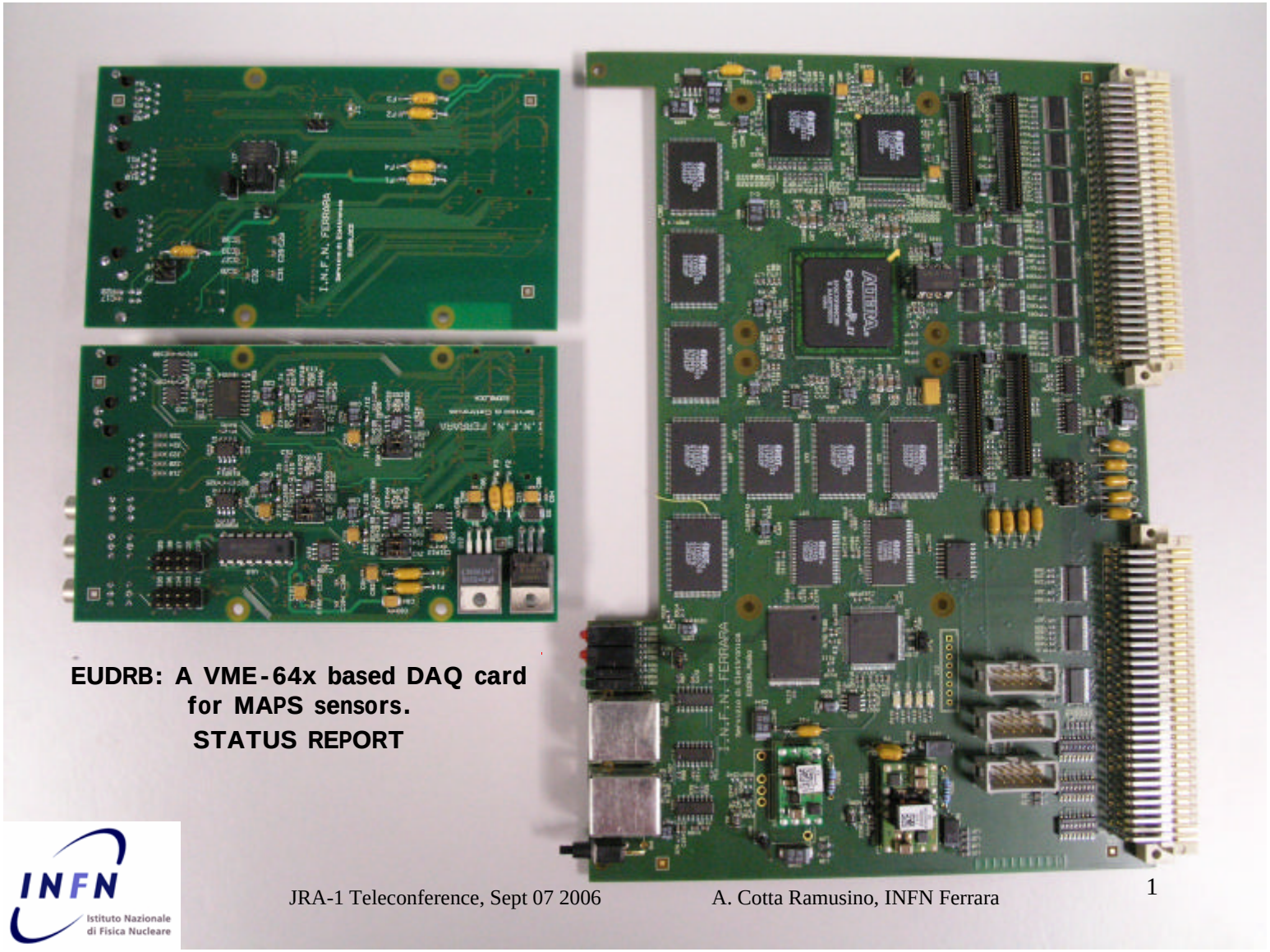


EUDRB: A VME-64x based DAQ card for MAPS sensors.

STATUS REPORT

A VME-64x based DAQ card for MAPS sensors

Overview of the operation of the EUDRB card

Basic features:

a) 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

b) VME64X slave interface

c) 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.

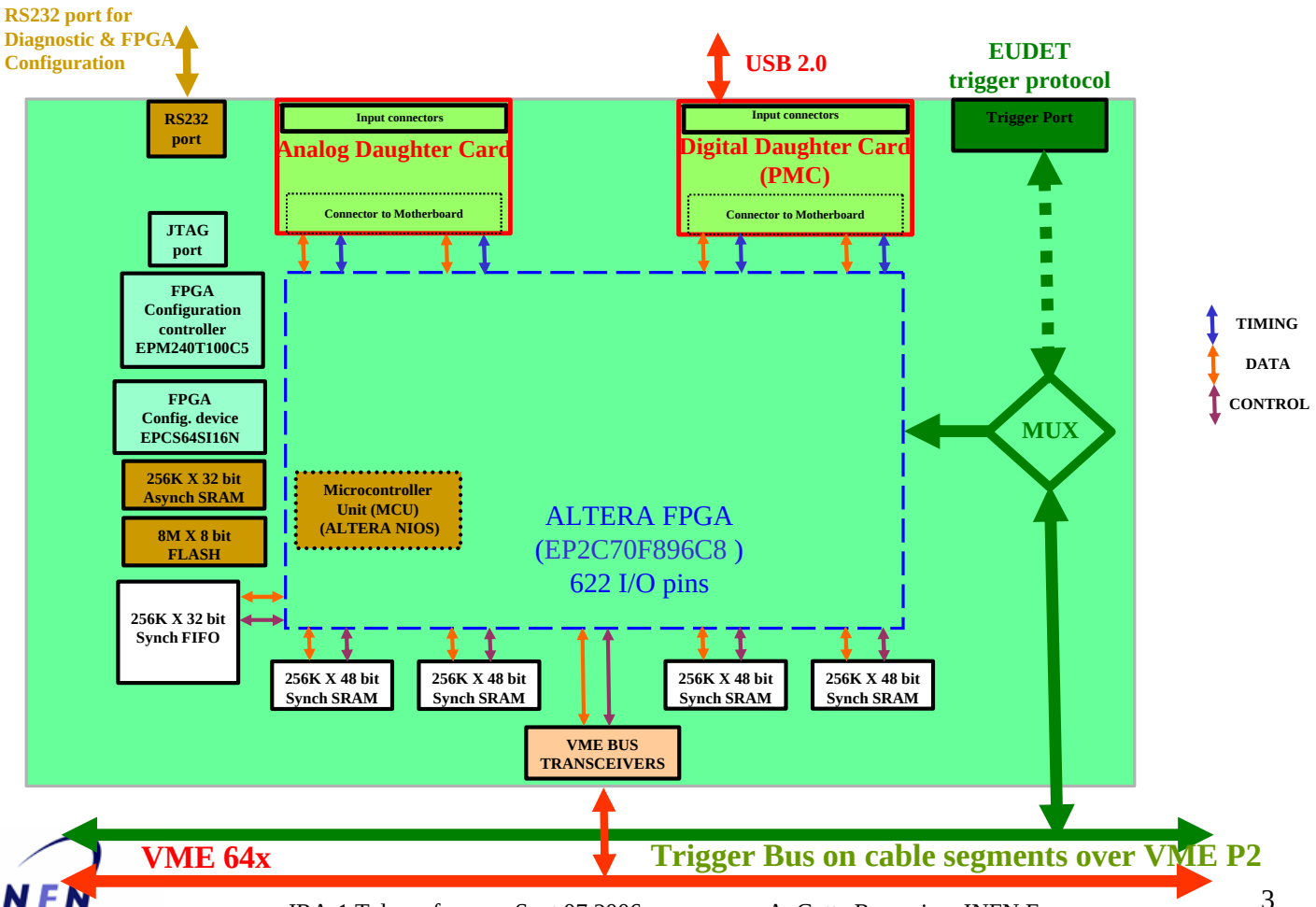
-Non Zero Suppressed readout of multiple frames for debugging or off-line pedestal and noise calculations

e) NIOS II, 32 bit "soft" microcontroller implemented in the CycloneII FPGA. It handles tasks like:

- on board 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



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 ....

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

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 EUDRB_MOBO: May 29th 2006
- Order to ARTEL for production of EUDRB_DCA, EUDRB_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 EUDRB configuration and diagnostics (L.Chiarelli)
- Testing the TLU interface and trigger distribution among EUDRBs: (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

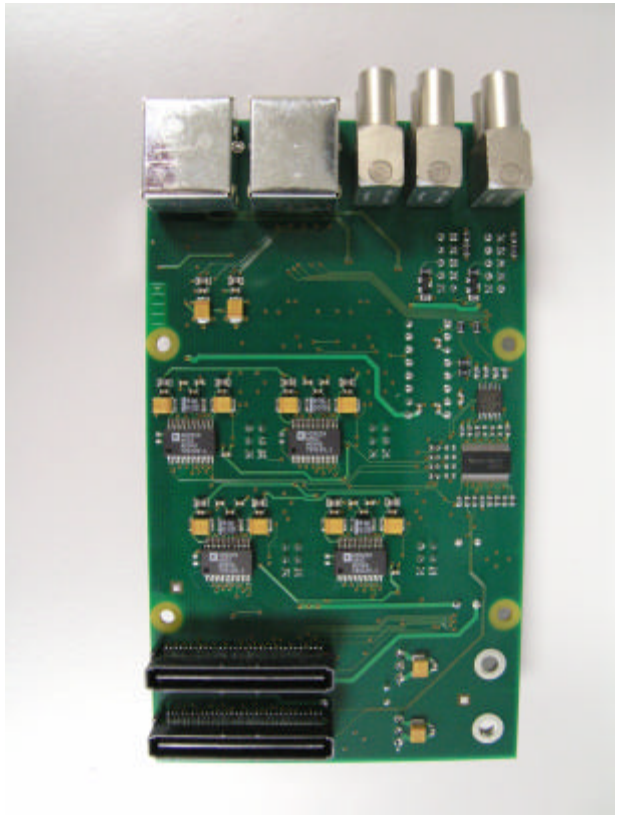
Status report

EUDRB_MOBO (delivered by ARTEL srl, June 27th 2006)

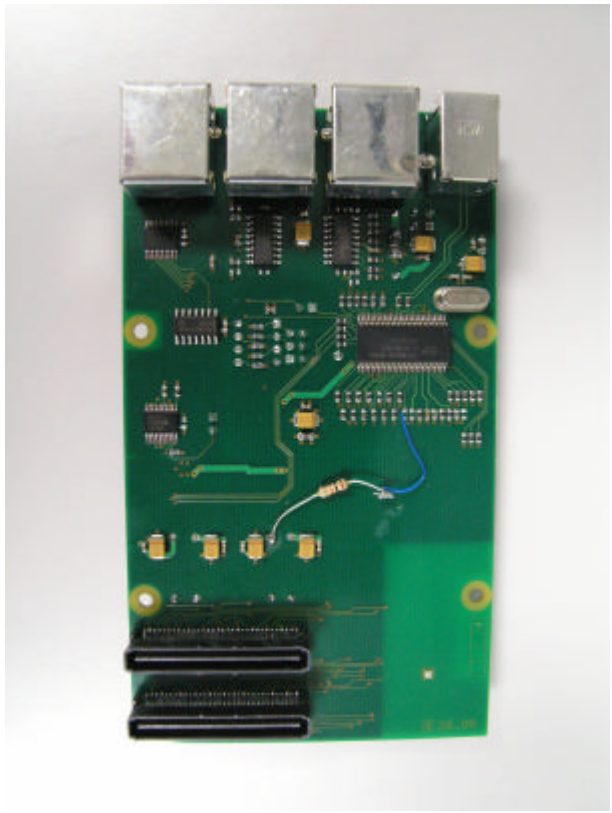
A VME-64x based DAQ card for MAPS sensors

Status report

EUDRB_DCA & EUDRB_DCD (delivered by ARTEL srl, June 27th 2006)

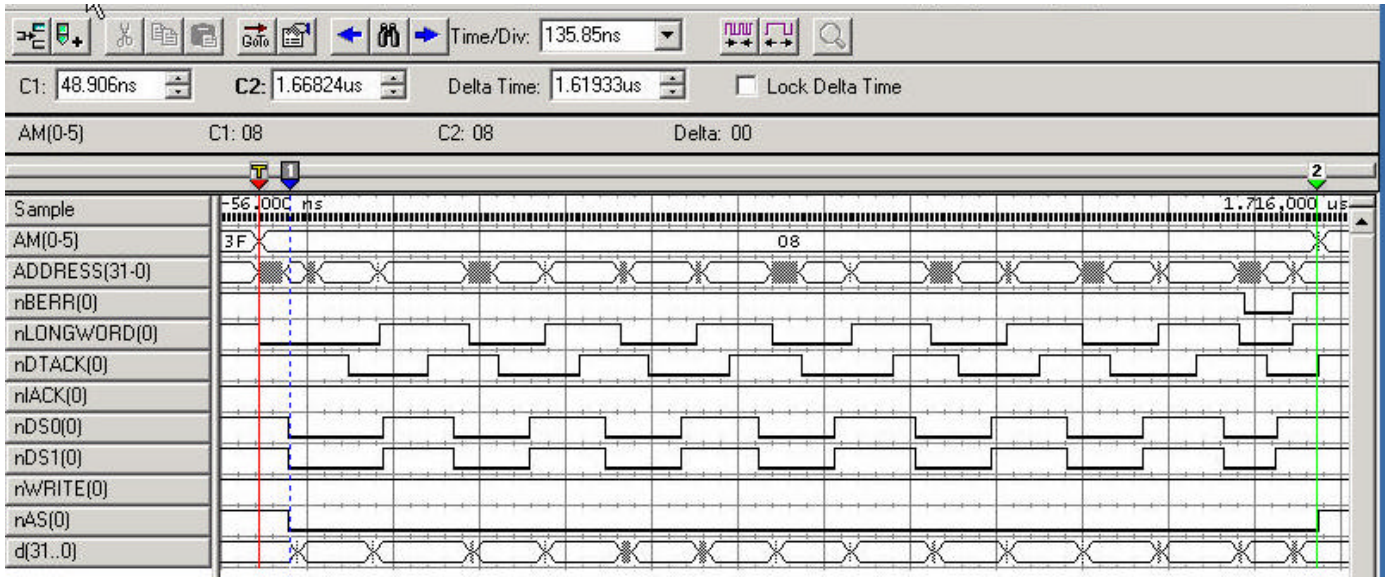


EUDRB_DCA: Analog daughter card
(bottom view)



EUDRB_DCD: Digital daughter card
(bottom view)

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.

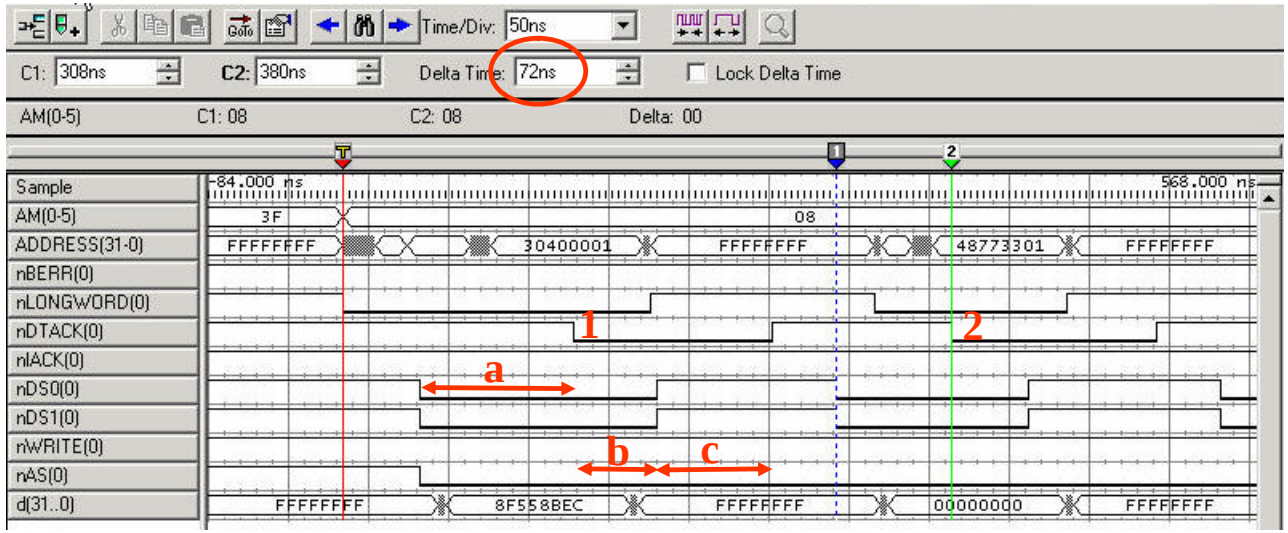
```
sh-2.05a# ./test2 -r0x30400000 -a3 -m32
Read output buffer 48773300
Read output buffer 0
Read output buffer aaaaaaaa
Read output buffer 55555555
Read output buffer 2000010
Read output buffer 4000020
Read output buffer 6000030
Read output buffer 8000040
Read output buffer 54773300
Read output buffer 0
```

Header

Trailer

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).

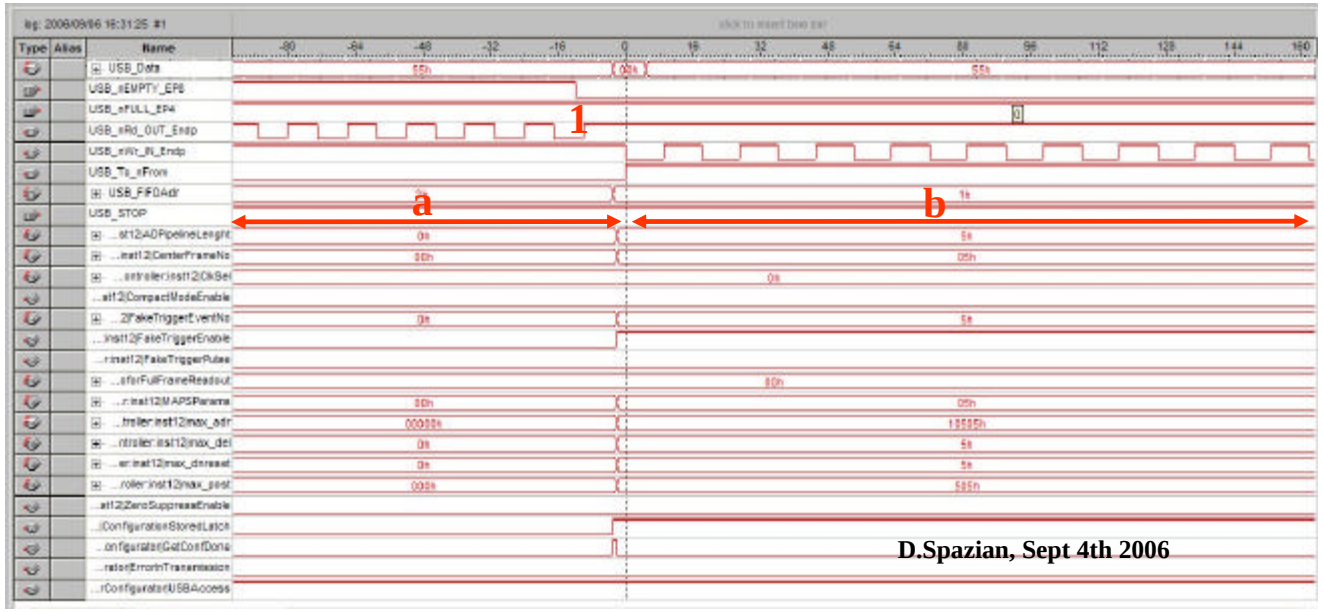
The cycle time between two successive transfers is 240ns (-> 33 MB/s peak transfer rate to be compared to the maximum theoretical rate of 80MB/s claimed by the standard). This time is made up of

- (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 and 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

A VME-64x based DAQ card for MAPS sensors

Some test results: USB 2.0 link



Detail of the activity of the USB interface inside the FPGA.

The waveforms are recorded “live” by the “SignalTap Logic Analyzer” tool provided by ALTERA.

“SignalTap” uses resources inside the FPGA to sample (at 80MHz in our case) the activity of the target signals and transmit them, after a trigger condition is met, to the host PC via the FPGA’s JTAG port.

During phase (a) the USB data is being received by the FPGA.

In (1) the whole bulk packet of 512 bytes has been completely read out (nEmpty line from the USB 2.0 PHY device goes ACTIVE LOW).

During phase (b) the FPGA returns to the host PC the same data just received for a consistency check.

A VME-64x based DAQ card for MAPS sensors

Additional information

EUDRB: front panel connector pinout

MOTHER BOARD connector pinout
DIMENSIONS 233 x 160 mm

PINOUT OF CONNECTOR MOBO_J1 (RJ-45)

RJ-45 conductor no.	Signal Name	Type	Standard
1	TriggerPulse_TriggerNumber_P	I	LVDS
2	TriggerPulse_TriggerNumber_N	I	LVDS
3	Busy_P	O	LVDS
4	TriggerReset_P	I	LVDS
5	TriggerReset_N	I	LVDS
6	Busy_N	O	LVDS
7	TriggerNumberClockOut_P	O	LVDS
8	TriggerNumberClockOut_N	O	LVDS

PINOUT OF CONNECTOR MOBO_J12(RJ-45)

RJ-45 conductor no.	Signal Name	Type	Standard
1	SpareLVDS_IN_P	I	LVDS
2	SpareLVDS_IN_N	I	LVDS
3	GND	Power	power
4	SpareLVDS_OUT_P	O	LVDS
5	SpareLVDS_OUT_N	O	LVDS
6	GND	Power	power
7	RJ_TX	O	RS-232
8	RJ_RX	I	RS-232

ANALOG DAUGHTER CARD connector pinout
DIMENSIONS 75 x 130 mm

PINOUT OF CONNECTOR A_J1 (RJ-45)

RJ-45 con	Signal Name	Type	Standard
1	ASG3p	In	analog
2	ASGL3n	In	analog
3	ASGL1p	In	analog
4	ASGL1n	In	analog
5	ASGL2p	In	analog
6	ASGL2n	In	analog
7	ASGL0p	In	analog
8	ASGL0n	In	analog

PINOUT OF CONNECTOR A_J2 (RJ-45)

RJ-45 con	Signal Name (normal/alternate) (*)	Type	Standard
	(*) Alternate pin functions to drive the MIMOSA V with analog board alone		
1	DAC_OUT1 / MKOFF1	Out /Out	analog/LVTTL
2	GND		
3	DAC_OUT2 / MKOFF2	Out /Out	analog/LVTTL
4	GND / OFAST	Pwr /Out	power/LVTTL
5	DAC_OUT3 / OSHUT	Out /Out	analog/LVTTL
6	GND		
7	DAC_OUT4 / CDS_TP	Out /Out	analog/LVTTL
8	GND		

PINOUT OF CONNECTOR A_J3 (LEMO)

LEMOconn	Signal Name	Type	Standard
1	ASGL0p	In	analog
2	GND		

PINOUT OF CONNECTOR A_J4 (LEMO)

LEMOconn	Signal Name	Type	Standard
1	ASGL1p	In	analog
2	GND		

PINOUT OF CONNECTOR A_J5 (LEMO)

LEMOconn	Signal Name	Type	Standard
1	ASGL2p	In	analog
2	GND		

PINOUT OF CONNECTOR A_J6 (LEMO)

LEMOconn	Signal Name	Type	Standard
1	ASGL3p	In	analog
2	GND		

PINOUT OF CONNECTOR A_J7 (LEMO)

LEMOconn	Signal Name	Type	Standard
1	External Pixel Scan Clock **(MIMO-ROMA)**	In	analog
2	GND		

PINOUT OF CONNECTOR A_J8 (LEMO)

LEMOconn	Signal Name	Type	Standard
1	External "Start Of Frame" **(MIMO-ROMA)**	In	analog
2	GND		

DIGITAL DAUGHTER CARD connector pinout
DIMENSIONS 75 x 130 mm

ALTERNATE PINOUT OF CONNECTOR D_J1 (RJ-45)
FOR MIMOSTAR / MIMOSA 5

RJ-45 conductor no.	Signal Name	Type	Standard
1	MSTAR_TCK / MKOFF1	O/O	3.3V TTL
2	GND		
3	MSTAR_TMS / MKOFF2	O/O	3.3V TTL
4	MSTAR_TDI / OFAST	O/O	3.3V TTL
5	GND		
6	MSTAR_TDO / OSHUT	I/O	3.3V TTL
7	GND		
8	MSTART_Ntrst / CDS_TP	O/O	3.3V TTL

ALTERNATE PINOUT OF CONNECTOR D_J2 (RJ-45)
FOR MIMOSTAR / MIMOSA 5

RJ-45 conductor no.	Signal Name	Type	Standard
1	MSTAR_CLKRDn / M5_CLKRDn	O/O	LVDS
2	MSTAR_CLKRDp / M5_CLKRDp	O/O	LVDS
3	GND		
4	GND		
5	MSTAR_CLK10Xp / ----------	O/O	LVDS
6	MSTAR_CLK10Xn / ----------	O/O	LVDS
7	MSTAR_SYNCp / M5_nRSTp	O/O	LVDS
8	MSTAR_SYNCn / M5_nRSTn	O/O	LVDS

ALTERNATE PINOUT OF CONNECTOR D_J3 (RJ-45)
FOR MIMOSTAR / MIMOSA 5 / MIMO-ROMA

RJ-45 conductor no.	Signal Name	Type	Standard
1	LVDSOUT4n / LVDSOUT4n / MR_CLKOUTn	I / I	LVDS
2	LVDSOUT4p / LVDSOUT4p / MR_CLKOUTp	I / I	LVDS
3	LVDSOUT3n / LVDSOUT3n / ----------	I / I	LVDS
4	LVDSOUT3p / LVDSOUT3p / ----------	I / I	LVDS
5	LVDSOUT2n / LVDSOUT2n / ----------	I / I	LVDS
6	LVDSOUT2p / LVDSOUT2p / ----------	I / I	LVDS
7	LVDSOUT1n / LVDSOUT1n / MR_SOF_OUTn	I / I	LVDS
8	LVDSOUT1p / LVDSOUT1p / MR_SOF_OUTp	I / I	LVDS

PINOUT OF CONNECTOR D_J4 (USB-B)
FOR MIMOSTAR / MIMOSA 5 / MIMO-ROMA

USB-B conductor no.	Signal Name	Type	Standard
1	+ 5V		power
2	- DATA		USB
3	+ DATA		USB
4	GND		power

INFN Istituto Nazionale di Fisica Nucleare

A VME-64x based DAQ card for MAPS sensors

Additional information

- Format of Data Block for "Zero Suppressed" readout mode:

(*) this last data would be meaningless in case the total HitCount were odd

"COMPACT" MODE packet structure					
Packet Header	Bit63..56 "H"	Bit55..48 Event Counter (from trigger scaler)	Bit47..40 Frame Counter	Bit39..32 Empty	Bit31..0 Empty
Data 1, Data0	Bit 63..52 Pixel Data	Bit 51..32 Pixel Address	Bit 31..20 Pixel Data	Bit 19..0 Pixel Address	
...					
DataN, Data N-1	Bit 63..52 Pixel Data (*)	Bit 51..32 Pixel Address(*)	Bit 31..20 Pixel Data	Bit 19..0 Pixel Address	
Packet Trailer	Bit63..56 "T"	Bit55..48 Event Counter (from trigger scaler)	Bit47..40 Frame Counter	Bit39..32 Empty	Bit31..0 HitCount

- Alternative (optional) format of Data Block for "Zero Suppressed" readout mode:

(*) this last data would be meaningless in case the total HitCount were odd

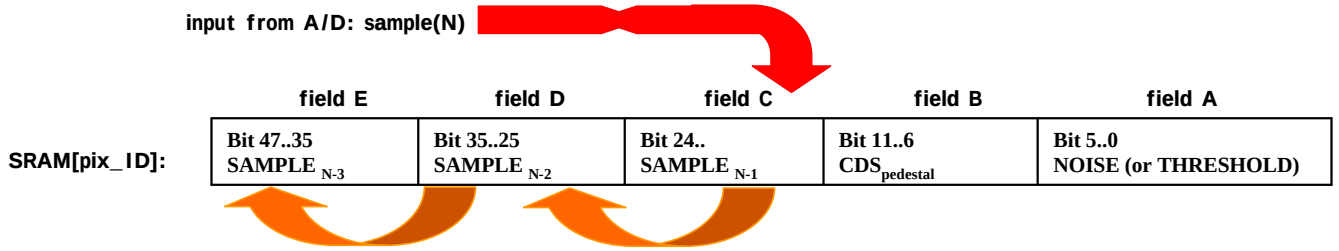
"Extended information" MODE packet structure							
Packet Header	Bit63..56 "H"	Bit55..48 Event Counter (from trigger scaler)		Bit47..40 Frame Counter		Bit39..32 Empty	Bit31..0 Empty
Data0	Bit 63..56 0	Bit 55..44 Pixel After	Bit 43..32 Pixel Before	Bit 31..26 Pixel pedestal	Bit 25..20 Pixel noise	Bit 19..0 Pixel Address	
...							
Packet Trailer	Bit63..56 "T"	Bit55..48 Event Counter (from trigger scaler)		Bit47..40 Frame Counter		Bit39..32 Empty	Bit31..0 HitCount

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....