Status of Readout Board Firmware

Guillaume Vouters
On behalf of the MiniDAQ developers

11 December 2014
LHCb Upgrade Electronics

cnrs IN2P3 Les deux infinis
lapp Laboratoire d'Annecy-le-Vieux de Physique des Particules

Outline

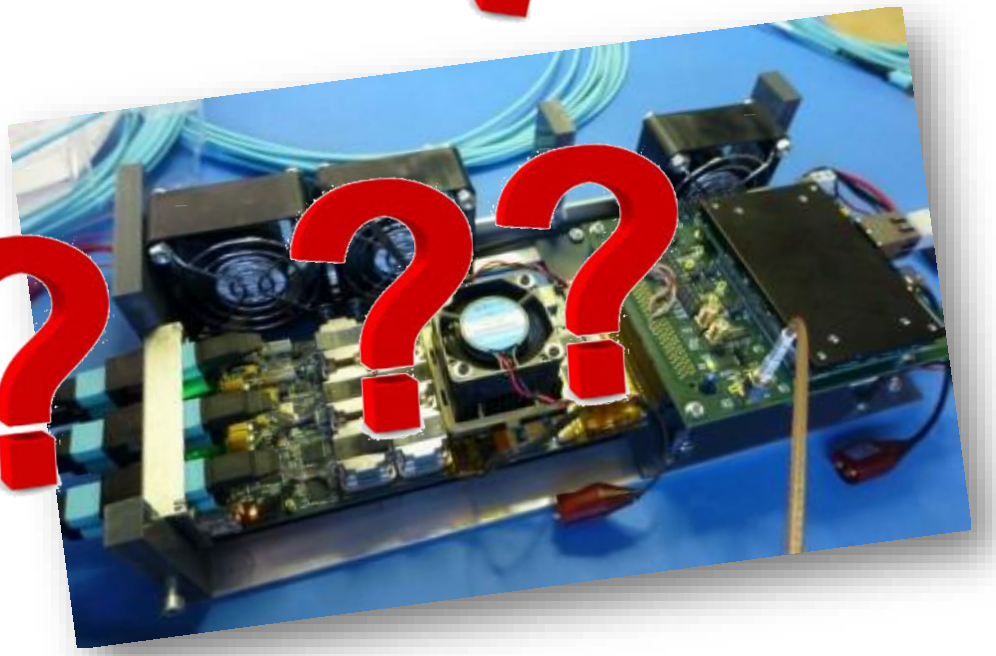
1. End of 2013
2. First part of 2014
3. Second part of 2014
4. Xmas 2014

1. End of 2013

What we got for Christmas 2013

FORGE ?

GIT ???



2. First part of 2014

We organized very useful meetings !!

2. First part of 2014

We organized very useful meetings !!

What really happened

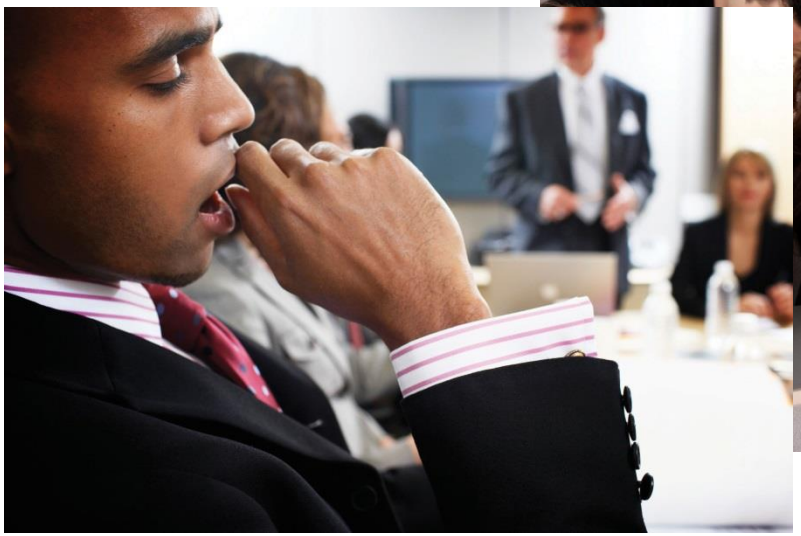
2. First part of 2014

Meetings were very successful !

2. First part of 2014

Meetings were very successful !

What really happened



2. First part of 2014

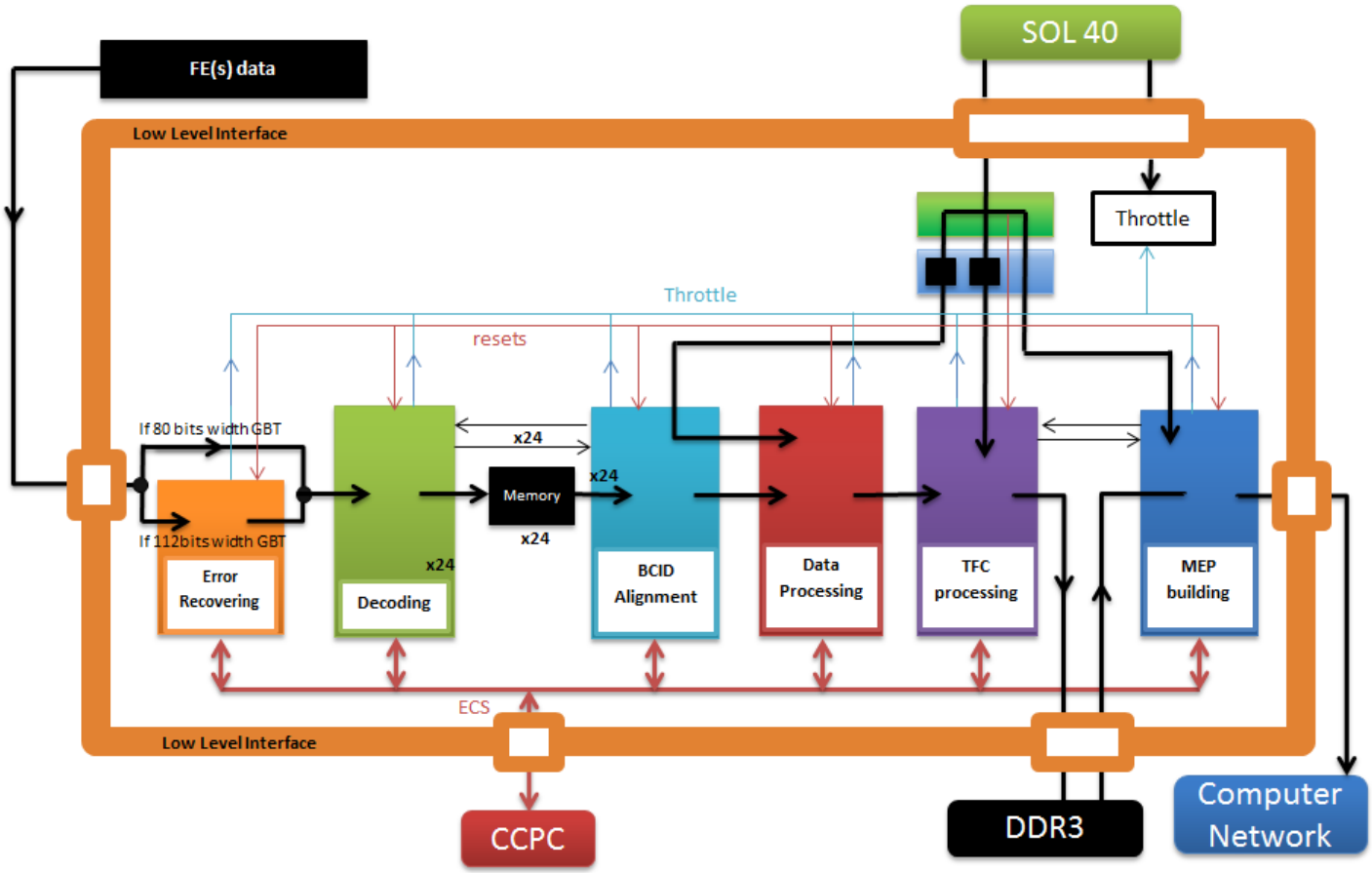
More seriously, after huge work, at the end of the first part of 2014 we had :

- A simulation framework working
- A first version of a firmware for the MiniDAQ (without data processing)
- Definition of a data format
- Documentations to use the FORGE, GIT and the framework
- Software for the CCPC
- Software to control the MiniDAQ.
- Full data flow test from the CCPC to a computer

But also a first glimpse of FPGA logic cell resources limitation

2. First part of 2014

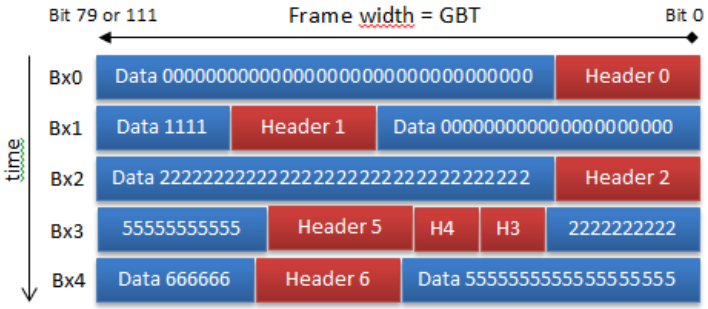
Previous firmware architecture

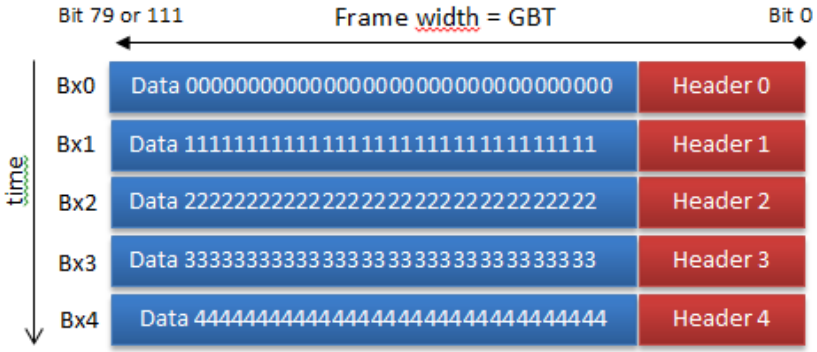


2. First part of 2014

Previous data format

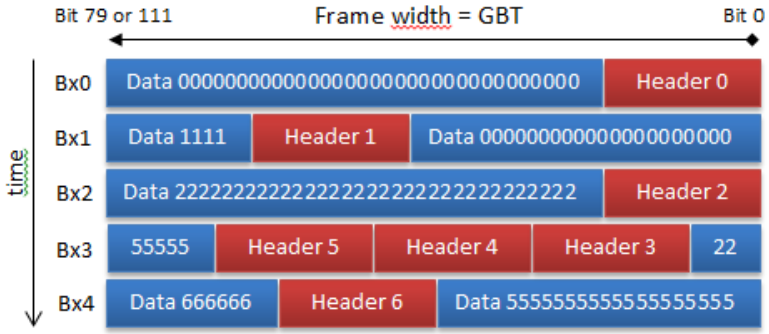
Variable header length and
Variable data length





Fixed header length and
Fixed data length

Fixed header length and
Variable data length



2. First part of 2014

Documentation listing

---------- LHCb Upgrade ----------

→ System-level Specifications of the Timing and Fast Control system for the LHCb Upgrade (on CDS)

→ Electronics Architecture of the LHCb Upgrade (on CDS)

---------- CCPC booting ----------

→ https://lbredmine.cern.ch/projects/ccpc-common/wiki/Kernel_and_distribution

---------- Framework Documentation ----------

→ FORGE documentation (on the FORGE)

→ GIT documentation (on the FORGE)

---------- Data Format ----------

Data format documentation (on the FORGE)

---------- MiniDAQ Handbook ----------

→ MiniDAQ documentation (on the FORGE)

→ MiniDAQ documentation (on CDS)

3. Second part of 2014

We had intense workshops

3. Second part of 2014

We had intense workshops

What really happened

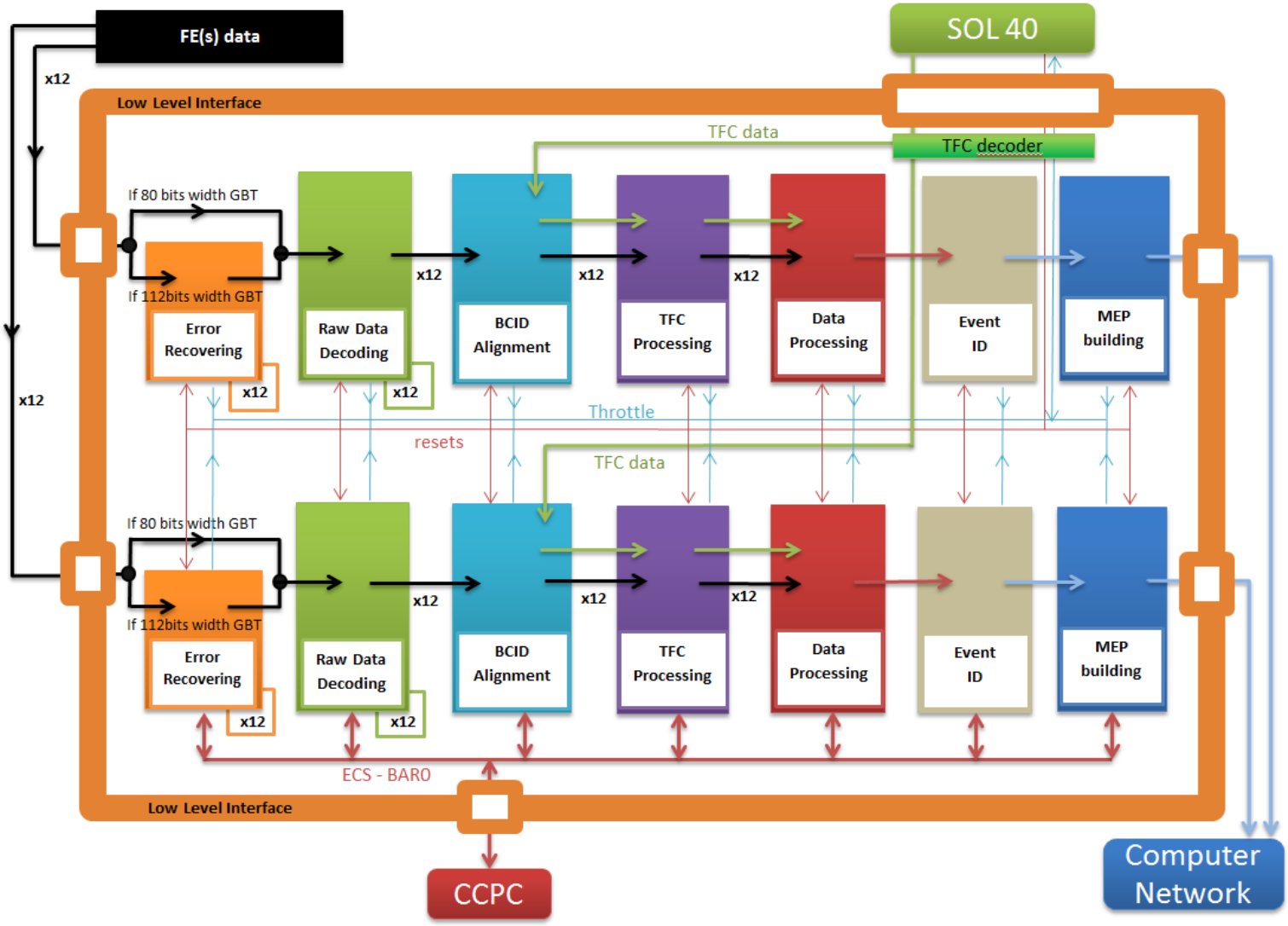
3. Second part of 2014

What we developed in the first part of the year allows us to test a lot the system from the MiniDAQ control Software to the Firmware and the Hardware which leads us to:

- A New Firmware Architecture
- A Firmware for the MiniDAQ with a Generic Data Processing
- A New a data format
- A Software to control the MiniDAQ developed with WinCCOA
- Estimations about the filling of the FPGA according to influential parameters
- Work environment training for sub-detector groups

3. Second part of 2014

New firmware architecture



3. Second part of 2014

2 differences between the previous and the new firmware architecture:

- The TFC processing block has been moved before the data processing (as it was decided last meeting 08.10) and split with the add of the Event ID (since the Event ID is added after the processing of the data)

- There are now 2 data flows receiving half the number of inputs fiber each.

There are 2 mains reasons for having 2 data flows :

→ To be able to use completely the 2 PCIe output interfaces, with one data flow, we need at a certain point to be able to send :

- Either 256bits @ 2xPCIe clock = 500MHz
- Or 2x256bits = 512bits @ PCIe clock

Both solutions have very few chances to fit with the timing constraints.

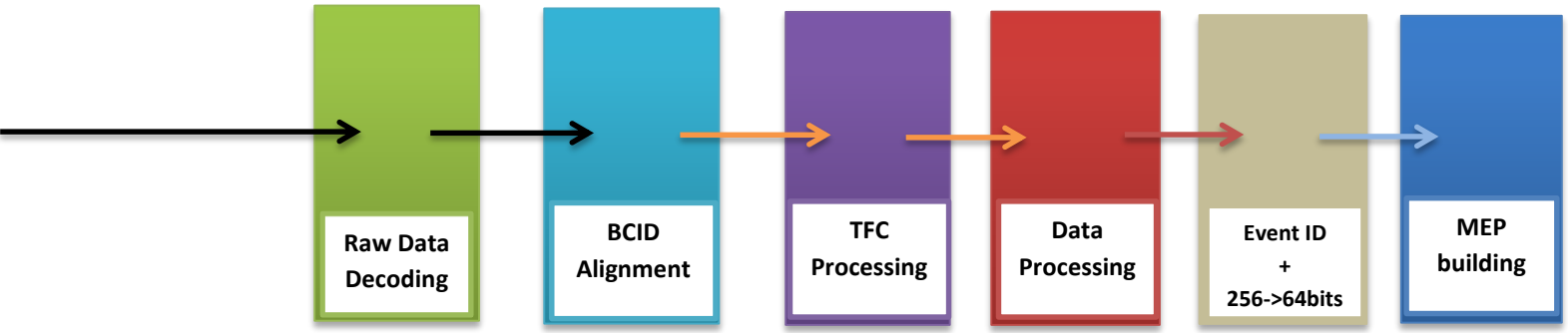
Using them independently is easier and will reduce effort on timing

→ It will reduced the complexity of the alignment and the data processing blocks since less fibers are handled in the same time. Then it will save FPGA cell resources.

It's very easy to duplicate one data flow process, then it's possible to focus on only one data flow like we do in the MiniDAQ and to duplicate it when it will be ready.

3. Second part of 2014

The output interface of the data processing is changing too up to the MEP building



Data (80 or 112bits) / data valid / ready
GBT clock

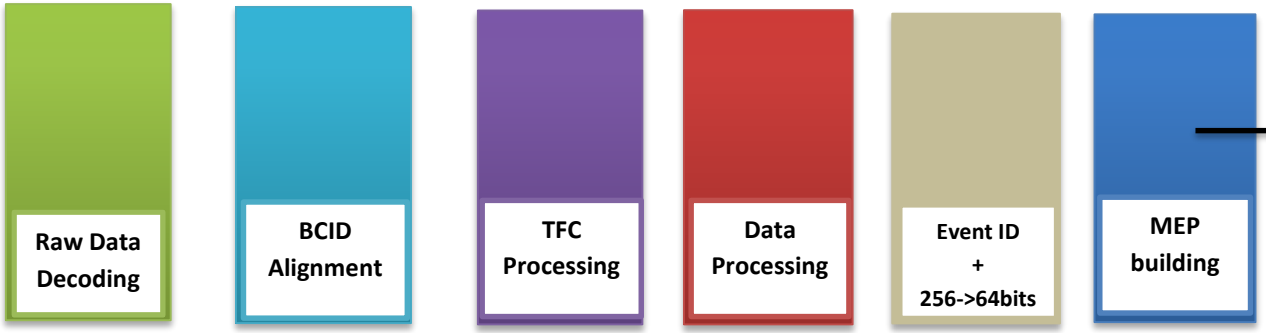
Data (256bits) / data valid / ready
+ SOF /EOF
Clock 10GbE or PCIe

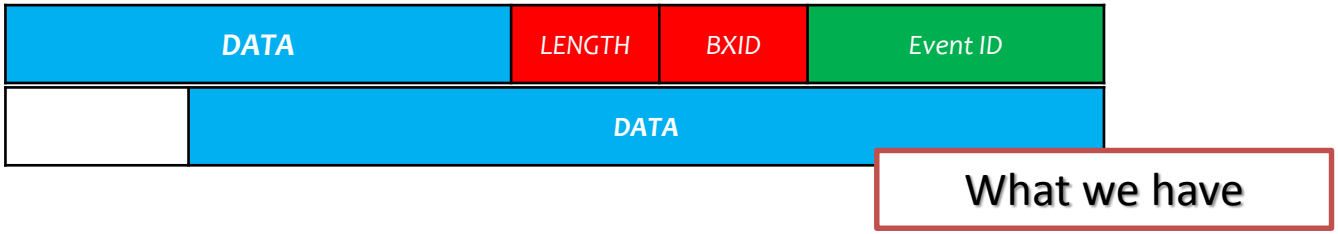
Data (Xbits) / data valid / ready
Internal clock
X = 80 or 112bits + (BCID full length - BCID header)

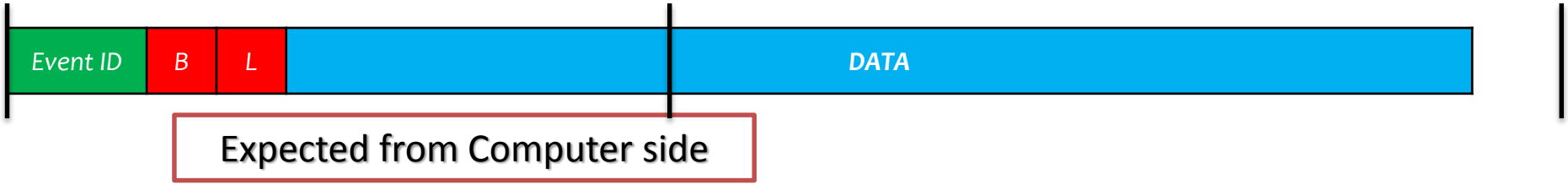
Data (256bits or 64bits) / data valid / ready
+ SOF /EOF
Clock 10GbE or PCIe

3. Second part of 2014

Current Data Format







3. Second part of 2014

Likely future Data Format

Then, the data format could change from FE boards and here are the consequences :

→ *Modification of the code for the user code blocks:*

→ ***Change of the FE data format rules***

- *The header will be in position MSB and not in position LSB=0 anymore*

Cluster 1	HDR	Cluster 4	Cluster 3	Cluster 2	Cluster 1	HDR
Cluster 8	Cluster 7	Cluster 6	Cluster 5	Cluster 4	Cluster3	Cluster 2
Cluster 1	HDR	Cluster 2	Cluster 1	HDR	Cluster 10	Cluster 9
Cluster 2	Cluster 1	HDR	Cluster 5	Cluster 4	Cluster3	Cluster 2
		Cluster 7	Cluster 6	Cluster 5	Cluster 4	Cluster3

This still need to be studied and could be done in the TELL40 firmware
Need to see with each sub-detector groups what could be the best option for each of them

HDR	Cluster 1	Cluster 2	Cluster 3	Cluster 4	HDR	Cluster 1
Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Cluster 9	Cluster 10	HDR	Cluster 1	Cluster 2	HDR	Cluster 1
Cluster 2	Cluster 3	Cluster 4	Cluster 5	HDR	Cluster 1	Cluster 2
Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7		

3. Second part of 2014

A document (EDMS#1451624) about the FPGA resources required by each block of the firmware has been written to help the sub-detector groups to define the % of resources used by their firmware.

Some graphs have been created to define the resources required by the blocks according to influential parameters such as :

- number of links
- GBT protocol
- data length unit
- data packing algorithm

Everything is estimation, values can evolve increasing or decreasing in the future after more precise studies. The differences can be quickly significant with many links.

3. Second part of 2014

GBT Wide-Bus mode protocol:

If a sub-detector group chooses to use the **FV data format** with an **120bits GBT width**, a **data unit value of 4** and **24 input links,** they will fill the FPGA up to **:**
7.8% (LLI) + 34% (D&A) + 1% (TFCP) + 5% (MEP) + 50% (GDP) =**97.8%**

This estimation is not compliant with the specifications. The number of links has to be reduced such as:

If a sub-detector group chooses to use the **FV data format** with an **120bits GBT width**, a **data unit value of 4** and **12 input links,** they will fill the FPGA up to **:**
5.9% (LLI) + 17% (D&A) + 1% (TFCP) + 5% (MEP) + 26% (GDP) =**54.9%**

4. Xmas 2014

What are we going to have for Christmas 2014 ?
Let's spy Santa

2014 Xmas
LHCb upgrade gifts

PCIe40 boards
~~More money~~
~~Bigger FPGA~~
~~More manpower~~

4. Xmas 2014

Then let's continue with the small short but efficient team and let's thanks them for the huge work which has been done this year for the firmware developments:

- Jean-Pierre Cachemiche, Frédéric Hachon, Julian Mendez, Frédéric Rethore, Michel Jevaud, Pierre-Yves Duval, Renaud Le Gac (Marseille)
- Federico Alessio, Paolo Durante, Clara Gaspard, Niko Neufeld, Rainer Schwemmer, Ken Wyllie (CERN)
- Sebastien Cap, Stephane T'Jampens (Annecy)

And the VELO, SciFi and UT teams for their welcoming during the trainings and their work with the environment we have set up.
Hope the MUON and RICH teams will join soon.

Conclusion

We have now a complete chain of the firmware at least for the generic one. And we have improved the definition of the specifications.

In the first part of 2015, all together, we will need to :

- Define the number of PCIe40 boards required by each group
- Define precisely the new data format
- Deeply test and debug the generic firmware (in particular the Generic Data Processing)
- Help the sub-detector groups to develop the specific data processing for the ones which are not going to use the generic data processing.
- Switch from the 10GbE MiniDAQ to the PCIe MiniDAQ
- Develop a complete WINCC0A control of the MiniDAQ