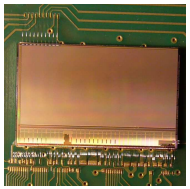
Mimosa 26 Readout proposal with NI Flex RIO boards
JRA1 Meeting Geneva October 2009



Mimosa 26

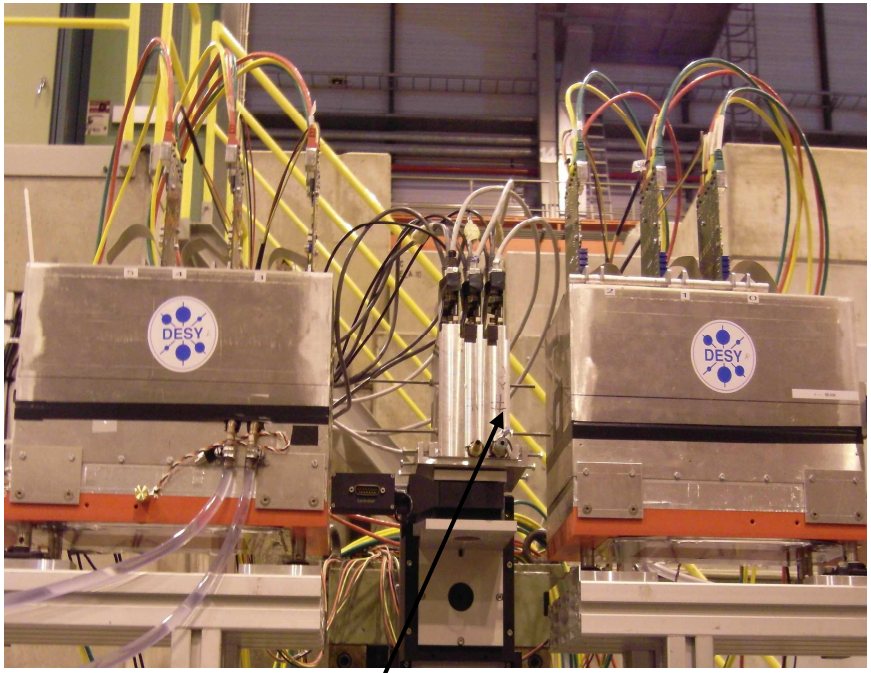
IPHC
Institut Pluridisciplinaire Hubert CURIEN
STRASBOURG

OUTLINE

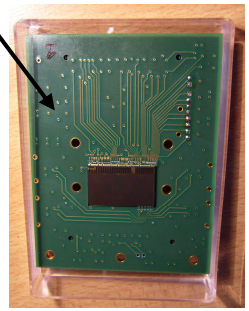
- The questions
  - EUDET JRA1 final beam Telescope acquisition boards ?
  - Test of NI PXI board PXI 6562 for IPHC beam Telescope
  - Solution for final Telescope based on NI PXI Flex RIO ?
- Evaluation of Flex RIO board – Collaboration with NI
  - Step No 1 → Emulate PXI 6562 with a Flex RIO board
  - Step No 2 → Evaluate dead time (Board / CPU – Disk transfer)
  - Step No 3 → Implementation of deserializer on Flex RIO
- Status & Conclusion
- Backup slides
  - Flex RIO board
  - Readout of Mi26 with flex RIO (Project slides for NI)

EUDET JRA1 Final Beam Telescope

Beam Telescope with 3 planes of Mimosa 26 sensor as DUT



Module with one Mimosa 26



Mimosa 26 mounted on PCB

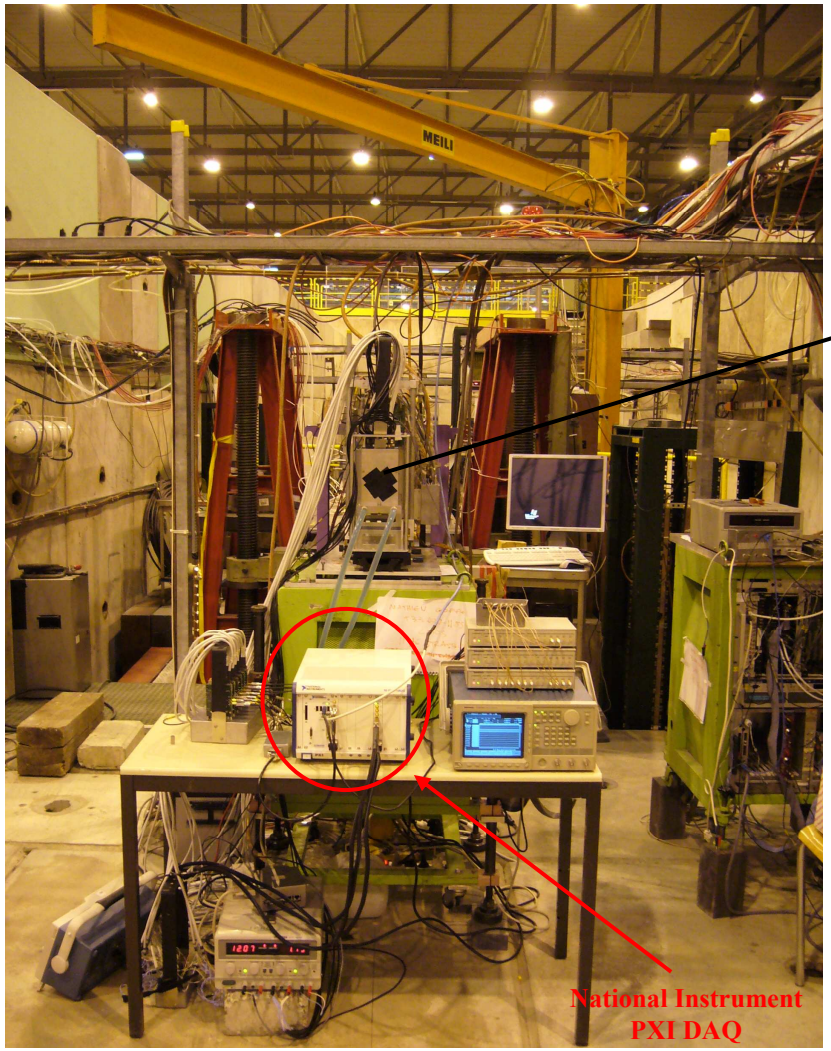
Try to use COTS for the final Telescope DAQ

- Telescope equipped with 6 planes of Mimosa 26
- Continuous readout without dead time
  - 1 x Mi26 → ~ 25 MB/s (~10 kframe/s * ~ 10 kb/frame)
    → It's a bit less (19 MB/s) but it's easier to use 25 MB/s for calculation
  - 6 x Mi26 → 150 MB/s
  - 1 x Ultimate → ~ 32 MB/s
  - 6 x Ultimate → 192 MB/s
- DAQ HW must be easy to copy → Telescope copies

The Idea

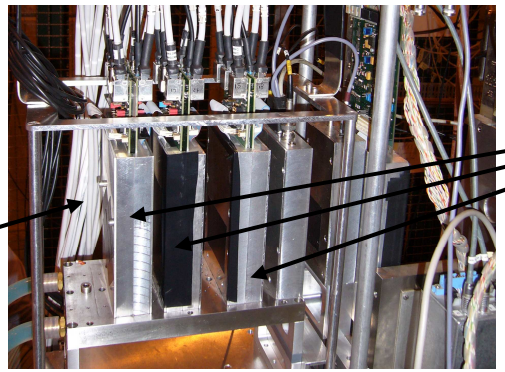
- Use National Instrument PXI or PXIe Board ?
  - Now it can be PXI but we should move to PXIe
  - PXI ~ 60 - 80 MB/s → 40 % to 53 % of full speed
  - PXIe ~250 MB/s → 100 % of full speed also for ULTIMATE
- Flex RIO boards → PXI 7953R ?
  - On board FPGA Virtex 5 – LX85
  - On board FW via LabView FPGA or VHDL
  - On board DRAM 128 MB

Test of PXI Board for IPHC beam Telescope



Telescope with 6 planes of Mimosa 26
Running with NI PXI 6562 Acquisition board
Monopix IPHC Beam Test – September 2009

Telescope with 6 planes of Mimosa 26



Using NI PXI 6562 on our Mi26 BT

- Board PXI 6562
  - No on board FW – But on board 32 MB SRAM
  - Can store up to 1800 frames of 6 Mimosa 26
  - Data clock up to 200 MHz
- How does it works
  - Frames stored on board during each spill
  - Data deserialisation by SW (→ CPU time)
- Performances
  - 1 to 3 Blocs of 1800 consecutives frames / spill
  - Average 540 frames/s
    - Mi26 = 8680 frames/s → ~ 94 % dead time

The Idea : Why Flex RIO ?

Limitation of PXI 6562 Board (Used on IPHC Telescope)

- Average event rate of ~ 540 Hz → 94 % dead time

  → Deserialisation must be implemented on board FW

- Synchronization with a DUT (other sensor < > MAPS) is not handled by this DAQ

  → Synchronization between Mimosa 26 & DUT CAN by done by on board FW

Flex RIO may be a good candidate for final DAQ

- Deserialization → Board FW
- Synchronization / DUT → Trigger handling and / or Record event index & time stamp → Board FW
- A LVDS frond-end (Mimosa 26 → 80 MHz BUT Ultimate → 125 MHz) is required for Flex RIO

Evaluation of Flex RIO : Tests done at IPHC in collaboration with NI

Step No 1 → Emulate PXI 6562 with flex RIO

- In order to check continuous readout on sensor side by board
  - Store blocs of ~ 1800 events in board memory (6 planes of Mimosa 26)
  - " Easy " to integrate in our current DAQ system to make the test
  - Should be done for end of October
- We can replace PXI 6562 board by a Flex RIO on our laboratory DAQ

Step No 2 → Evaluate dead time

- In order to check continuous readout on DAQ side
  - We don't need to have a deserializer on board → Event size is the same for raw and deserialized data
  - Should be done for end of November

Step No 3 → Implementation of deserializer on board

- In order to check deserialializer implementation
  - Should be done for end of December

Status : First tests done

Step No 1 → Emulate PXI 6562 with flex RIO

- In order to check continuous readout on sensor side by board
  - Store blocs of ~ 1800 events in board memory (6 planes of Mimosa 26)
  - " Easy " to integrate in our current DAQ system to make the test
  - Should be done for end of October
- We can replace PXI 6562 board by a Flex RIO on our laboratory DAQ

Status on 16 October 2009

- We are able to read Mimosa 26 frame via FPGA FIFO
- Transfer to DRAM (Mi26 → DRAM → CPU) is not ready BUT work in progress
- Strong involvement of NI
  - Development of FW done by Mr ALBERTINI (from NI)
  - Three days spent at IPHC during last two weeks
  - Flex RIO PXI 7953 R system at IPHC for ~ 10 Days
- Resources on IPHC side
  - Someone is able to run this system (install, compile FW …) at ~ 0,5-1 day / week
  - We plan to buy one system → PXI 7953 + PXIe CPU + Crate

Conclusion

Performances

- Flex RIO board – PXI version - for Mimosa 26 readout seems to be possible
  - Maximum event rate will be ~ 50 % of full speed (40 – 53 %)
  - Work still needed to have a system FOR TESTS No 1, 2, 3 and a lot of work to have a RUNNING SYSTEM
    - Good news → It will not be a ready to use black box → There is work for an engineer BUT on fw NOT on hw :-)
  - Evaluation in progress at IPHC in collaboration with NI
    → We hope to have results for the end of 2009 or very beginning of 2010
- Next generation of Flex RIO will use PXIe bus
  - No event rate limitation (100 % full speed) on HW side (Mi26 → Disk)

Next Steps

- It's to early to claim that Flex RIO is the solution → We must perform tests No 1, 2, 3 before
- Human resources on IPHC side
  - We plan to perform the three tests for the end of 2009 or early 2010
- Do we need to work faster ?
  - Human resources in EUDET collaboration ?
- My idea : Finish current tests (No 1, 2, 3) at IPHC → Involvement of EUDET starting in 2010

Backup slides

- NI Flex RIO boards (JRA1 meeting 11 June 2009)

- Proposal of Mimosa 26 readout with NI Flex RIO
  - Written by G.CLAUS on 6-8 September 2009
  - Discussion with Ingrid & Daniel on 8 September 2009
  - Sent to National Instrument on 9 September 2009

Commercial DAQ board : NI Flex RIO ?

NI FlexRIO – Custom I/O for LabVIEW FPGA and PXI

NI PXI-795xR *NEW!*

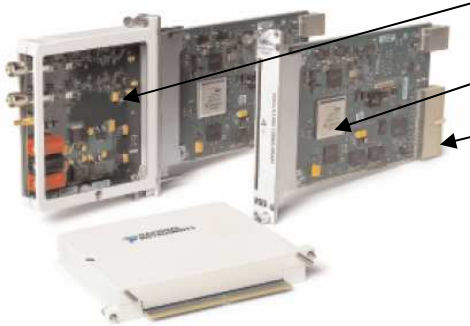
- Virtex-5 FPGA-programmable with the LabVIEW FPGA Module
- Up to 128 MB onboard DDR2 DRAM
- Access to 132 single-ended I/O lines, configurable as 66 differential pairs
- Customizable I/O with the NI FlexRIO Adapter Module Development Kit (MDK)
- 100 MHz digital I/O with the NI 6581 adapter module
- 3 DMA channels for high-speed data streaming

Operating Systems
- Windows Vista/XP/2000
- LabVIEW Real-Time

Required Software
- LabVIEW
- LabVIEW FPGA Module
  - LabVIEW code compiler for FPGAs
  - Emulated debugging mode

Recommended Software
- LabVIEW Real-Time Module

Driver Software (included)
- NI-RIO



Model	Bus/Form Factor	FPGA	FPGA I/O	Onboard Memory (DRAM)
PXI-7951R	PXI	Virtex-5 LX30	66 differential or 132 single-ended	0 MB
PXI-7952R	PXI	Virtex-5 LX50	66 differential or 132 single-ended	128 MB
PXI-7953R	PXI	Virtex-5 LX85	66 differential or 132 single-ended	128 MB
PXI-7954R	PXI	Virtex-5 LX110	66 differential or 132 single-ended	128 MB

Table 1. NI FlexRIO FPGA Modules

Future ?

- PXIe ? 250 MB/s / lane … X 1, 2, 4, 8, 12, 16, 32 → 6 GB/s …
  - Can't be handled by software …
  - Writing to disk + % of monitoring by software
- Fast serial links Gb/s
  - Direct handling on FPGA ?
  - User HW on Adaptater Module

Current Status

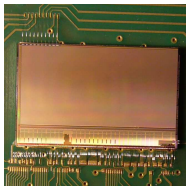
- 66 Differential inputs
- User defined Adaptator Module
- User defined on-board fw (deserialisation)
- PXI bus → Average ~ 60 MB/s

- 1 Mi26 plane @ 10 kframe/s → ~ 25 MB/s
  - 4 Planes ~ 100 MB/s → 60 % of full speed
  - 6 Planes ~ 150 MB/s → 40 % of full speed

Cost ?

- PXI-7951R (No RAM) ~ 2,8 K€
- PXI-7952R (128 MB RAM) ~ 3,7 K €
- Adaptator module SE 100 MHz ~ 1 K€
  - LVDS foreseen for end of summer
- PXIe Crate + CPU ~ 8 K €
- Adaptater Module Development Kit ~ 4,6 K€
  - Overhead of 4,6 K € (CAD files etc …)
  - ~ 60 € / enclosure

Mimosa 26 Readout proposal for Evaluation of NI Flex-Rio boards
Geneva September 2009



Mimosa 26

IPHC
Institut Pluridisciplinaire Hubert CURIEN
STRASBOURG

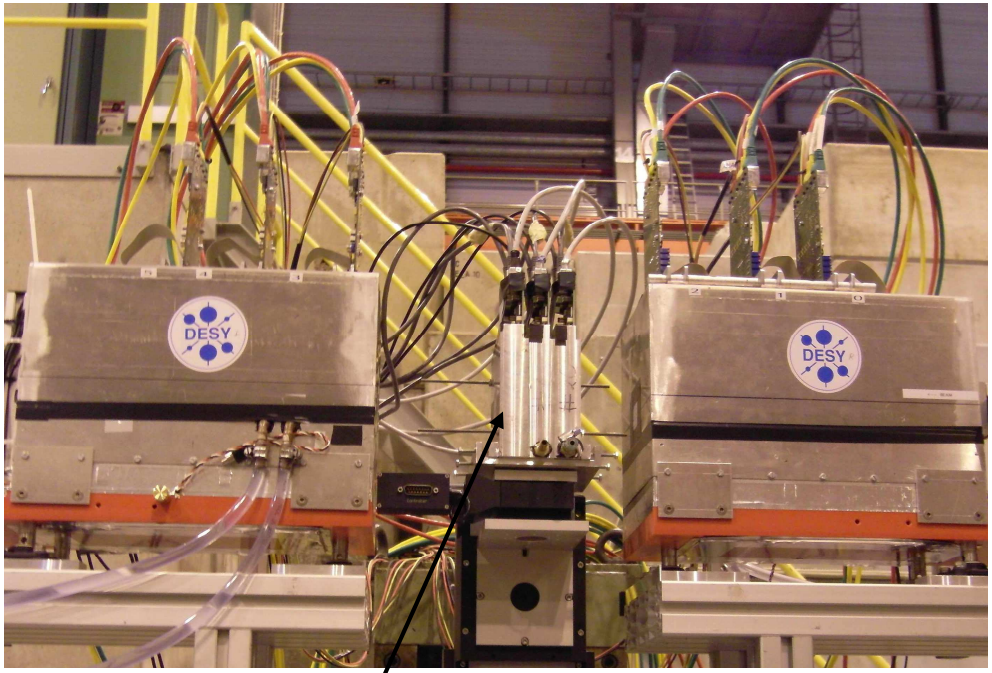
OUTLINE

- The goal of the project
  - EUDET JRA1 beam Telescope
  - First test of PXI HW on IPHC beam Telescope
- Mimosa 26 Steering & Readout
- First step → Emulate PXI 6562 with a Flex RIO board
  - Store Mi 26 data as is on board → Deserialisation done by SW
- Second step → Deserialize data on Flex RIO
  - Deserialize – Bufferize – Read board by SW *
- Third step → Trigger handling on Flex RIO
  - Deserialize – Bufferize ONLY events with trigger – Read board by SW*

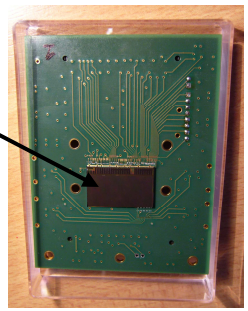
* Board readout by SW can be done after Mimosa 26 data acquisition OR in parallel (double buffers) it will depend on the board possibility and time needed to implement it.

The goal of the project : EUDET JRA1 Beam Telescope

Beam Telescope with 3 planes of Mimosa 26 sensor as DUT



Module with one Mimosa 26



Mimosa 26 mounted on PCB

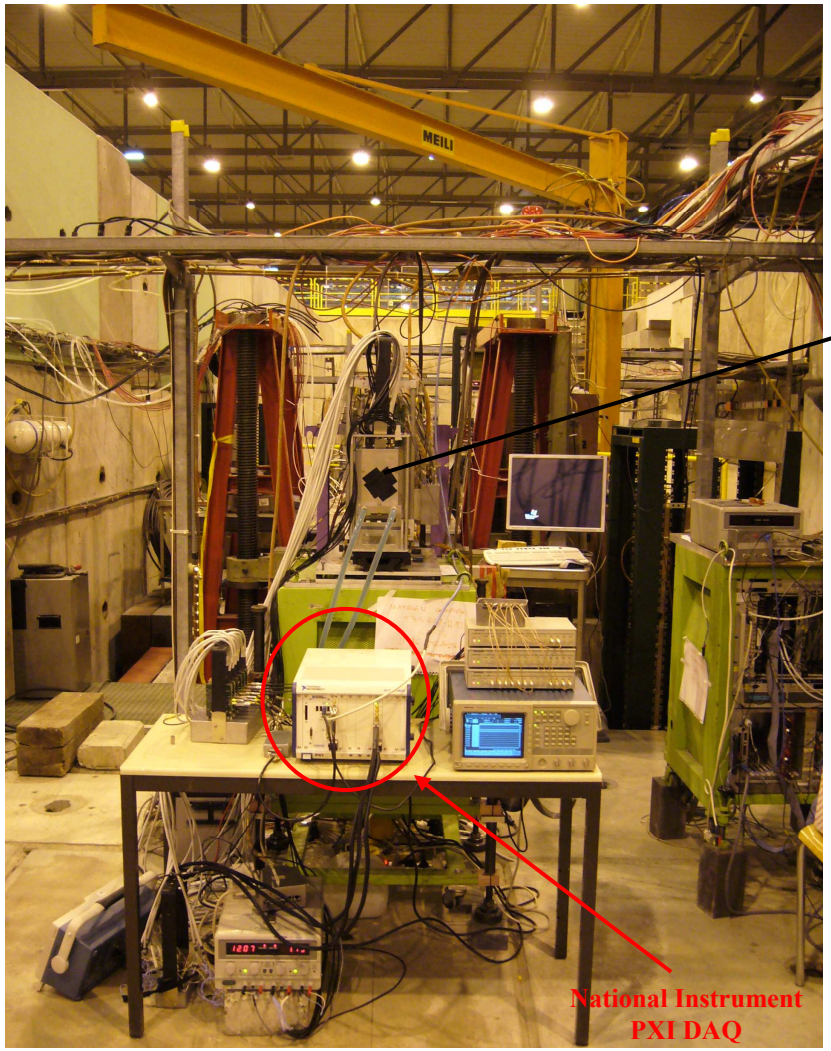
Try to use COTS for the final Telescope DAQ

- Telescope equipped with 6 planes of Mimosa 26
- Continuous readout without dead time
- DAQ HW must be easy to copy
  - Distribution of Telescope copies to collaborators

The Idea

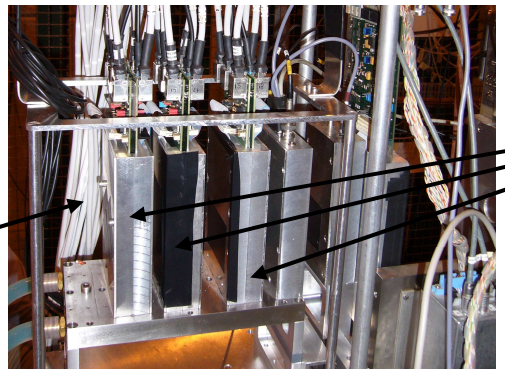
- Use National Instrument PXI DAQ ?
- Flex RIO boards ?
- We MUST evaluate boards performances
  - Sensor readout side
  - Data processing by on board FW
  - Data throughput Board / CPU Memory - Disk

The goal of the project : Test of PXI HW on IPHC beam Telescope



Telescope with 6 planes of Mimosa 26
Running with NI PXI 6562 Acquisition board
Monopix IPHC Beam Test – September 2009

Telescope with 6 planes of Mimosa 26



Modules
with two Mimosa 26

How does it works ?

- Acquisition of 12 LVDS links at 80 MHz
  - 6 Mimosa 26 x 2 Links = 12 links
- Acquisition synchronized to beginning of spill
  - Polling by SW of trigger signal
- Trigger position recording
  - Sampled at Mimosa 26 clock rate
  - Use one more LVDS input of PXI 6562
- Deserialization done by SW
  - Board acquires data as a // bus of 13 links
  - C software deserialize data on PXI CPU

The goal of the project : Why Flex RIO ?

Limitation of PXI 6562 DAQ (Used on IPHC Telescope)

- Average event* rate of ~ 540 Hz (* Event = one readout of Mimosa 26 matrix)
  - Store blocs of 1800 events in board memory (6 planes of Mimosa 26)
  - Record on disk 3 blocs / spill (~ 10 s) = 3 x 1800 = 5400 events → Average ~ 540 Hz event rate
  - Fine for our beam test BUT not to run Mimosa 26 in continuous readout

    … Readout in 115,2 µs => 8680 events/s … (compare to 540 event/s with PXI 6562)

→ Deserialisation must be implemented on board FW

- Synchronization with a DUT (other sensor < > MAPS) is not handled by this DAQ
  - IPHC Telescope → All sensors including DUT are Mimosa 26
  - EUDET Telescope → DUT is a detector (not a MAPS) with his own DAQ

→ Synchronization between Mimosa 26 & DUT MUST by done on board FW

Flex RIO may be a good candidate for final DAQ

- Deserialization → Board FW
- Synchronization / DUT → Trigger handling and / or Record event index & time stamp → Board FW
- A LVDS frond-end (Now 80 MHz BUT 125 MHz for later upgrade) is required for Flex RIO

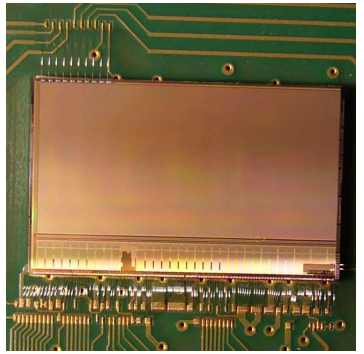
Steering & Readout : Mimosa 26 ?

Mimosa 26 is a MAPS (Monolithic Active Pixel Sensor)

- **Matrix of 663 552 pixels :** 576 lines x 1152 col – 10,6 mm x 21,2 mm – 18,4 µm pitch - 115,2 µs integration time
- **Integrated A/D conversion (by discriminator) & Data reduction (ZS : Zero Suppression)**

Steering

- **Configuration (operating modes, bias etc …) by JTAG slow control**
- **Main clock 80 MHz → Provided by external oscillator**
- **Start command → Provided by external HW signal**



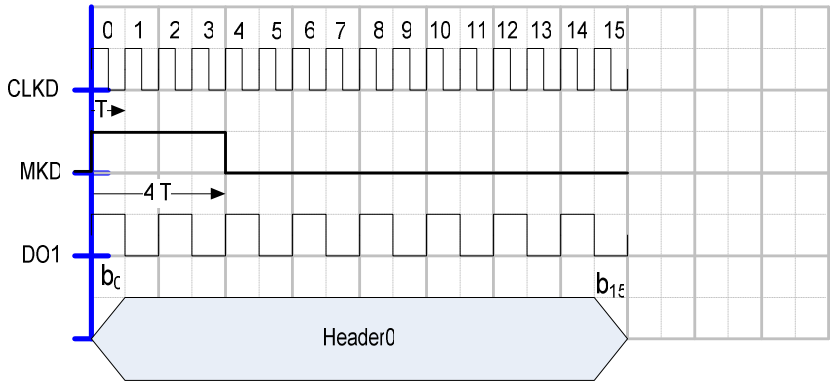
Readout

- **Continuous readout of the matrix frame by frame**
  - **Acquisition of current image by Mimosa 26 while reading previous one by DAQ**
- **Normal ZS output → Mimosa 26 provides**
  - **Two serial links at 80 MHz - 9216 bits / frame**
    - **Data stream clocks number is fixed (9216) BUT useful data size is variable (depend of the number of hits on the matrix)**
  - **Clock at 80 Mhz**
  - **Synchronization signal on first 4 bits of data stream**

Readout configuration : 2 Links @ 80 MHz

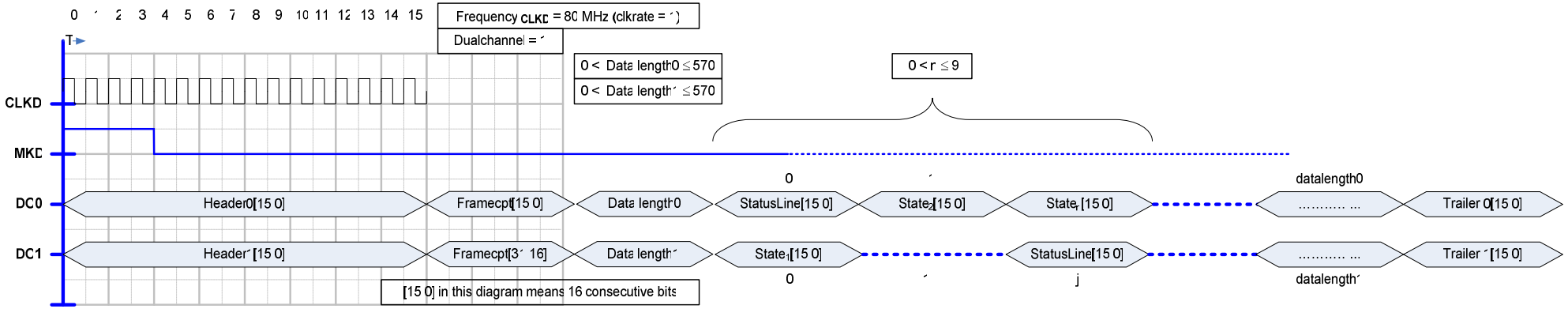
Summary

- Data generated on rising edge of Mimosa 26 clock
- Synchronization signal MKD – Duration 4 clocks cycles
- Data LSB first
- First bit of frame = Header LSB



Readout configuration : 2 Links @ 80 MHz

- Maximum data stream length is 1140 W16 (word of 16 bits) – 570 W16 on each link (D00, D01)



Summary

- Data generated on rising edge of Mimosa 26 clock
- Header → 16 bits / link
- Frame counter → 16 bits / link
- Data length (useful part of data) → 16 bits / link (Sum the 2 W16 to get matrix W16 size)
- Data (format on next slide) → Max = 570 x 16 bits / link
- Trailer → 16 bits / link
- Padding zero → 32 bits / link
- Maximum data stream size per link : 9216 bits = 576 W16 = 1152 W8
- Maximum data part size 570 W16 / link – But can be less ! → Size – in W16 - given by "Data length" field

First step : Emulate PXI 6562 with flex RIO ?

Why ?

- In order to check continuous readout on sensor side by board
  - Store blocs of ~ 1800 events in board memory (6 planes of Mimosa 26)
    - Is it possible to store data without dead-time in on board DRAM ?
  - " Easy " to integrate in our current DAQ system to make the test
    - SW reads bloc of the same size (as PXI 65462) & Process it in the same way
- We can replace PXI 6562 board by a Flex RIO on our laboratory DAQ
  - Same board on all systems : Calibration test bench at lab & Beam Telescope

How ?

- The acquisition of one bloc start upon reception of SW request (→ No trigger handling)
- The board acquires N samples (config by SW) of a 16 LVDS inputs bus
  - Each sample is stored as a W16 in memory
- At end of acquisition the board is read by SW
- The Telescope provides the following signals to the board (Slide 6 & 7) → Synchronous readout of all Mimosa 26
  - A 80 MHz clock
  - A synchronization signal
  - 13 LVDS links (6 Mimosa 26 x 2 links + Trigger = 13 inputs → 3 spare inputs / 16)

Second step : Deserialize data on flex RIO ?

Why ?

- In order to check deserializer implementation → Is it possible to …
  - Deserialize current bloc of events while sending on output previous bloc ?
    - In this case we must store deserialized data in DRAM
  - If not, can we have enough throughput to do it on the fly ?
    - Minimum bufferization on FPGA

How ?

- The acquisition should be controlled in two ways
  - Acquisition of one bloc – Deserialization – Readout by SW → Keep compatibility with our SW for first Tests
  - Continuous acquisition and data saving on disk → NI should propose SW to perform this test
- The board acquires N samples (config by SW) of a 13 LVDS link
  - 6 x Mimosa 26 data over 12 links + A reserved link for event tag
- Running one deserializer per link in // → 12 deserializers
  - Deserialize the input LSB first - W16 by W16 - and store results in W16 list in memory
  - Don't care about "Data length" field → Convert the whole event
- The Telescope provides the same signals to the board as written on slide 9

Third step : Trigger handling on flex RIO ?

Why ?

- In order to check external event handling by FW
  - Note : There is ONLY one trigger for the six Mimosa 26

How ?

- One event is continuously deserialized and overwritten in memory
- On trigger reception (Edge on trigger signal)
  - This event and next events (window length to be defined) are stored in memory
  - Wait on next trigger
- Board readout by SW → Two options
  - When on board buffer is full
  - Continuous readout → NI should propose a SW

Remarks : Open questions ?

- **Why variable event size implementation is not done on this test ?**
  - Each Mimosa 26 will provide a different event length
  - It will cost time to implement it NOW – It is not mandatory to evaluate board
- **Can we acquire the 6 planes of Telescope with a single board ?**
  - 1 Plane @ 10 KHz ~ 25 MB/s (full frame) → 6 Planes ~ 150 MB/s – Bus BW ~ 60 MB/s → 40 % of full speed (5 à 10 % now)
  - Overestimation of event size & underestimation of bus BW → Should be a bit better
  - Need to evaluate average event size from September beam test to have a better estimation (lower event size ?)
- **Can we use more than one board in one PXI crate ?**
  - Yes but it will not help → Split BW between boards
  - Use one crate / 3 planes → ~80 % of full speed
  - PXI Express → Solve BW problem BUT No board ready Now
- **Which Flex RIO to use ?**
  - It's up to NI to propose which one fit to the application
- **Crate & CPU → PXI or PXI Express ?**
  - Despite the fact that Flex RIO may be PXI, we should use PXI Express because
    - It will allow upgrade of the DAQ performances
    - Direct data streaming modules from board to disk may not be available on PXI

Important aspect : SW & FW should not be a " black box "

- **Firmware**
  - It can be written with Labview FPGA tools to demonstrate feasibility
  - It can stay in LabView FPGA if integration with VHDL is possible & easy
  - We must learn how to use this interface Labview FPGA / VHDL
- **Software**
  - The easiest way seems to be …
    - Building a Labview application which acts as a data producer running on PXI carte processor
    - Using Labview board drivers (VI) to control board
    - Handling Ethernet in C code (via DLL call in LabView application)
  - But … the easiest way may not be the best one on performances side …
    - A direct data streaming from board to disk may solve ALL continuous readout problems
    - But this link will not be integrated in EUDET DAQ SW
    - This may be solved like this
      - Keep EUDET SW for "Run control" of this NI Data producer
      - Send a fraction of data to EUDET DAQ for monitoring ONLY
  - The main question about SW integration is
    - Can we build a continuous readout (zero dead time) by a simple integration of "NI producer" in current EUDET DAQ ?
    - We should evaluate NI "Ethernet VIs" ? Use them ? Make the bridge LabView / EUDET on supervisor PC ?

Next step : Telescope sensor upgrade

- **Mimosa 26 may be replaced by Ultimate 1**
  - **Active area X ~ 2 (~ 4cm² instead of ~ 2 cm²)**
  - **Integration time of ~ 200 µs (instead of 115,2 µs)**
  - **Memory size X ~ 2,5**
  - **Output data stream should be ~ 125 MHz (instead of 80 MHz)**
  - **Maximum data rate / sensor ~ (115,2 µs / 200 µs) * 2,5 * 25 MB/s = 36 MB/s**
    - **(115,2 / 200) ratio of readout time (Ultimate slower)**
    - **2,5 ratio of memory size (Ultimate has a larger memory)**
    - **25 MB/s (Mimosa 26 data stream)**

- **Consequences on DAQ board**
  - **Deserializer should be able to run at up to 125 MHz**
  - **Data flow can be 44 % higher than with Mimosa 26**

Planning & Resources

- **Step 1 → Emulation of PXI 6562 with Flex RIO = Validation of continuous readout on sensor side**
  - **Should be finished before end of October**
  - **Manpower ?**
    - **Is it possible for NI to do the FW development on Flex RIO (for free) & provide a board ready to use ?**
    - **Tests can be done at IPHC**
- **Step 2 → Deserialization on Flex RIO = Validation of continuous readout on DAQ side**
  - **End of 2009**
  - **Manpower ?**
    - **How much NI can be involved in deserializer development ?**
      - **Best case → NI provide a board ready to use OR a FW template with one deserializer**
    - **Is there resources available on EUDET side ?**
    - **Is there resources available on IPHC side ?**
    - **Tests MAY be done at IPHC ???**
- **Step 3 → Trigger handling = Validation of possibility to " easily " upgrade FW**
  - **Beginning of 2010**
  - **Manpower ?**
    - **Resources should be found in EUDET collaboration**