I. OPERATION MANUAL FOR LYRTECH DIGITAL DATA ACQUISITION SYSTEM AT YORK

This system consists of 2 VHSADC modules each equipped with Virtex-II fpga and 16 channels input at the front. Each input is internally terminated at 50 Ω. Optionally an external trigger and clock inputs are also available but most of the day to day applications would not require these because the internal on board clock of 105MHz and internal trigger on the fpga would be sufficient for most of the applications.

The cPCI crate would switch on using the on/off button at the front. This would also start the processor which is one of the cards installed in the crate. The screen, keyboard and mouse are connected to this card and Windows XP login prompt on the screen would be displayed once the processor is booted. You must login with the username **Lyrtech** with passward same as the username. (Alhough the account has administrator privilege, one must not use it to open any other account or try to install any new program on the system). Use the following steps to run the data acquisition program.

1. **Turn on the cPCI crate.**
2. **login using username** *Lyrtech* **and password the same as username.**
3. **Click on the following three icons (or four icons if listmode data is also needed) on the desktop.** Wait for few seconds after any click till any new window is opened and all the outputs are displayed on them.
   (a) System
   (b) RTDEXV4
   (c) TapeServer (optional)
   (d) MIDAS
4. On the **Lyrtech Base frame** window click on "Lyrtech VHS-ADC". A new window "Lyrtech VHS-ADC Client" would open. Here do the following:
   (a) Fill in **VHSADC WebServer** as *localhost* (generally set as default).
   (b) Set **Board Number** = *1* (set as default but use selection if have more than one boards).
   (c) Click on **Report Board information**. If the board is identified then relevant information would be displayed there.
   (d) Click on **FPGA programming** and a new window will appear where following settings have to be done (leave other settings as default):
      i. **Browse** for Bitstream file (default is *C:/MIDAS/Lyrtech/bit/Virtex2/SAGE_rel_4_2_12ch_v1_21Sep10.bit*).
      ii. Click on **"Upload & Program FPGA"** to upload the bit file (firmware) and wait for a message to tell you that fpga has been programmed successfully. If result was not successful then it should be generally due to incorrect firmware.
      iii. Select **ADC Clock = 105MHz** and click on **"Program ADC Clock Now"**.
      iv. select **Frame Size** = *8*. (this would ensure that raw traces are not stored in the fpga buffer. If Raw traces are required to be collected then select higher value of frame size. There are some limitations on this option. For example: As the buffer is only 2048 samples deep, therefore, highest value (2048) would work only if one channel is used while 1024 could be used with two channels.
   (e) Now click on the **Firmware Setup(User)** and set the required parameters, as below, in the new window which is opened.
      i. Select correct **Board No.** and correct **channel No**.
      ii. Click on the channels being used in front of **Apply Settings to channels**.
      iii. Click **enable** next to "Channel". Also make sure that the unused channels are not selected.
      iv. Select "Trigger Polarity" this is actually the polarity of your input signal since the trigger is internal in most of the applications.
      v. Select **Trigger Mode** as *CFD only* for internal trigger.
      vi. Baseline Subtraction : *On* (unless you want to set it off).
      vii. Energy Digital Gain : Upto 8 can be used (if desired).

viii. Still in the firmware set-up window fill the following values :

A. CFD Threshold : This depends upon your signal.

B. TFA Shaping Time (ns) : This should be slightly longer than the risetime of the preamp signal.

C. Shaping Time (μs) : this is shaping time for your signal and it is equal to the sum of rise time (see below) and the flat top time of the trapezoid.

D. Rise time : This would be the risetime of the trapezoid. This is also the window of integration so it should be longer than the pulse rise-time and sufficient to have as many as possible points within the window of integration.

E. Peak Sample : This is time corresponding to the point on the flat top portion of trapezoid from where energy would be sampled. This parameter defines location of this point with respect to the CFD trigger. As it may not always be possible to visualize these signals (unless using TDRI module) this should be set slightly longer than the rise time but shorter than the shaping time. This can further be optimized by minimizing the energy resolution.

F. Peak Separation : Two peaks arriving in time interval shorter than this value would be considered piled up.

G. Decay TIme (μs) : This is the decay time of the exponential pulse from the preamp. This parameter affects the energy resolution. Therefore, This should be set close to the actual value and then should be optimized further by minimizing the energy resolution.

5. Now click on Experiment Control in **Lyrtec Base frame** window. The Experimental control window will open up. In this window enable the histograming and then click on yellow coloured SETUP button and select "setup everything". After this define a Volume name which is just the name of directory which will be created in the *C/TapeData* area and then click **Continue** Now enable both Tape-Server and Tape-Transfer in the Experimental control window and then proceed for pressing the yellow coloured SETUP button in that window. before clicking the yellow SETUP button. After this the data acquisition can be controlled by the Experiment Control window. Now one can monitor statistics or access the energy histograms via controls provided on the same window. If you want listmode data to be written on tape then You will need to enable tape server before the SETUP. For doing this click on Tape control and in the Tape control window select the drive by clicking on button next to Drive named *file0* so that it displays "Yes". Now click on **Load Tape** and select *initialise (New Tape)*.

   For any further suggestions contact pj9@npg.york.ac.uk

II. FIRMWARE

The firmware is very detailed. However the basic operation can be summarized as the block diagram shown in the fig 1. The PCI backplane is used for reading the data or read/write of users/daq parameters into the fpga. As shown in the block diagram the event buffer generally comrises of the energy and timestamp values. The data read process is enabled using a handshake mechanism involving control registers. Under this mechanism once a buffer is ready, a specific register on the fpga is set to 1. This register is polled by the DAQ program continuously and upon finding a valid value that buffer is collected by the DAQ using RTDEx channel via the PCI backplane. The lower part of figure 1 shows the processing block each channel is individually going to pass through. Each block comprises a Trapezoid filter (TF) for energy measurement and a TFA ad CFD so that CFD could be used for timestamp. Every valid CFD trigger will increment a 48 bit timestamp which is passed along with the event to the event buffer and then to the computer. Further description of the TF, TFA and CFD is given below.

A. Trapezoid Filter

The trapezoid filter (TF) us very commonly used in nuclear physics and is the digital counterpart of the shaping amplifier. Only the mathematical formulation is given here while its digital implementation, which may be accomplished in various ways, is not discussed. An implementations of this filter is also called Moving Window Deconvolution method, however, as mentioned earlier, the essence is still the same.

The TF performs two things

1. Filtering: The TF should be able to filter out any random noise from the signal. This is performed using a simple FIR filter in which the gains of all the taps are same (see fig 2). Mathematically it means a simple averaging

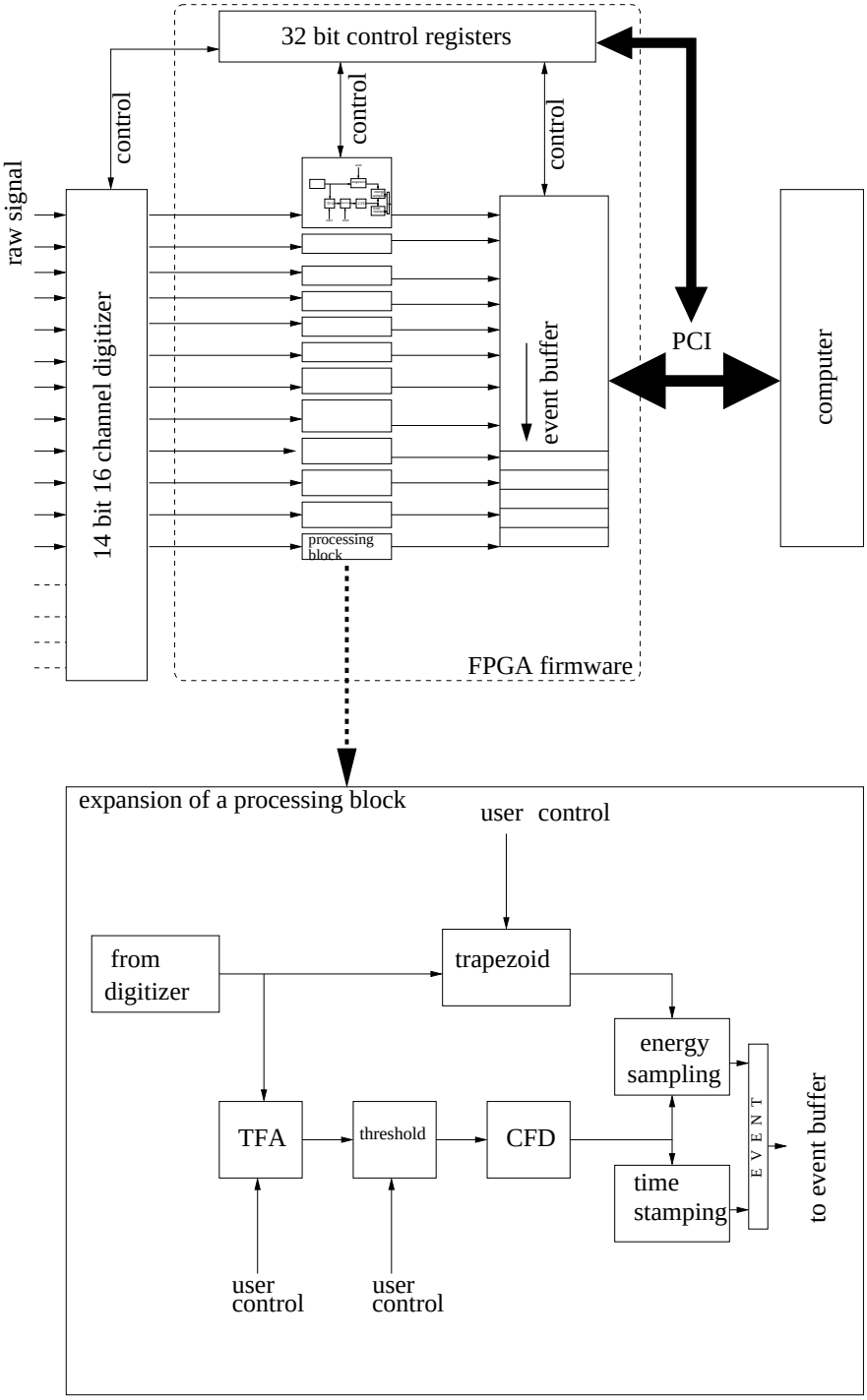

FIG. 1: A simple block diagram showing the process of data acquisition using Lyrtech system. The PCI is the only gateway used for data acquisition as well as control purposes. user control of the various blocks is performed using a block of 32 bit registers. These registers which exist on the FPGA can be written or read by the DAQ program and depending upon various flags their values could be further distributed to specific locations within the firmware to control various tasks. The firmware is actually more detailed than this. One main thing which is not shown here is circular buffer to store traces. However, in most of the applications do not involve recording the traces.

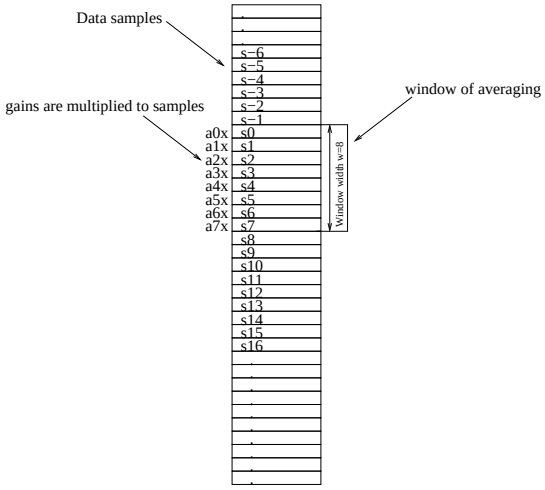

FIG. 2: A general implementation of an FIR filter with 8 taps (a window width $w = 8$ is used for averaging) is shown. The gains (or weights) of the taps (a_i's) would define the frequency response of the filter. In case of random noise filtering, which is mostly the case in nuclear physics, all the gains are set to be same (=1). If i^{th} input is s_i then any k^{th} output o_k from the filter can be calculated as $o_k = \sum^{i=0,w-1}(a_i * s_{i+k})$. By changing k in this formula one can slide the window of integration ($k = 0$ is the case on display). Due to this reason this is also called sliding scale averaging.

of the signal within a given time window. Therefore, before performing this averaging task, the exponentially falling input function should have been shaped to a constant height function (like a step function) the height of which could then be averaged. Therefore, converting an exponentially falling function into a step function is the first tasks to be done.

2. Shaping: It should generate a pulse in the shape of a trapezoid. The height of trapezoid is proportional to height of the signal at the input. Ideal input function required for this shaping operation is a function with very fast rise time and an exponential fall time. The signals from charge sensitive preamplifier coupled to modern detectors are generally of this form. In reality the parameters of shaping have to be chosen keeping in mind of the actual rise time and fall time of the input pulse.

Shaping algorithm: In order to convert an exponential into a step function one can first integrate the original falling exponential function A as shown in fig 3a. This would result in an increasing exponential. The relative amplitude of this is much higher then original signal due to the effect of integration. However after suitably normalizing its height it (shown as B in fig 3a) can be added to original signal in order to obtain a step function (fig 3b). You can see from the figure that the input pulses may be coming before the exponential decay of the previous pulse has died out. In such cases the height of pulse can not be estimated from the raw data unless a shaping is performed. The shaped step pulse on the other hand, provides a good measure of the height of exponential since it is now riding on a step of previous pulse rather than an exponential. Once the step shaping is performed it is also advised to differentiate the pulse once. The result of differentiation is to generate a barrier shaped pulse (fig 3c) from the step. This is helpful because this helps to restore the baseline to zero after every pulse which was otherwise not possible with the step function. In the step function every new pulse would add a step over the previous step thereby the height of the final step is ever increasing. This problem is taken care by differentiating the pulse. Finally a sliding scale average (as described in fig 1) is performed and one obtains the trapezoid (fig 3d). This operation removes the noise. Any noise of frequency smaller than 1/(integration window) is not possible to be removed.

Shaping parameters: The shaping time parameters are *rise time* and *shaping time*. These and some more explanation is given in figure 4.

Pileup: It can be seen from the fig 5 that the sometimes pulses which are close together in the time can generate pile-up. These pile-up pulses will have a larger width than the normal pulses whose width is precisely known. This information can be used to remove pile-up pulses. Alternative way to detect a pileup pulse is to see how close in time the pulses are separated (please see fig 5). A timing pick-off by a CFD would tell when a pulse might arrive. If two pulses arrive in a time duration which is less than a value set by the user then it could be considered a pileup.

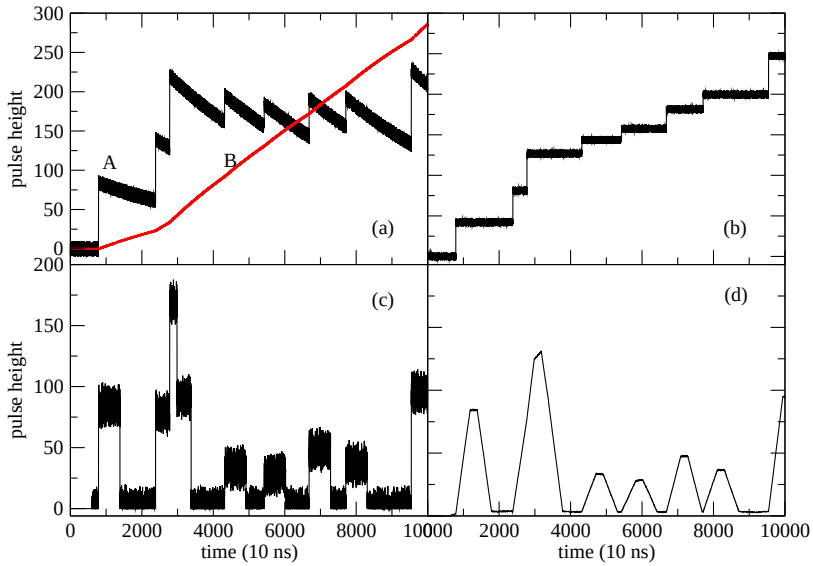

FIG. 3: The process of generation of trapezoid starting from a raw preamplifier signal is shown. For details please see the relevant part of the text. The simulation is done using a fortran program called trap.f which is available alongwith (at york npg wiki page) this document and it can allow to simulate the effects of various shaping parameters in the Lyrtech. A slightly different C++/root code trapezoid.cc which can do similar operation is also available at the wiki page.

This logic is used in the Lyrtech firmware and the corresponding parameter is *Peak Separation* can be defined in the Firmware Setup(User) menu.

TFA shaping: TFA shaping is performed on the raw preamp signal so that it can be used in for CFD for the timing pickoff. The TFA algorithm involves exactly similar shaping as performed for the trapezoid. The only difference is that the integration and differentiation times for TFA are very small compared to the Trapezoid. This would mean that the TFA will have only minimal noise filtering. However, this narrow width would ensure that the TFA output is just only as wide enough as to not to overlap with another pulse coming shortly after it. The parameter called *CFD Threshold* in the Lyrtec firmware set-up(User) is actually the threshold on TFA just before it goes into the CFD. This means pulses smaller than the threshold will not trigger on the CFD. In addition to removing the low energy/noise from the spectrum this also ensures that any multiple triggers are not generated by the CFD due to ringing, after a genuine signal.

CFD: The CFD action is performed in the usual way as performed in the analog electronics. The zero crossover point is detected in the firmware and this is used for timestamping the event in the particular channel.

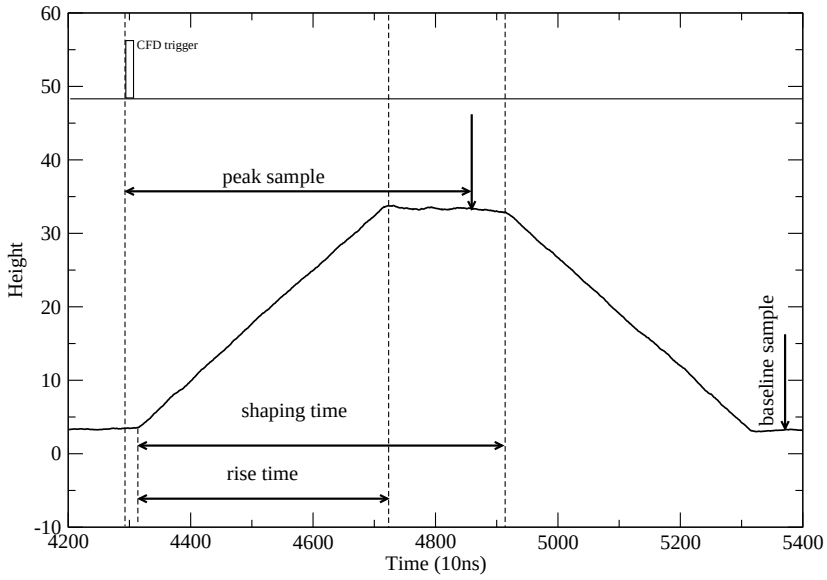

FIG. 4: The shaping parameters for a trapezoid are shown. The integration time is the same as rise-time while the differentiation time is called shaping-time. Please note that the CFD trigger may not occur exactly where the trapezoid is starting. Also note that the risetime and shaping time are measured from the edge of the trapezoid while the peak sampling time is measured from the CFD trigger. Furthermore the peak sampling can be done at any point on the flat portion of the trapezoid and the sampling point does not need to be exactly in the middle of the flat portion of the trapezoid. For these reasons it is advised to see the signals on the scope if such a facility is available.

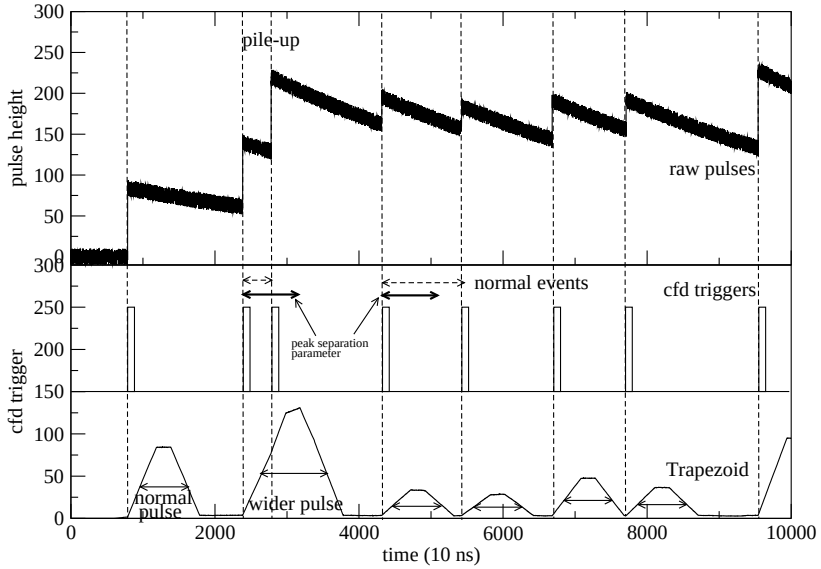

FIG. 5: The pile-up rejection procedure is shown. One can reject pulses which are wider than the shaping time. Alternately one can set a user defined parameter so that pulses separated in time less than this parameter are rejected.