

**EUDRB: A VME-64x based DAQ card for  
MAPS sensors in EUDET.**

**STATUS REPORT**

**Angelo Cotta Ramusino,  
Davide Spazian, Lorenzo Chiarelli**



# Scope of the work: a readout for a pixel beam-telescope

- ❖ The EUDET JRA1 project: a pixel-based beam telescope for the test of a pixel sensor ( and other detectors?) suitable for the ILC
- ❖ Plan is to use MAPS sensor MimoStar 3, now in development stage. Existing detector is Mimosa 5: 1 Mpixel, analog output to be digitized in 12 bits; pixel size is  $17\mu\text{m} \times 17\mu\text{m}$ .
- ❖ A "frame" contains 12 Mbits; at least 2 –better 3 – frames NEED to be read to extract signal ==> 36 Mbits per sensor per event. MASSIVE data.
- ❖ Analog readout of pixels takes place at maximum 20MHz: this means that in a single line the analog voltage of a pixel lasts 50ns before it is replaced by the next pixel.
- ❖ The sum of the times of reading and writing data limits DAQ at few tens Hz.
- ❖ 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, which can implement real-time sparsification on the readout board..
- ❖ ...AND, which uses the fastest possible VME readout to reduce dead times.

# A VME-64x based DAQ card for MAPS sensors

## Overview of the operation of the EUDRB card

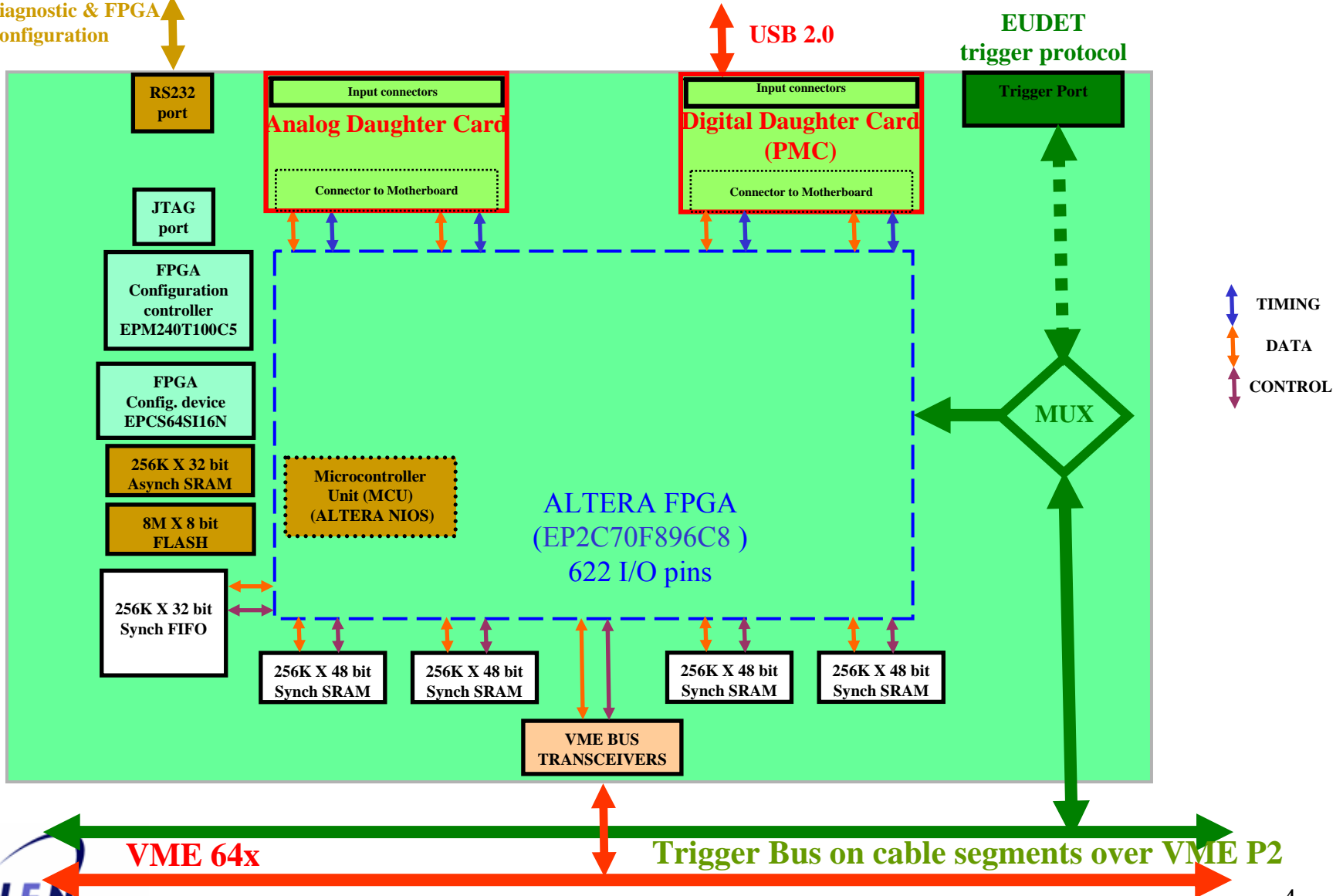
### Basic features:

- a) VME64X slave interface
- b) USB2.0 interface
- d) Interface to the EUDET trigger bus ( trigger request, event number, busy)
- e) 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
- e) NIOS II, 32 bit "soft" microcontroller implemented in the CycloneII FPGA. It handles tasks like:
  - on board MODULARITY:
  - **One** mother board (**EUDRB\_MOBO**), built around an ALTERA CycloneII EP2C70F896C8N FPGA and hosting the core resources (SRAMs, FIFOs, VME64x interface, trigger port, diagnostic UART)
  - **One** analog daughter card (**EUDRB\_DCA**), based on the successful LEPSI and SUCIMA designs
  - **One** digital daughter card (**EUDRB\_DCD**), featuring a standard "PCI Mezzanine Card" interface to the mother board. It drives/receives control signals for the detectors and it features a USB 2.0 link
  - diagnostics
  - on-line calculation of pixel pedestal and noise (not interfering with data taking operations)
  - remote configuration of the FPGA via RS-232, VME, USB2.0

# A VME-64x based DAQ card for MAPS sensors

## Overview of the operation of the EUDRB card

RS232 port for Diagnostic & FPGA Configuration



# 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:  $262.144 * 50 \text{ ns} \approx 13 \text{ ms}$

-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 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 times.

The processing time of a trigger includes:

- the data transfer time ( 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 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 a 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

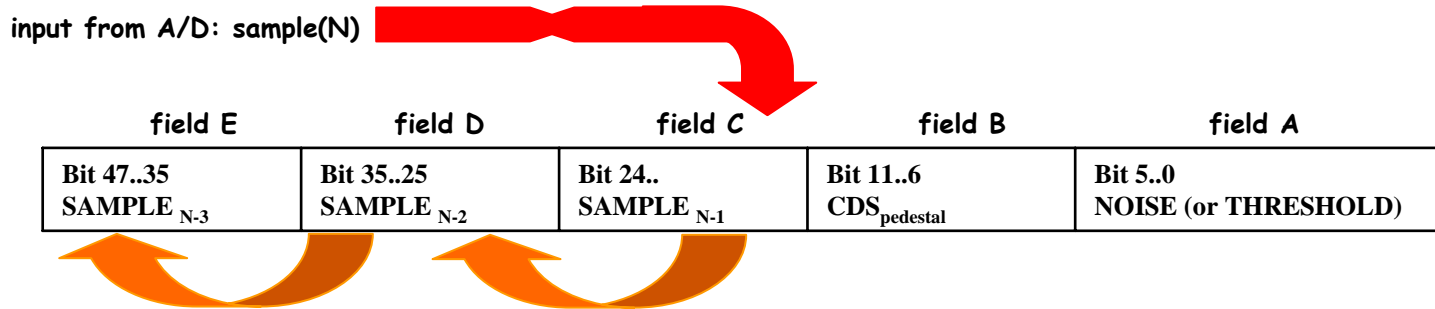
## Additional information

### 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, let's say:  $\text{pix\_ID}_{\text{Trig}}$ .
- 2) Then it evaluates the pedestal subtracted CDS as follows:

$$\text{CDS}_{\text{ped\_sub}} = \text{sample}_N(\text{pix\_ID}) - \text{sample}_{N-1}(\text{pix\_ID}) - \text{CDS}_{\text{pedestal}}$$

storing the result to an embedded FIFO if the pedestal subtracted CDS is above threshold.  $\text{sample}_{N-1}(\text{pix\_ID})$  is fetched from field C of the  $\text{SRAM}[\text{pix\_ID}]$  contents

- 3) Step 2 is repeated for all pixels until:

$$\text{pix\_ID}_{\text{Last}} = \text{pix\_ID}_{\text{Trig}} - 1$$

Eventually  $\text{pix\_ID}$  has overflowed and restarted from 0. By then, the contents of the field C at all locations of the SRAMs have been updated to  $\text{sample}_N(\text{pix\_ID})$ .

This is OK, since after the rollover, the quantity to evaluate is:

$$\text{CDS}_{\text{ped\_sub}} = \text{sample}_{N+1}(\text{pix\_ID}) - \text{sample}_N(\text{pix\_ID}) - \text{CDS}_{\text{pedestal}}$$

i.e. again the new sample from the A/D converter minus the content of field C in the active location pointed by  $\text{pix\_ID}$

continues....

# A VME-64x based DAQ card for MAPS sensors

## Status report

### Milestones:

- Fitting and pin assignment (622 I/Os) of FPGA design: **March 2006** (A.Cotta Ramusino, D. Spazian)
- Schematic of mother board: **May 10 2006** (A.Cotta Ramusino)
- Schematic of daughter cards: **May 23rd 2006** (A.Cotta Ramusino)
- Order to ARTEL for production of EU DRB\_MOBO: **May 29th 2006**
- Order to ARTEL for production of EU DRB\_DCA, EU DRB\_DCD: **June 5th 2006**
- Setup of a MVME6100 VME CPU in a VME64x crate and preliminary tests on VME block transfer with a CAEN v1290 (Multihit TDC) as a target device: **June 2006**. (L. Chiarelli, A. Cotta Ramusino)
- Delivery of boards by ARTEL: **July 26th 2006**
- Hardware/Firmware functionality tests (as of **Sept 7th 2006**):
  - NIOS II: **OK** (A. Cotta Ramusino)
  - pixel memories: **not OK for one quadrant** (A. Cotta Ramusino): problems with the integrity of a few bits
  - output FIFO: **OK** (A. Cotta Ramusino)
  - USB 2.0 link: **OK** (A. Cotta Ramusino, D. Spazian)
  - VME interface: **OK** (L.Chiarelli,A. Cotta Ramusino)
  - analog card: **OK** (A. Cotta Ramusino)
  - digital interface to MIMOSA V detector (timing and controls): **OK** (A. Cotta Ramusino, M.Jastrzab)

### Work In Progress:

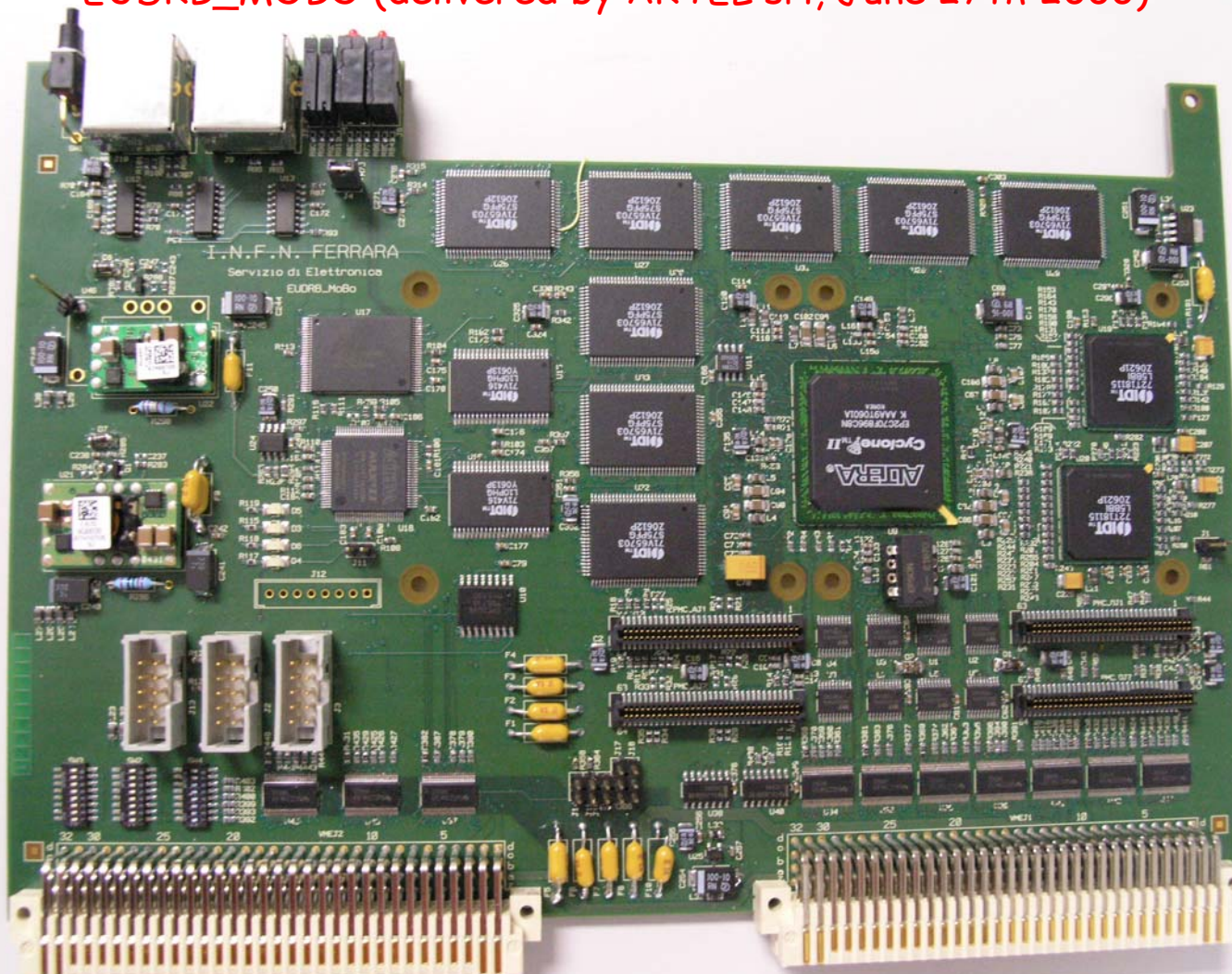
- Debugging of design with a MIMOSA detector: (D. Spazian, M.Jastrzab, A.Cotta Ramusino)
- Investigation and workaround for the of the data memory problem: (A.Cotta Ramusino)
- Testing communication between the host PC and the NIOS through the USB 2.0 and the VME interfaces: (L.Chiarelli,A. Cotta Ramusino,D.Spazian)
- Testing 2eVME data transfers (L.Chiarelli,A. Cotta Ramusino)
- Writing code for the MVME6100 CPU for EU DRB configuration and diagnostics (L.Chiarelli)
- Testing the TLU interface and trigger distribution among EU DRBs: (A.Cotta Ramusino, D.Spazian)
- JTAG interface to MIMOSTAR: (A.Cotta Ramusino)
- Changing present design to adapt to MIMOSTAR operation: (A.Cotta Ramusino)



# A VME-64x based DAQ card for MAPS sensors

## Status report

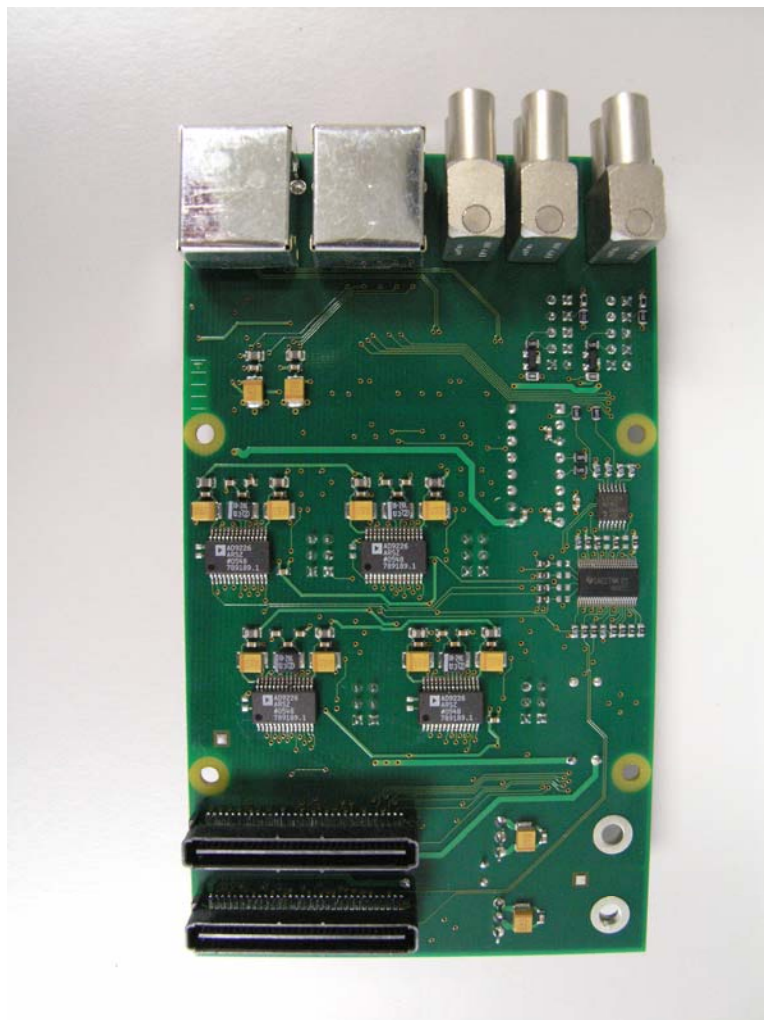
EUDRB\_MOBO (delivered by ARTEL srl, June 27th 2006)



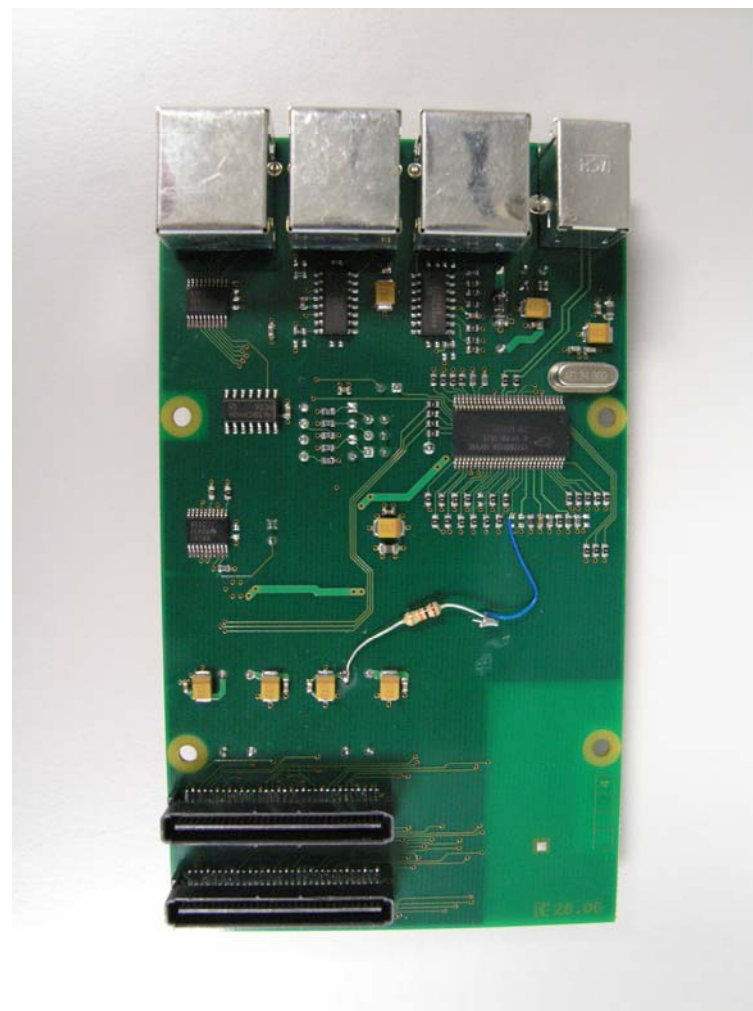
# A VME-64x based DAQ card for MAPS sensors

## Status report

EUDRB\_DCA & EUDRB\_DCD (delivered by ARTEL srl, June 27th 2006)



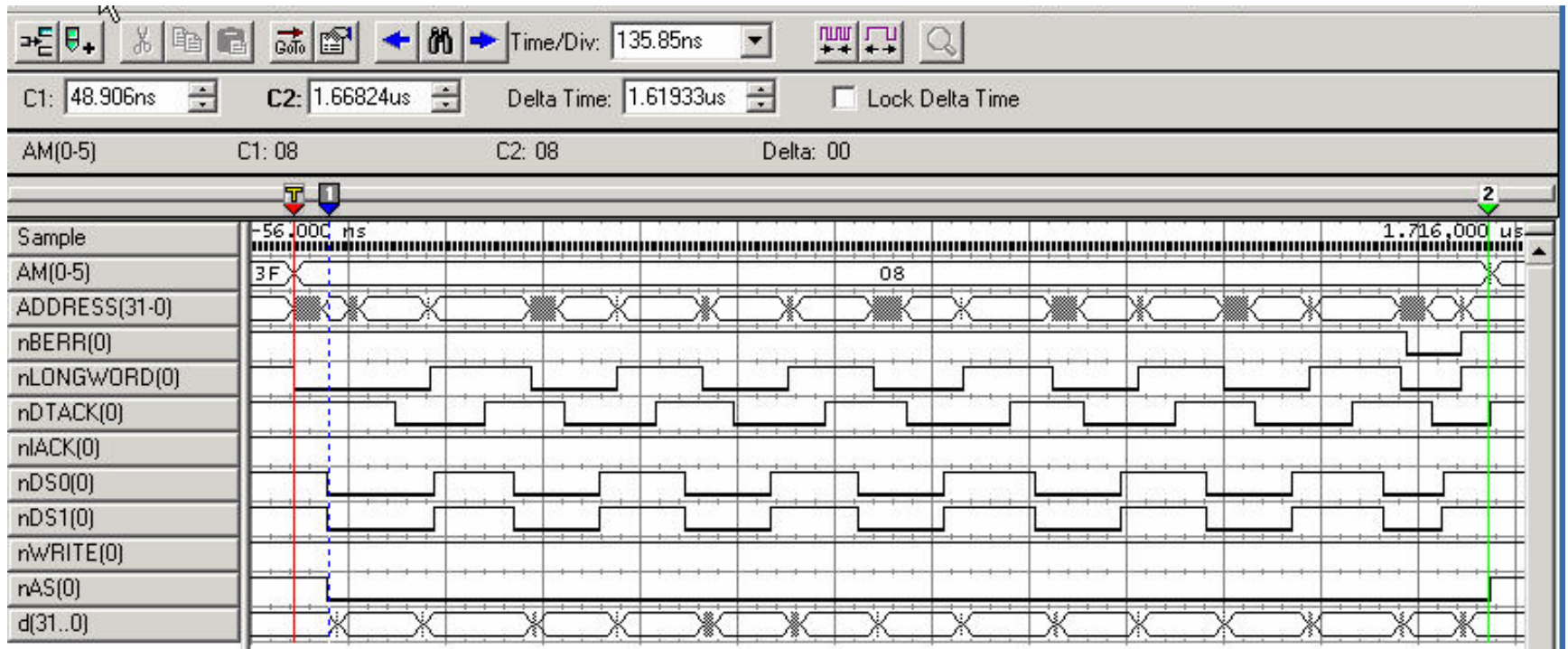
EUDRB\_DCA: Analog daughter card  
(bottom view)



EUDRB\_DCD: Digital daughter card  
(bottom view)

# A VME-64x based DAQ card for MAPS sensors

## Some test results: VME interface



L.Chiarelli, A.Cotta R., Sept 1st 2006

Screenshot from a TLA714 logic analyzer: an MBLT block read of a 40 byte packet, initiated by the CPU when nAS goes low and terminated when the EUDRB shows, by driving nBERR low, that it holds no more data to be read.

```
sh-2.05a# ./test2 -r0x30400000 -a3 -m32
```

```
Read output buffer 48773300
```

```
Read output buffer 0
```

```
Read output buffer aaaaaaaa
```

```
Read output buffer 55555555
```

```
Read output buffer 2000010
```

```
Read output buffer 4000020
```

```
Read output buffer 6000030
```

```
Read output buffer 8000040
```

```
Read output buffer 54773300
```

```
Read output buffer 0
```

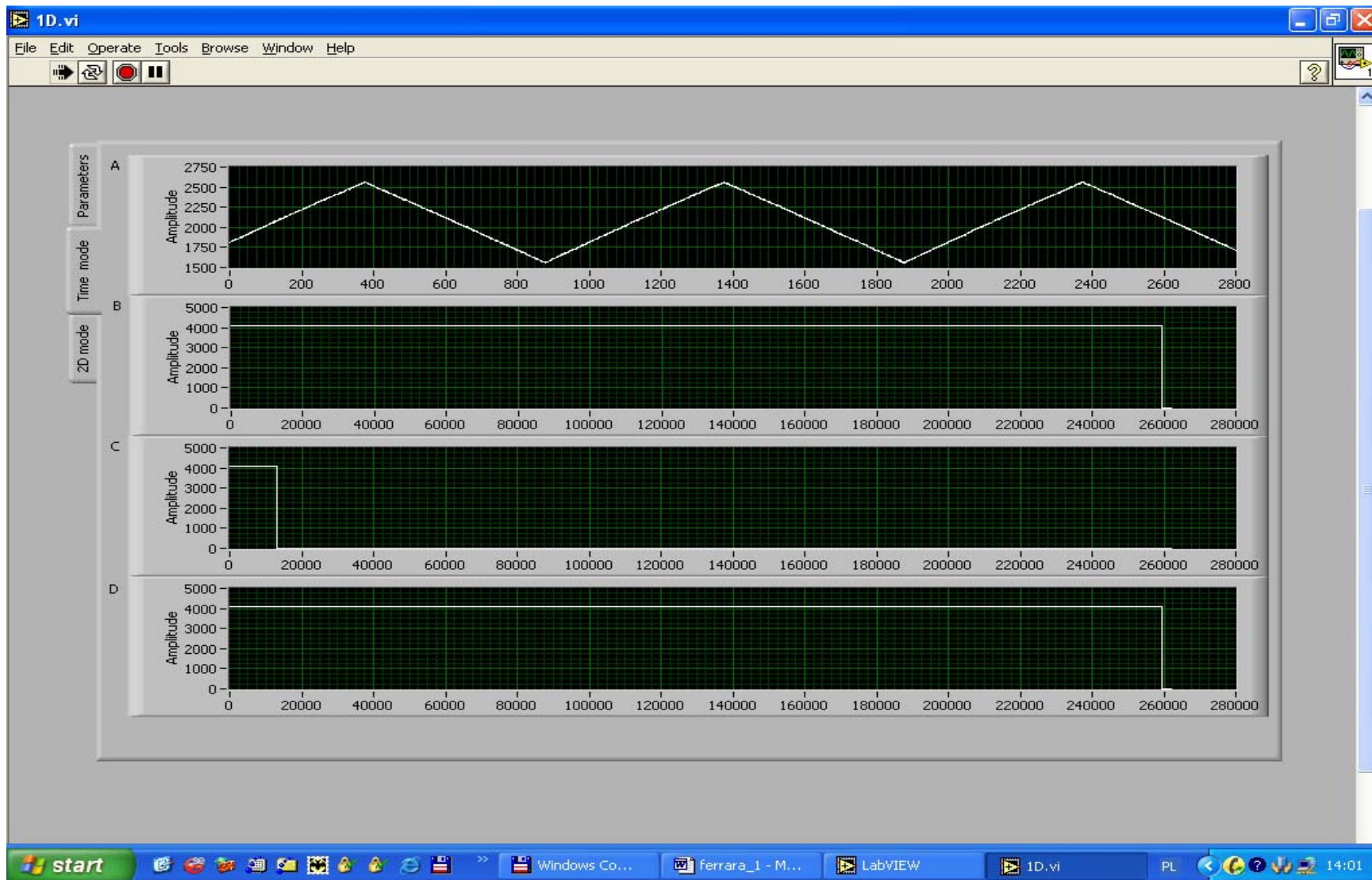
**Header**

**Trailer**

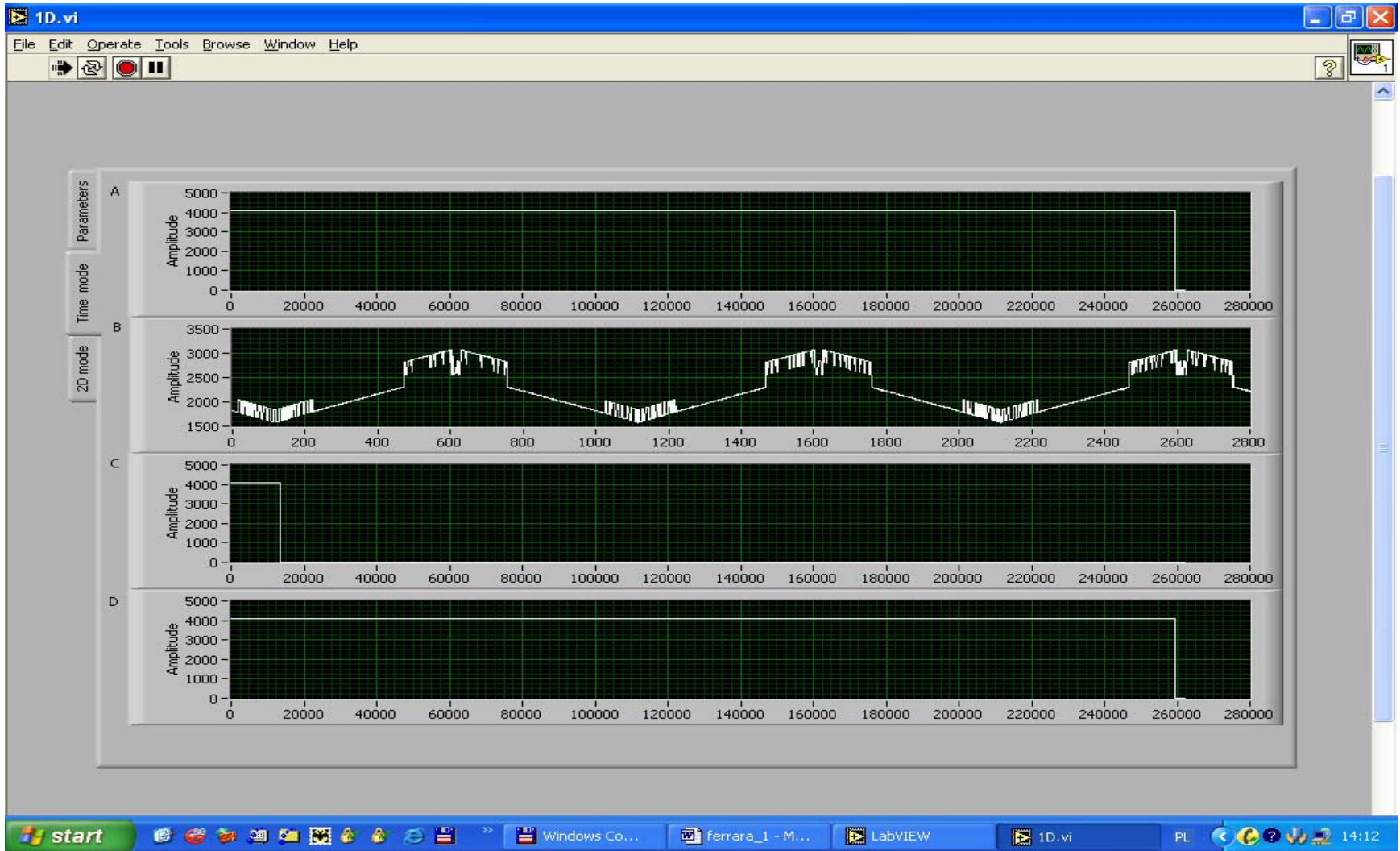
# Since the JRA1 September 06 meeting

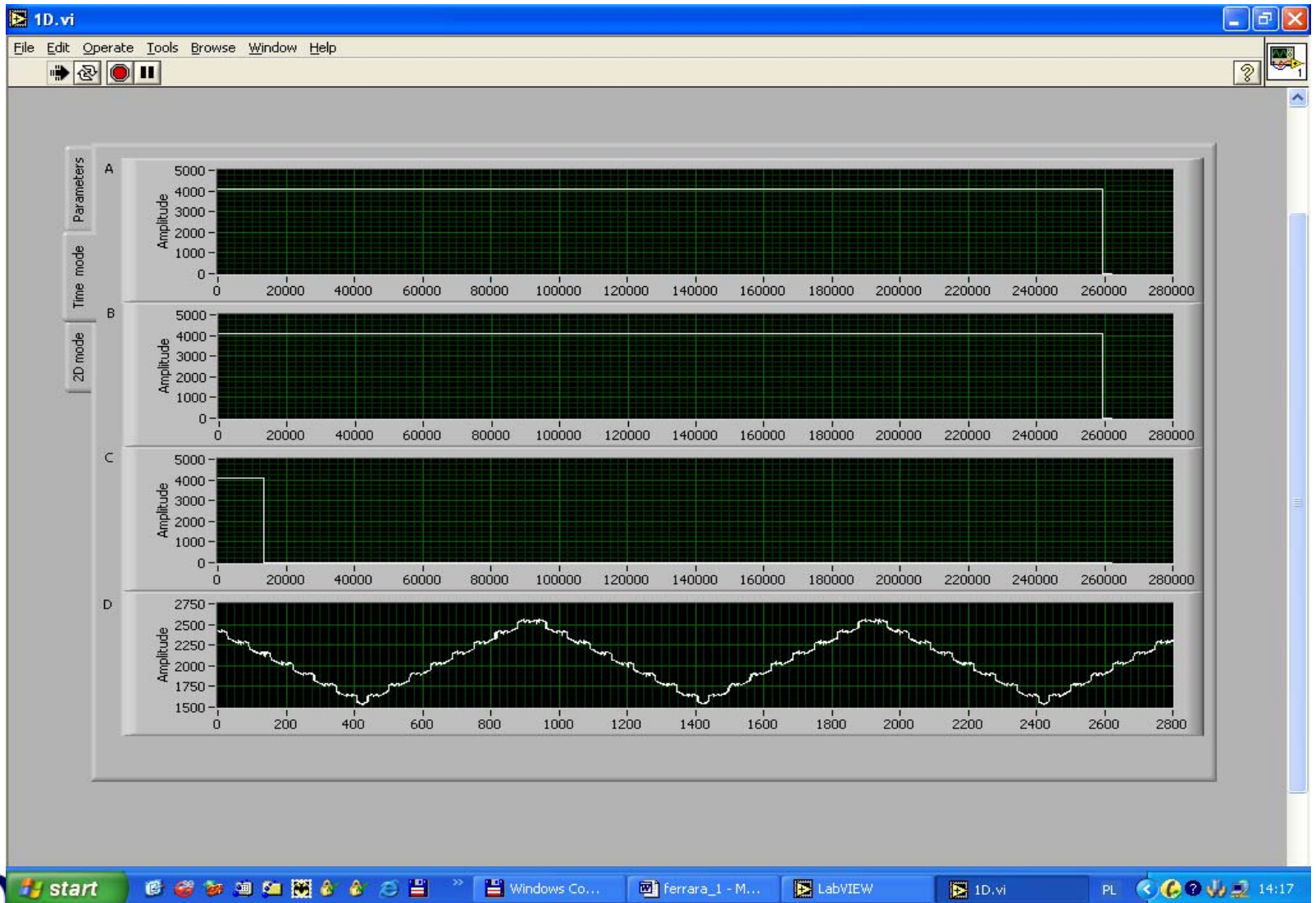
- Been working on the FPGA program: data acquisition and storage in memory, readout from memory via USB2.
- The program **exists**, is already written, and contains two versions: writing of full data and of sparsified data.
- Debugging of the acquisition and readout of full frames, i.e. **non\_sparsified** data
- Data have been simulated by sending to input of analog daughter card output from a pulse generator
- Sent Mimoso configuration from pc to FPGA – and then read it back – via USB
- Start readout of data from ADCs and store them in RAMs
- Generate a fake trigger and then start the procedure of completing readout of 3 frames, freeze the writing of new data in memory and write data out via USB.
- Digitized data at 5MHz (at first try). Data readout achieved so far at 12 Mbytes/s. Data displayed on a GUI.

This is the good news....



...and this is the less good one





## WorkPlan:

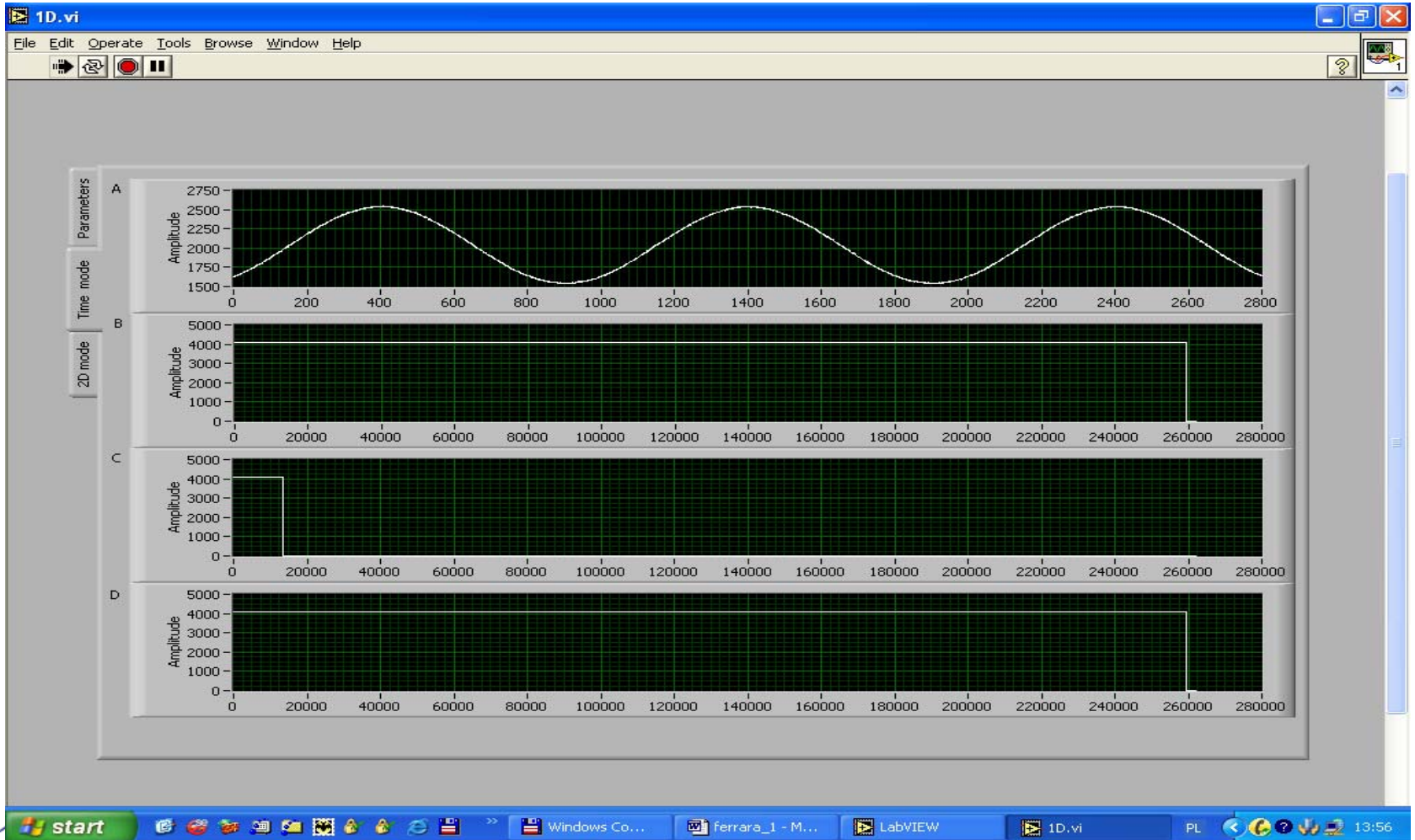
- ❖ Commission USB interface (Done)
- ❖ Commission VME interface (Fe) – 10/2006
- ❖ Read MIMOSA5 in full transparent mode – 11/2006
- ❖ Interface Mimo\*2 (Fe, Mi) – end 2006
- ❖ Commission (debug?) FPGA sparsification firmware (signal\_over\_threshold) – end 2006
- ❖ Compare with clustering-based sparsification – spring 2007
- ❖ Interface to Mimo\*3 (baseline of the “demonstrator” EUDET telescope) – spring 2007
- ❖ VME readout of several MIMO\*3 chips in sequence – summer 2007
- ❖ commissioning of the “demonstrator” EUDET telescope at DESY – end summer 2007



# Conclusion

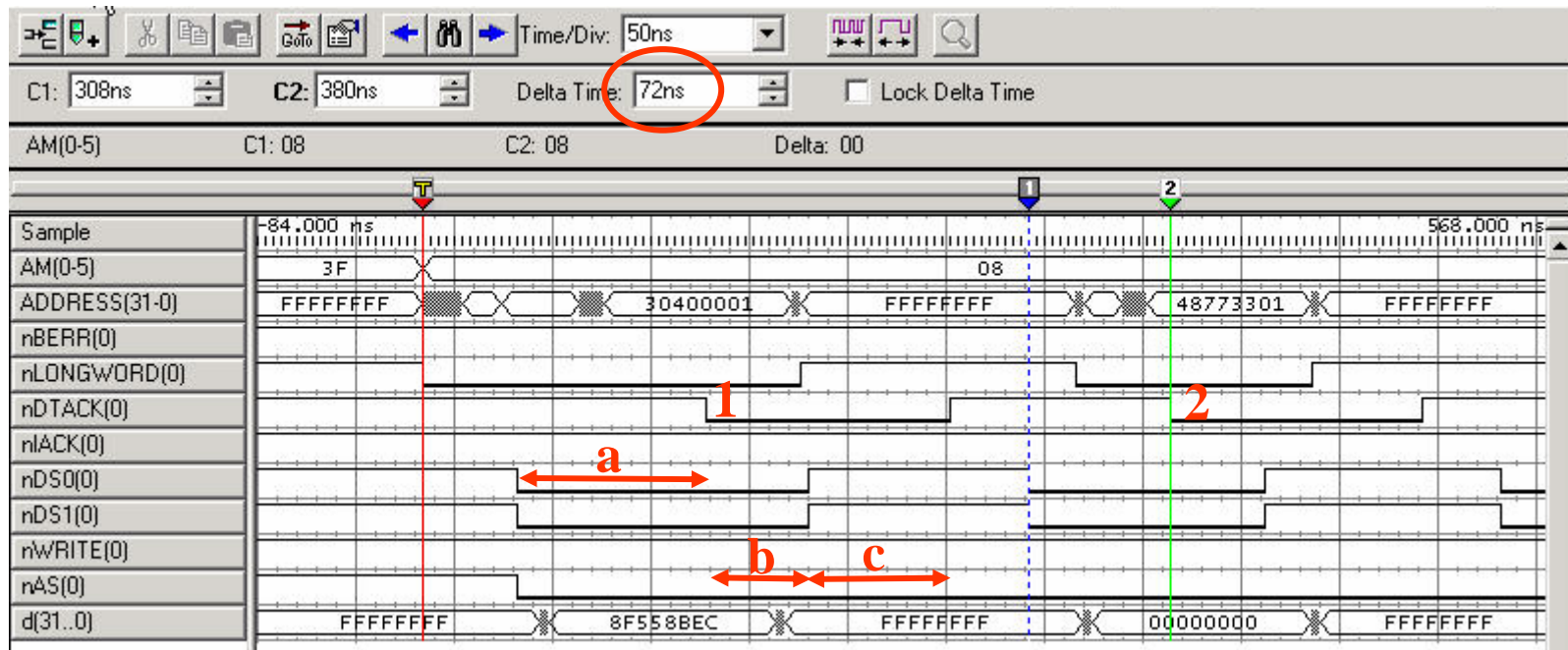
- DAQ board prototypes have been made, FPGA software has been written, and both are in an advanced stage of test and debugging.
- Tested the whole readout chain from the encoding in the ADCs to the data transmission to a pc via USB and an ad hoc GUI.
- Proved overall good performance in one channel, other 3 channels still to be debugged.
- 12 Mbyte/s writeout frequency – with a system still in debugging mode – achieved so far.
- Problems still to be addressed: program gets stuck after (correct) reading of one event. And – bad coding of ADC channels 2, 3 and 4.
- *Have plan for completion of project in summer 2007.*

# SPARE



# A VME-64x based DAQ card for MAPS sensors

## Some test results: VME interface



L.Chiarelli, A.Cotta R., Sept 1st 2006

Detail of the MBLT block read of 64 bytes.

The first nDTACK = low strobe shows that the slave has attached (1); the following nDTACK (2) strobe indicates valid data on both the 31 address lines + nLONGWORD and the 32 data lines (0x4877330000000000).

The cycle time between two successive transfers is 240ns (-> 33 MB/s peak transfer rate to be compared to the maximum theoretical rate of 80MB/s claimed by the standard). This time is made up of

- (a) Delay from nDS0,nDS1 active to nDTACK active: about 70ns. It is determined by the EUDRB.
- (b) Delay from nDTACK active to nDS0,nDS1 unactive : about 40ns. It is determined by the CPU
- (c) Delay from nDS0,nDS1 unactive to nDTACK unactive : about 70ns. It is determined by the EUDRB and it can be improved with a different coding for the VME interface in the FPGA.

By gaining an additional 40ns in the response (c) of the EUDRB board we could probably achieve 40MB/s peak in MBLT mode and double that in 2e-VME transfers