User's Manual

Digital Gamma Finder (DGF)

DGF-4C

Version 3.08, March 2007

XIA LLC

31057 Genstar Road
Hayward, CA 94544 USA

Phone: (510) 401-5760; Fax: (510) 401-5761
http://www.xia.com

Disclaimer

1 Overview

The Digital Gamma Finder (DGF) family of digital pulse processors features unique capabilities for measuring both the amplitude and shape of pulses in nuclear spectroscopy applications. The DGF architecture was originally developed for use with arrays of multi-segmented HPGe gamma ray detectors, but has since been applied to an ever broadening range of applications.

The DGF-4C is a 4-channel all-digital waveform acquisition and spectrometer card. It combines spectroscopy with waveform digitizing and on-line pulse shape analysis. The DGF-4C accepts signals from virtually any radiation detector. Incoming signals are digitized by 14-bit 40 MSPS ADCs. Waveforms of up to 100 μs in length for each event can be stored in a FIFO. The waveforms are available for onboard pulse shape analysis, which can be customized by adding user functions to the core processing software. Waveforms, timestamps, and the results of the pulse shape analysis can be read out by the host system for further off-line processing. The DGF-4C can process up to 200,000 counts per second (combined for all 4 channels) into spectra with up to 32K channels. It supports coincidence spectroscopy and can recognize complex hit patterns. Front panel I/O connections allow external vetoing and provide trigger and multiplicity information. Several DGF-4C modules can be combined into groups with distributed timing and trigger signals.

1.1 Features

- Designed for high precision γ-ray spectroscopy with HPGe detectors.
- Directly compatible with scintillator/PMT combinations: NaI, CsI, BGO, and many others.
- Simultaneous amplitude measurement and pulse shape analysis for each channel.
- Wide range of filter rise times: from 50 ns to 45 μs, equivalent to 22 ns to 20 μs shaping times.
- Programmable gain and input offset.
- Excellent pileup inspection: double pulse resolution of 100 ns. Programmable pileup inspection criteria include trigger filter parameters, threshold, and rejection criteria.
- Digital oscilloscope and FFT for health-of-system analysis.
- Triggered synchronous waveform acquisition across channels, modules and crates.
- Dead times as low as 1 μs per event are achievable (limited by DSP algorithm complexity). Events at even shorter time intervals can be extracted via off-line ADC waveform analysis.
- Digital constant fraction algorithm measures event arrival times down to a few ns accuracy.
- External global first level trigger ("GFLT" or gate) and delayed global second level trigger ("GSLT" or validation) facilitate complex data acquisition system construction.

- Analog type multiplicity input and output can be daisy chained across multiple DGF-4C modules.
- Supports 16 bit Level-1 FAST CAMAC transfers (5 Mbytes/second).

1.2 Specifications

- ***Inputs (Analog)***
  Signal Input: 4 inputs. Selectable input impedance: 50Ω, 250Ω and 1kΩ, ±5V pulsed, ±2V DC. Selectable input attenuation 1:21, 1:5 and 1:1.

- ***Inputs (Digital)***
  Clock Input/Output: Daisy-chained 40MHz clock on auxiliary bus.
  Triggers: Two wired-or trigger buses on auxiliary bus. One for synchronous waveform acquisition, one for event triggers.
  Next-neighbor logic: One pair of next-neighbor signals for distributed next-neighbor logic, on auxiliary bus.
  Busy-Synch pair: NIM level output (Busy) and input (Synch). Used to synchronize timers and run start/stop to 50 ns precision.
  Multiplicity in/out: Analog multiplicity signal, 37 mV/hit; can be daisy-chained.
  GFLT: Global first level trigger/veto. Suppresses event triggering.
  GSLT: Global second-level triggering. Stores arrival time of GSLT signal.
  DSP Trigger: Signal to indicate DSP busy time.

- ***Interface***
  CAMAC: 16-bit Read/Write, fast CAMAC level-1, 5Mbyte/s.

- ***Digital Controls***
  Gain: 100:1 gain range in fine steps.
  Shaping: Digital trapezoidal filter. Rise time and flat top set independently: 50 ns – 45 μs in small steps.
- ***Data outputs***
  Spectrum: 1024-32768 channels, 24-bit deep (16,777,215 counts/bin).
  Other: Real time, live time, input and throughput counts. List mode data comprising of energies, timestamps, waveform data, pulse shape analysis results and ancillary data like hit patterns.

2 Setting up

2.1 Scope of document

The scope of this document is DGF-4C modules with serial numbers:
D1100 through D1199,
E1200 through E1399.

2.2 Installation

Put the CAMAC controller in the right most slot of your CAMAC crate. Currently, the DGF-4C software, DGF-4C Viewer, supports both the Jorway 73A SCSI controller and the Wiener CC32 PCI controller. Place the DGF-4C modules into any free slots. Connect your host computer to the CAMAC controller, and switch the CAMAC crate on first. Then power up, or reboot, your PC.

Make sure the SCSI adapter (for the Jorway J73A) or the PCI card (for the CC32) is installed properly, without hardware conflicts. If using the J73A, Windows operating system may detect it as a new device and try to find and install a driver. Do not install a driver – it is provided by the DGF-4C Viewer software as explained below.

The DGF-4C Viewer, XIA's graphical user interface to set up and run the DGF-4C modules, is based on WaveMetrics' IGOR Pro. To run the DGF-4C Viewer, you have to have IGOR Version 4.0 or higher installed on your computer. By default, the IGOR Pro should be installed at C:\Program Files\WaveMetrics\IGOR Pro Folder.

The CD-ROM with the DGF-4C software distribution contains

1) an installation program Setup.exe,
2) the DGF-4C software in the folder XIA\DGF4C and its subfolders.

The DGF-4C software can be installed by running the installation program. Follow its instructions shown on the screen to install the software to the default folder selected by the installation program, or a custom folder. This folder will contain the IGOR control program (DGF4C.pxp), online help files and 7 subfolders including Configuration, DOC, Drivers, DSP, Firmware, MCA, and PulseShape. Make sure you keep this folder organization intact, as the IGOR program and future updates rely on this. Feel free, however, to add folders and subfolders at your convenience.

2.3 Getting Started

To start the DGF-4C Viewer, double-click on the DGF4C.pxp file in the installed folder. If IGOR starts up without error messages, the DGF-4C Start Up Panel should be prominently displayed in the middle of the desktop.

In the panel, first select the number of DGF-4C modules in the system. Then specify the CAMAC slot number in which each module resides. Keep the FIPPI file name for each module intact in the moment. Detailed discussions about how to select appropriate FIPPI files will be given later.

Then, select your CAMAC controller type. Choose “Offline” if you want to try the software without a crate attached. For the Jorway 73A, you have to select the proper SCSI bus number and Crate ID. The SCSI ID usually is either 0 or 1 and may vary between 0 and 7. If it is unknown, set it to 0. After the system is boot up, it will return the correct SCSI bus number and automatically correct it on the DGF-4C Start Up Panel. For CC32 controllers, you only need to select the Crate ID. The Crate ID for both Jorway 73A and CC32 controllers should match the Crate number that you set on the controllers.

At this moment, you can ignore the advanced options which will be discussed later. Then you click on the “Start Up System” button to initialize the modules. This will download DSP code and FPGA configuration to the modules, as well as the module parameters. If you see messages similar to “Module 1 in slot 5 started up successfully!” in the IGOR history window, the DGF-4C modules have been initialized successfully. Otherwise, refer to the troubleshooting section for possible solutions.

After the system is initialized successfully, you will now see the main DGF-4C Control Panel from which all work is conducted. The tabs in the Control Panel are arranged in logical order from left to right. For most of the actions the DGF-4C Viewer interacts with one DGF-4C module at a time. The number of that module is displayed at the top right corner of the Control Panel (inside the "Module" control). Next to the “Module” control is the “Channel” control which indicates the current channel the DGF-4C viewer is interacting with. Proceed with the steps below to configure your system.

1. In the *Calibrate* tab, click on the Oscilloscope button. This opens a graph that shows the untriggered signal input. Click "Refresh" to update the display. The pulses should fall in the display range. If no pulses are visible or if they are cut off out of the display range, click "Adjust Offsets" to automatically set the DC offset. There is a control called “Baseline [%]” on the Oscilloscope which can be used to set the DC-offset level for each channel. If the pulse amplitude is too large to fall in the display range, decrease the "Gain" in the *Calibrate* tab of the DGF-4C Control Panel. If the pulses are negative, toggle the “Trigger positive” checkbox in the Channel CSRA Edit Panel which can be accessed by clicking on the Edit button next to “Chan. CSRA” on the *Settings* tab.

2. In the *Calibrate* tab, enter an estimated preamplifier exponential RC decay time for Tau then click on "Auto Find" to determine the actual Tau value for the current channel of the current module. Repeat this for other channels if necessary. The Tau finder works best for a Tau value from 20 μs to 200 μs. You can also enter a known good Tau value directly in the control.

3. Save the modified parameter settings to a file. To do so, click on the “Save” button on the *Settings* tab to open a save file dialog. Create a new file name to avoid overwriting the default settings file.

4. Click on the *Run* tab, set "Run Type" to 0x301 MCA Mode, “Polling time” to 1 second, and “Run time/time out” to 30 seconds or so, then click on the "Start Run" button. A spinning wheel will appear in the lower left corner of the screen as long as the system is waiting for the run to finish. After the run is complete, select the *Analyze* tab and click on the “MCA Spectrum” button. The MCA spectrum shows the MCA histograms for all four channels. You can deselect other channels while working on only one channel. You can do a Gauss fit on a peak by entering values in the "Min" and "Max" fields as the limits for a Gauss fit. You can also use the mouse to drag the Cursor A and B in the MCA spectrum to the limits of the fit. Click the popup menu "Fit" to perform the fit on a single channel or all four channels if their peaks are all within the fit limits. Enter the true energy value in the "Peak" field to calibrate the energy scale. But note: this energy calibration only rescales the MCA spectrum without changing the gain or other settings in the DGF module; and after a new data acquisition run, the spectrum will change back to its original scaling which only depends on the parameter “Binning Factor” on the *Calibrate* tab.

At this stage, you may not be able to get a spectrum with good energy resolutions. You may need to adjust some settings such as energy filter rise time and flat top etc. as described in Section 4.3.

3 Navigating the DGF-4C Viewer

3.1 Overview

The DGF-4C Viewer consists of a number of graphs and control panels, linked together by the main "DGF-4C Run Control" panel. The "DGF-4C Run Control" panel is divided into 4 tabs, corresponding to the 4 topics summarized below. The *Settings* tab contains controls used to initialize the module, and the file and directory settings. The *Calibrate* tab contains controls to adjust parameters such as gain, DC-offset, preamplifier decay time, histogram control. The *Run* tab is used to start and stop runs, and in the A*nalyze* tab are controls to analyze, save and read spectra or event traces. Below we describe the concepts and principles of using the DGF-4C Viewer. Detailed information on the individual controls can be found in the online help for each panel.

3.2 Settings

The DGF-4C Viewer comes up in exactly the same state as it was when last saved to file using *File->Save Experiment*. However, the DGF-4C module itself loses all programming when it is switched off. When the DGF-4C is switched on again, all programmable components need code and configuration files to be downloaded to the module. Clicking the *Start System* button in the main "DGF-4C Run Control" panel performs this download.

The DGF-4C being a digital system, all parameter settings are stored in a settings file. This file is separate from the main IGOR experiment file, to allow saving and restoring different settings for different detectors and applications. Parameter files are saved and loaded with the corresponding buttons in the *Settings* tab. After loading a settings file, the settings are automatically downloaded to the module. At module initialization, the settings are automatically read and applied to the DGF-4C from the current settings file.

Internally, the module parameters are handled as binary numbers and bitmasks. The *Settings* tab gives access to user parameters in meaningful physical units. Values entered by the user are converted by the DGF-4C Viewer to the closest value in internal units. Refer to the Online Help for detailed descriptions of these parameters.

3.3 Calibrate

The *Calibrate* tab is used to calibrate or diagnose the system. You can adjust the Gain and DC-Offset on the channel by channel basis. You can use the automatic Tau-Finder routine to find the "Decay Time" of the preamplifier. You can also control the histogram by setting the cut-off energy and binning factor.

The *Calibrate* tab also has an *Oscilloscope* button linking to a diagnostic graph. The *Oscilloscope* shows a graph of ADC samples which are untriggered pulses from the signal input. The time intervals between the samples can be adjusted; for intervals greater than 0.275 μs the samples will be averaged over the interval. The main purpose of the

Oscilloscope is to make sure that the signal is in range in terms of gain and DC-offset. The *Oscilloscope* is also useful to estimate the noise in the system. Clicking on the *FFT Display* button opens the "ADC Trace FFT", where the noise spectrum can be investigated as a function of frequency. This works best if the *Oscilloscope* trace contains no pulses, i.e. with the detector attached but no radioactive sources present.

3.4 Run

The *Run* tab is used to start and stop runs. Before you start a run, you need to select the run type, polling time (the time interval for polling the run status), run time for MCA runs, and time out limit and the number of spills (repeated runs) for list mode runs.

In a multi-module system, you can set all modules to start and stop simultaneously and to reset the timers in all modules with the start of the next data acquisition run by selecting the two options in the *Synchronization* group.

You can choose a base name and a run number in order to form an output file name. The run data will be written to a file whose name is composed of both. The run number is automatically incremented at the end of each run if you select "Auto update run number" on the *Record* panel, but you can set it manually as well. Data are stored in files in either the MCA folder if the run is a MCA run or the PulseShape folder if the run is a List Mode run. These files have the same name as the output file name but different extension. For list mode runs, buffer data are stored in a file with name extension ".bin". For both list mode runs and MCA runs, MCA spectrum data are stored in a file with name extension ".mca" if you select "Auto store spectrum data" on the *Record* panel, and module settings are stored in a file with name extension ".set" after each run if you select "Auto store settings" on the *Record* panel. In the end of a list mode run, the output data file (.bin) will be automatically parsed and event data such as energy, trigger time, etc. will be written to a text file (.dat) in the PulseShape folder for a quick review of the list mode data.

3.5 Analyze

The *Analyze* tab is used to investigate the spectrum or to view list mode traces. It also shows the run statistics such as run time, event rate, and live time and input count rate for each channel. You can perform Gauss fits on peaks to find the resolution, and calibrate the energy spectrum by entering a known energy value for a fitted peak. You can also view individual event trace and its energy from a standard list mode run.

4 Data Runs and Data Structures

4.1 Run types

DGF-4C can be configured to perform different types of data runs depending on the intended applications. If only MCA spectra are required for monitoring purposes such as checking energy resolutions of radiation detectors, MCA runs (RUNTASK = 0x301) can be used. MCA spectrum (up to 32K bins), live time and input count rate are available for each channel using this run type. On the other hand, if individual event information is necessary for applications like pulse shape analysis, list mode runs (RUNTASK = 0x100, 0x101, 0x102, or 0x103) are good choices. Multi-parametric or list mode data are collected during these runs on an event-by-event basis, including energies, time stamps, pulse shape analysis values, and waveforms. Fast list mode runs (RUNTASK = 0x100, 0x101, 0x102, and 0x103) are very similar to the standard list mode runs except without waveform acquisition and certain data quality control measures (e.g. coincidence pattern check) in order to make the runs faster. As a result of these exceptions, there are limitations for using the fast list mode runs.

4.1.1 MCA Runs

MCA mode puts all modules into a typical spectrum-only acquisition mode in which there are no list-mode data required. The event data is not stored in DSP's circular Level-1 buffer, but only used to calculate the energy for incrementing the spectrum. Runs end after the time specified in the "RunTime/TimeOut" control counts down to zero. The "Maximum no. of Events" control is set to zero for MCA runs since it is not used to end the run.

4.1.2 List Mode Runs

List mode is the general data acquisition run. Waveforms, energies and time stamps are collected on an event-by-event basis. Since available memory limits the number of events that each module can store in its buffer, the DGF-4C Viewer computes the maximum number of events for each list mode run type. When the maximum is reached, the run is stopped and the buffer is read out. For a longer run in list-mode, you can request several spills, or buffer fills. At the start of the first run all previous run history, e.g. MCA memory or run statistics, is cleared. All other sub-runs are started with a Resume Run command, which leaves previous run information intact. The output data for list mode runs can be stored in standard or compressed format as described in Section 4.2. In standard list mode runs (RUNTASK = 0x100), the DSP's linear I/O buffer is selected as the intermediate buffer, and energies and time stamps are reported in the standard format. In other list mode runs (RUNTASK = 0x101, 0x102, 0x103), the circular Level-1 buffer is selected as the intermediate buffer, and energies and time stamps are reported in the compressed data format. Even though traces are not reported, they are still read out for optional pulse shape analysis such as the computation of a constant fraction time. Since the readout takes time, set the trace length to zero if you do not need pulse shape analysis for these runtypes. Also, since the circular buffer is only 2k words long, make sure that the maximum combined tracelength from all four channels is less than 50 microseconds.

4.1.3 Fast List Mode Runs

Fast list mode is an event-by-event data acquisition run without waveforms. The trace length for all channels should be set to 0 to obtain the best performance. Also for each channel, bit 10 (Compute constant fraction timing) of ChanCSRA should be cleared, which can be modified by clicking the "Edit" button next to the Chan. CSRA control on the *Settings* tab. Since no traces are read out, the data acquisition is faster than a regular list mode; however, no pulse shape analysis (PSA) values are available. There will also be no check if the circular Level-1 buffer is filled faster than the event processing rate. So keep the average trigger rate well below the processing rate. Otherwise, the data from the remainder of the run will be corrupted. The output can again be stored in standard or compressed format as described in Section 4.2.

Table 4.1: Summary of run types and data formats.

	Intermediate buffer	Output data	DSP Variables
List Mode (standard)	Linear buffer	Energies, time stamps, 5 PSA values, and waveforms in linear buffer. Spectra in MCA block	RUNTASK = 256 MAXEVENTS = <calculate> (CHANHEADLEN = 9)
List Mode - Compression 1	Circular buffer	Energies, time stamps, and 5 PSA values in linear buffer. Spectra in MCA block	RUNTASK = 257 MAXEVENTS = <calculate> (CHANHEADLEN = 9)
List Mode - Compression 2	Circular buffer	Energies, time stamps, and 2 PSA values in linear buffer. Spectra in MCA block	RUNTASK = 258 MAXEVENTS = <calculate> (CHANHEADLEN = 4)
List Mode - Compression 3	Circular buffer	Energies and time stamps in linear buffer. Spectra in MCA block	RUNTASK = 259 MAXEVENTS = <calculate> (CHANHEADLEN = 2)
Fast List Mode - (standard)	Linear buffer	Energies, time stamps, and 5 PSA values in linear buffer. Spectra in MCA block	RUNTASK = 512 MAXEVENTS = <calculate> (CHANHEADLEN = 9)
Fast List Mode - Compression 1	Circular buffer	Energies, time stamps, and 5 PSA values in linear buffer. Spectra in MCA block	RUNTASK = 513 MAXEVENTS = <calculate> (CHANHEADLEN = 9)
Fast List Mode - Compression 2	Circular buffer	Energies, time stamps, and 2 PSA values in linear buffer. Spectra in MCA block	RUNTASK = 514 MAXEVENTS = <calculate> (CHANHEADLEN = 4)

Fast List Mode - Compression 3	Circular buffer	Energies and time stamps in linear buffer. Spectra in MCA block	RUNTASK = 515 MAXEVENTS = <calculate> (CHANHEADLEN = 2)
MCA Mode	Circular buffer	Spectra in MCA block	RUNTASK = 769 MAXEVENTS=0

4.2 Output data structures

Output data are available in two different memory blocks. The multichannel analyser (MCA) block resides in memory external to the DSP. The linear I/O buffer is located in data memory and consists of 8192 16-bit words.

4.2.1 MCA histogram data

The MCA block is fixed to 32K words (24-bit deep) per channel, residing in the external memory. The memory is organized into 8 pages of 4K words. To read out the spectra to the host, each page has first to be transferred from the external memory to the linear I/O buffer in the DSP memory, and then read out by a DMA transfer.

4.2.2 List mode data

The list mode data in the linear I/O buffer can be written in a number of formats. User code should access the three variables BUFHEADLEN, EVENTHEADLEN, and CHANHEADLEN in the configuration file of a particular run to navigate through the data set.

The data organization of the linear I/O buffer is as follows. The buffer content always starts with a buffer header of length BUFHEADLEN. Currently, BUFHEADLEN is six, and the six words are:

Table 4.2: Buffer header data format.

Word #	Variable	Description
0	BUF_NDATA	Number of words in this buffer
1	BUF_MODNUM	Module number
2	BUF_FORMAT	Format descriptor = RunTask + 0x1000
3	BUF_TIMEHI	Run start time, high word
4	BUF_TIMEMI	Run start time, middle word
5	BUF_TIMELO	Run start time, low word

Following the buffer header, the events are stored in sequential order. Each event starts out with an event header of length EVENTHEADLEN. Currently, EVENTHEADLEN=3, and the three words are:

Table 4.3: Event header data format.

Word #	Variable	Description
0	EVT_PATTERN	Hit pattern
1	EVT_TIMEHI	Event time, high word
2	EVT_TIMELO	Event time, low word

The hit pattern is a bit mask, which tells which channels were read out. The LSB (bit 0), if set, indicates that channel 0 has been read. Bit number *n*, if set, indicates that channel *n* has been read. The number of bits set, 0…3, tells for how many channels data have been recorded following the event header. After the event header there follow the channel information as indicated by the hit pattern, and in order of increasing channel numbers.

The data for each channel are organized into a channel header of length CHANHEADLEN, which may be followed by waveform data. CHANHEADLEN depends on the run type and on the method of data buffering, i.e. if raw data is directed to the intermediate Level-1 buffer or directly to the I/O buffer. Offline analysis programs should therefore check the value of RUNTASK, which are reported in the run settings file. All currently supported data formats are defined below.

1. For List Mode or Fast List Mode, either standard or compression 1, (RUNTASK = 256, 257, 512 or 513), CHANHEADLEN=9, and the nine words are

Table 4.4: Channel header, possibly followed by waveform data.

Word #	Variable	Description
0	CHAN_NDATA	Number of words for this channel
1	CHAN_TRIGTIME	Fast trigger time
2	CHAN_ENERGY	Energy
3	CHAN_XIAPSA	XIA PSA value
4	CHAN_USERPSA	User PSA value
5	CHAN_GSLTHI	GSLT time stamp, high word
6	CHAN_GSLTMI	GSLT time stamp, middle word
7	CHAN_GSLTLO	GSLT time stamp, low word
8	CHAN_REALTIMEHI	High word of the real time

Any waveform data for this channel would then follow this header. An offline analysis program can recognize this by computing N_WAVE_DATA=CHAN_DATA-9. If N_WAVE_DATA is greater than zero, it indicates the number of waveform data words to follow.

In the current software version, the XIA PSA value contains the result of the constant fraction trigger time computation (CFD). The format is as follows: the upper 8-bit of the word point to the ADC sample before the CFD, counted from the beginning of the trace. The lower 8 bits give the fraction of an ADC sample time between the sample and the CFD time. For example, if the value is 0x0509, the CFD time is 5 + 9/256 ADC sample steps away from the beginning of the recorded trace.

2. For compression 2 List Mode or Fast List Mode (RUNTASK = 258 or 514), CHANHEADLEN=4, and the four words are:

Table 4.5: Channel header for compression 2 format.

Word #	Variable	Description
0	CHAN_TRIGTIME	Fast trigger time
1	CHAN_ENERGY	Energy
2	CHAN_XIAPSA	XIA PSA value
3	CHAN_USERPSA	User PSA value

3. For compressed List Mode or Fast List Mode (RUNTASK = 259 or 515), CHANHEADLEN=2, and the two words are:

Table 4.6: Channel header for compression 3 format.

Word #	Variable	Description
0	CHAN_TRIGTIME	Fast trigger time
1	CHAN_ENERGY	Energy

4.3 Optimizing Parameters

Optimization of the DGF-4C's run parameters for best resolution depends on the individual systems and usually requires some degree of experimentation. The DGF-4C Viewer includes several diagnostic tools and settings options to assist the user, as described below.

4.3.1 Noise

For a quick analysis of the electronic noise in the system, you can view a Fourier transform of the incoming signal by selecting *Oscilloscope* → *FFT Display* in the *Calibrate* tab. The graph shows the FFT of the untriggered input sigal of the *Oscilloscope*. By adjusting the "dT" control in the *Oscilloscope* and clicking the *Refresh* button, you can investigate different frequency ranges. For best results, remove any source from the detector and only regard traces without actual events. If you find sharp lines in the 10 kHz to 1 MHz region you may need to find the cause for this and remove it. If you click on the "Apply Filter" button, you can see the effect of the energy filter simulated on the noise spectrum.

4.3.2 Energy Filter Parameters

The main parameter to optimize energy resolution is the rise time of the energy filter. Generally, longer rise times result in better resolution, but reduce the throughput. Optimization should begin with scanning the rise time through the available range. Try 2μs,

4μs, 8μs, 11.2μs, take runs of 60s or so and note changes in energy resolution. Then fine tune the rise time.

The flat top usually needs only small adjustments. For a typical coaxial Ge-detector we suggest to use a flat top of 1.2μs. For a small detector (20% efficiency) a flat top of 0.8μs is a good choice. For larger detectors flat tops of 1.2μs and 1.6μs will be more appropriate.

In general the flat top needs to be wide enough to accommodate the longest typical signal rise time from the detector. It then needs to be wider by one filter clock cycle than that minimum, but at least 3 clock cycles. Note that the filter clock cycle ranges from 0.05 to 1.6μs, depending on the filter time range, so that it is not possible to have a very short flat top together with a very long filter rise time.

XIA's DGF-4C Viewer software (Version 3.04 or higher) provides a tool which automatically scans all possible combinations of energy filter rise time and flat top and finds the combination that gives the best energy resolution. This tool can be accessed by clicking the *Optimize* button on the Settings tab. Please refer to the DGF-4C Online Help documentation for more details.

4.3.3 Threshold and Trigger Filter Parameters

In general, the trigger threshold should be set as low as possible for best resolution. If too low, the input count rate will go up dramatically and "noise peaks" will appear at the minimum and maximum edge of the spectrum. If the threshold is too high, especially at high count rates, low energy events below the threshold can pass the pile-up inspector and pile up with larger events. This increases the measured energy and thus leads to exponential tails on the ideally Gaussian peaks in the spectrum. Ideally, the threshold should be set such that the noise peaks just disappear.

The settings of the trigger filter have only minor effect on the resolution. However, changing the trigger conditions might have some effect on certain undesirable peak shapes. A longer trigger rise time allows the threshold to be lowered more, since the noise is averaged over longer periods. This can help to remove tails on the peaks. A long trigger flat top will help to trigger on slow rising pulses and thus result in a sharper cut off at the threshold in the spectrum.

4.3.4 Decay time

The preamplifier decay time τ is used to correct the energy of a pulse sitting on the falling slope of a previous pulse. The calculations assume a simple exponential decay with one decay constant. A precise value of τ is especially important at high count rates where pulses overlap more frequently. If τ is off the optimum, peaks in the spectrum will broaden, and if τ is very wrong, the spectrum will be significantly blurred.

XIA's DGF-4C Viewer software (Version 3.04 or higher) provides several tools which would help find or fine tune the decay time. The first and usually sufficiently precise estimate

of τ can be obtained by clicking the *Auto Find* button in the *Calibrate* tab. The *Auto Find* routine tries to measure the decay time 10 times and report the average τ value and its standard deviation (Sigma). DGF-4C users can also use the *Manual Fit* routine to manually find the decay time through exponentially fitting the untriggered input signals. The last tool which can be used to find the decay time is the *Optimize* routine. Similar to the routine for finding the optimal energy filter times, this routine can be used to automatically scan a range of decay times and find the optimal one. Please refer to the DGF-4C Online Help documentation for more details.

4.4 Settings File

Starting from DGF-4C software version 3.0, the format of settings file (*.itx) has been changed from IGOR text format to binary file format. The new settings file consists of settings for 23 modules (the maximum number of DGF modules that can be installed in a 24-slot CAMAC crate is 24; however, the CAMAC controller holds one slot, so the maximum number of usable slots is 23). The settings for each module are stored sequentially in this file, from module #1 to #23, 416 unsigned 16-bit integers for each module, resulting in a file with exactly 19,136 bytes.

5 Hardware Description

The DGF-4C is a 4-channel unit designed for gamma-ray spectroscopy and waveform capturing. It incorporates four functionally building blocks, which we describe below. This section concentrates on the functionality aspect. Technical specification can be found in Section 1.2.

5.1 Analog signal conditioning

Each analog input has its own signal conditioning unit. The task of this circuitry is to adapt the incoming signals to the input voltage range of the ADC, which spans 1.00 V. Input signals are adjusted for offsets, and there is a computer-controlled gain stage. This helps to bring the signals into the ADC's voltage range and set the dynamic range of the channel.

The ADC is not a peak sensing ADC, but acts as a waveform digitizer. In order to avoid aliasing, we remove the high frequency components from the incoming signal prior to feeding it into the ADC. The anti-aliasing filter, a 4-th order passive Gaussian filter, cuts off sharply at the Nyquist frequency, namely half the ADC sampling frequency.

Though the DGF-4C can work with many different signal forms, best performance is to be expected when sending the output from a charge integrating preamplifier directly to the DGF-4C without any further shaping.

5.2 Real-time processing units

The real time processing units, one per channel, consist of a field programmable gate array (FPGA) and a FIFO memory. The data stream from the ADCs is sent to these units at the full ADC sampling rate. Using a pipelined architecture, the signals are also processed at this high rate, without the help of the on-board digital signal processor (DSP).

The real-time processing units (RTPUs) apply digital filtering to perform essentially the same action as a shaping amplifier. The important difference is in the type of filter used. In a digital application it is easy to implement finite impulse response filters, and we use a trapezoidal filter. The flat top will typically cover the rise time of the incoming signal and makes the pulse height measurement less sensitive to variations of the signal shape.

Secondly, the RTPUs contain a pileup inspector. This logic ensures that if a second pulse is detected too soon after the first, so that it would corrupt the first pulse height measurement, both pulses are rejected as piled up. The pileup inspector is, however, not very effective in detecting pulse pileup on the rising edge of the first pulse, i.e. in general pulses must be separated by their rise time to be effectively recognized as different pulses. Therefore, for high count rate applications, the pulse rise times should be as short as possible, to minimize the occurrence of pileup peaks in the resulting spectra.

If a pulse was detected and passed the pileup inspector, a trigger may be issued. That trigger would notify the DSP that there are raw data available now. If a trigger was issued the data remain latched until the RTPU has been serviced by the DSP.

The third component of the RTPU is a FIFO memory, which is controlled by the pile up inspector logic. The FIFO memory is continuously being filled with waveform data from the ADC. On a trigger it is stopped, and the read pointer is positioned such that it points to the beginning of the pulse that caused the trigger. When the DSP collects event data, it can read any fraction of the stored waveform, up to the full length of the FIFO.

5.3 Digital signal processor (DSP)

The DSP controls the operation of the DGF-4C, reads raw data from the RTPUs, reconstructs true pulse heights, applies time stamps, and prepares data for output to the host computer, and increments spectra in the on-board memory.

The host computer communicates with the board, via the CAMAC interface, using a direct memory access (DMA) channel. Reading and writing data to DSP memory does not interrupt its operation, and can occur even while a measurement is underway.

The host sets variables in the DSP memory and then calls DSP functions to program the hardware. Through this mechanism all gain and offset DACs are set and the RTPUs are programmed.

The RTPUs process their data without support from the DSP, once they have been set up. When any one or more of them generate a trigger, an interrupt request is sent to the DSP. It responds with reading the required data from the RTPUs and storing those in memory. It then returns from the interrupt routine without processing the data to minimize the DSP induced dead time. The event processing routine works from the data in memory to generate the requested output data.

In this scheme, the greatest processing power is located in the RTPUs. Implemented in FPGAs each of them processes the incoming waveforms from its associated ADC in real time and produces, for each valid event, a small set of distilled data from which pulse heights and arrival times can be reconstructed. The computational load for the DSP is much reduced, as it has to react only on an event-by-event basis and has to work with only a small set of numbers for each event.

5.4 CAMAC interface

The CAMAC interface through which the host communicates with the DGF-4C is implemented in its own FPGA. The configuration of this gate array is stored in a PROM, which is placed in the only DIP-8 IC-socket on the DGF-4C board.

The interface conforms to the regular CAMAC standard, as well as the newer Level-1 fast CAMAC with a cycle time of 400 ns per read operation. The interface moves 16-bit data words at a time. The upper 8 bits of the read and write bus are ignored.

6 Theory of Operation

6.1 Digital Filters for γ-ray detectors

Energy dispersive detectors, which include such solid state detectors as Si(Li), HPGe, HgI_2, CdTe and CZT detectors, are generally operated with charge sensitive preamplifiers as shown in Figure 6.1 a). Here the detector D is biased by voltage source V and connected to the input of preamplifier A which has feedback capacitor C_f and feedback resistor R_f.

The output of the preamplifier following the absorption of an γ-ray of energy E_x in detector D is shown in Figure 6.1 b) as a step of amplitude V_x (on a longer time scale, the step will decay exponentially back to the baseline, see Section 6.3). When the γ-ray is absorbed in the detector material it releases an electric charge $Q_x = E_x/\varepsilon$, where ε is a material constant. Q_x is integrated onto C_f, to produce the voltage $V_x = Q_x/C_f = E_x/(\varepsilon C_f)$. Measuring the energy E_x of the γ-ray therefore requires a measurement of the voltage step V_x in the presence of the amplifier noise σ, as indicated in Figure 6.1 b).

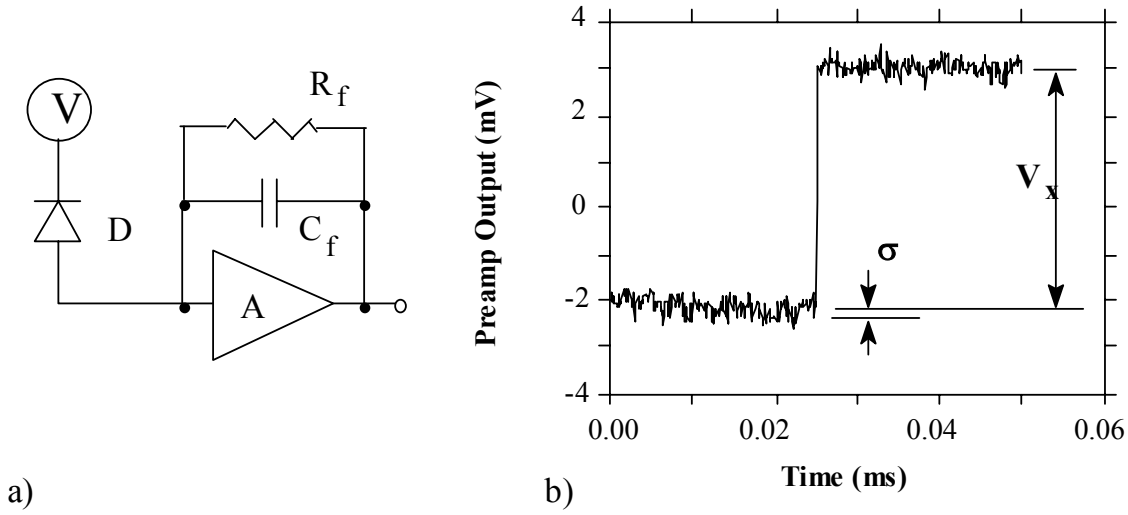


Figure 6.1: a) Charge sensitive preamplifier with RC feedback; b) Output on absorption of an γ-ray.

Reducing noise in an electrical measurement is accomplished by filtering. Traditional analog filters use combinations of a differentiation stage and multiple integration stages to convert the preamp output steps, such as shown in Figure 6.1 b), into either triangular or semi-Gaussian pulses whose amplitudes (with respect to their baselines) are then proportional to V_x and thus to the γ-ray's energy.

Digital filtering proceeds from a slightly different perspective. Here the signal has been digitized and is no longer continuous. Instead it is a string of discrete values as shown in

Figure 6.2. Figure 6.2 is actually just a subset of Figure 6.1 b), in which the signal was digitized by a Tektronix 544 TDS digital oscilloscope at 10 MSA (megasamples/sec). Given this data set, and some kind of arithmetic processor, the obvious approach to determining V_x is to take some sort of average over the points before the step and subtract it from the value of the average over the points after the step. That is, as shown in Figure 6.2, averages are computed over the two regions marked "Length" (the "Gap" region is omitted because the signal is changing rapidly here), and their difference taken as a measure of V_x. Thus the value V_x may be found from the equation:

$$V_{x,k} = -\sum_{i(before)} W_i V_i + \sum_{i(after)} W_i V_i \tag{6.1}$$

where the values of the weighting constants W_i determine the type of average being computed. The sums of the values of the two sets of weights must be individually normalized.

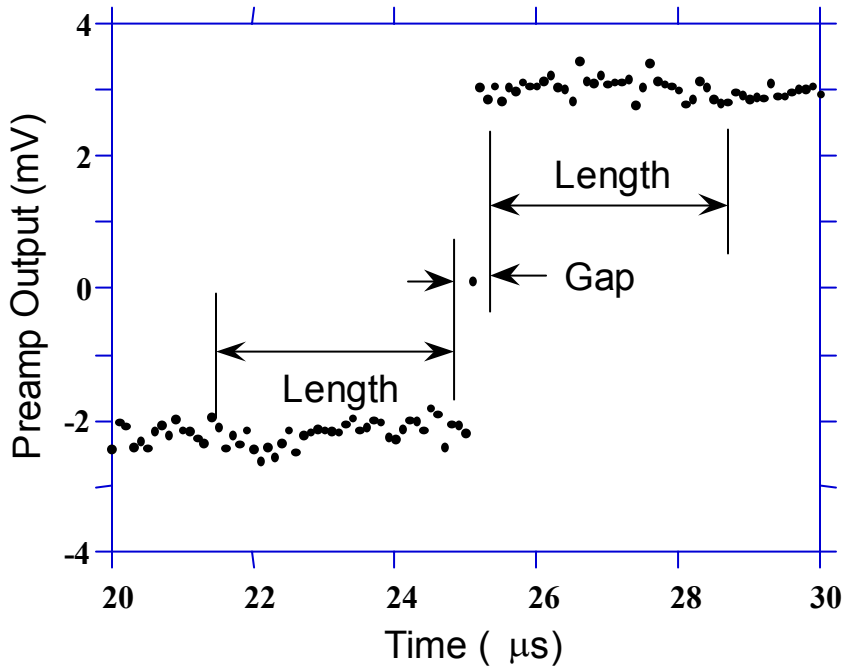


Figure 6.2: Digitized version of the data of Figure 6.1 b) in the step region.

The primary differences between different digital signal processors lie in two areas: what set of weights $\{W_i\}$ is used and how the regions are selected for the computation of Eqn. 6.1. Thus, for example, when larger weighting values are used for the region close to the step while smaller values are used for the data away from the step, Eqn. 6.1 produces "cusp-like" filters. When the weighting values are constant, one obtains triangular (if the gap is zero) or trapezoidal filters. The concept behind cusp-like filters is that, since the points nearest the step carry the most information about its height, they should be most strongly weighted in the averaging process. How one chooses the filter lengths results in time variant (the lengths vary

from pulse to pulse) or time invariant (the lengths are the same for all pulses) filters. Traditional analog filters are time invariant. The concept behind time variant filters is that, since the γ-rays arrive randomly and the lengths between them vary accordingly, one can make maximum use of the available information by setting the length to the interpulse spacing.

In principal, the very best filtering is accomplished by using cusp-like weights and time variant filter length selection. There are serious costs associated with this approach however, both in terms of computational power required to evaluate the sums in real time and in the complexity of the electronics required to generate (usually from stored coefficients) normalized $\{W_i\}$ sets on a pulse by pulse basis.

The DGF-4C takes a different approach because it was optimized for very high speed operation. It implements a fixed length filter with all W_i values equal to unity and in fact computes this sum afresh for each new signal value k. Thus the equation implemented is:

$$LV_{x,k} = -\sum_{i=k-2L-G+1}^{k-L-G} V_i + \sum_{i=k-L+1}^{k} V_i \tag{6.2}$$

where the filter length is L and the gap is G. The factor L multiplying $V_{x,k}$ arises because the sum of the weights here is not normalized. Accommodating this factor is trivial.

While this relationship is very simple, it is still very effective. In the first place, this is the digital equivalent of triangular (or trapezoidal if $G \neq 0$) filtering which is the analog industry's standard for high rate processing. In the second place, one can show theoretically that if the noise in the signal is white (i.e. Gaussian distributed) above and below the step, which is typically the case for the short shaping times used for high signal rate processing, then the average in Eqn. 6.2 actually gives the best estimate of V_x in the least squares sense. This, of course, is why triangular filtering has been preferred at high rates. Triangular filtering with time variant filter lengths can, in principle, achieve both somewhat superior resolution and higher throughputs but comes at the cost of a significantly more complex circuit and a rate dependent resolution, which is unacceptable for many types of precise analysis. In practice, XIA's design has been found to duplicate the energy resolution of the best analog shapers while approximately doubling their throughput, providing experimental confirmation of the validity of the approach.

6.2 Trapezoidal Filtering in the DGF-4C

From this point onward, we will only consider trapezoidal filtering as it is implemented in the DGF-4C according to Eqn. 6.2. The result of applying such a filter with Length L=1μs and Gap G=0.4μs to a γ-ray event is shown in Figure 6.3. The filter output is clearly trapezoidal in shape and has a rise time equal to L, a flat top equal to G, and a symmetrical fall time equal to L. The basewidth, which is a first-order measure of the filter's noise reduction properties, is thus 2L+G.

This raises several important points in comparing the noise performance of the DGF-4C to analog filtering amplifiers. First, semi-Gaussian filters are usually specified by a *shaping time*. Their rise time is typically twice this and their pulses are not symmetric so that the basewidth is about 5.6 times the shaping time or 2.8 times their rise time. Thus a semi-Gaussian filter typically has a slightly better energy resolution than a triangular filter of the same rise time because it has a longer filtering time. This is typically accommodated in amplifiers offering both triangular and semi-Gaussian filtering by stretching the triangular rise time a bit, so that the *true* triangular rise time is typically 1.2 times the selected semi-Gaussian rise time. This also leads to an apparent advantage for the analog system when its energy resolution is compared to a digital system with the same nominal rise time.

One important characteristic of a digitally shaped trapezoidal pulse is its extremely sharp termination on completion of the basewidth 2L+G. This may be compared to analog filtered pulses whose tails may persist up to 40% of the rise time, a phenomenon due to the finite bandwidth of the analog filter. As we shall see below, this sharp termination gives the digital filter a definite rate advantage in pileup free throughput.

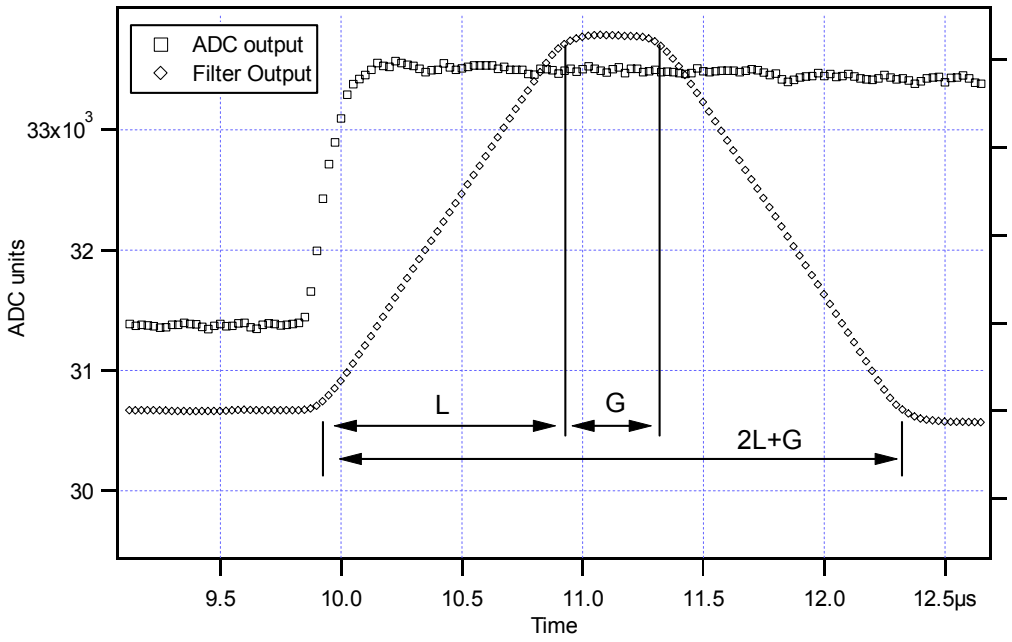


Figure 6.3: Trapezoidal filtering of a preamplifier step with L=1µs and G=0.4µs.

6.3 Baselines and preamplifier decay times

Figure 6.4 shows an event over a longer time interval and how the filter treats the preamplifier noise in regions when no γ-ray pulses are present. As may be seen the effect of the filter is to reduce both the amplitude of the fluctuations and their high frequency content. This signal is termed the *baseline* because it establishes the reference level from which the γ-

ray peak amplitude V_x is to be measured. The fluctuations in the baseline have a standard deviation σ_e which is referred to as the *electronic noise* of the system, a number which depends on the rise time of the filter used. Riding on top of this noise, the γ-ray peaks contribute an additional noise term, the *Fano noise*, which arises from statistical fluctuations in the amount of charge Q_x produced when the γ-ray is absorbed in the detector. This Fano noise σ_f adds in quadrature with the electronic noise, so that the total noise σ_t in measuring V_x is found from

$$\sigma_t = \text{sqrt}(\sigma_f^2 + \sigma_e^2) \tag{6.3}$$

The Fano noise is only a property of the detector material. The electronic noise, on the other hand, may have contributions from both the preamplifier and the amplifier. When the preamplifier and amplifier are both well designed and well matched, however, the amplifier's noise contribution should be essentially negligible. Achieving this in the mixed analog-digital environment of a digital pulse processor is a non-trivial task, however.

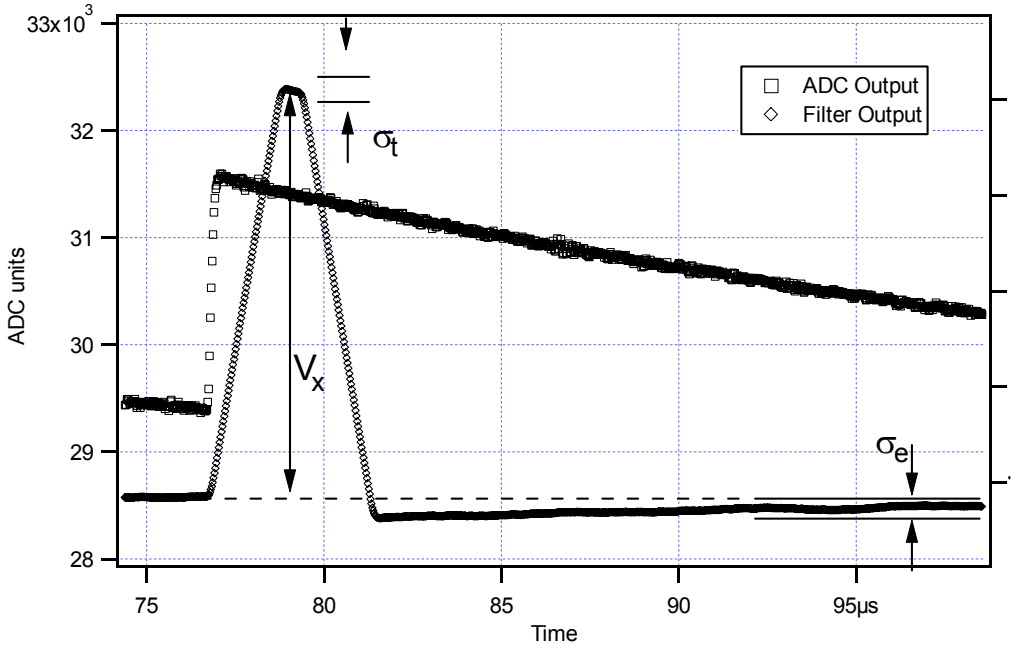


Figure 6.4: A γ-ray event displayed over a longer time period to show baseline noise and the effect of preamplifier decay time.

With a RC-type preamplifier, the slope of the preamplifier is rarely zero. Every step decays exponentially back to the DC level of the preamplifier. During such a decay, the baselines are obviously not zero. This can be seen in Figure 6.4, where the filter output during the exponential decay after the pulse is below the initial level. Note also that the flat top region is sloped downwards.

Using the decay constant τ, the baselines can be mapped back to the DC level. This allows precise determination of γ-ray energies, even if the pulse sits on the falling slope of a

previous pulse. The value of τ, being a characteristic of the preamplifier, has to be determined by the user and host software and downloaded to the module.

6.4 Thresholds and Pile-up Inspection

As noted above, we wish to capture a value of V_x for each γ-ray detected and use these values to construct a spectrum. This process is also significantly different between digital and analog systems. In the analog system the peak value must be "captured" into an analog storage device, usually a capacitor, and "held" until it is digitized. Then the digital value is used to update a memory location to build the desired spectrum. During this analog to digital conversion process the system is dead to other events, which can severely reduce system throughput. Even single channel analyzer systems introduce significant deadtime at this stage since they must wait some period (typically a few microseconds) to determine whether or not the window condition is satisfied.

Digital systems are much more efficient in this regard, since the values output by the filter are already digital values. All that is required is to take the filter sums, reconstruct the energy V_x, and add it to the spectrum. In the DGF-4C, the filter sums are continuously updated by the RTPU (see Section 5.2), and only have to be read out by the DSP when an event occurs. Reconstructing the energy and incrementing the spectrum is done by the DSP, so that the RTPU is ready to take new data immediately after the readout. This usually takes much less than one filter rise time, so that no system deadtime is produced by a "capture and store" operation. This is a significant source of the enhanced throughput found in digital systems.

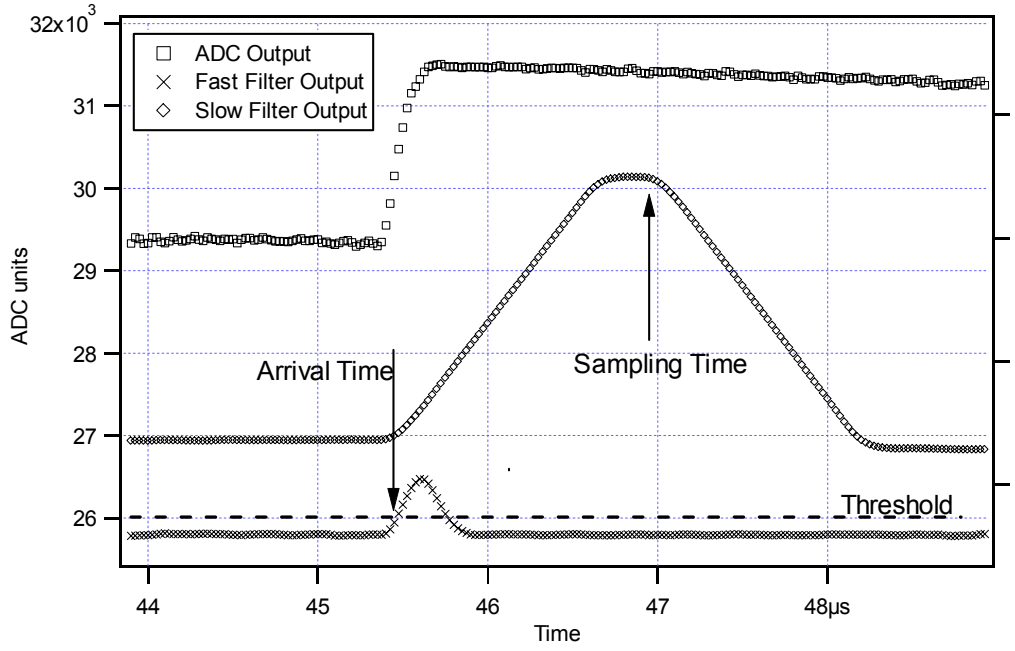


Figure 6.5: Peak detection and sampling in the DGF-4C.

The peak detection and sampling in the DGF-4C is handled as indicated in Figure 6.5. Two trapezoidal filters are implemented, a *fast filter* and a *slow filter*. The fast filter is used to detect the arrival of γ-rays, the slow filter is used for the measurement of V_x, with reduced noise at longer filter rise times. The fast filter has a filter length L_f = 0.1μs and a gap G_f =0.1μs. The slow filter has L_s = 1.2μs and G_s = 0.35μs.

The arrival of the γ-ray step (in the preamplifier output) is detected by digitally comparing the fast filter output to THRESHOLD, a digital constant set by the user. Crossing the threshold starts a counter to count PEAKSAMP clock cycles to arrive at the appropriate time to sample the value of the slow filter. Because the digital filtering processes are deterministic, PEAKSAMP depends only on the values of the fast and slow filter constants and the rise time of the preamplifier pulses. The slow filter value captured following PEAKSAMP is then the slow digital filter's estimate of V_x.

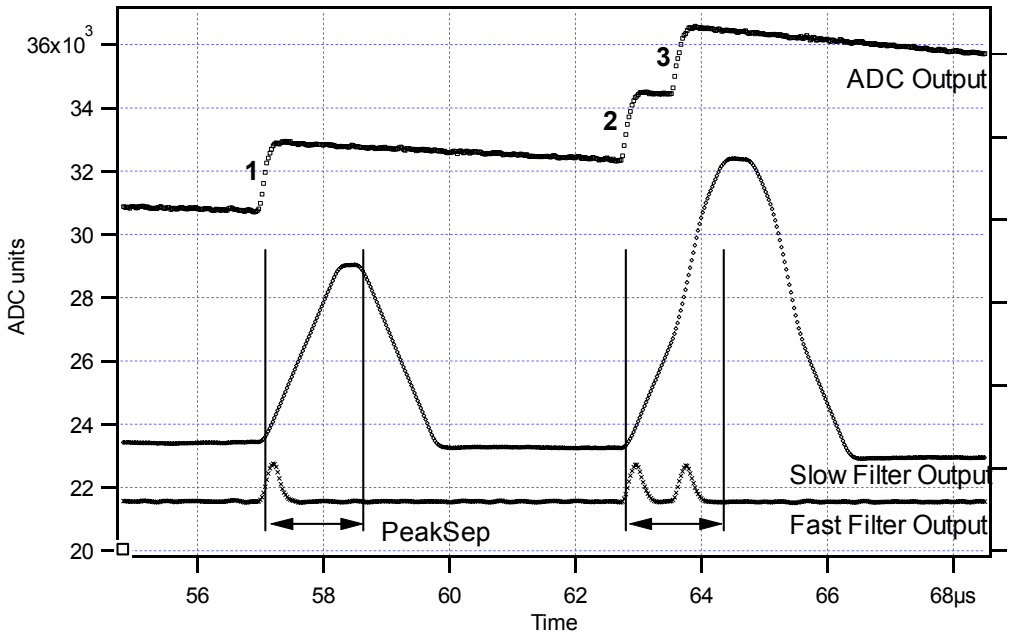


Figure 6.6: **A sequence of 3 γ-ray pulses separated by various intervals to show the origin of pileup and demonstrate how it is detected by the DGF-4C.**

The value V_x captured will only be a valid measure of the associated γ-ray's energy provided that the filtered pulse is sufficiently well separated in time from its preceding and succeeding neighbor pulses so that their peak amplitudes are not distorted by the action of the trapezoidal filter. That is, if the pulse is not *piled up*. The relevant issues may be understood by reference to Figure 6.6, which shows 3 γ-rays arriving separated by various intervals. The fast filter has a filter length L_f = 0.1μs and a gap G_f =0.1μs. The slow filter has L_s = 1.2μs and G_s = 0.35μs.

Because the trapezoidal filter is a linear filter, its output for a series of pulses is the linear sum of its outputs for the individual members in the series. Pileup occurs when the rising edge of one pulse lies under the peak (specifically the sampling point) of its neighbor. Thus,

in Figure 6.6, peaks 1 and 2 are sufficiently well separated so that the leading edge of peak 2 falls after the peak of pulse 1. Because the trapezoidal filter function is symmetrical, this also means that pulse 1's trailing edge also does not fall under the peak of pulse 2. For this to be true, the two pulses must be separated by at least an interval of L + G. Peaks 2 and 3, which are separated by less than 1.0 μs, are thus seen to pileup in the present example with a 1.2 μs rise time.

This leads to an important point: whether pulses suffer slow pileup depends critically on the rise time of the filter being used. The amount of pileup which occurs at a given average signal rate will increase with longer rise times.

Because the fast filter rise time is only 0.1 μs, these γ-ray pulses do not pileup in the fast filter channel. The DGF-4C can therefore test for slow channel pileup by measuring the fast filter for the interval PEAKSEP after a pulse arrival time. If no second pulse occurs in this interval, then there is no trailing edge pileup. PEAKSEP is usually set to a value close to L + G + 1. Pulse 1 passes this test, as shown in Figure 6.6. Pulse 2, however, fails the PEAKSEP test because pulse 3 follows less than 1.0 μs. Notice, by the symmetry of the trapezoidal filter, if pulse 2 is rejected because of pulse 3, then pulse 3 is similarly rejected because of pulse 2.

6.5 Filter decimation

To accommodate the wide range of filter rise times from 0.1μs to 44μs, the filters are implemented in the RTPUs or FPGA configurations with different clock decimation (filter ranges). The ADC sampling rate is always 25ns, but in higher clock decimations, several ADC samples are averaged before entering the filtering logic. In decimation 1, 2^1 samples are averaged, 2^2 samples in decimation 2, and so on. Since the sum of rise time and flat top is limited to 31 decimated clock cycles, filter time granularity and filter time limits are listed in Table 6.1.

Table 6.1: RTPU clock decimations and filter time granularity.

Decimation	Filter granularity [μs]	max. $T_{rise}+T_{flat}$ [μs]	min. T_{rise} [μs]	min. T_{flat} [μs]
1	0.05	1.55	0.1	0.15
2	0.1	3.1	0.2	0.3
3	0.2	6.2	0.4	0.6
4	0.4	12.4	0.8	1.2
5	0.8	24.8	1.6	2.4
6	1.6	49.6	3.2	4.8

As the decimations are implemented in different FPGA configurations, different files have to be downloaded to the FPGA to change decimation.

7 Operating multiple DGF-4C modules synchronously

When many DGF-4C modules are operating as a system, it may be required to synchronize clocks and timers between them and to distribute triggers across modules. It will also be necessary to ensure that runs are started and stopped synchronously in all modules.

To distribute clocks and triggers it is necessary to connect the modules together via the back plane bus. In a revision-D DGF system there will be a clock master on one end of the bus, a number of slaves, and the clock terminator at the far end of the bus. In a revision-E DGF system there will be a clock master on the far right side of the bus and a number of clock repeaters, but no slave modules. The bus is a positive ECL (PECL) bus, which needs to be terminated properly by installing the appropriate jumpers or setting the FET switches through the software. All jumpers associated with the auxiliary bus are located near the 16-pin connector at the back of the board.

Note: if several modules are in a CAMAC crate, you have to initialize all modules, even if you work with only one of them. A module that is not initialized can disturb CAMAC communication, even if it is not specifically addressed.

7.1 Multiplicity unit

The outputs from the digital leading edge discriminators, called fast triggers elsewhere in this document, are summed together in the multiplicity unit to produce a positive multiplicity signal at the MultOut front panel connector. When terminated into 50Ω, its amplitude is 35mV per trigger. The pulse width can be adjusted for each channel independently via the variable “Pulse width” on the Channel CSRA Edit Panel of the DGF-4C viewer. The default is 4 which translates into a width of 4*25ns=100ns. Through the MultIn front panel input you can add the multiplicity signal of another unit to the current sum. This way it is possible to daisy chain many modules to build up multiplicity sums. The signal delay per module is about 5-10ns. If precise timing is required for larger groups of modules, connections should be made in parallel to an external summing unit, rather than daisy chaining the modules.

By pulling jumpers JP14 through JP17 and installing jumpers JP10 through JP13, it is possible to sum the analog signals from the four channels and make these available at the MultOut front panel output.

7.2 Clock distribution

Clocks are distributed via the backplane connector and/or the 4-pin Firewire connector at the back of the board. Depending of the module revision, connections can be made either by a continuous flat cable or by module-to module jumper cables, as detailed below.

7.2.1 Clock distribution for revision-D modules

When jumpers JP1 and JP2 are installed, the local 40MHZ clock is fed into the board circuitry. Jumpers JP5 and JP6 connect the clock lines of the bus to 100Ω termination resistors. The recommended jumper settings for master, middle, and terminator modules are:

Table 7.1: On-board jumper settings for the clock distribution on Rev. D modules.

Module	JP1 and JP2	JP3 and JP4	JP5 and JP6
Master	Yes	Yes	No
Middle (slave)	No	Yes	No
Terminator	No	Yes	Yes
Standalone	Yes	Yes	Yes

Clocks are distributed via the 16 pin backplane connector. The connector pins are organized into two rows. Row-A consists of the even-numbered pins and is the row which is physically closest to the PC-board. The other row is row-B and consists of the odd-numbered pins. When inserted into a CAMAC crate the orientation of the connector is such that the pins 15 and 16 are found at the top end of the connector; row A is to the right. Pins 13 and 15 carry the clock signal on row B, pins 14 and 16 on row A.

The clock lines on row-B are always connected to the differential on-board clock receivers. The clock lines on row-A can be disconnected via the jumpers JP3,4 (located next to the connector). It is advised, however, that JP3,4 be set for all DGF-4Cs in a system. The breaking of the clock bus is of use only in very large systems, where the drive capability of a single PECL driver may prove insufficient to supply all modules. In this case, an external clock driver may supply a PECL clock through the 4-pole Firewire connector (on the fire wire lines 1A and 1B). For the clock master, the Firewire connector can be used as a second clock output.

The clock connection between Rev. D modules is best made with a continuous flat cable, together with the trigger lines (see below).

7.2.2 Clock distribution for revision-E modules

When jumpers JP5 is set to “board clock”, and JP1 and JP2 are installed, the local 40 MHz clock is fed into the board circuitry. With JP5 in the “external” position and JP 1,2 removed, the module will use the incoming clock signal from the backplane connector, and repeat the clock signal to the outgoing lines. Jumpers JP3 and JP4 connect the incoming clock lines of the bus to 100Ω termination resistors. The recommended jumper settings for master, middle (repeater), and terminator modules are:

Table 7.2: On-board jumper settings for the clock distribution on Rev. E modules.

Module	JP1 and JP2	JP3 and JP4	JP5	Crate Position
Master	Yes	No	"board clock"	Most right
Repeater	No	Yes	"external"	Middle
Terminator	No	Yes	"external"	Most left
Standalone	Yes	Yes	"board clock"	Any

Clocks are distributed via the 16 pin backplane connector. The connector pins are organized into two rows. Row-A (incoming) consists of the even-numbered pins and is the row which is physically closest to the PC-board. The other row (outgoing) is row-B and consists of the odd-numbered pins. When inserted into a CAMAC crate the orientation of the connector is such that the pins 15 and 16 are found at the top end of the connector; row A is to the right. Pins 13 and 15 carry the clock signal on row B, pins 14 and 16 on row A.

The clock connection between Rev. E modules should be made using 2x jumper cables from row-B of one module to row-A of the left neighbor. Consequently, the clock master must be in the most right position, the terminator in the most left position. Unlike in Rev. D modules, the connector pins are always connected to the differential translators. A 4-pole Firewire connector provides an alternative clock input (lines 1A and 1B). If the clock signal is provided through row-A of the backplane bus, the Firewire connector can be used to branch off the clock signal to a secondary crate.

The outgoing clock signal of the master module is available on both row-A and row-B, as well as on the 4-pole Firewire connector. This can be used to connect clock signals to a secondary crate of modules, if necessary. No incoming clock signal is allowed on row-A or the Firewire connector of the clock master module.

7.2.3 Mixed systems (revision-D and revision-E)

Since clock distribution between revision-D modules is limited to about 12-16 modules, revision-E repeater modules can be interspersed between the revision-D modules to connect larger groups. The backplane connection should be made with a 16x flat cable, but lines have to be cut as follows for the repeater to work properly:

To the right of a revision-E repeater module, cut lines 13 and 15,
To the left of a revision-E repeater module, cut lines 14 and 16.

If only one revision-D module is placed between the revision-E repeaters, the connection can also be made with jumper cables from row-A to row-B; eliminating the need to cut lines.

7.3 Trigger distribution

This section describes how to use local and distributed triggers. It assumes that a fault in the trigger distribution circuitry of revision-D DGFs has been properly repaired.

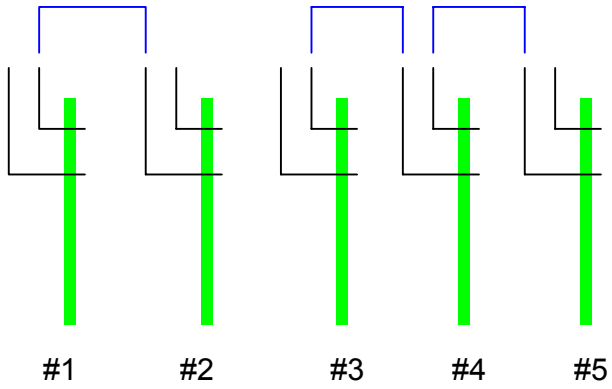
7.3.1 Local triggering

For revision-D and revision-E modules that need to use only local triggers set the Module CSRA on the *Settings* tab to 0x2400. This causes the local PECL trigger bus to be properly terminated into 100Ω. If all channels are to trigger independently, uncheck the checkbox "Respond to group triggers only" for each channel on the Channel CSRA Edit Panel. If the triggers should be distributed among channels on the same board, check the checkbox "Respond to group triggers only" for those channels that should share the triggers.

7.3.2 Distributed triggers

For revision-D and revision-E modules, you can distribute triggers and clocks by connecting modules with a flat cable on the auxiliary 16-pin header on the backside of the module. Cut the trigger lines between modules to define trigger groups (cf. Section 9.2 for the pinout of the backplane connector).

Alternatively, it is possible to daisy-chain the trigger lines. Daisy chain the trigger bus section for all DGFs within a group using jumper cables connecting row-B pins of any module in the group to row-A pins of its immediate neighbor to the left. Break the daisy chain between different trigger groups.



Top view of 5 DGFs split into two trigger groups

For example, if 5 DGFs were to be placed in the CAMAC slots 1 through 5 and there were to be two trigger groups encompassing slots 1,2 and slots 3,4,5, the trigger bus connections would be as shown in the figure above. The fat green lines indicate the DGF PC-boards. The dual row right angle headers are shown in black, and the jumper cable positions are shown in blue at the top. Note that row-A and row-B pins are connected on the board.

Within each trigger group exactly one DGF needs to have its ICSR set to 0x2400. All other DGFs need to have their ICSRs set 0x0000.

Note that for the trigger timing to be correct, all modules and channels in a trigger group have to be set to the same energy filter rise time and flat top.

7.3.3 Front Panel Trigger – Trig Out

On the DGF's front panel, the LEMO connector labeled "Trig OUT" signals the live status of the DSP. During a run, the DSP is live, except when processing an event interrupt. The signal can thus be used as an event trigger output. However one has to keep in mind that the live status is toggled one more time *after* a run has finished in order to read out the live time counters in a consistent manner. To avoid registering triggers when no run is in progress, look at the combination of "Busy OUT" and "Trig OUT": event triggers are valid only if "Busy OUT" is logic 1.

7.4 The busy–synch loop

It is possible to make all DGF-4Cs in a system start and stop runs at the same time. To do this the *Busy* outputs of all modules have to be OR'ed together and the result has to be routed back to the *Synch* inputs. The signals are NIM-level logic signals, and for the OR a LeCroy logic fanin/fanout unit (no. 429A), or similar, will prove useful. The checkbox "Simultaneously start/stop modules" on the *Run* tab should be checked in order to set the DSP variable SYNCHWAIT in all modules to 1. If the busy–synch loop is not used that checkbox should be unchecked so that SYNCHWAIT in all modules will be set to 0.

When the host requests a run start in the modules all of them execute their run initiation sequence and then idle in a tight loop (100ns cycle time) waiting for the last module in the system to be ready. It is the last module ready that allows the run to start. By the same mechanism, the first module to end the run will stop the run in all other modules.

If the timers in all modules are to be synchronized with the next run, check the checkbox "Synchronize clocks" on the *Run* tab, and that will set the DSP variable INSYNCH to 0. This instructs the DSP to reset all timers to zero when the next run starts. From then on they will remain in synch if the system is operated from one master clock.

The busy-synch loop works as follows. The host computer requests a run start. If the requested run is a new run, the run initialization may take a considerable amount of time as a large number of derived quantities have to be computed and histogram memory has to be cleared. On the other hand, if a resume run is requested, the run initialization is much shorter because the assumption in this case is that none of the settings have changed, and that histograms in memory should be retained. At the end of the run initialization the DSP sets the ACTIVE bit (bit 13) in its control status register (CSR) and enters a waiting loop. In this loop it checks the value of the variable SYNCHWAIT. The loop continues until SYNCHWAIT==0. The front panel busy output is logic 1 if the ACTIVE bit is zero, and drops to logic 0 when the ACTIVE bit is set.

When the synch input detects a logic 1 → 0 transition, a synch-interrupt routine is launched, which sets SYNCHWAIT to zero. Upon return from the interrupt the DSP leaves the wait loop to begin the data acquisition run.

Whenever a module encounters an end-of-run condition and stops the run it clears the ACTIVE bit in the CSR. This causes a logic 0→1 transition on its busy output. One can force an end-of-run condition by creating a logic 0→1 transition at the synch input while a run is going on.

Note that if the BUSY-SYNCH loop is not operating properly and there was a run start request with the checkbox "Simultaneously start/stop modules" checked, the DSP will be caught in an infinite loop. In this case, DGF modules should be rebooted so that they can jump out of this loop.

7.5 External gate — GFLT

It is common in larger applications to have dedicated electronics to create event triggers or vetoes. While the DGF-4C does not accept an external fast trigger, it does accept a global first level trigger (GFLT). This signal acts as a validation for an event already recognized by the DGF-4C. Based on multiplicities and other information the dedicated trigger logic needs to make the decision whether to accept or reject a given event. If that decision can be made within a filter rise time of all DGF-4C channels involved, then the GFLT input can be used. The GFLT signal applied to the DGF-4C input must be logic 1 at the time when the event data are latched in the RTPUs. This happens not before a filter rise time has passed since the event arrival and not later than a rise time plus the flat top. Therefore the trigger logic should generate a GFLT pulse that is logic 1 during the filter flat top. The GFLT input expects a NIM-level signal, so logic 1 means –0.8V.

Each involved channel can be programmed individually to require the presence of a GFLT in order to latch event data. This is achieved by setting bit 6 in the variables CHANCSRA0,1,2,3.

7.6 Late event validation — GSLT

If the dedicated trigger logic cannot make the decision within the time window allowed for the GFLT, it is possible to issue a global second level trigger. If this feature is activated the GSLT has to arrive within a predefined time window after the event. Events are continuously stored in the intermediate buffer, but processing is deferred. When a GSLT arrives, the DSP checks if any event in the intermediate buffer satisfies the time constraint. If there is one, it will be processed. Older events will be discarded, younger ones remain in the queue. This feature is however not part of this software distribution. The interrupt routine responding to the GSLT input does, however, record the arrival time of the GSLT and stores it as a 48-bit time in the variable set GSLTTIMEA, GSLTTIMEB and GSLTTIMEC.

In a list mode run, unless compression is 2 or 3, the GSLT time is also stored in the I/O buffer for each event (see Section 4.2).

8 Troubleshooting

8.1 Startup Problems

The following describes solutions to common startup problems.

8.1.1 IGOR compilation error

Every time the DGF-4C Viewer is started, IGOR compiles the functions and procedures contained in the DGF4C.pxp file. If IGOR is installed properly and all driver files are in the correct folders, the Viewer will start successfully in 10-30 seconds. Otherwise you may encounter the following errors:

- IGOR reports a "Function Compilation Error"
  Go to Program Files\Wavemetrics\IGOR Pro Folder\IGOR Extensions and verify that "DGF4C.xop" exist in this folder. If not, copy it from the "drivers" directory in the XIA software distribution.

8.1.2 SCSI hardware problems

SCSI hardware problems can show one or more of the following symptoms:

- IGOR reports "downloading system FPGA was not successful" at startup.
- IGOR reports unlikely SCSI bus and crate number at startup in the history window.
- Inhibit LED on Jorway J73A does not go off.
- The LED on the DGF does not flash at startup.

To find the cause for these problems, verify if

- The SCSI card and its drivers are installed properly on the host PC.
- No hardware conflict between the SCSI card and another device is reported in Windows Device Manager (open Control Panels -> System, go to the Device Manager Tab and look for exclamation marks).
  If you have a "SCSI Explorer" that came with your SCSI card, use it to scan for other hardware conflicts. Sometimes even seemingly harmless devices such as an internal zip drive are treated as SCSI devices.
  Note that the Jorway controller is flagged as having no driver installed. This is correct, as the DGF4C.xop file acts as the driver. If a driver is installed for the Jorway, remove the driver and the device from your system and reboot the PC. Do not search for and install a driver when prompted to do so during the booting process.
- Both the CAMAC crate and the Jorway controller have been powered and connected to the PC while the PC was booting. The Jorway controller should be recognized during the booting process. (The crate can be power cycled after booting without problem).

8.1.3 Jorway problems

Communication problems can sometimes also be caused by the Jorway controller. In normal operation, the Inhibit LED on the Jorway should be on after powering the crate, but go off after booting the system. Symptoms for problems with the Jorway are similar to the more likely SCSI problems. To find the cause for these problems, besides checking for above SCSI problems, verify if

- No windows driver is installed for the Jorway controller (open Control Panels -> System, go to the Device Manager Tab and check the properties for the Jorway controller. It should read "No drivers are required or have been loaded for this device".)
- No other CAMAC module interferes with the Jorway controller. Remove all other modules besides the Jorway and DGF modules from the crate for the initial setup.
- Of the Jorway's piano switch, only switch 2 (Fast) is in the "ON" position.
- The Jorway's serial number is greater than 400.
- The Jorway's internal EPROMS are **not** the "Fermilab" type (see the Jorway manual for details).
- The Jorway's internal jumpers are set for Windows byte ordering (see the Jorway manual for details).

8.1.4 Software problems

Further communication problems can be caused by some software settings. Again, symptoms are similar to SCSI problems and include the following:

- IGOR reports "downloading system FPGA was not successful" at startup.
- The Inhibit LED on the Jorway does go off, but the LED on the DGF does not flash at startup.

To find the cause for these problems, verify if

- The Slot Number on the DGF-4C Start Up Panel is correctly set. The FIPPI File column should list files that exist in the "Firmware" directory of the software distribution, and match the revision of DGF used (suffix –E for revision E, suffix –D for revision D).
- The crate number set in the Start Up Panel matches the number shown on the Jorway controller; and it must not be zero.
- Crate number and SCSI number set in the Start Up Panel match the values reported in the history window at startup.
- The correct controller (J73A or CC32) is selected in the Start Up Panel.

8.1.5 Other problems

Though unlikely, other issues having caused problems in the past include the following:

- The CAMAC crate did not provide sufficient power.
  Note that a DGF-4C module draws about 2A of current on the +6V supply. If you operate a full crate of DGF-4C modules, you need a high powered crate.

- The DGF-4C was operated on a CAMAC extender with too restrictive fusing.
  Make sure the extender can provide the 2A necessary for the DGF-4C on the +6V supply. You can test the DGF-4C's supply power by probing 4 test points at the bottom of the circuit board. Test points "V+5" and "V-5" should show +5V and –5V, respectively; "VCC5" should show +5V and "VCC" should show +3.3V.

9 Appendix A

This section contains hardware-related information.

9.1 Jumpers

The jumpers for the auxiliary bus are as follows (JP3 and JP4 are located directly next to the back connector):

Table 9.1: Jumper settings for the clock and trigger distribution bus on revision-D modules.

Auxiliary Bus Jumper for revision-D modules:	
JP1, JP2	Connect local clock oscillator to the DGF board. Set jumpers in a 1-module system. In a multi-module system set these jumpers only for the clock master, which resides on one end of the clock distribution bus.
JP3, JP4	Connect DGF clock input to auxiliary bus, i.e. to the even pins on the back connector for signals coming from the right neighboring module (seen from the front).
JP5, JP6	100Ω terminators for clock lines on the auxiliary bus. Always set in a 1-module system. In a multi-module system set these jumpers only for the module which resides at the far end of the clock distribution bus, away from the clock master.

Table 9.2: Jumper settings for the clock and trigger distribution bus on revision-E modules.

Auxiliary Bus Jumper for revision-E modules:	
JP1, JP2, JP5	Connect local clock oscillator to the DGF board. Install jumpers JP1 and JP2, and set JP5 to “board clock” in a 1-module system. In a multi-module system set these jumpers only for the clock master, which resides on one end of the clock distribution bus. For the other modules, remove JP1 and JP2, and set JP5 to “external”.
JP3, JP4	100Ω terminators for clock lines on the auxiliary bus. Always set for a single module. Do not set for the clock master module in a multi-module system, but always set for repeater (middle) modules and terminator (far end) module.

At the front of the DGF-4C modules there are four sets of 4 jumpers, one set for each channel, to select the input impedance and attenuation. Jumper JPx00 bypasses a 1kΩ attenuation resistor; Jumpers JPx01, JPx02 and JPx03 connect to ground via 1kΩ, 250Ω and 50Ω, respectively. The jumper settings are as follows:

Table 9.3: Input impedance selection jumpers on revision-D and revision-E modules.

Channel input jumpers for revision-D and revision-E modules:	
JPx00	Remove only if you require attenuation. Attenuation will be 1:2 if JPx01 is set. 1:5 if JPx02 is set. 1:21 if JPx03 is set.
JPx01	Set for input impedance of 1kΩ.
JPx02	Set for input impedance of 250Ω.
JPx03	Set for input impedance of 50Ω.

Table 9.4: Miscellaneous jumpers on revision-D and revision-E modules.

Additional jumpers for revision-D and revision-E modules:	
JP10-JP13	Connect if channel should contribute to analog sum (disconnect JP14-JP17).
JP14-JP17	Connect if channel might contribute to multiplicity sum (disconnect JP10-JP 13).
JP19	Set to connect Firewire Vcc to main board Vcc.
JP20	Set to connect Firewire GND to main board GND.
JP21	Set to connect Firewire Vcc to power from pin-6 of the Firewire connector.

9.2 Pin out of the auxiliary connector in the back for modules

When the DGF-4C is inserted into the CAMAC crate, pin no. 16 is at the top end and closest to the PC board. The connector is a daisy chain connector and can be thought of as consisting of two rows. The row closest to the PC board consists of the even numbered pins and is called row-A. The other row, row-B, has the odd numbered pins. For triggers, row-A and row-B pins of the same signal are always connected on the board; for clocks, row-A and row-B pins of the same signal are connected on the board only if the related jumpers are set properly.

Table 9.5: Pin out of the auxiliary connector on revision-D and revision-E modules.

Pin #	Row	Description	Auxiliary bus section
1	B	Nearest Neighbor Signal B	Nearest Neighbor logic NOT YET IMPLEMENTED
2	A	Nearest Neighbor Signal A	
3	B	Nearest_Neighbor Validation B	
4	A	Nearest_Neighbor Validation A	
5	B	Ground	Trigger bus
6	A	Ground	
7	B	Fast_Trigger_B	
8	A	Fast_Trigger_A	
9	B	DSP_Trigger_B	
10	A	DSP_Trigger_A	
11	B	Ground	
12	A	Ground	
13	B	PECL clock	Clock bus
14	A	PECL clock	
15	B	PECL clock*	
16	A	PECL clock*	

9.3 Control and Status Register Bits

Table 9.6: Map of the Control and status register (CSR).

CSR, revision-D and revision-E modules:		
0x0001	RunEna	Set to 1 to start data acquisition or 0 to stop. Automatically cleared when DSP de-asserts Active to end run.
0x0002	NewRun	Set to 1 to perform run start-up actions such as clearing MCA spectra and statistics, 0 to resume run without clearing.
0x0004	Unused	Reserved for future use.
0x0008	EnaLAM	Set to enable LAM interrupts.
0x0010	DSPReset	Write only. Set to reset DSP processor to initiate program download
0x0020	Unused	Reserved for future use. Note difference to earlier modules
0x0040	Unused	Reserved for future use. Note difference to earlier modules
0x0080	Unused	Reserved for future use. Note difference to earlier modules
0x0100	Unused	Reserved for future use. Note difference to earlier modules
0x0200	Unused	Reserved for future use. Note difference to earlier modules
0x0400	Unused	Reserved for future use. Note difference to earlier modules
0x0800	Unused	Reserved for future use. Note difference to earlier modules
0x1000	DSPErr	Read only. If set DSP has raised error flag.
0x2000	Active	Read only. If set there is a run in progress.
0x4000	LAMState	Read only. If set LAM is set internally.
0x8000	Unused	Reserved for future use.

Table 9.7: Map of the Interface Control and status register (ICSR).

ICSR, revision-D and revision-E modules:		
0x0001	SystemFPGA	Set to reset/configure system FPGA. Changes to 0 if download successful
0x0002	Unused	Reserved for future use.
0x0004	Unused	Reserved for future use.
0x0008	Unused	Reserved for future use.
0x0010	Fippi FPGA0	Set to reset/configure trigger/filter FPGA of channel0. Changes to 0 if download successful
0x0020	Fippi FPGA1	Set to reset/configure trigger/filter FPGA of channel1. Changes to 0 if download successful
0x0040	Fippi FPGA2	Set to reset/configure trigger/filter FPGA of channel2. Changes to 0 if download successful
0x0080	Fippi FPGA3	Set to reset/configure trigger/filter FPGA of channel3. Changes to 0 if download successful
0x0100	Switchbus0	Reserved
0x0200	Switchbus1	Reserved
0x0400	Switchbus2	Set to terminate DSP trigger bus with 100 Ohm
0x0800	Switchbus3	Reserved
0x1000	Switchbus4	Reserved
0x2000	Switchbus5	Set to terminate fast trigger bus with 100 Ohm
0x4000	Unused	Reserved for future use.
0x8000	Unused	Reserved for future use.