

## MAPS- DAQ

**A proposal for the baseline features of a VME based DAQ card for MIMOSA-V  
sensors to be used on test beam experiments**

## A proposal for the baseline features of a VME based DAQ card for MIMOSA-V sensors to be used on test beam experiments



This note outlines the basic features of the VME based card being developed at INFN Ferrara for the acquisition of data from Monolithic Active Pixel Sensors (MAPS) to be characterized or employed in a test beam foreseen for June 2006.

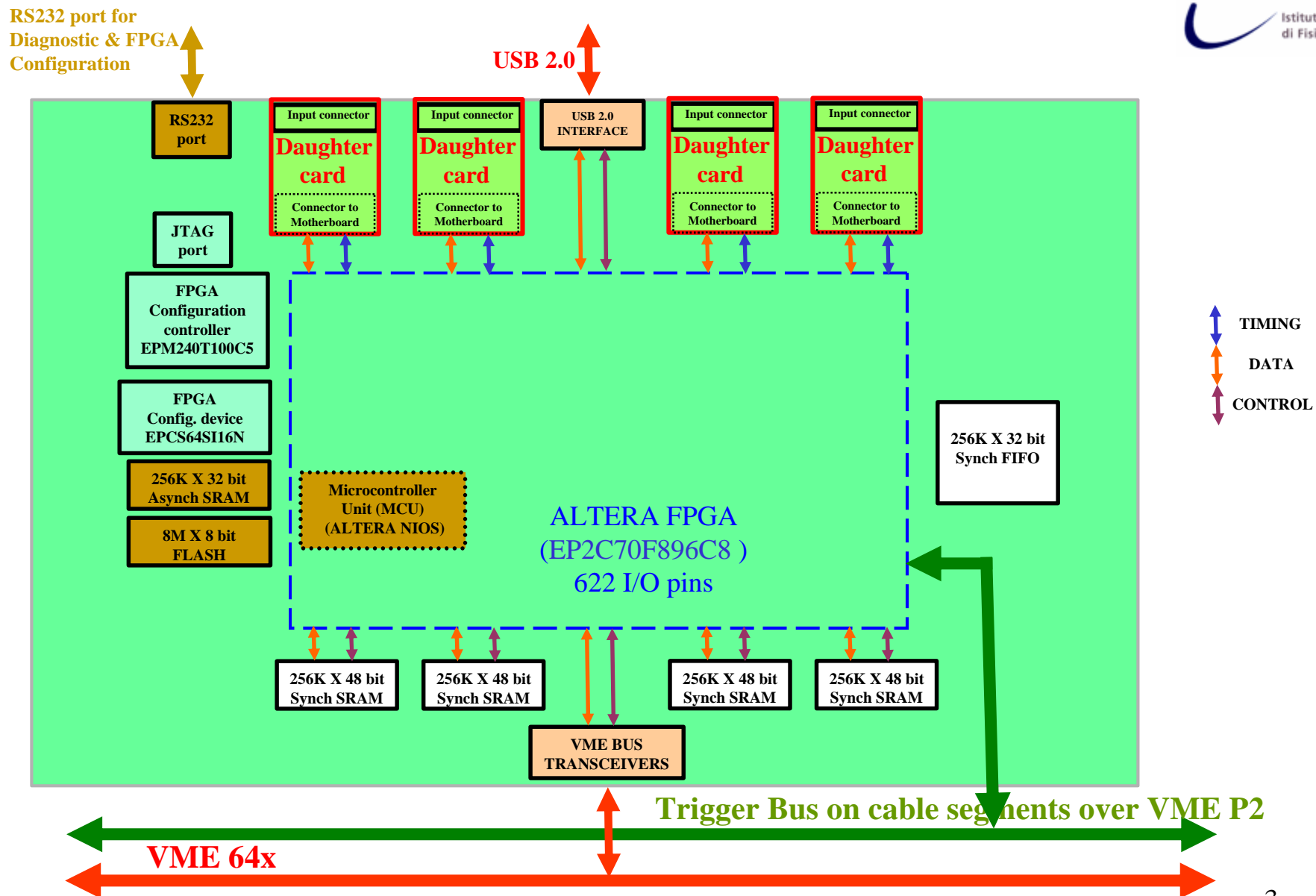
Its first application will be to readout the MIMO-Roma MAPS in a test beam foreseen for next June and thus the development is being carried out in collaboration with INFN-Roma3 and University of Insubria.

The board could also be a candidate for reading out the MAPS in the EUDET telescope and in this view its main features are presented here.

### Acknowledgments:

While some of the features presented here are quite novel, the general outline of the card is inspired by previous works by the People and Institutions involved with the SUCIMA project.

# A proposal for the baseline features of a VME based DAQ card for MIMOSA-V sensors to be used on test beam experiments



## A proposal for the baseline features of a VME based DAQ card for MIMOSA-V sensors to be used on test beam experiments



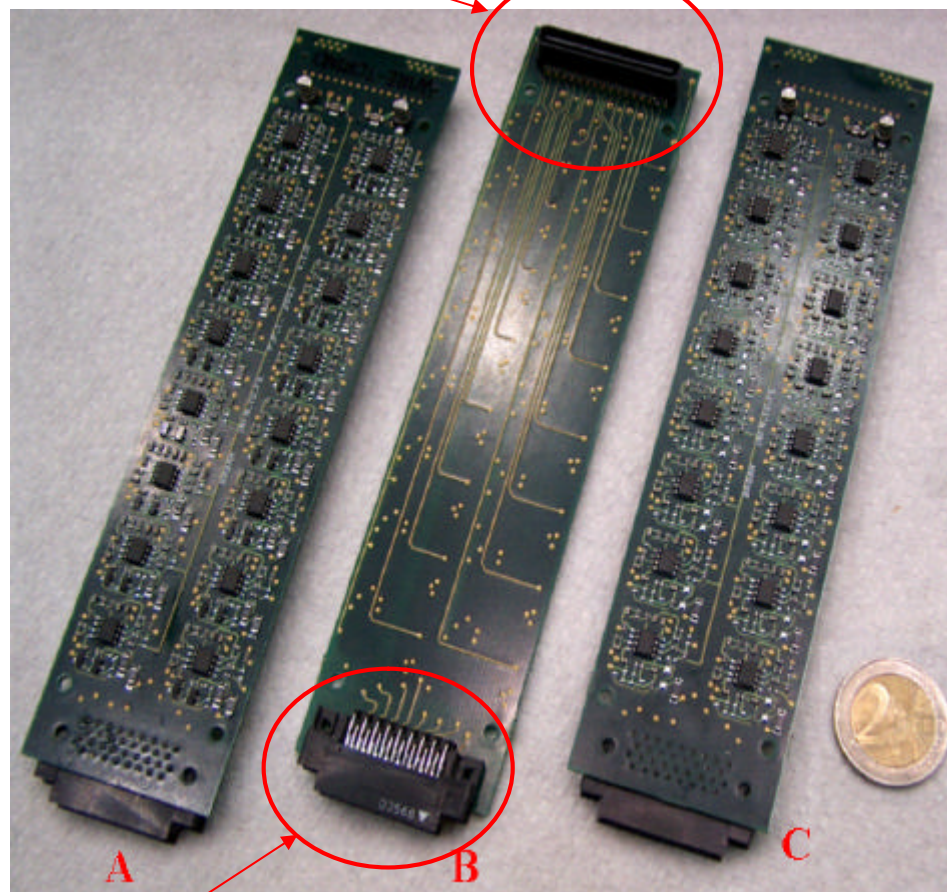
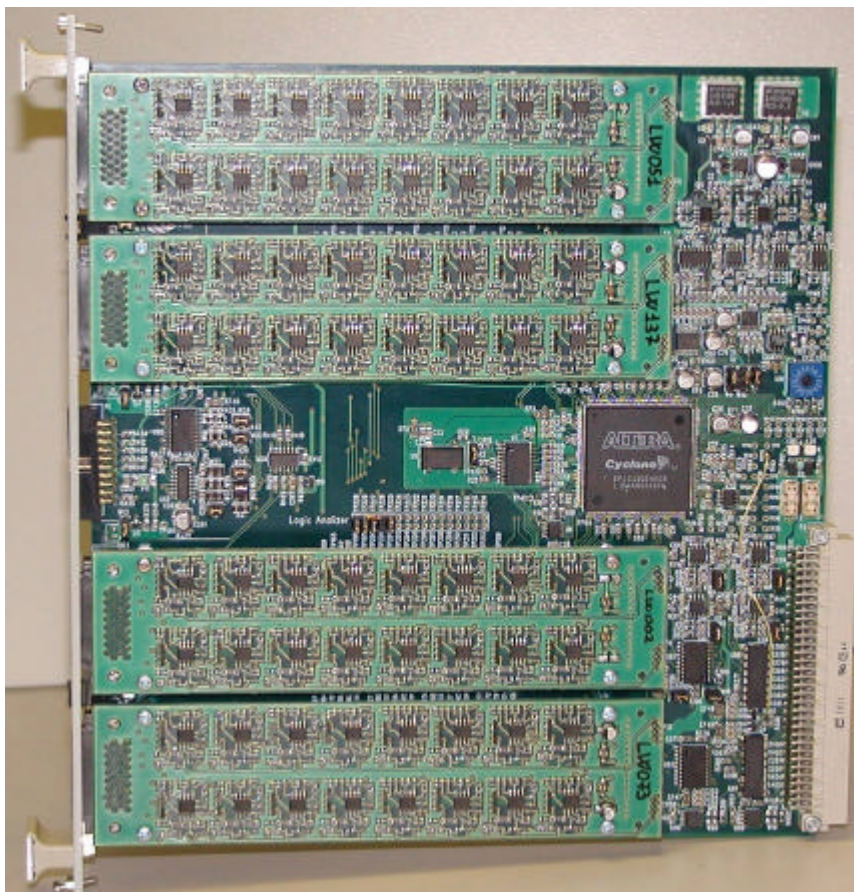
### Basic features of the present proposal:

- a) the card carries daughter cards with a standard interface to the motherboard and different flavours of input connectors and implemented functions, to adopt to different type of sensors.
  - b) The amplifier and A/D converter daughter card will be based on SUCIMA's design
  - c) VME64x slave bus interface with MBLT capability (VME64-2e block transfer mode)(\*).
  - d) Interface to a simple trigger protocol ( trigger request, event number, busy, holdoff ) on cable segments accessible via the user-defined rows of the VME P2 connector
  - e) Zero suppressed readout mode to minimize the readout dead-time while in normal data taking.
  - f) Full frame readout mode for debugging or off-line pedestal and noise calculations
  - g) USB2.0 interface for stand-alone operation of the board
  - h) NIOS II -32, 32 bit "soft" microcontroller implemented in the Cyclone I FPGA. It handles tasks like:
    - a) on-line calculation of pixel pedestal and noise by tapping data samples as they are read from the sensors
    - b) remote configuration of the FPGA via RS-232, VME, USB2.0
    - c) on board diagnostic
- (\*) burst transfer rate of 160MB/s. Sustained bandwidth to be determined with the CPU being purchased for the test setup, a Motorola MVME6100-0163

A proposal for the baseline features of a VME based DAQ card for MIMOSA-V sensors to be used on test beam experiments

Just to show (with another board as an example) what this assembly could look like

IEEE-1386 board to board connectors



KEL 8825E-034-175D connectors for 0.635mm pitch ribbon cable

# A proposal for the baseline features of a VME based DAQ card for MIMOSA-V sensors to be used on test beam experiments



## More details on the FPGA:

The MAPS-DAQ card handles the control of MAPS, the processing of the experiment's triggers and the consequent data output toward the data acquisition CPU by exploiting a single FPGA, an ALTERA EP2C70F896C8 in a FineLine BGA package with 896 pins of which 622 are I/O pins.

The FPGA is the top performance device in the ALTERA's Cyclone I family and contains 70K Logic Elements (1 LE = one flip flop + one 4-input LUT + fast carry chain and a few other resources) and in addition 250 M4K embedded memory blocks which can be organized in RAMs, FIFOs, DualPorts as needed. The total embedded memory capacity is about 1Mbit. One EP2C70F896C8 costs about 260\$.

### I/O pin balance and port description:

The FPGA's 622 I/O pins may seem many, but are almost all used up, as shown below:

- 1 port for: clock, board resets, MAPS timing / control signals  
total I/O pin count this section = 6
- 4 ports of parallel synchronous interface to the daughter cards  
in : 4 x 12bit data bus  
i/o : common control lines for the above (A/D Clock @ 20MHz, StartOfFrame, DetectorReset, Detector Clock)  
total I/O pin count this section = 52
- 4 ports for synchronous 256k x 48bit SRAMs to be used as a revolving buffer for input data  
i/o : 4 x 48bit data bus  
out : 1 x 18bit address bus common to the 4 sections  
out : 4 x individual chip select (nCS1)  
out : common control lines  
total I/O pin count this section = 224
- 1 port of parallel interface to the trigger bus, distributed in the DAQVME crate via private bus over J2  
in : 16bit data bus (trigger number)  
in : trigger strobe  
out : trigger BUSY ( flags that the DAQ card is busy processing one trigger but can accept more )  
out : trigger XOFF ( flags that the DAQ card is busy processing one trigger and cannot accept more )  
total I/O pin count this section = 19

continues ....

# A proposal for the baseline features of a VME based DAQ card for MIMOSA-V sensors to be used on test beam experiments



... continuing

- 1 interface to the VME bus (VME64-2e <-> 160MB/s peak transfer rate)		
i/o : 32bit data bus		
i/o : 31bit address bus + nLWORD		
in : addressing control lines		
out : acknowledge signals + transceiver controls		
	total I/O pin count this section	= 91
- 1 interface to the USB2 link (480Mb/s peak transfer rate)		
	total I/O pin count this section	= 26
- 1 RS-232 interface for remote configuration and debugging of the DAQ card via the NIOS processor		
	total I/O pin count this section	= 2
- 1 interface to the 512k x 36bit SRAM (2x IDT 71V67603-S-166-BQ-G) and 8M x 8FLASH (Amd29LV065D) memories serving the NIOS (NIOS SRAM/flash RAM will store temp/NonVolatile pedestal_noise values for all pixels)		
	total I/O pin count this section	= 71
- 1 interface to a 256k x 36bit FIFO (2x IDT 72T18115-L-5-BB240-1) to provide output buffering - common to all quadrants (mostly effective in sparsified-readout mode)		
	total I/O pin count this section	= 96
- 1 interface to diagnostic LEDs		
	total I/O pin count this section	= 8
	<b>grand total I/O pin count</b>	<b>= 587</b>

**Note:** If the expected occupancy (I mean, by it, the number of hits to extract in response to a trigger when the DAQ card operates in sparsified mode) in the test beam application is reasonably low, then one could just use the FPGA's embedded memory resources and would not need the external output FIFO

## More details on the operation of the MAPS-DAQ card:

- **A/D converter clock rate** (for the daughter cards with analog input from the sensors): 20MHz.

- **frame acquisition time**: for a MI MOSA-V 1Mpixel sensor with 4 independent outputs sampled @20MHz:  
 $262.144 * 50ns = 13,1072ms$  (four quadrants work simultaneously)

- **readout modes and trigger processing times**

- **“Full Frame” readout mode:**

In this operating mode the card responds to a trigger by sending out the pixel readings (12bits+padding to 16bits) for 3 frames: the frame being acquired at trigger time, the preceding one and the following one, a total of about 6MB per event.

I am assuming that in this “development” 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.

Also, assuming an average data bandwidth of 120MB/s (somewhat less than the peak bandwidth) on the VME backplane each event will be acquired by the VME CPU in about 1/20s per sensor.

The total trigger processing time at the crate level will then depend from the total number of MAPS to be read out. The latency in the response can be no less than a frame acquisition time, since each detector needs to complete the sampling of the frame being captured at trigger time and of the following frame.

The total trigger processing time should probably include the time to reset and restart the detectors at the end of the lengthy readout phase.

continues ....



# A proposal for some baseline features of the super-pixel digital processing blocks



... continuing

- **"Sparsified" (zero suppressed) readout mode:**

In this operating mode the card responds to a trigger by sending out a packet that could be formatted as follows (for 64 bits VME transfers):

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

"COMPACT" MODE packet structure				
Packet Header	Bit63..56 "H"	Bit55..24 Empty	Bit23..16 Frame Counter	Bit15..0 Event Counter (from trigger scaler)
Data 1, Data0	Bit 61..50 Pixel Data (w padding)	Bit 49..32 Pixel Address	Bit 31..20 Pixel Data	Bit 19..0 Pixel Address
...				
DataN, Data N-1	Bit 61..50 Pixel Data (w padding)(*)	Bit 49..32 Pixel Address(*)	Bit 31..20 Pixel Data	Bit 19..0 Pixel Address
Packet Trailer	Bit63..56 "T"	Bit55..24 HitCount (32 bit)	Bit23..16 Frame Counter	Bit15..0 Event Counter (from trigger scaler)

Or, alternatively,  
we could send out the two raw data from which CDS can be calculated off-line:

"Extended information" MODE packet structure				
Packet Header	Bit63..56 "H"	Bit55..24 Empty	Bit23..16 Frame Counter	Bit15..0 Event Counter (from trigger scaler)
Data0	Bit 63..56 Padding with 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..24 HitCount (32 bit)	Bit23..16 Frame Counter	Bit15..0 Event Counter (from trigger scaler)

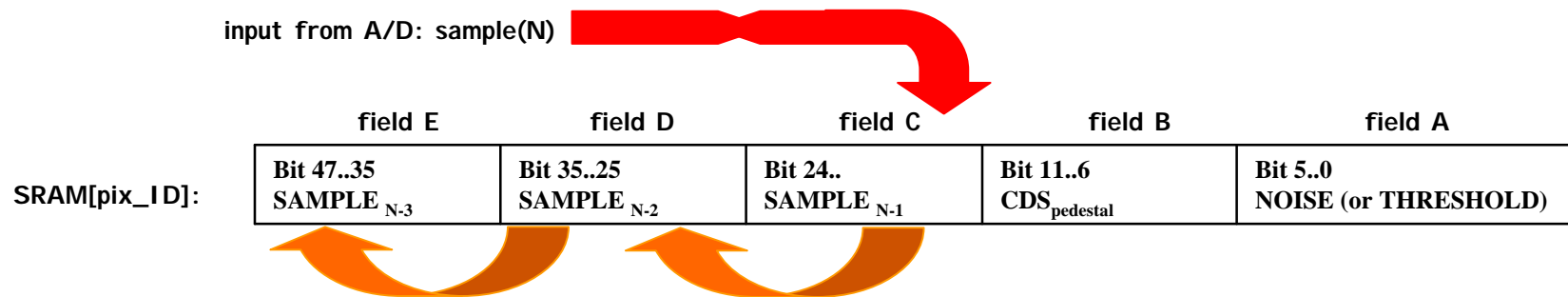
# A proposal for some baseline features of the super-pixel digital processing blocks



The packet contains the data only for those pixels whose signal was found above threshold after **ON-LINE** CDS and (pedestal+noise) subtraction.

I am assuming that 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 ):



I am assuming that in the "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 SRAM[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 proposal for some baseline features of the super-pixel digital processing blocks



... continuing

Assuming all of the above it follows that, in “sparsified” acquisition mode, the preparation of the output data packet in response to a trigger can start as soon as the trigger is received and that it only needs to last 1Mega / 4 cycles of detector clock after the trigger (plus some overhead and plus the data transfer time toward the data acquisition CPU).

Processing of ONE trigger needs not to interrupt the sampling of pixel data at the input port and it is, for reasonable channel occupancy, deadtime-less.

The DAQ card issues a Busy signal while processing a trigger request and an XOFF signal if another trigger is received while one is being serviced.

If resources for more than one deadtime-less trigger processor cannot be implemented in the FPGA, then the DAQ card will issue the XOFF after the first trigger request received.

An XOFF signal could also be generated at the crate level by a suitable “Trigger-Interface” card, which would distribute the trigger signals over a private backplane built on the free pins of the VME J2 connector and monitor the collection of data from the DAQ cards by the crate's CPU.

An estimate of the detector occupancy is needed at this point to evaluate the crate readout deadtime at different values of average trigger rate.