Subject: Templates for Signal Processing
Posted by Lars Schmitt on Thu, 22 Jul 2004 09:59:56 GMT
View Forum Message <> Reply to Message

Dear colleagues,

I would like to start to collect information on prior experience in our groups on all types of detector oriented signal processing.

I would first like to do this in an informal brainstorming way via this forum hoping that people can contribute a description in a few sentences. In addition you could also report on articles you read or talks you heard.

The topics I would like to cover in this first round are the following (but you may add others you think are important, too):

* All types of noise reduction:
  - Zero suppression via thresholds
  - Common mode noise subtraction
  - Amplitude reconstruction (MIPS vs. reaction/delta electrons)
* Signal time reconstruction
* Clusterisation algorithms (1D and 2D, maybe 3D for TPCs...)
* Pre-pattern recognition with limited scope
  (e.g. ring matching for RICHes, tracklets in close chambers)

Previously, some of these items were done only by the reconstruction software. The algorithms used there are however of equal interest, since in future(DAQ) they can be implemented in FPGAs on the frontend.

Regards,
Lars

{some markup added by moderator}

Subject: Re: Templates for Signal Processing
Posted by Alexander Mann on Thu, 26 Aug 2004 10:09:25 GMT
View Forum Message <> Reply to Message

Dear colleagues,

here is a first collection of the algorithms I implemented in our FPGA based frontend modules for
a PET detector readout (some of them were also mentioned in
http://www.gsi.de/documents/DOC-2004-Apr-17-1.pdf):

1. Pulse detection / Zero suppression
The analog detector signals are continuously sampled by ADCs and the sampled data are processed
in FPGAs. To distinguish a signal pulse from background noise two summation windows are used, which

are compared with a threshold level. The size of the windows is selected in a way that the uncorrelated noise should cancel out.
If there is no signal pulse, the sum values of both windows are nearly equal. If a pulse enters the
first window, the sum starts to rise and a trigger is generated if the sum is bigger than the second window sum plus a threshold.
The value of the second window can also be used for pedestal correction of the sampled data.

2. Time reconstruction
After the pulse detection some pedestal corrected sample points are processed by the time reconstruction
algorithm. Until now there are two algorithms implemented:

- SAD Fit
In this iterative algorithm the detected rising edge of the signal is compared with a previously generated set of data points with different phase shifts to the sample clock. Therefore an error
value is calculated for each data set which indicates its difference to the actual sampled data.
The minimum error value identifies the phase shift of the sampled data. The pulse time is then
expressed by: pulse detection clock count + phase shift.

- Mean Weighted Sum of Derivation
This algorithm is also operating on the rising edge of the detected signal pulse. At the beginning,
the first derivation of the data stream is calculated, followed by a search for the maximum value.
This gives the position of the turning point of the rising signal edge. Supposed the signal is almost
symmetric, the maximum and the adjacent values carry information about the phase shift of the signal.
Therefore, a sum is calculated from these values which are weighted with a time index and a binning value.
After the summation it is divided by the sum of sample values which results in a mean time value for
the rising edge. This value is then corrected to get again the time information in the format: clock
count + phase shift.

The timing resolutions are for both algorithms around 5 ns (SNR 10) and can reach up to 1 ns and shorter
for good SNR values (> 50).

Regards,

Alexander Mann