TEL62 and TDCB firmware status

Roberto Piandani

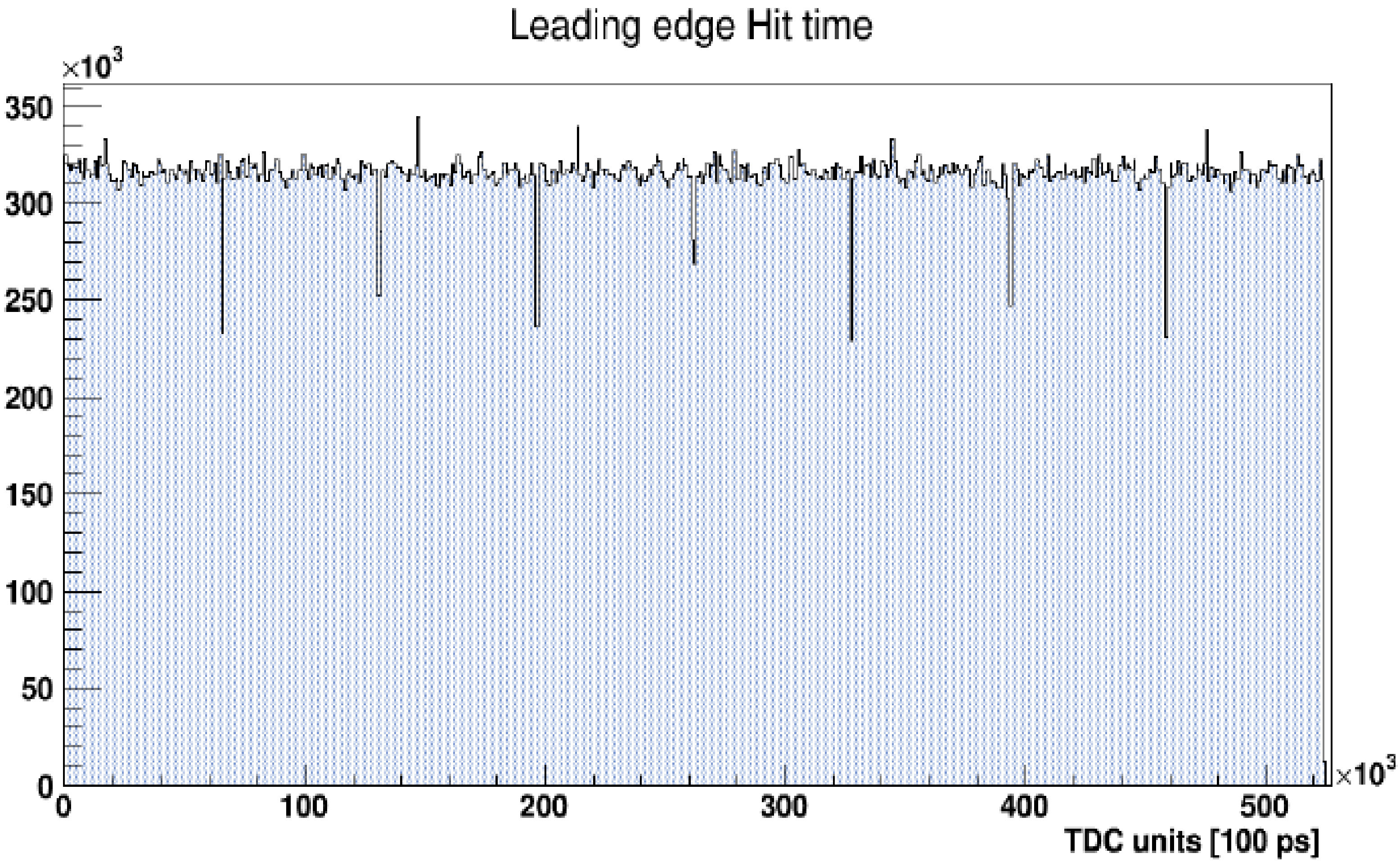
INFN PI

NA62 Collaboration Meeting in Ferrara – TDAQ Working Group

TEL62/TDCB Performances - 1

From Venelin presentation in Liverpool

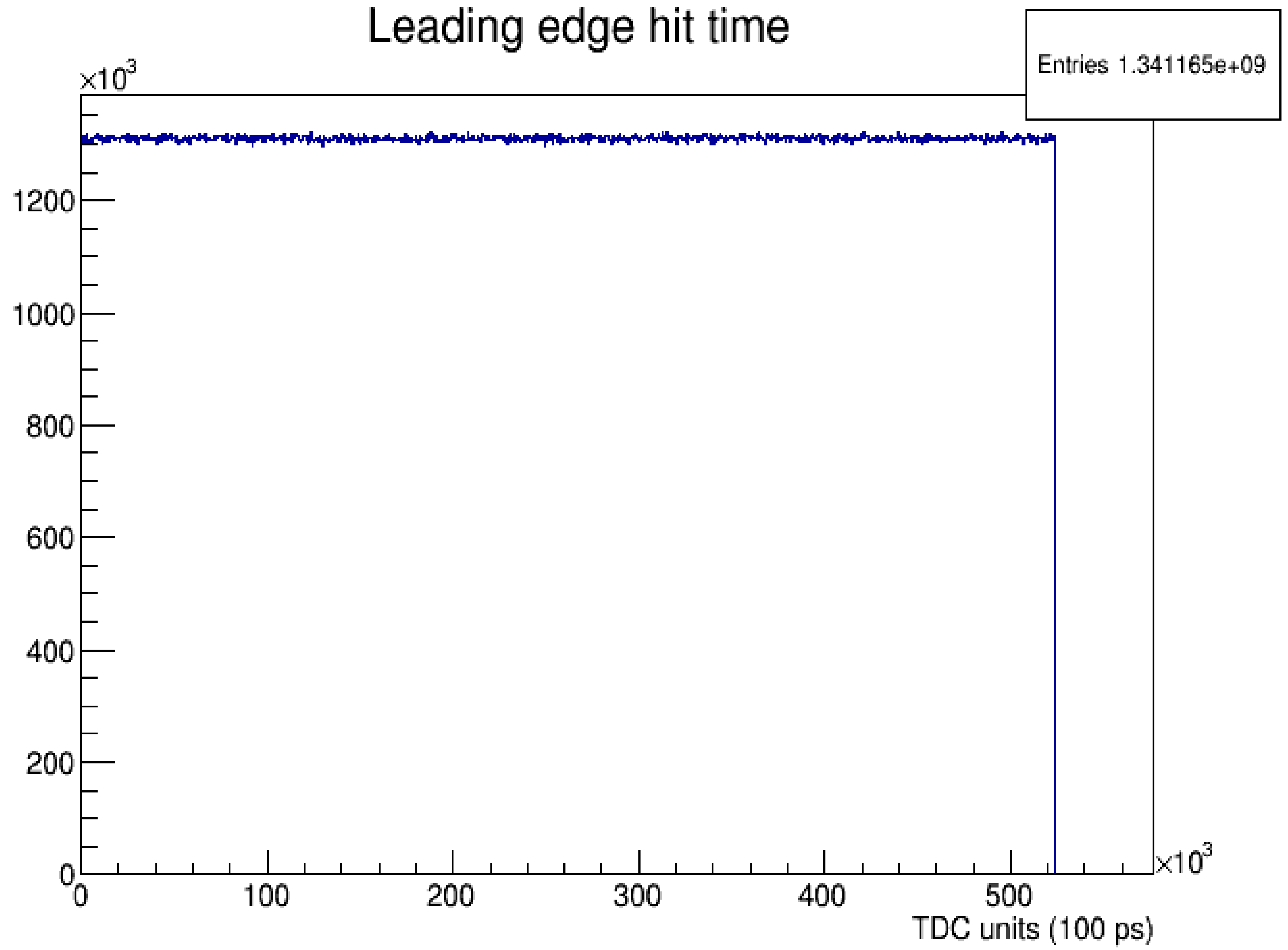
- Visible (and unexplained) structure at 6.4 us



The structure is associated to the TDC fifo saturation: when we have a big amount of hits entering to the TDC (fifo saturation), we have a word's lost at the and of the 6.4 μs readout frame

TEL62/TDCB Performances - 2

If we have a normal hits rate entering the TDC we have a flat distribution as expected



TEL62/TDCB Performances - 3

Is the cross-talk bettwen channels present?

We have checked the presence of cross-tlak in the following to ways:

1) Appearance of channels not pulsed on the data

Looking to a statics of about 107 events: **no spurious channels**

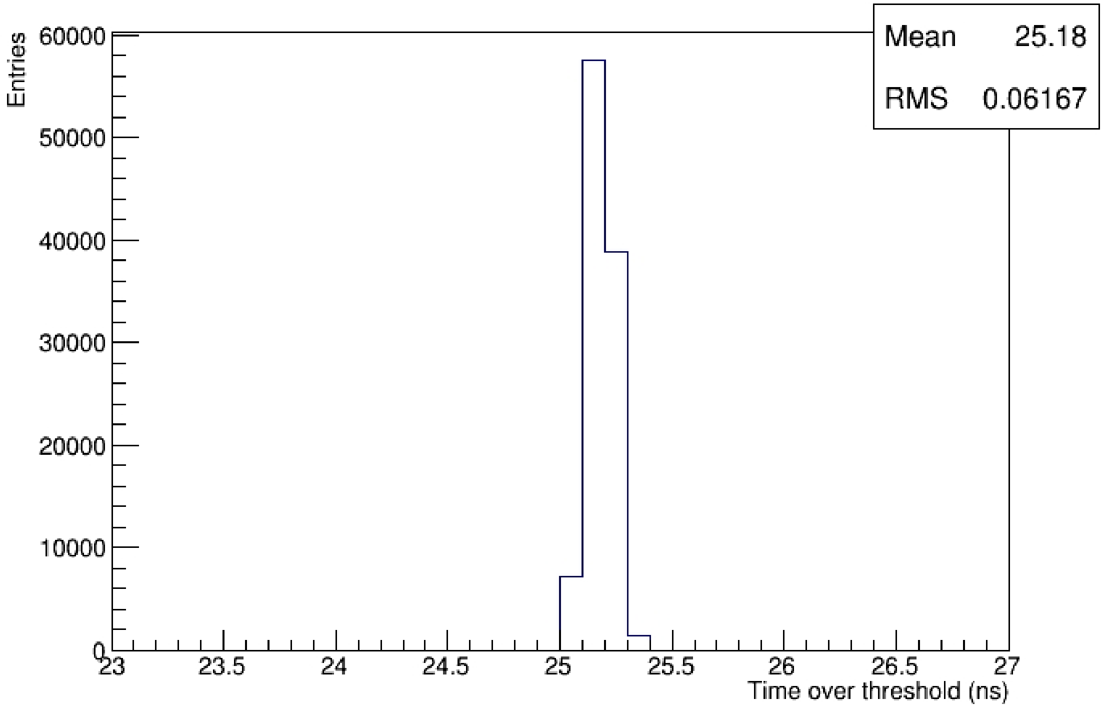
2) Look for possible difference on the time resolution in two different cases

a) Measuring the time over threshold of hits coming from an isolated channel

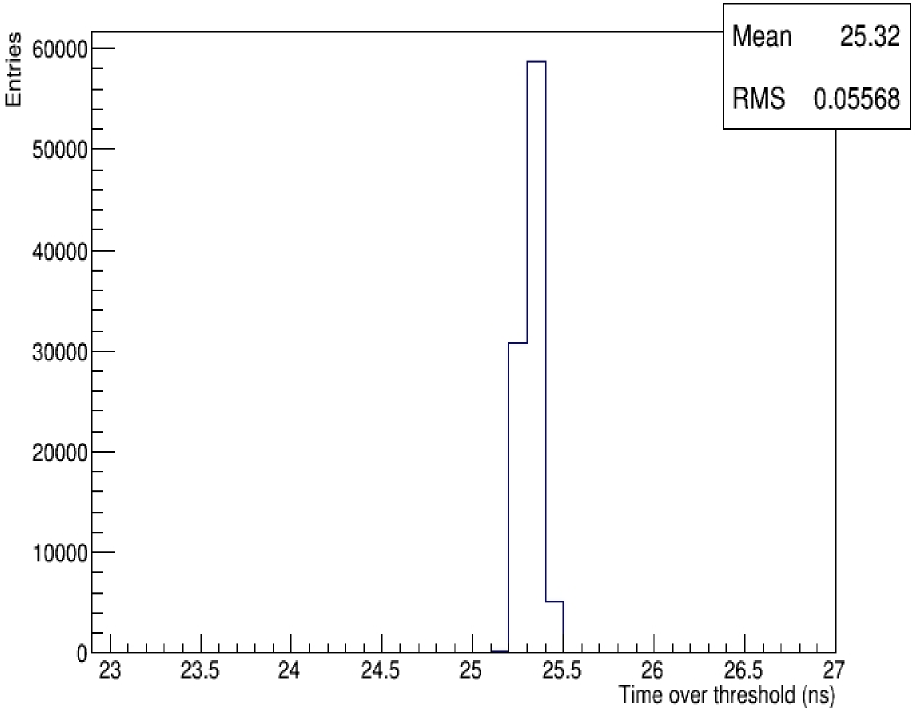
b) Measuring the time over threshold of hits coming from a not isolated channel

No differences found

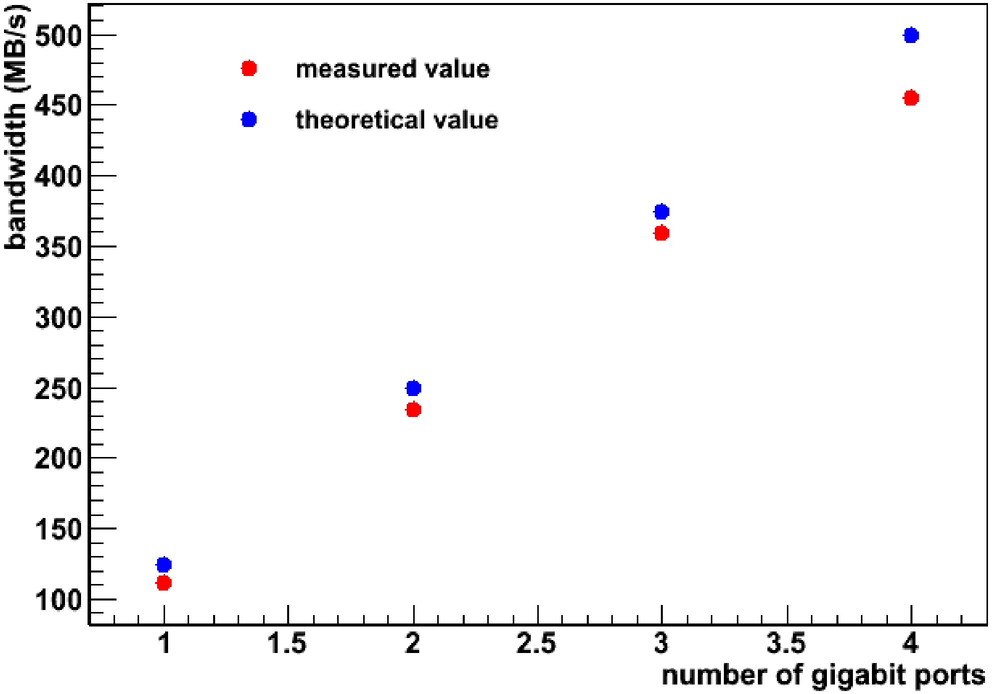
Isolated channel



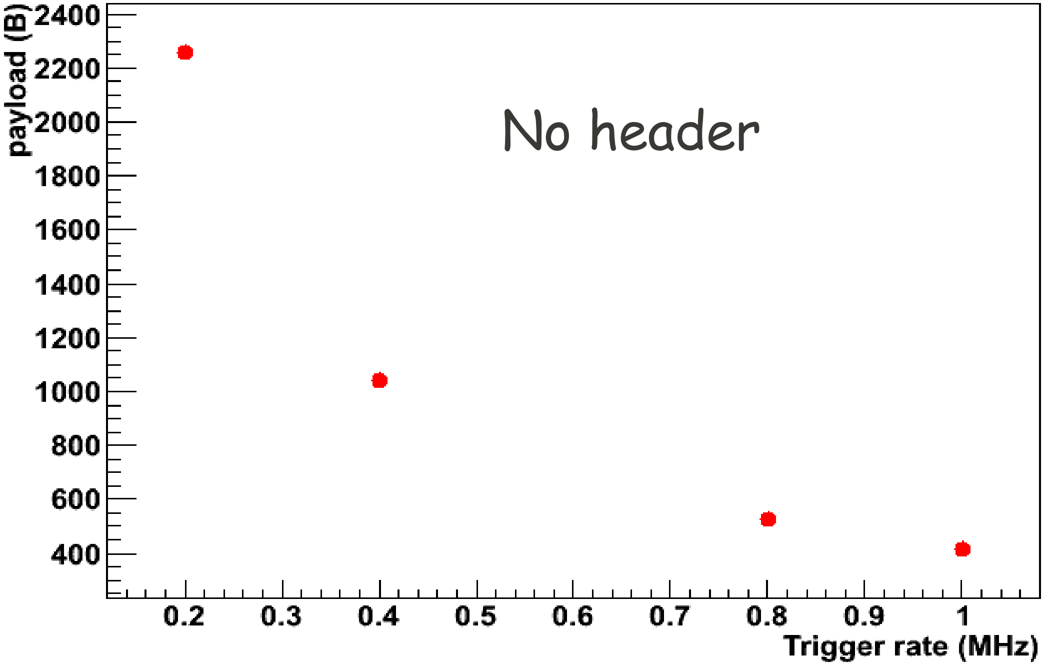
Not isolated channel



TEL62/TDCB Performances - 4



We are practically sautrated the gigabit Bandwidth apart in the last point



No header

TEL62 generic firmware - 1

In order to improve the performances of the tel62 in particular at high trigger rate we have decdied to apply a change in the data format

FPGA header	Flags	FPGA ID	Number of non-empty frames	Number of frames
Frame 1 header	Number of words		Frame timestamp (lower bits)	
Frame 1 data	TDC hit or frame-specific error word			
	TDC hit or frame-specific error word			
	…			
Frame 2 header	Number of words		Frame timestamp (lower bits)	
Frame 2 data	TDC hit or frame-specific error word			
	TDC hit or frame-specific error word			
	TDC hit or frame-specific error word			
	…			
Frame N header	Number of (error) words		*Reserved*	
Error words	Error word			
	Error word			
	…			
	31 24	23 16	15 8	7 0

The idea is to remove from the packet the empty slots

As an example, assuming to ask 10 slot around the trigger and only 2 with 4 words each:

Now → (1 + 10 + 8) words for a total of 76 B

Future → (1 + 2 + 8) words for a total of 44 B

TEL62 generic firmware - 2

To do list:

Understand the real reason and solve the problem of the ddr overflow raised by Tonino and reproduced in Pisa

Understand the reason of the “burst not completed” error, everybody using the tel62 at some point can see the appearance of “BURST” in the status of the sl and PPs

Understand if the automatic configuration of the tel62 and TDCB using the XML procedure load the right parameters for the various registers

TEL62 firmware - 1

Missing parts:

Apart the informations on the TDCB counters, no others specific EoB structure coming from the detectors
Probably will be necessary also a proper decoding of the informations: not in the usual format

The run is approaching and now we have fw for different detectors, we can start to set the ccpc home with different users to use the proper tdspy (already present the structure)

TDCB firmware - 1

The TDCB fw apart minor modification is the same with respect the one dated 13/11/2013

The relevant modification, not so important for the run, is a correction on the tdc data emulator: now is possible to set the number of repetitions in the reading of the memory in which the fake data have been loaded

LAV firmware - 1

The code for the LAV primitives generation written by Francesco in Frascati has been tested several times during the last few months giving good resalts.

In particular has been used also to test the L0TP in the following way:

Starting from the pulse generated with the LAV FEE entering in the TDCB, the fw produce the primitive inside the PP, once the merging of the PPs in the SL is done, the primitives are sent to the L0TP.
The trigger processor elaborates the information and produces a trigger sent to the LTU, from the LTU the trigger returns to the TEL62 to acquire the corresponding data written in the DDR.

The comparison between the input data (the pulses of the LAV FEE) and the output data (data read from the ddr) shows that both the LAV fw and the L0TP work in the proper way (for detail concerning the L0TP see the talk of Dario)

RICH firmware - 1

Also the code for the RICH primitives generation written by Cristiano in Perugia has been tested several times during the last few months giving good resalts (details in the talk of Francesca).

In this case there was a problem in the integration of the specific part of the RICH fw with the generic one.

The integration process:
Starting from the generic structure of the fw (written in Pisa) you have to create a separate structure which is a copy of the generic one where you have to add the specific parts of your fw. In this procedure you have to recreate the same structure of configuration files.

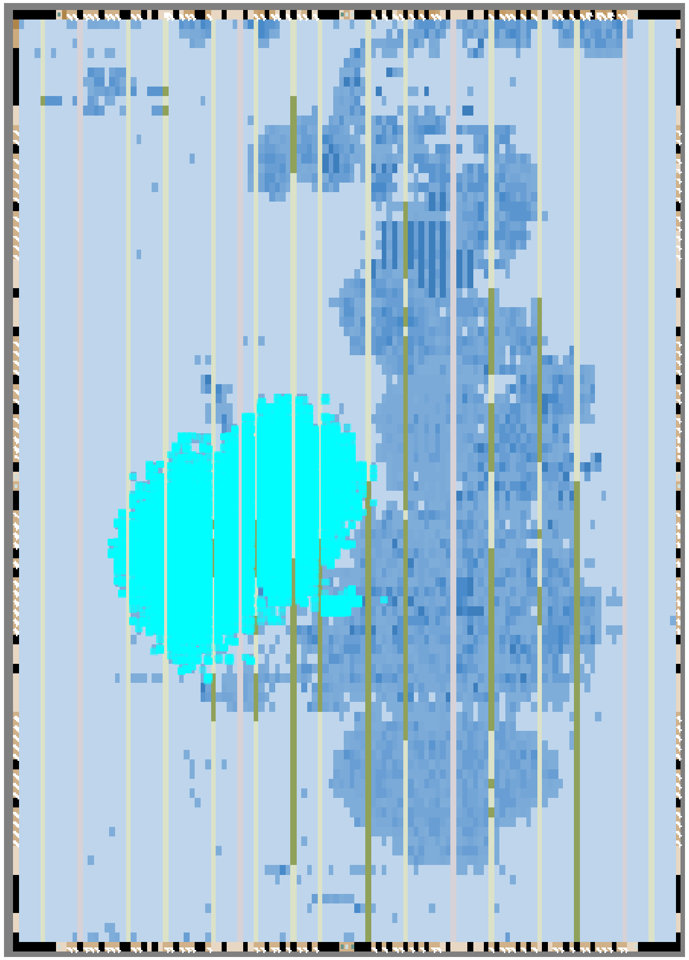
The problem of the RICH seems associate to a missimg configuration file not yet discover by Cristiano, in any case the problems doesn't prevent the possibility of testing the primitives generation part of the fw.
In order to better understand the problem Cristiano, instead to create the alternative structure for the RICH, added his part directly inside the generic fw obtaining the disappearance of the previous problem.

RICH firmware - 2

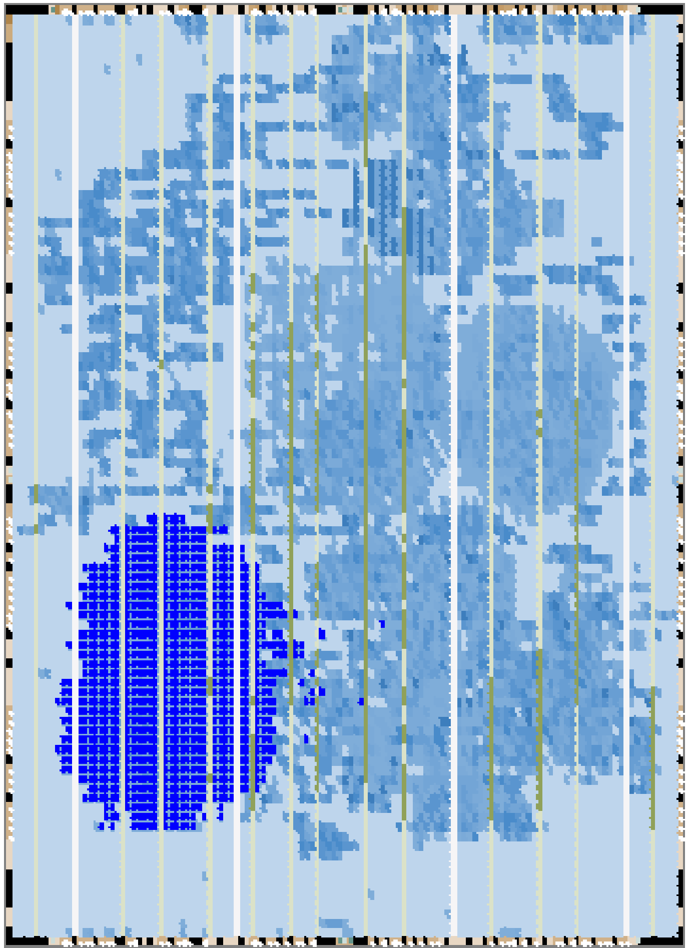
Running this new fw has appeared a new problem in the comunication between the PP and the SL. The problem is concerning the appearance of timing violation in the PP to SL comunication.

In general is usual to set some timing constraints to the delay added in particular in the comunication line between different modules.

To solve the problem Cristiano had to decrease a little bit the range of this constraints.



FPGA occupancy
Generic (left)
Generic + RICH (right)



RICH firmware - 3

In order to prevents certain problems Cristiano is investiagting another way to compile the firmware.

Since now when we add a part of code we practically recompile the entire project, having a differnts positions of the various components inside the fpga.

Another way to proceed is the Incremental compilation: in this case you can write a code compile and perform the test, if the fw is working well you can freeze that part of code and when you add other code and compile you don't touch the previous parts (all the componets remain in the same position)

Conclusions

There are some things to do in order to solve residual problems of the generic TEL62 firmware

There are some missing parts in the detector specific firmware (not so relevant for the operation of the fw)

The LAV and the RICH firmware seems in a good shape

Some problems (under investigation) in the integration of the RICH firmware