EUDRB: the data reduction board of the EUDET pixel telescope

Lorenzo Chiarelli, Angelo Cotta Ramusino,
Livio Piemontese, Davide Spazian
Università & INFN Ferrara

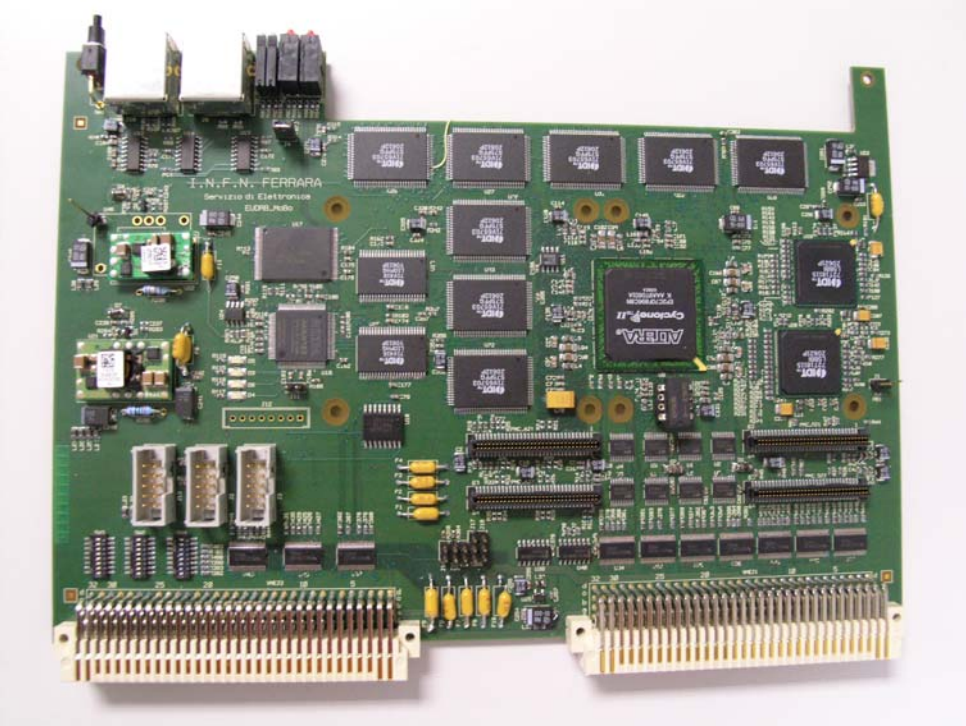
Presented by Concezio Bozzi

Context

- The EUDET JRA1 project: a pixel-based beam telescope to test pixel sensors suitable for the ILC
- In 2007: build and operate a “demostrator” telescope
  - Beam test campaign at DESY (June, August) and CERN (September-October)
- MAPS sensor for demonstrator: MimoTEL, 256x256, 30umx30um pitch

- For the final telescope (2008), aim at a ***sensor*** which has sparsification ON THE CHIP to reduce, with the bulk of data, also its readout time.
- To test such sensors, we have designed a DAQ electronics which
  - maximizes velocity by using superfast VME,
  - …AND can implement real-time sparsification on the readout board..
  - …AND uses the fastest possible VME readout to reduce dead times.

A VME64x/USB2.0-based DAQ card for MAPS sensors



mother board built around an ALTERA CycloneII FPGA (clock rate: 80MHz) and hosting the core resources and Interfaces (VME64X slave, USB2.0, EUDET trigger bus)

NIOS II, 32 bit “soft” microcontroller (clock rate: 40Mz) implemented in the FPGA for

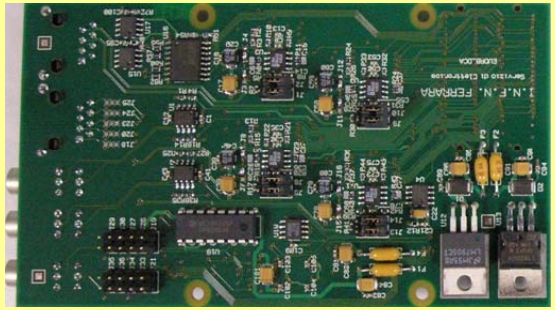
- on board diagnostics
- on-line calculation of pixel pedestal and noise
- remote configuration of the FPGA via RS-232, VME, USB2.0

Two readout modes:

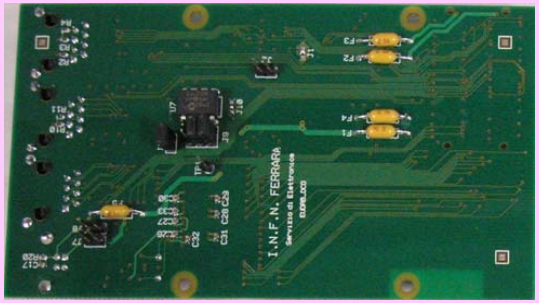
Zero Suppressed readout to minimize the readout dead-time while in normal data taking.

Non Zero Suppressed readout of multiple frames for debugging or off-line pedestal and noise calculations

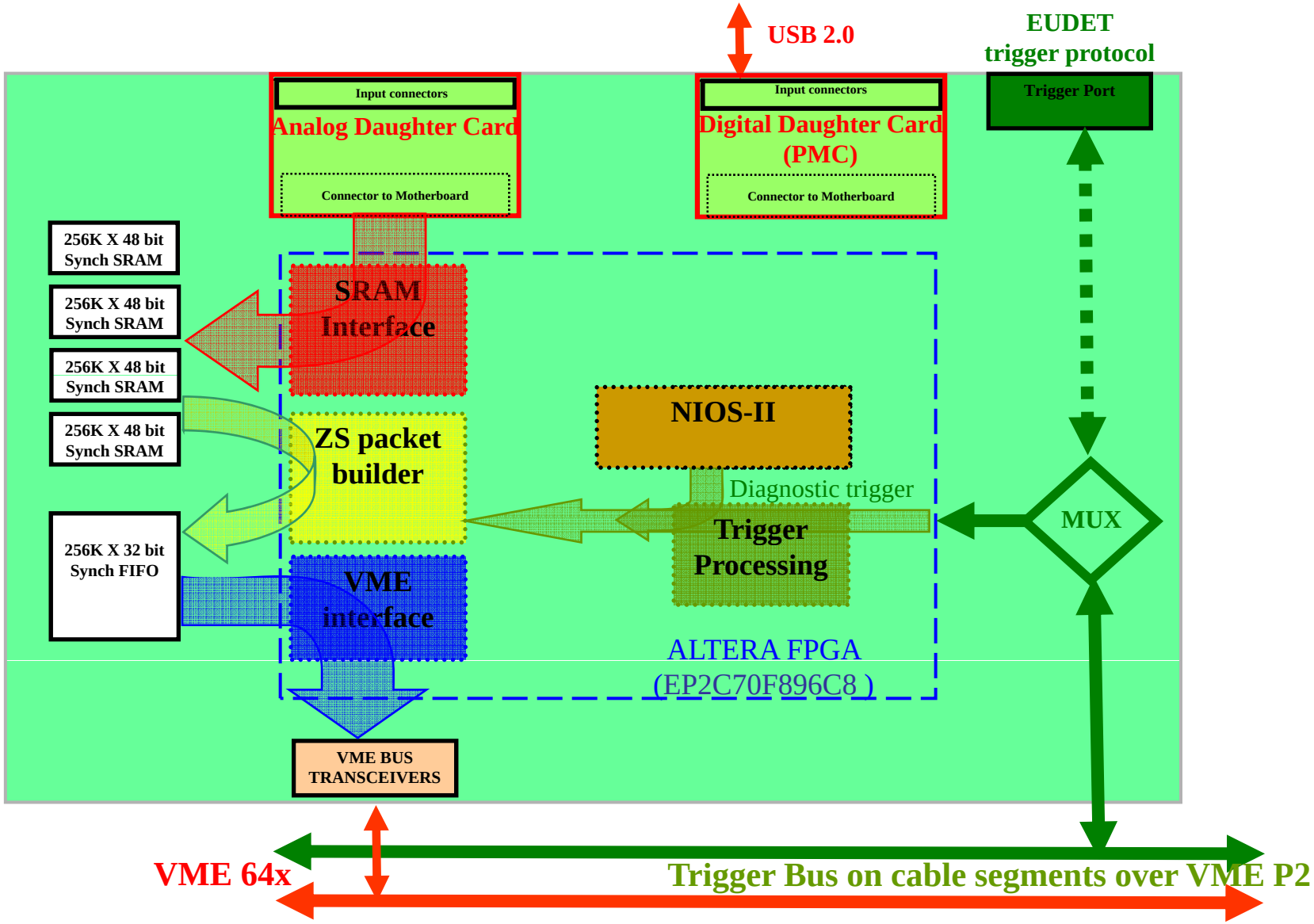
analog daughter card based on the successful LEPSI and SUCIMA designs clock rate up to 20 MHz



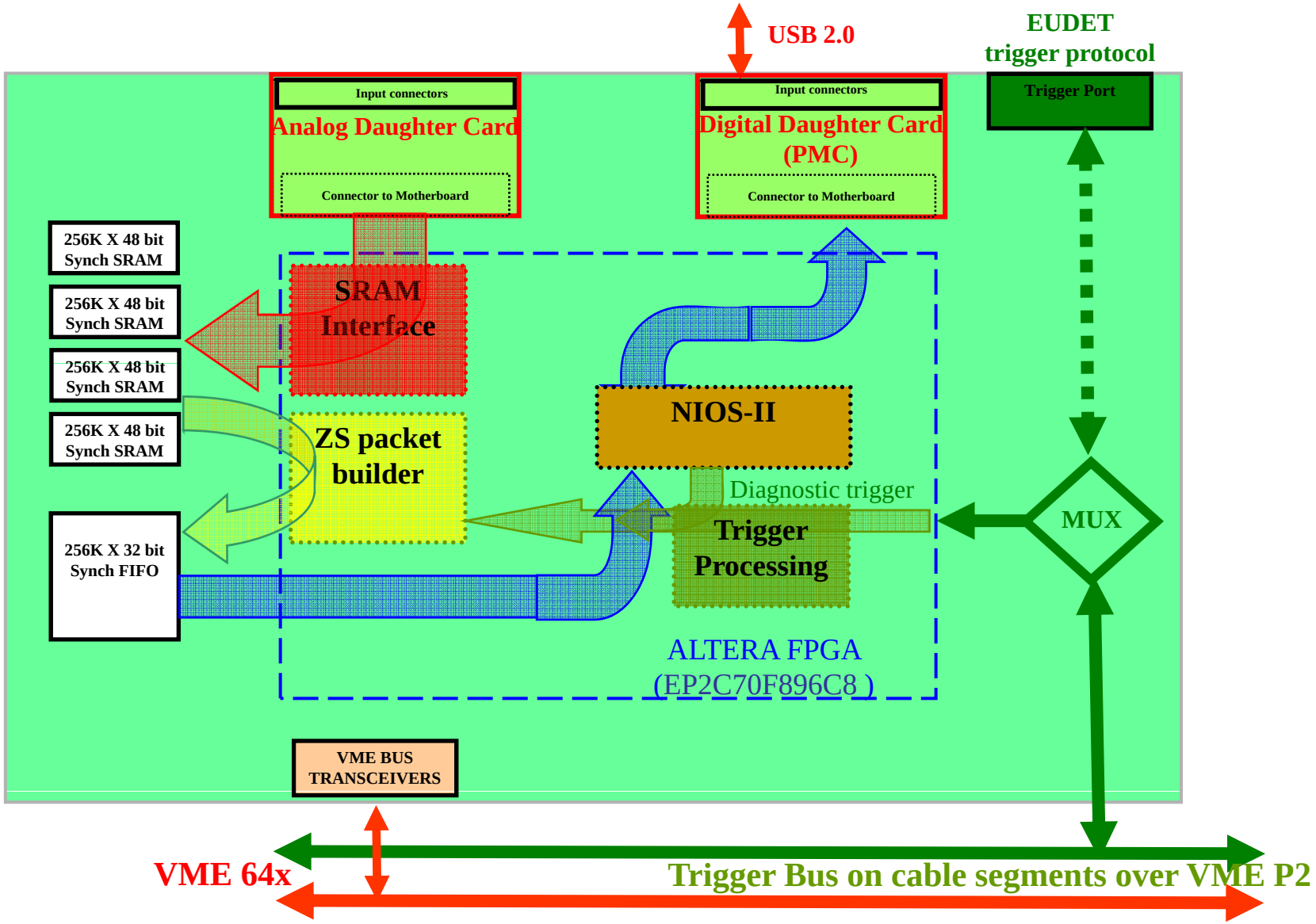
digital daughter card drives/receives control signals for the detectors and features a USB 2.0 link



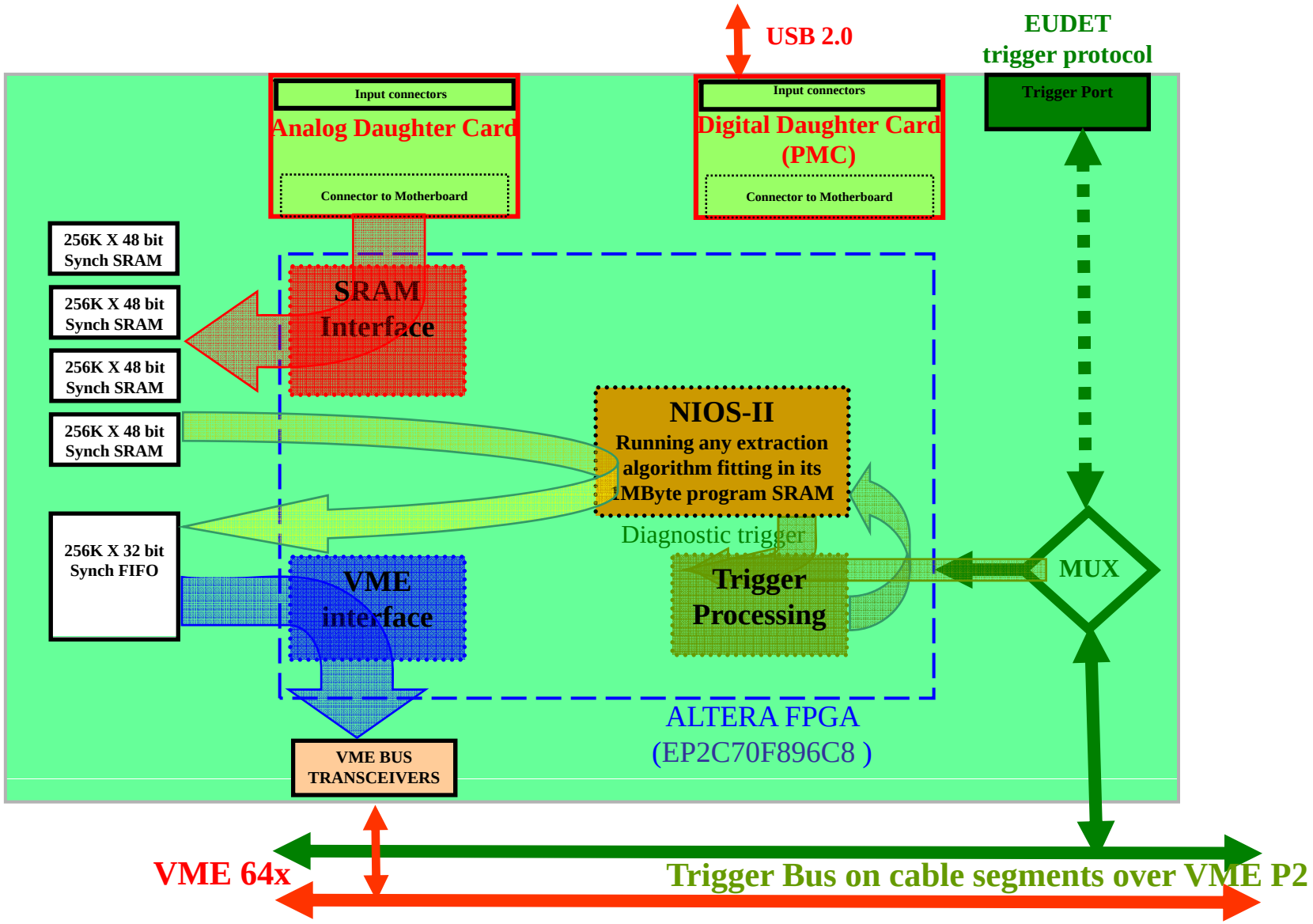
Data Flow for ZS operation (mode for real data taking)



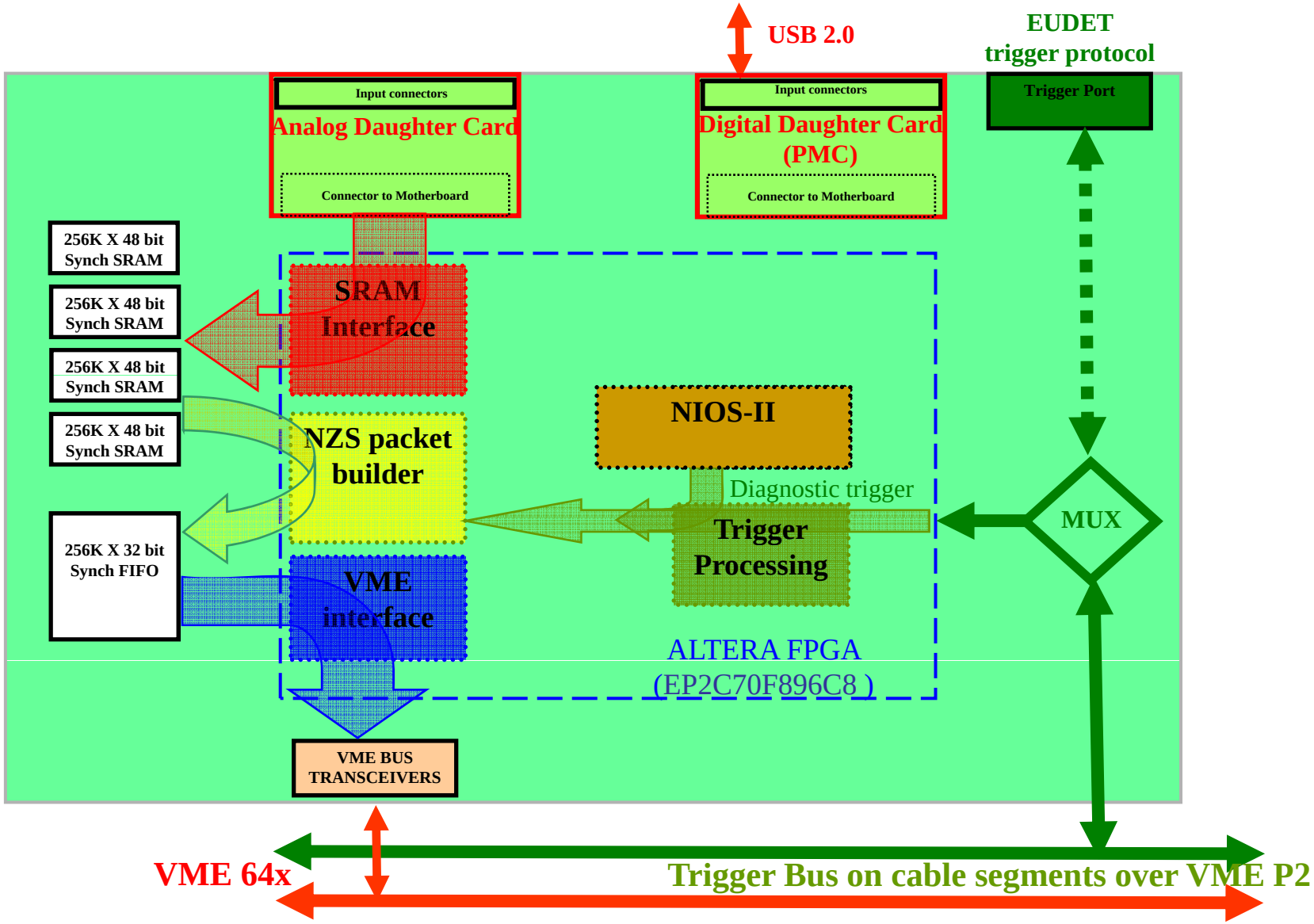
Data Flow for ZS operation for benchtop DAQ (slow) via USB2.0



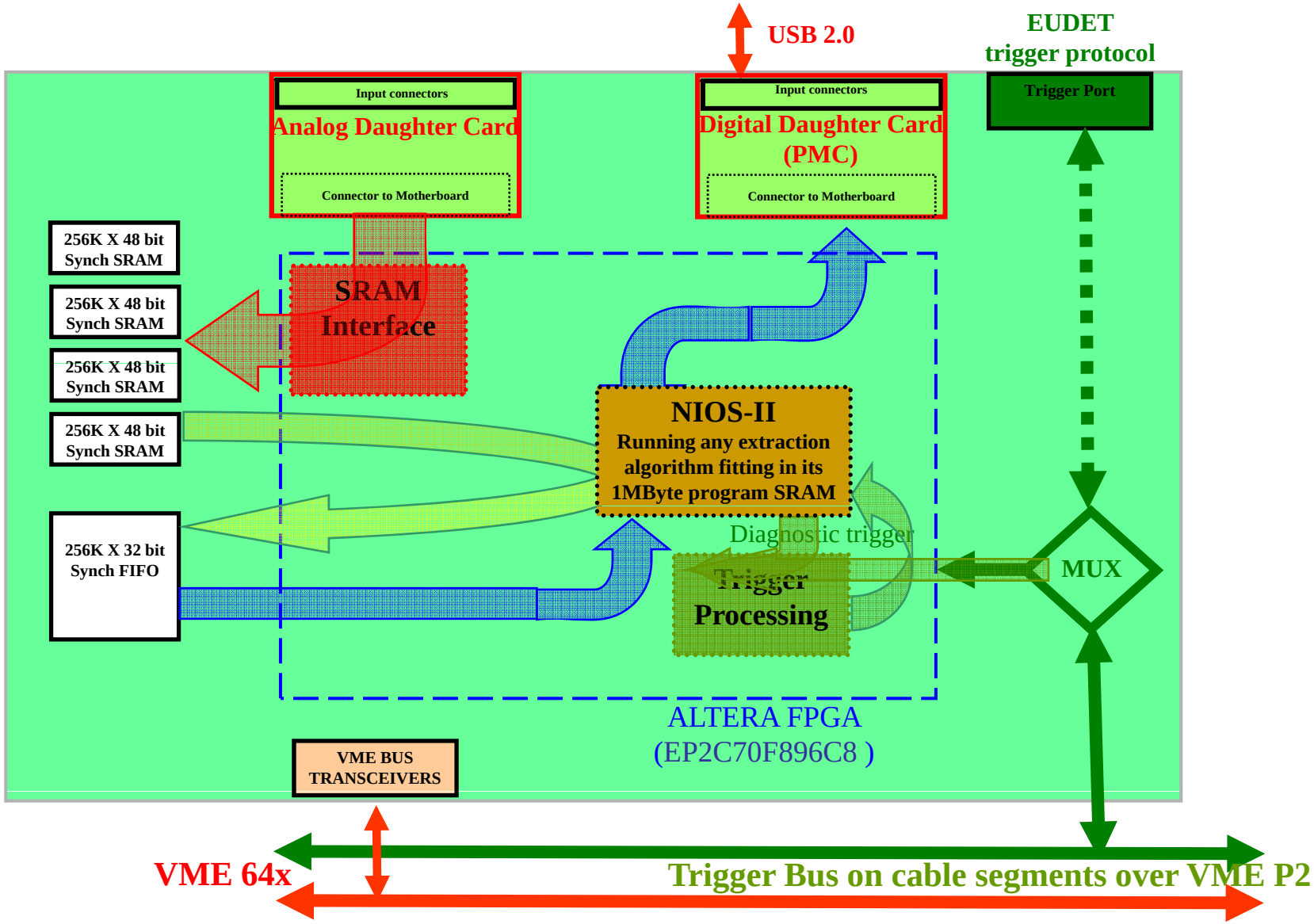
Data Flow for NZS readout via VME (for detector characterization). Current implementation



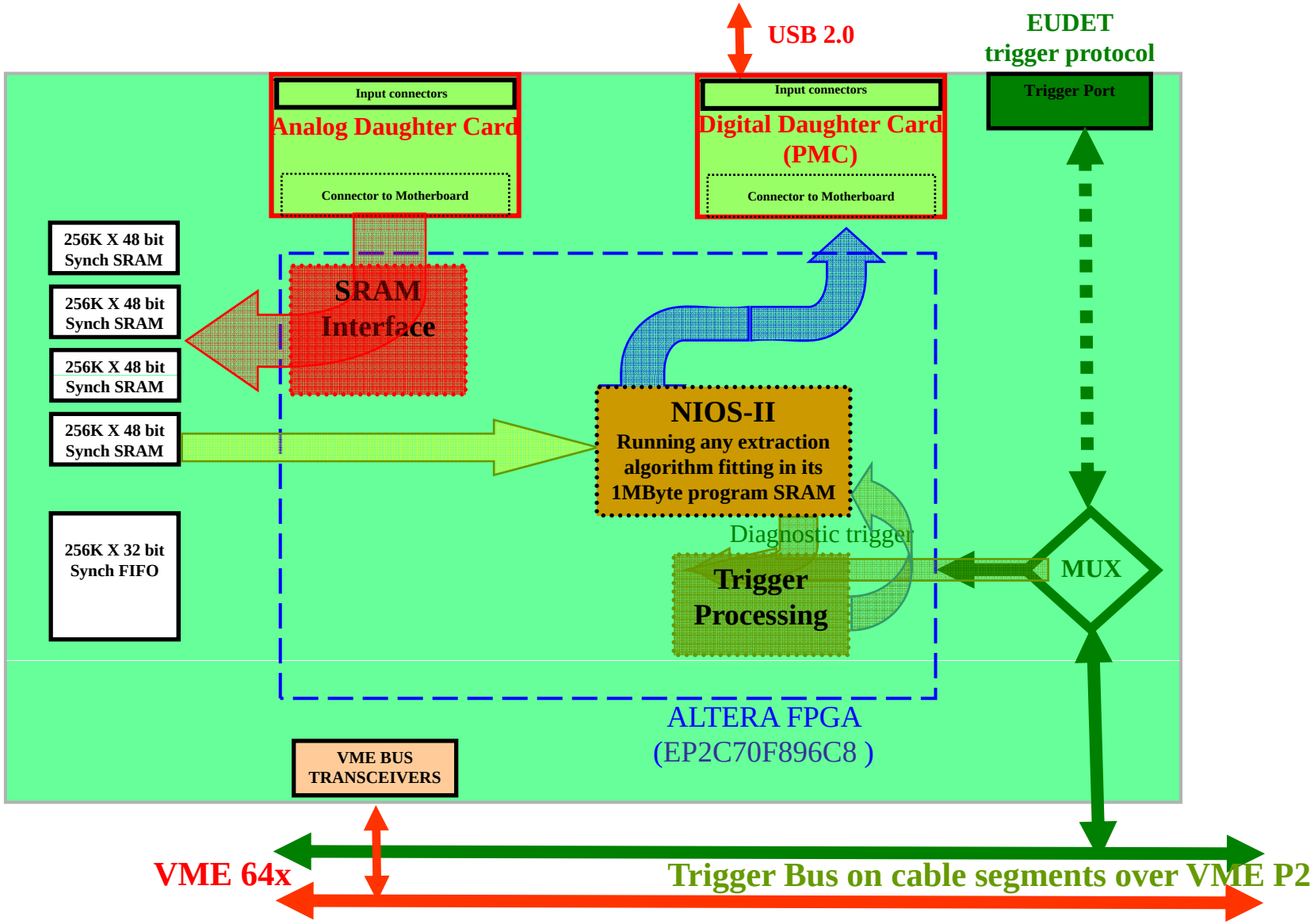
Improved Data Flow for NZS readout via VME



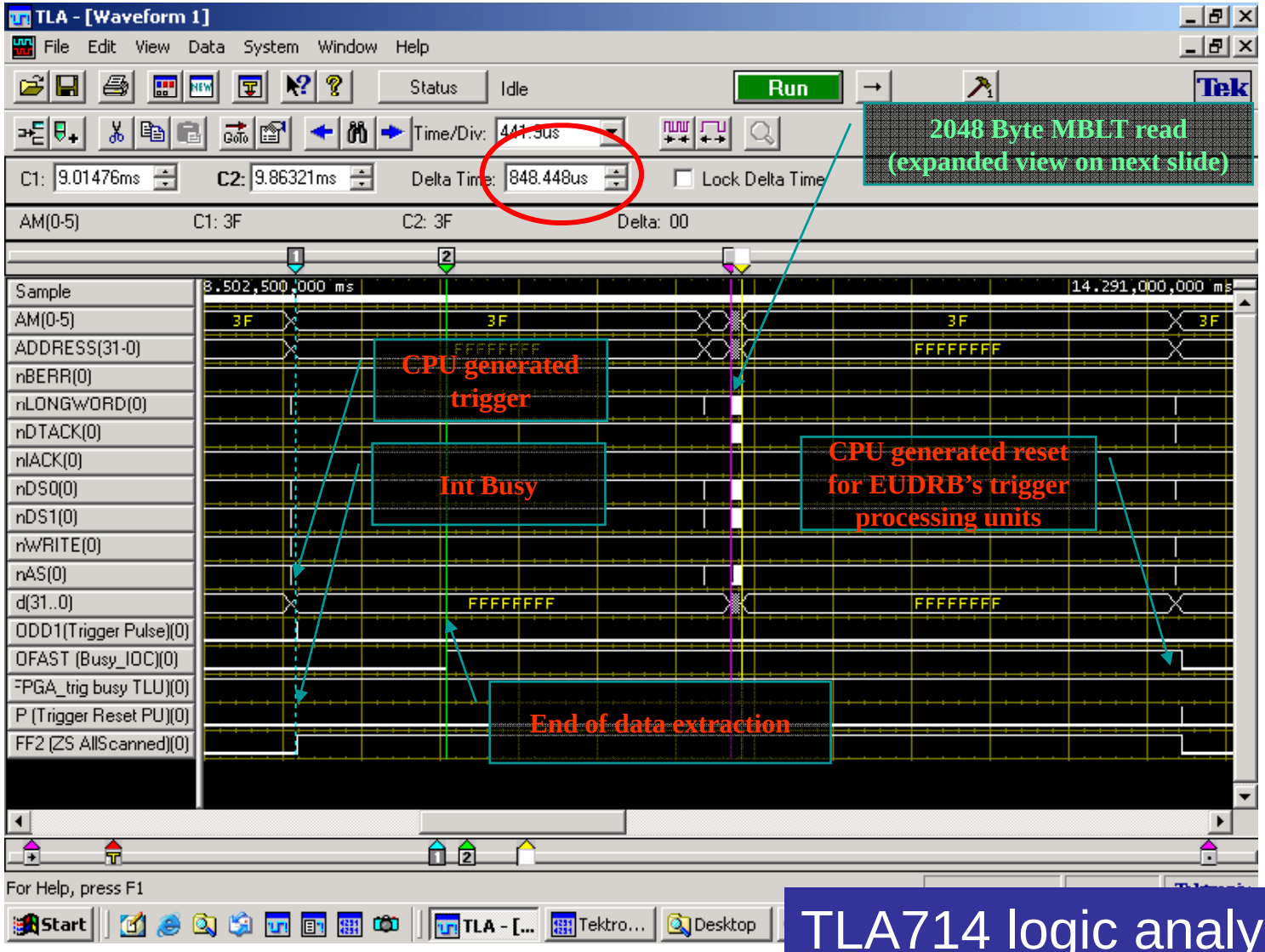
Data Flow for NZS readout via USB2.0 (for detector characterization). Current implementation:



Data Flow for NZS readout via USB2.0 (for detector characterization)
Final implementation: will work also with large sensors (MIMOSA V)

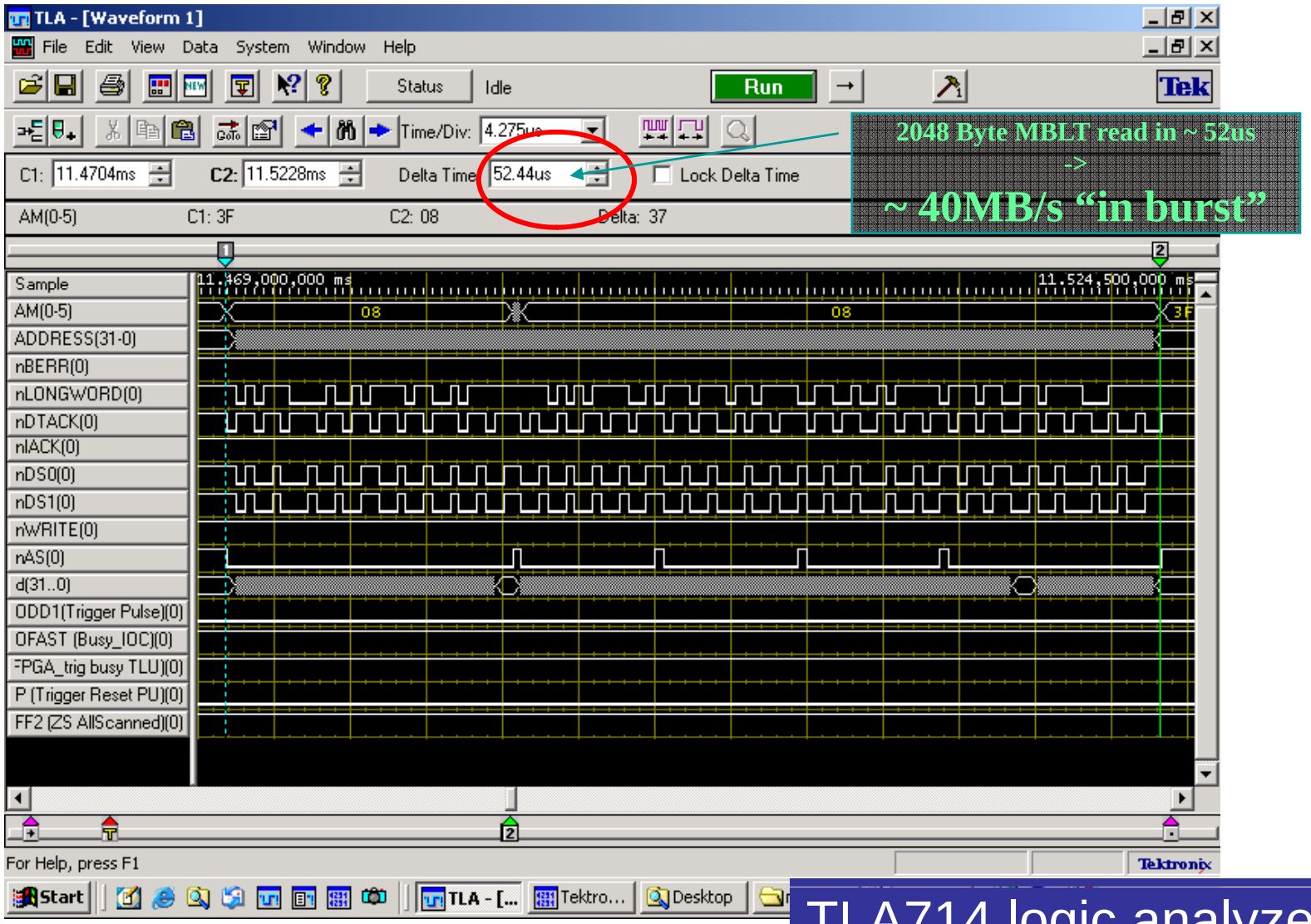


Example of VME data extraction



TLA714 logic analyzer

Results from the workshop in Ferrara (Feb 26th): the VME CPU is running the "mimoloop" program by L.Chiarelli

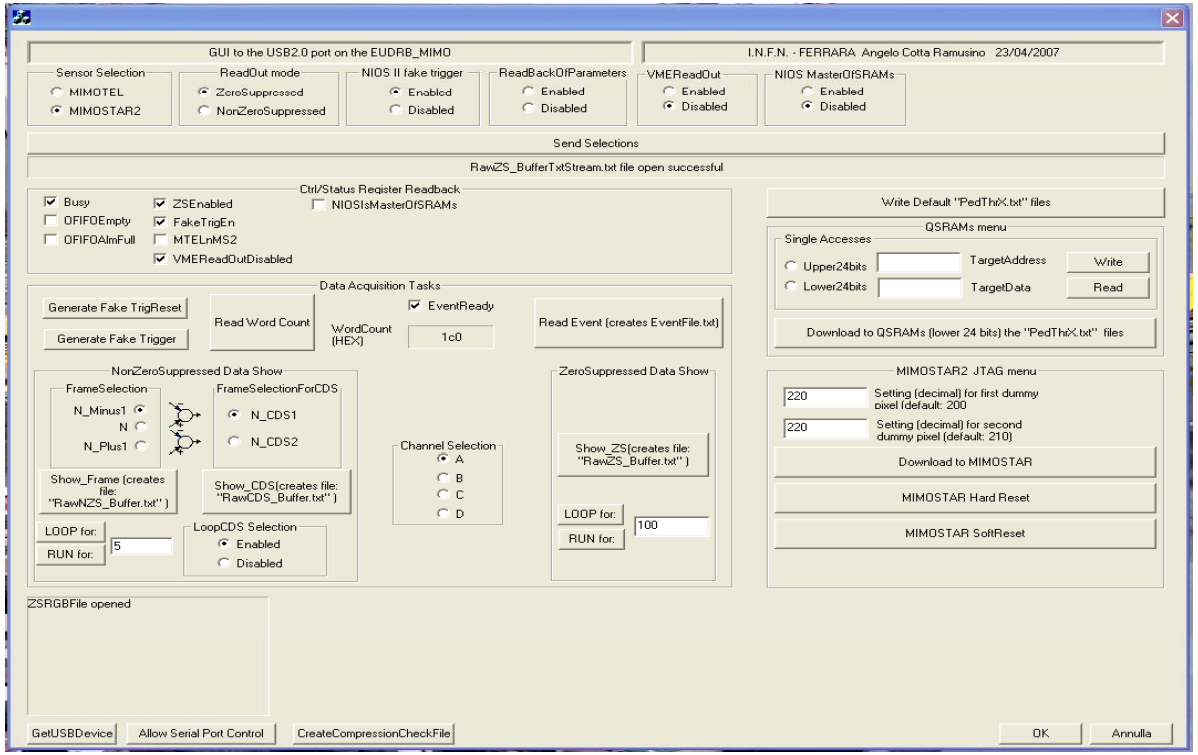


TLA714 logic analyzer

Recent results

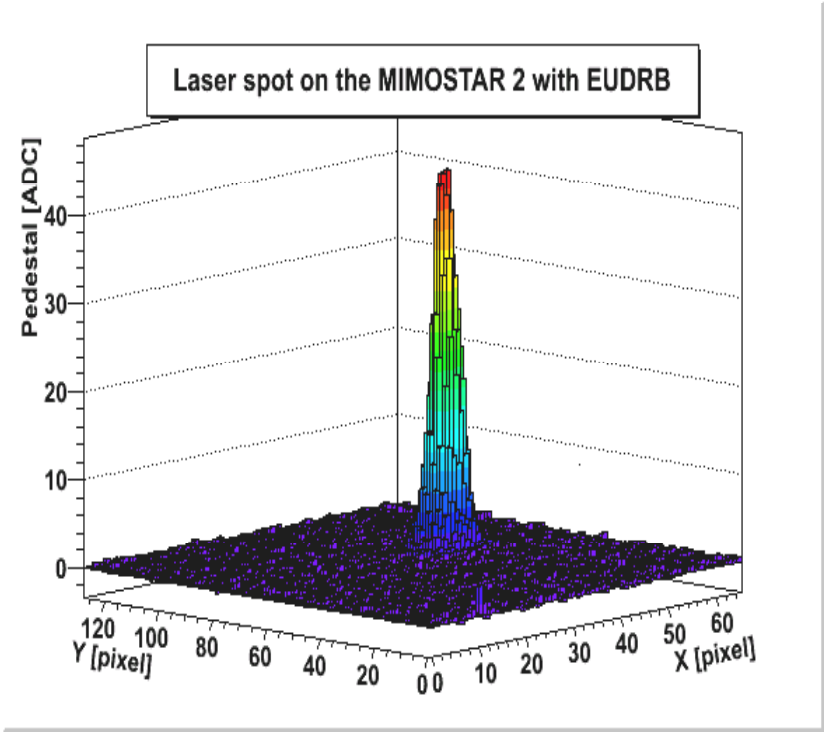
- We used a pulsed laser beam (set up in Ferrara by a team from University of Insubria – Como) to test the readout of a MimoStar2 via a EUDRB

- We developed a C++ GUI for debugging (and slow DAQ) via the USB2.0 port



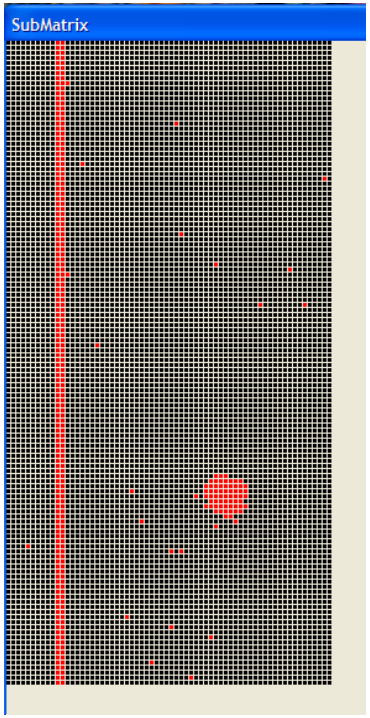
Laser beam results: Offline CDS

- laser pulses synchronized to the fake trigger generated with each loop of the acquisition routine (controlled by the GUI written in Visual C++ 6.0 and running on notebook PC)
- the routine looped 100 times, saving 3 raw frames per channel per event into a single “run” file.
- The “run” file was processed by A. Bulgheroni with “SUCIMA Pix” to perform off line CDS and statistical analysis

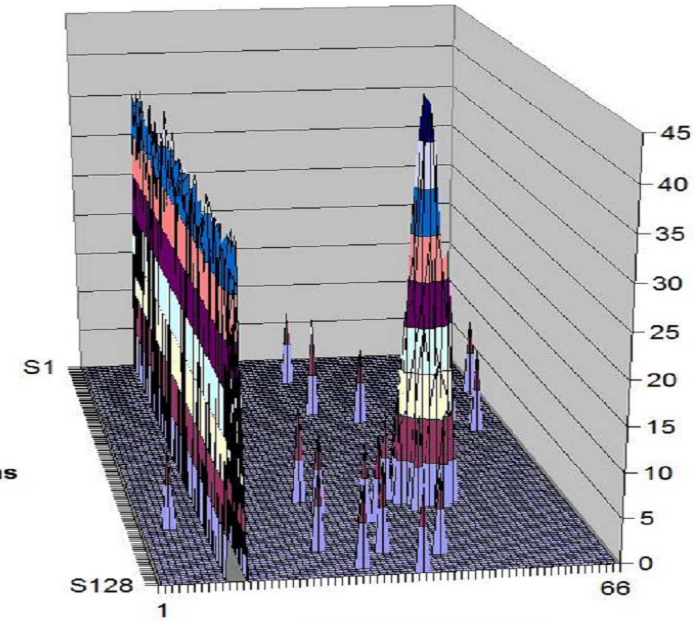


Laser beam results: on board ZS

- The image is obtained by plotting the data from a single laser event captured by the EUDRB in Zero Suppressed mode.
- The EUDRB is configured with ZS threshold=8 for each pixel and ZS pedestal=0 for each pixel except the 10th and 11th of each row (for channel A), for which the ZS pedestal=31.
- Thus in the ZS event captured one can see the “confidence” pattern of pixels 10th and 11th of each row above threshold and then, of course, the signal from the laser pulse.

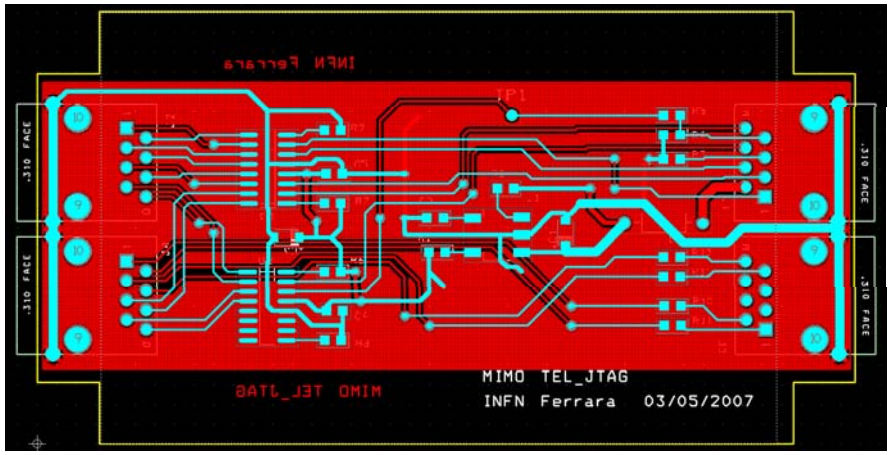


TestPattern and LaserSpot acquired in ZS mode



Integration efforts

- The EUDRB trigger interface to the TLU has been successfully tested
- 5 EUDRB delivered by manufacturer
- The DAQ team at the University of Geneva has set up a VME crate with a Motorola CPU MVME6100 as the crate controller.
- Presently the crate is hosting 3 EUDRBs:
  - 1 EUDRB configured as TLU interface board
  - The other 2 EUDRB receive trigger via a private backplane
- Preliminary loop tests achieved a sustained trigger rate of just above 1 Hz, with triggers generated by the TLU under software control as soon as the events were read from all 3 EUDRBs (3 NON-ZERO-SUPPRESSED frames per channel per board)
- Built PCBs of level adapters for Mimotel JTAG and timing signals



Conclusion/Outlook

- EUDRB fully functional
  - Zero- and non-zero-suppressed modes
  - VME64x and USB
- EUDRB tested OK on MimoStar2 with laser system
- Working to interface MimoTel and characterize the A/D section of EUDRB
- Some optimizations are under way
  - “NZS packet builder block”
  - Library of functions for the VME CPU to perform
    - Generic housekeeping of EUDRB
    - Continuous data taking with pedestal noise analysis
- Integration into DAQ of EUDET pixel telescope ongoing
  - Should be ready for beam test on June 11th

Spares

A VME-64x based DAQ card for MAPS sensors

Overview of the operation of the EUDRB card

Clock rate of the FPGA : 80MHz (40Mhz for the NIOS II processor)

- Clock rate of the A/D converter : up to 20MHz.
  => frame acquisition time: for a MIMOSA-V 1Mpixel sensor with 4 independent outputs sampled @20MHz: 262.144 * 50 ns ≈ 13 ms

-readout modes and trigger processing times

- "Full Frame" readout mode:

  The card responds to a trigger by sending out ALL RAW pixel data for at least 3 frames(*): the frame being acquired at trigger time, the preceding one and the following one, for a total of 6MB per event.
  In this readout mode the MAPS-DAQ it is allowed to stop the recording of new data from the MAPSs until the three frames selected by the trigger have been sent to the data acquisition CPU.

  The latency in the EUDRB response to a trigger can thus be no less than ONE and up to TWO frame time

  The processing time of a trigger includes:
  • the data transfer time (assuming a sustained bandwidth of 80MB/s for block transfers in 2e-VME mode each 3-frames event (6MB) can be acquired in about 1/13s per sensor)
  • the time to reset the MIMOSA V detectors at the end of the readout phase.

  The "TRIGGER_BUSY" is set as soon as the trigger is received and released when data has been transferred to the host PC

  (*) a design specification by Eleuterio Spiriti, INFN-ROMA III

continues ….

A VME-64x based DAQ card for MAPS sensors

Overview of the operation of the EUDRB card

… from previous slide

- "Zero Suppressed" readout mode:

  The card responds to a trigger by sending out a formatted block which includes an header and a trailer identifying the event number and the number of hits (the final implementation will feature the wordcount in the header).
  Processing of a trigger in this mode does not stop the scan of the detector -> no loss of data due to trigger processing

  The latency is virtually none, since the extraction of sparsified data from the pixel memories starts as soon as a trigger is received.

  The processing time of a trigger includes:
  • the extraction time (1 frame time): data is read from the pixel memories while they continue to be updated with new samples of pixel voltages. Address and data of "hit" pixels are stored into an output FIFO memory.

  • the data transfer time: the output FIFO memory is read and its contents transferred to the host PC

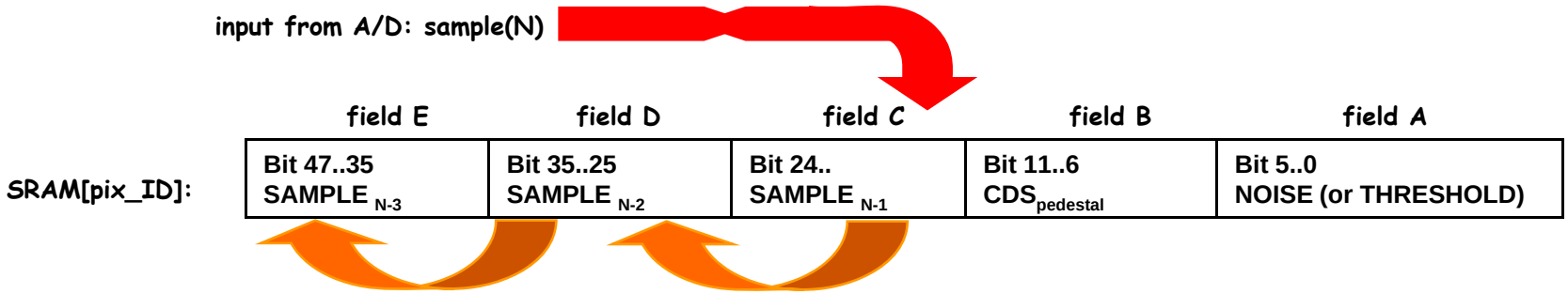
  The "TRIGGER_BUSY" is set as soon as the trigger is received and released when data has been transferred to the host PC.
  In this mode it would also be possible to release the "TRIGGER_BUSY" right after filling the output FIFO, overlapping the readout phase of a trigger with the processing of the next

A VME-64x based DAQ card for MAPS sensors

Additional information

Organization of the SRAM memories for pixel data and CDS operation

The packet contains the data only for those pixels whose signal was found above threshold after ON-LINE CDS and (pedestal+noise) subtraction.
Each SRAM location for pixel data is organized as follows
(the graphics shows the data flow for updating the pixel data with Frame(N)'s new sample):



When the board operates in "sparsified" mode the CDS (correlated double sampling) is made according to the following rule:
"
1) When the trigger signal arrives, let's say in the middle of sampling Frame N's data, the sampling controller on the FPGA latches the address of the pixel whose data is currently being updated, lets' say: pix_ID_{Trig}.
2) Then it evaluates the pedestal subtracted CDS as follows:

$$CDS_{ped\_sub} = sample_N(pix\_ID) - sample_{N-1}(pix\_ID) - CDS_{pedestal}$$

storing the result to an embedded FIFO if the pedestal subtracted CDS is above threshold. $Sample_{N-1}(pix\_ID)$ is fetched from field C of the SRAM[pix_ID] contents
3) Step 2 is repeated for all pixels until:

$$pix\_ID_{Last} = pix\_ID_{Trig} - 1$$

Eventually pix_ID has overflowed and restarted from 0. By then, the contents of the field C at all locations of the SRAMs have been updated to $sample_N(pix\_ID)$.
This is OK, since after the rollover, the quantity to evaluate is:

$$CDS_{ped\_sub} = sample_{N+1}(pix\_ID) - sample_N(pix\_ID) - CDS_{pedestal}$$

i.e. again the new sample from the A/D converter minus the content of field C in the active location pointed by pix_ID
"

continues….