DCH FEE STATUS

Some ideas for Level 1 Triggered Data Flow

&

A preliminary study on Cluster Counting Data Processing Algorithm

G. Felici

Outline

- *DCH TRIGGERED DATA FLOW*
  - *Trigger & OL Specs (reminds)*
  - *ADB main blocks & data frame (remind)*
  - *The trigger burst case – 2 possible readout implementation*
  - *A pushing-mode front-end readout architecture*
- *Cluster Counting Data Processing algorithm*
  - *The preamplifier*
  - *Noise filtering based on DSP correlation technique*
  - *The correlation algorithm*
  - *Example of the technique applied to acquired waveform*
- *Conclusions*

Trigger & OL specs (remind)

Specs

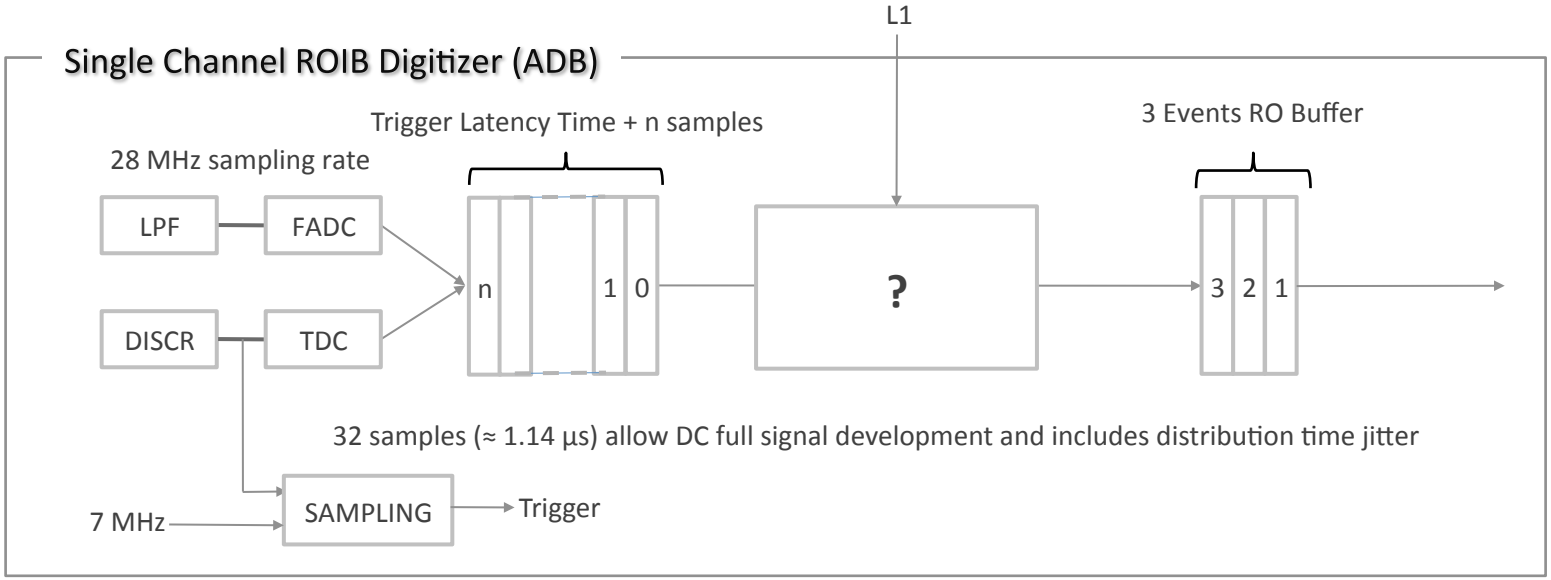
System

- Trigger rate (average): ***150 kHz***
- Trigger (fixed) latency : ≈ ***4 (6 !?) µs***
- Data OL BW : 16 OL @ 2 Gbits/sec
- ECS OL BW : 16 OL @ 2 Gbits/sec
- Trigger OL BW : 64 OL @ 1.2 Gbit/sec
- Trigger spacing (min) ≈ ***60 ns (!?)***
- *Trigger burst : 3 events (?)*

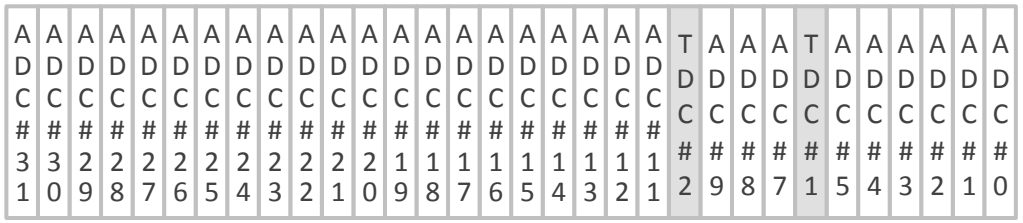
Detector

- Number of cells (guess): ≈ ***9216***
- Chamber occupancy : ***15%*** (Inner layers)
- Chamber gain : $5 \cdot 10^4$ - $1 \cdot 10^5$
- Sense wire parasitic (C_D) ≈ ***25 pF***

ADB main blocks & data frame (remind)



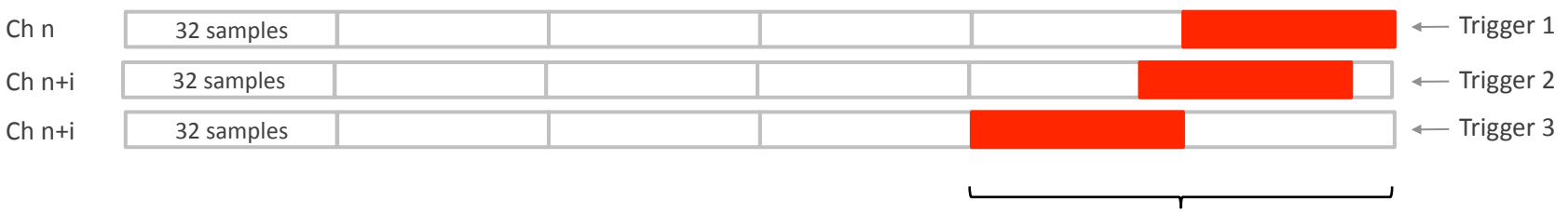
Digitized Data Frame Example



Position inside the frame → coarse time measurement (5 bits)
Content → fine time measurement (6 bits)

The trigger burst case

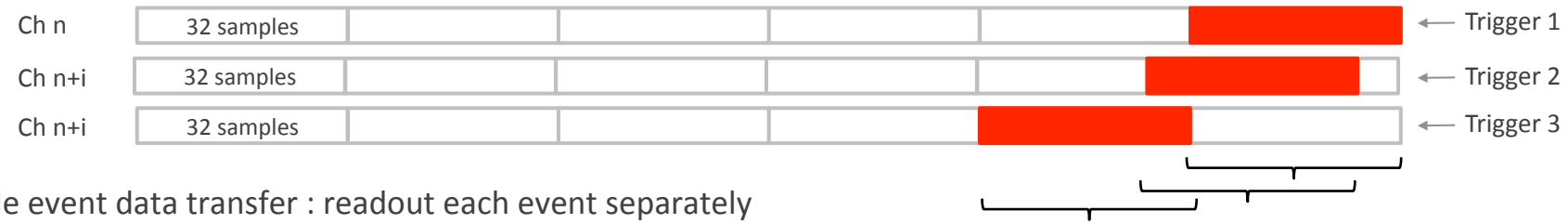
Data transfer optimization



FEE data transfer optimization : download the 3 events as a single “big” event

- Pro : Front-End data transfer optimization
- Cons :
  - events are overlapped in the same data frame (further elaboration required to split single events)
  - data frames do not have the same lengths (L1 trigger occurrences)

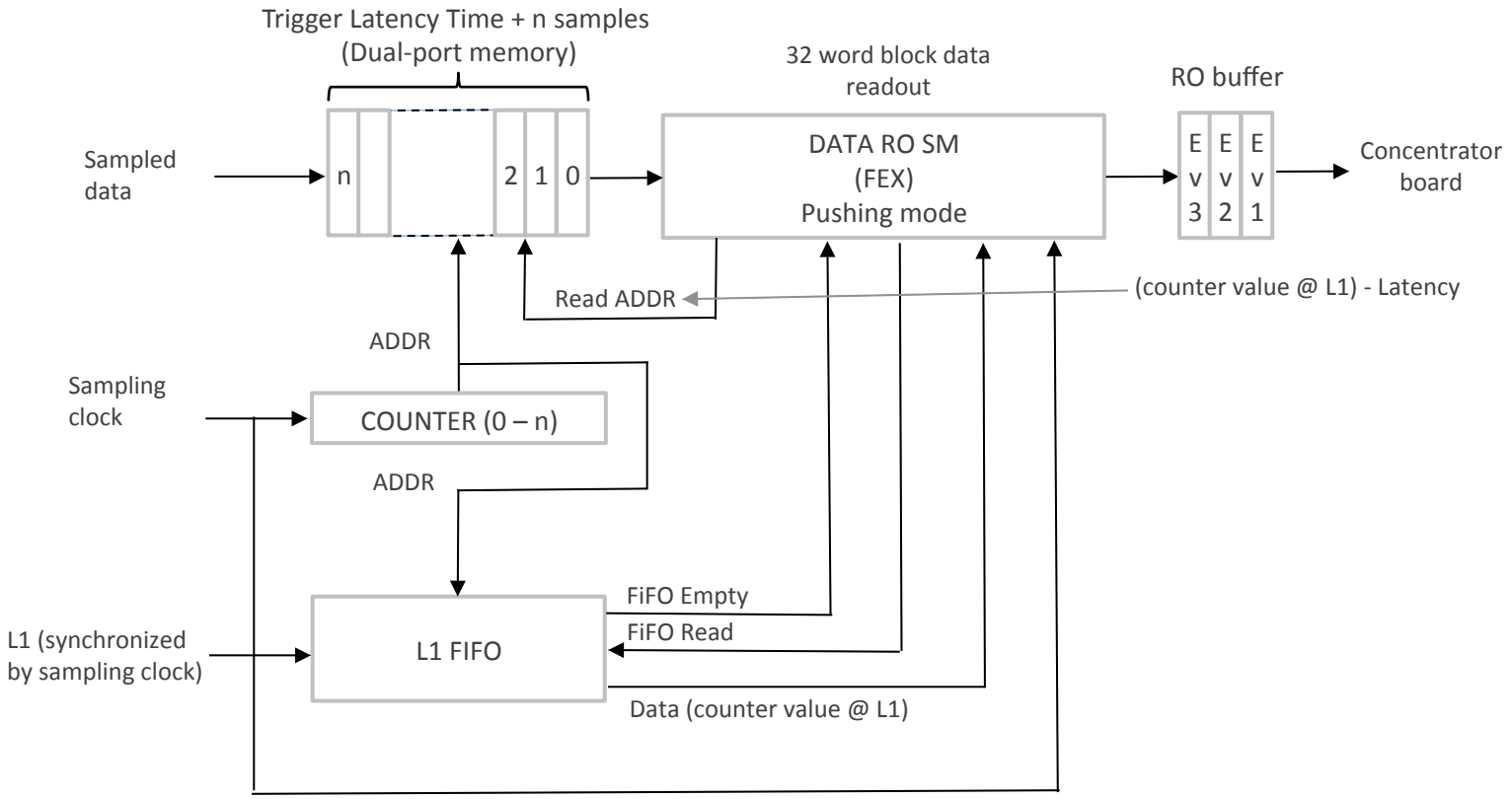
Event management optimization



Single event data transfer : readout each event separately

- Pro :
  - Front-End events have the same size
  - FEX can be applied while reading the event
  - Readout procedure provides event de-randomization
- Cons :
  - Partial (previous) L1 event re-reading

A pushing-mode FE readout architecture

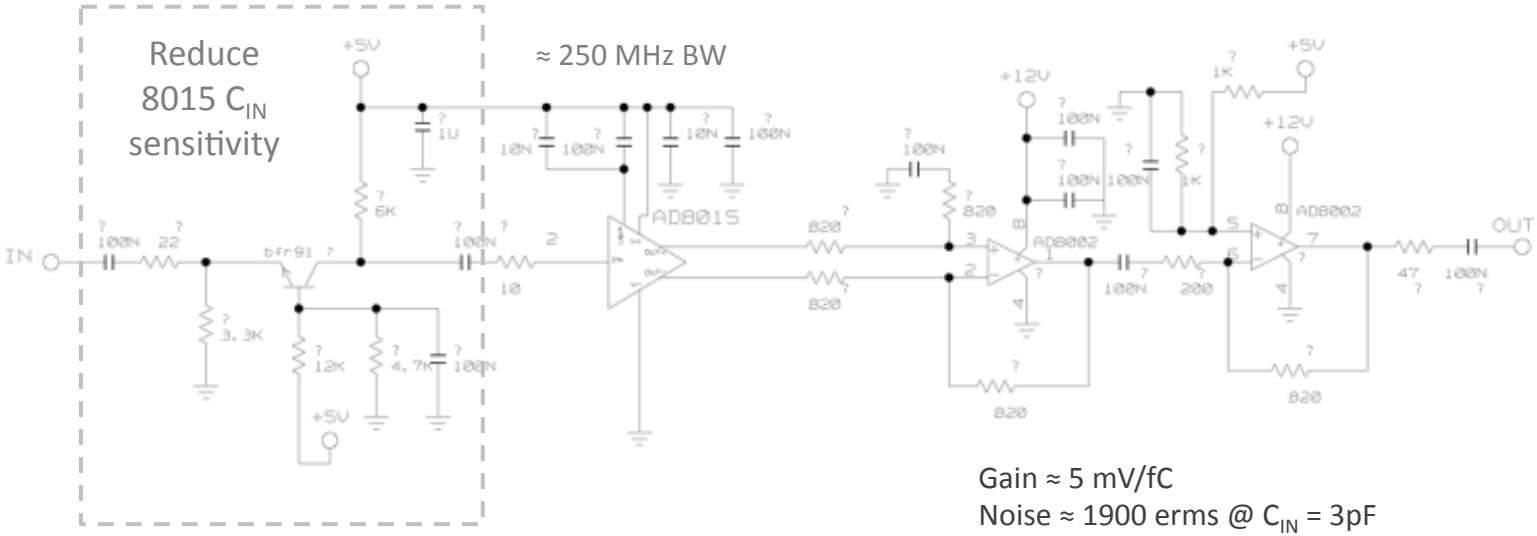


NB : minimum trigger spacing > sampling period (≈ 36 ns)

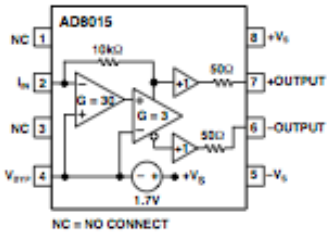
A simulation of the readout architecture will be carried out in the next weeks

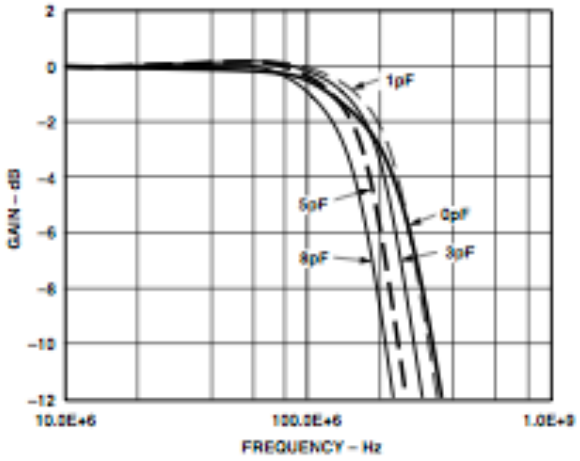
A preliminary study on Cluster Counting Data Processing Algorithm

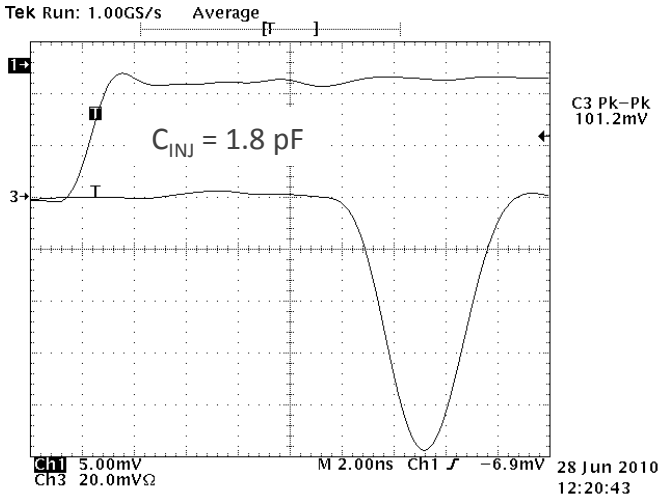
Preamplifier



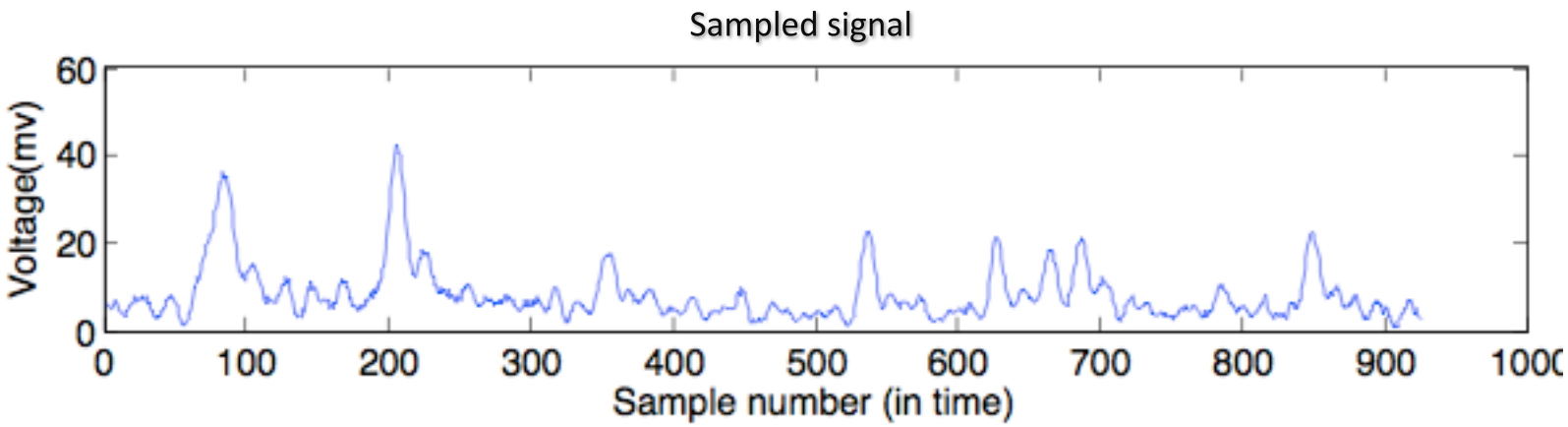
Gain ≈ 5 mV/fC
Noise ≈ 1900 erms @ C_{IN} = 3pF





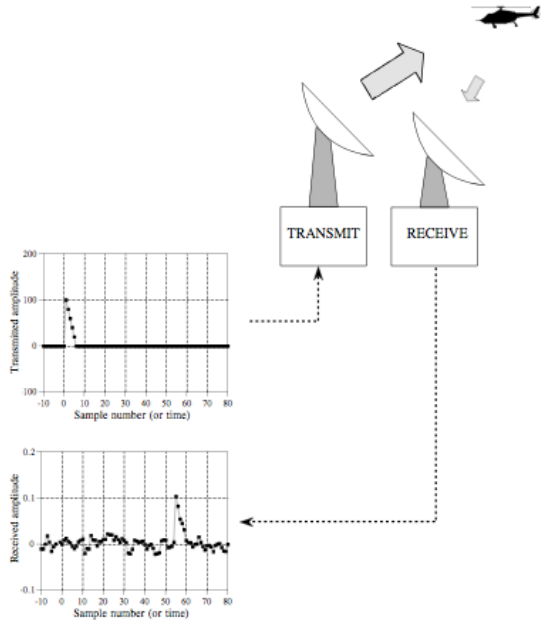


Cluster Counting – noise filtering



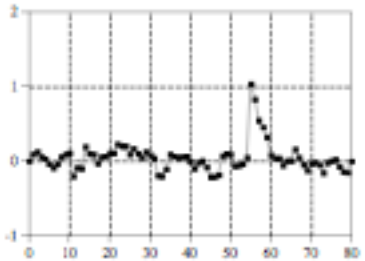
noise filtering using an algorithm with low area consumption in a FPGA

We are trying to use a DSP technique based on the correlation concept to determine if a target signal (a signal with a shape close to a well separated cluster) is present inside another signal (the digitized signal). This technique is generally used in radar systems where the received signal consist of a shifted and scaled version of the transmitted pulse and overlapped random noise (interference, thermal noise etc)



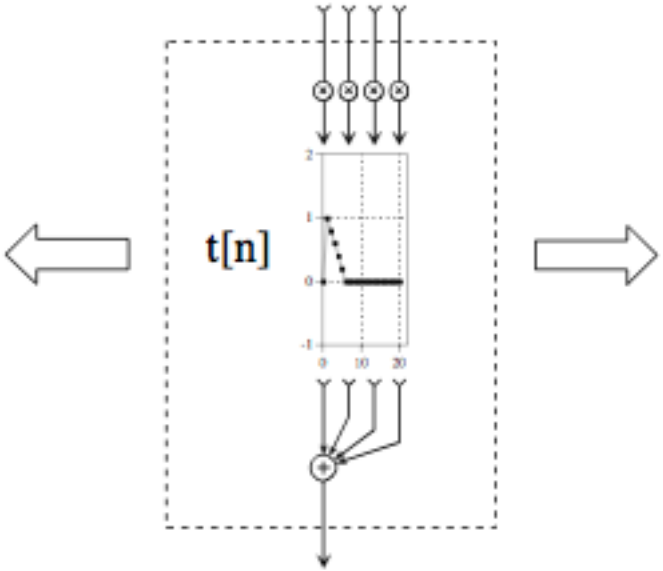
Cluster Counting – The correlation machine

Received signal **x[n]**



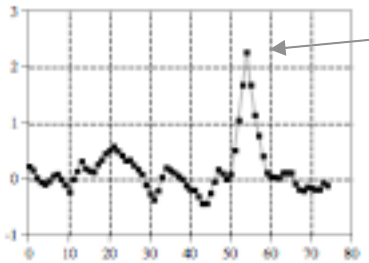
Correlation is the optimal technique for detecting a known waveform in random noise (matched filtering technique)

Target signal



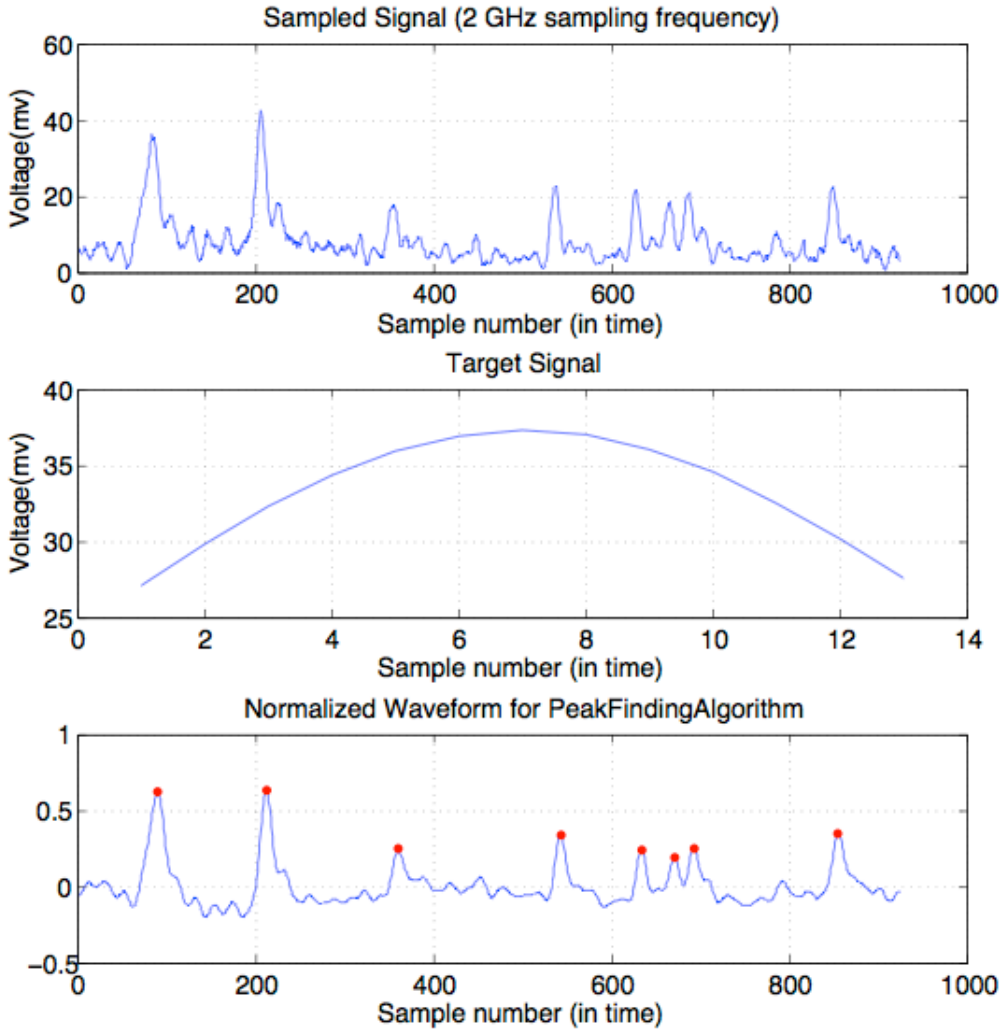
$x[n]*t[-n] = y[n]$

Cross-correlated signal **y[n]**



The value of the cross-correlation is maximized when the target signal is aligned with the same features in the received signal

Cluster Counting – An example



- The Peak Finding algorithm is under study

Conclusions

- A (very preliminary) front-end data readout architecture capable to manage short dead-time (less than 100 ns) has been presented. We plan to validate the architecture by means of (VHDL) simulations in the next weeks

- Some work has been done to verify the possibility of using Cluster Counting for dE/dx measurements. A setup made of a short tube (≈ 40 cm), a fast preamplifier/ digitizer has been built.
- A very preliminary algorithm based on a matched filtering and peak finding procedure to count single ionization events has been implemented. First results look encouraging.