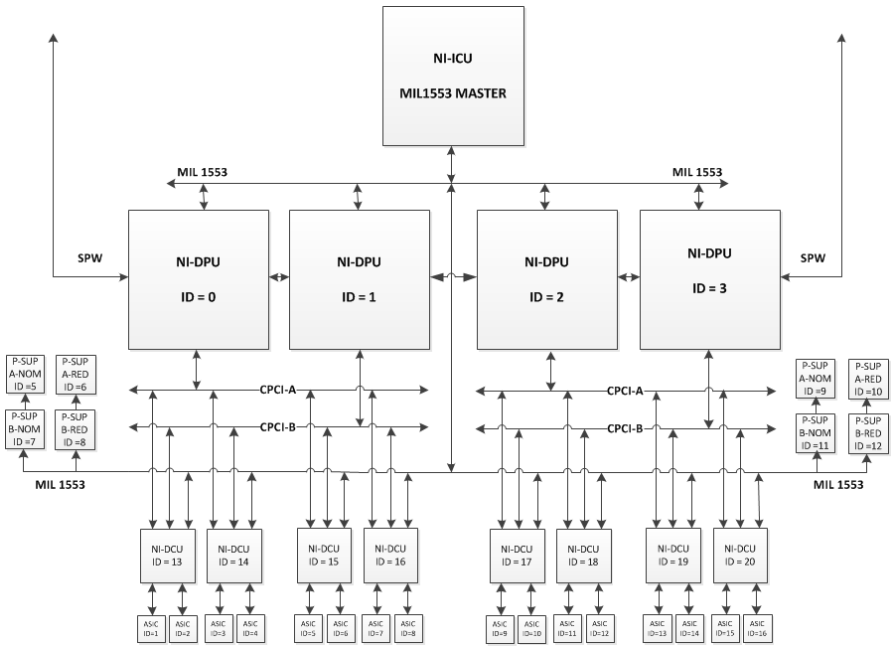

Figure 8. NISP hardware organization, from bottom: 4+4 detector I/F FPGA-modules (NI-DCU) supporting 8+8 Sidecar/H2RG pairs, 2+2 CPCI-CPUs (NI-DPU) served by dedicated CPCI buses, MIL1553 connection to a redundant master control-unit (NI-ICU) and SPW connection to Mass-Memory.

The needed buffering memory per detector is then as follows:

- An input buffer with size 2048x2048, 16 bits deep, holding DMA transfers at each MACC group (I.E. ~ 34 Mbytes)
- A couple of buffers with size 2048x2048, 32 bits deep, holding SXY and SY LSF partials (I.E. ~ 138 Mbytes)
- A buffer 2048x2048 with 16 bits deep holding the averaged hits pedestals $\Delta(k)$ (I.E. ~ 34 Mbytes)

For a total memory estimate of 206 Mbytes for 4 detector. Under normal operation each CPU processes 4x detector frames at a time arriving in DMA from the dedicated CPCI bus.

Each NI-ICU module can be switched from NI-ICU to one from two CPCI buses (CPCI A/B). This possibility allows operation recovery from total CPU fault condition: in such a case one single CPU will take the load for the processing of 4+4 detectors with a dedicated MACC procedure.

9. RESULTS

The pre-processing procedure, as outlined in the previous sections, has been written in C code and tested on a number of different systems with the aim to minimize the computational load and the memory occupancy.

Furthermore, as a support to the Euclid mission and verification, this pre-processing scheme was adopted in ESTEC and independently written in C code, optimized and tested on LEON II and MAXWELL SCS750 PPC-G3 platforms. Relevant results are reported in Table 4; they can be interpreted as a cross check and show, in the MAXWELL case, a spread of the results of up to 23%. This discrepancy can be explained by the fact the algorithms were independently implemented in the code producing faster or slower results for the same given platform at different CPU clock speed. For this reason, the final code has to be optimized for the specific chosen platform.

Table 4. MACC-kernel processing time, group-to-group, for single H2RG on different platforms. LEON data obtained in ESTEC, MAXWELL performance both in ESTEC and by courtesy of MAXWELL US. PPC-G2 evaluation is by courtesy of PATRIA Finland. Other verifications made in Padova Observatory.

	LEON II (ESTEC)	**MOTOROLA MVME6100 PPC-G4**	**MAXWELL SCS750 PPC-G3**		**PATRIA PPC-G2**
Op-Sys	None	VX-Works	VX-Works		RTEMS
Architecture Bus Width/Clock MHz	32	64/33	32/33		32/25
Clock Ghz	0.080	1.3	0.8	0.4	0.150
Memory	?	DRAM	DRAM	DRAM	SRAM
MACC-INT No CR Sec	19.2	1.9	2.8	4.8	7.1
MACC-FP No CR Sec	-	2.7	3.4	6	7.6
MACC-INT Full CR Sec	21.5	1.9	2.8 @ ESTEC 3.4	5 @ ESTEC 4.6	-
MACC-FP Full CR Sec	28.2	2.7	3.5 @ ESTEC 4.5	6.2 @ ESTEC 5.9	-
CPU Time Increase With Frame Size^2	-	Linear	Linear	Linear	-

Table 4 gives the processing time measured for different configurations on the platforms we had so far available; three basic tests have been done:

- Full integer arithmetic, with two 32-bit buffer storage, on frames with 0% and 100% of the pixels affected by a cosmic ray;
- Floating point arithmetic, with two 32-bit buffer storage, as before;
- Full integer arithmetic, with 1 32-bit + 1 byte buffer storage, on frames with no cosmic rays.

Within the differences between different implementations there is quite no difference processing data with or without cosmic ray correction, being the time spent mainly due to the least-square fit.

A decomposition of the processing time per steps is shown in Table 5 for the algorithm developed at ESTEC.
Between the two platforms, for a given step, we should expect a reduction of the processing time by a factor of 10 due to the increase of the CPU frequency by the same factor.
However, the measurements show that this assumption is too simple. One major reason for the deviations observed is the difference in the frequency for the access to memory between the two platforms. For the Leon, the memory access is in the same range as the internal frequency (80Mhz), rather than for the SCS750, the memory access is 100Mhz for 800Mhz of internal frequency. As a consequences, processing steps which require a lot of memory access, such as super-bias subtraction, will not see their total time decreased by a factor of 10. On the other side, steps such as non-linearity correction and cosmic ray detection, which require the highest number of operation gain a factor close to 10 between the two platforms.
Depending on the number of operations per step and the amount of memory access required, the platform could be tuned in terms of memory and internal frequency to results in an optimized total time per step.

Table 5. Processing time decomposition for crucial blocks of algorithm code.

Processing step	**Total time [s]**		**Operations per pixel on Leon platform**	
	Leon at 80Mhz	**SCS 750 at 800Mhz**	**Integer Operations**	**Float Operations**
Saturation detection	0.7	0.14	7	0
Super bias subtraction	0.9	0.33	7	0
Non-linearity correction	3.6	0.29	26	0
Reference pixel subtraction i) odd/even pixel correction	0.7	0.26	5	0
ii) 1/f noise reduction	0.9	0.22	7	0
Cosmic ray detection	13.3	1.58	57	4
Linear least square fit	3.0	1.07	22	0

10. CONCLUSIONS

A reference algorithm covering the needs for the Euclid NISP focal plane on-board pre-processing has been developed and independently tested on hardware compatible with present space qualified CPUs.

The algorithm has been generalized to cover the needs for the focal plane spectrometer and photometer modes, respectively based on up the ramp and Fowler readout processing of incoming frames. The overall hardware supporting structure can be implemented in gate-arrays hardware or in a hybrid structure where the section requiring more strict synchronization, basically the interface to the focal plane digital data outputs, is based on gate-arrays and the more algorithmic part is based on reprogrammable hardware. A similar problematic is present for final data compression where both approaches, dedicated hardware ICs and sequential processors, are both viable solutions. The final Euclid

consortium selection privileges the hybrid approach considering the benefit represented by algorithm refurbishing, possible only on sequential processors.

Tests have been made at two levels: end to end simulations based on high level data processing languages (IDL) to verify the correct algorithm behavior and dynamical simulations on standard C operating on several platforms (notably LEON and PowerPC processors) with the aim to select a suitable processor for the final architecture. Results shown clearly that the PowerPC architecture is the only, presently, able to cope with instrument requirements within a reasonable number of parallel units and with margins for operational and hardware optimization.

ACKNOWLEDGMENTS

We acknowledge the financial support from contract ASI-Uni Bologna-Astronomy Dept. Euclid-NIS I/039/10/0. Thanks are due to PATRIA (FIN) and Maxwell (US) companies for kind helpfulness during performance tests.

REFERENCES

[1] Laureijs, R.J., et al., "Euclid: ESAs mission to map the geometry of the dark universe," this conference
[2] Beletic, J.W., Blank, R., Gulbransen, D., Lee, D., Loose, M., Piquette, E. C., Sprafke, T., Tennant, W. E., Zandian, M. and J. Zino, "Teledyne Imaging Sensors: Infrared imaging technologies for astronomy and civil space," SPIE Proc. 7021(20), (2008)
[3] Rauscher, B.J., Isaacs, J.C., Long, K., "Cosmic Ray Management on NGST 1: The Effects of Cosmic Rays on Near Infrared Imaging Exposure Times"STScI-NGST-R-0003A, (2000)
[4] C. Bonoli, F. Bortoletto, E. Giro, L. Corcione, S. Ligori, L. Nicastro, , "A Frame Simulator for Data Produced by 'Multi-Accumulation' Readout Detectors", Proc. SPIE 7731, 77312U (2010)
[5] Rosati, P., Kümmel, M., Walsh, J., Franzetti, P., Cimatti, A., Garilli, B., Zamorani, G., Scodeggio, M., Rivera, H., " Slitless spectroscopic simulations for Euclid with aXeSIM," ST-ECF Newsletter, 44, 12-13, (2009)
[6] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., [Numerical Recipes: The Art of Scientific Computin], Cambridge University Press, (2007)
[7] R. Rice, "Adaptive Variable-Length Coding for Efficient Compression of Spacecraft Television Data, " IEEE Trans. (1971)
[8] Corcione, L., Ligori, S., Bortoletto, F., Bonoli, C., Valenziano, L., Toledo-Moreo, R., D'Alessandro. M., Trifoglio, M., Morgante, G., Colodro-Conte, C., Rebolo-Lopez, R., Muñoz, J., and Villò, I., "The on-board electronics for the Near Infrared Spectrograph and Photometer (NISP) of the EUCLID mission," this conference