

Product Description

The Digital Gamma Finder

DGF-4C

X-Ray Instrumentation Associates

2513 Charleston Road, Suite 207

Mountain View, CA 94043

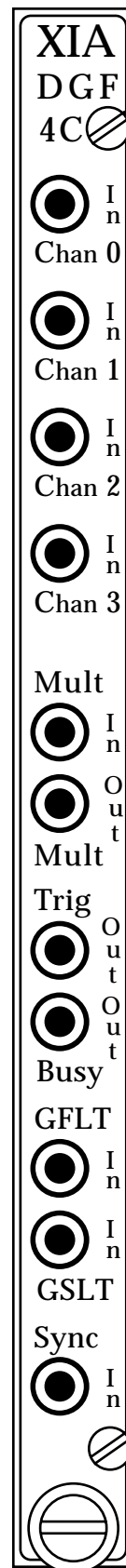
Phone: (650)903-9980

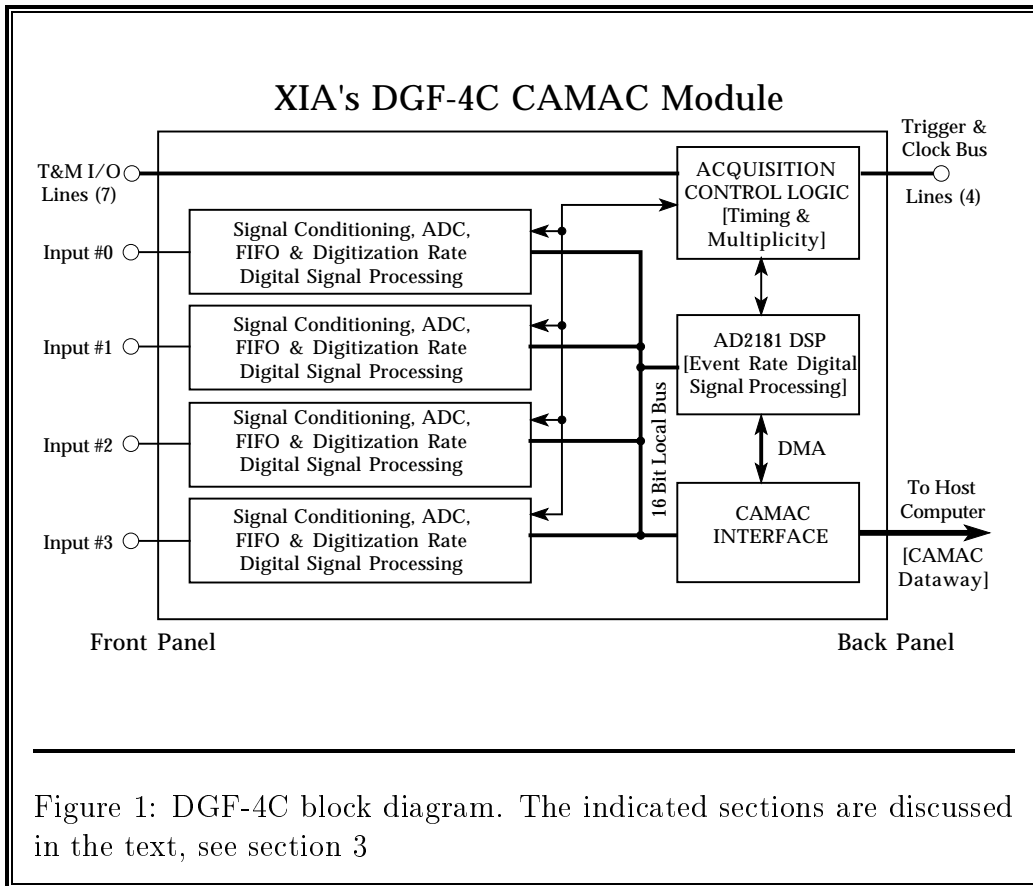
FAX: (650)903-9887

E.-mail: sales@xia.com

February 1999

The contents of this document and the device described therein are protected in part or whole by the following patents: USA Pat.No.s 5,873,054, 5,774,522, and 5,684,850; European Pat. Appl. no.: 96928138.5/0880821; Japan Pat. Appl. no. 9-509405





1 The DGF-4C — Overview

The DGF-4C is a single width CAMAC module that was originally developed for use in nuclear physics experiments involving large arrays of multi-segmented HPGe gamma-ray detectors. However, because its design employs a variety of advanced technologies for digital signal capture and processing, coupled with a flexible triggering system that can span multiple modules, the DGF-4C is a powerful instrument with possible applications to a wide variety of x-ray and gamma-ray detectors.

1.1 DGF-4C system organization

Figure 1 shows a high level block diagram of the DGF-4C. Very briefly, the module accepts four inputs directly from preamplifiers attached to the detectors. These input signals are processed in parallel, as indicated by the 4 identical boxes on the left side of figure 1. Here they are conditioned, digitized, and then subjected to first level digital processing, which handles them in real time at the full data rate (40 MSPS) of the analog-to-digital converter (ADC). Events which are identified and initially validated by the real time processing units (RTPU's) are then passed to a digital signal processor (DSP), the Ana-

log Devices 2181. The data passed include both quantities extracted by the RTPU and signal traces which are captured in First-In-First-Out (FIFO) memories. The DSP waits for the acquisition control logic to successfully complete more global validation tests and the subjects the event signals to a second level of digital signal processing. In this XIA patented scheme, because the DSP secondary processing occurs only at the accepted event rate, it can be substantially more comprehensive than is possible in the RTPUs. It thus becomes possible to perform sophisticated pulse shape analyses on the data close to the front end of the experiment and only pass extracted values back to the central data collection system. Because it relieves data transfer and analysis constraints, the DGF-4Cs' distributed preprocessing capability allows experimental systems to accommodate significantly higher data rates at significantly lower costs, than could previously be achieved using strictly analog instrumentation.

1.2 DGF-4C main features

Preamplifier signal inputs

- Accepts outputs of most charge sensitive preamplifiers
- Input signal range +/- 3 volts (+/- 15 when using input attenuation)
- Both polarity signals accepted

Signal conditioning

- All functions digitally controlled
- 40 dB gain range in steps of 0.0012 dB
- +/- 3V offset DAC, 15 bit precision
- 4th Order Gaussian anti-aliasing filter

Digitization

- 12 bit ADC at 40 MSPS

Spectral resolution

- Up to 15 bit precision (up to 32K spectrum length)
- Baseline corrected
- Detector-specific algorithms (e.g. ballistic deficit, rise time correction) can be implemented.

Signal Capture

- Short FIFO captures 800 ns of data (32 samples)
- Long FIFO captures 25.6 μ s of data (1024 samples)
- Trigger control over FIFO data capture
- Shared clock for timing accuracy

Triggering

- Local group fast triggers

- Multiplicity daisy chain
- Timer synchronization for accurate event arrival time measurements
- Fast trigger granularity of 25 ns (synchronous to system clock)
- Trigger qualified by complex hit patterns (coincidences)
- Fast Global First Level Trigger
- Programmable time Global Second Level Trigger

Second Level Processing - Pulse shape analysis

- Software discriminators (leading edge or constant fraction)
- Construction of waveform descriptors (peak heights, zero crossings, etc.)
- Arrival time analysis (accuracy depends on detector and algorithm)
- Extensible DSP code allows user additions of special purpose analyses

Data Input/Output

- CAMAC
- FERA compatible in conjunction with FERA auxiliary controllers

1.3 DGF-4C applications

1.3.1 Large arrays of segmented HPGe detectors.

In this application, each detector is serviced by a "local group" of several DGF modules which identify gamma-ray absorption events within the detector, extract their energy and time of arrival, and look for coincidences with absorption events in neighboring detectors. By capturing traces on the detector's multiple segments and extracting appropriate parameters, the DGFs allow the central data processing system to compute the locations of events within the detectors to a small fraction of the total detector volume.

A typical experimental use of such an array would be to study nuclear interaction products and their decays.

1.3.2 Multi-element crossed strip Si or Ge detectors

These detectors comprise of a block of semiconductor material with strip contacts on opposing sides, one set of strips (X) running orthogonal to the other (Y). A photon absorption event within the block excites a signal on a single strip on each side of the detector, which thereby define the event's X-Y location. The X and Y signals from the same are paired by their time coincidence. DGF-4Cs combine the ability to make the required coincidence discrimination with the ability to measure the event's deposited energy. Thus the detector is both position and energy resolving. Typical experimental uses of such crossed strip detector would be to collect gamma-ray images

(as in nuclear medicine or astrophysics) or make lifetime measurements of implanted short-lived nuclear isotopes.

1.3.3 Compton gamma telescopes

In Compton gamma camera, high energy gamma rays are scattered from a first detector (or detectors) into a second detector array. For each gamma, the deposited energy is measured in both detectors. Using a line drawn between these two detectors, the Compton scattering equation can be solved to determine the cone of possible directions about this axis on which the gamma ray must have entered the first detector. The intersection of cones from many events is then developed to locate gamma ray sources in the telescope's field of view. Obviously, only coincident events are considered, and the more accurately their energy can be determined, the less uncertainty there is in the thickness of their arrival cone. The DGF is well suited to this application, combining coincidence measurements across many detectors with very good energy resolution.

These detectors have been applied to various problems in astrophysics, nuclear medicine, and radioactive materials localization.

1.3.4 X-Y position sensitive x-ray detectors

One important class of x-ray detectors consists of a method for converting an x-ray into a charge cloud which then drifts toward a 2-dimensionally sensitive detector. The latter include collecting wires with delay lines, resistive anodes, etc. A set of 4 signals is output, 2 X signals and 2 Y signals, where the relative amplitude of the members in a pair encode the spatial locations (e.g. $X = Ax_1/(Ax_1 + Ax_2)$). Signal sets are determined by coincidence. The DGF is capable not only of determining the coincidences and measuring the pulse amplitudes, but the DSP can be programmed to compute the X and Y coordinates as well.

These detectors find application to a wide variety of x-ray scattering experiments.

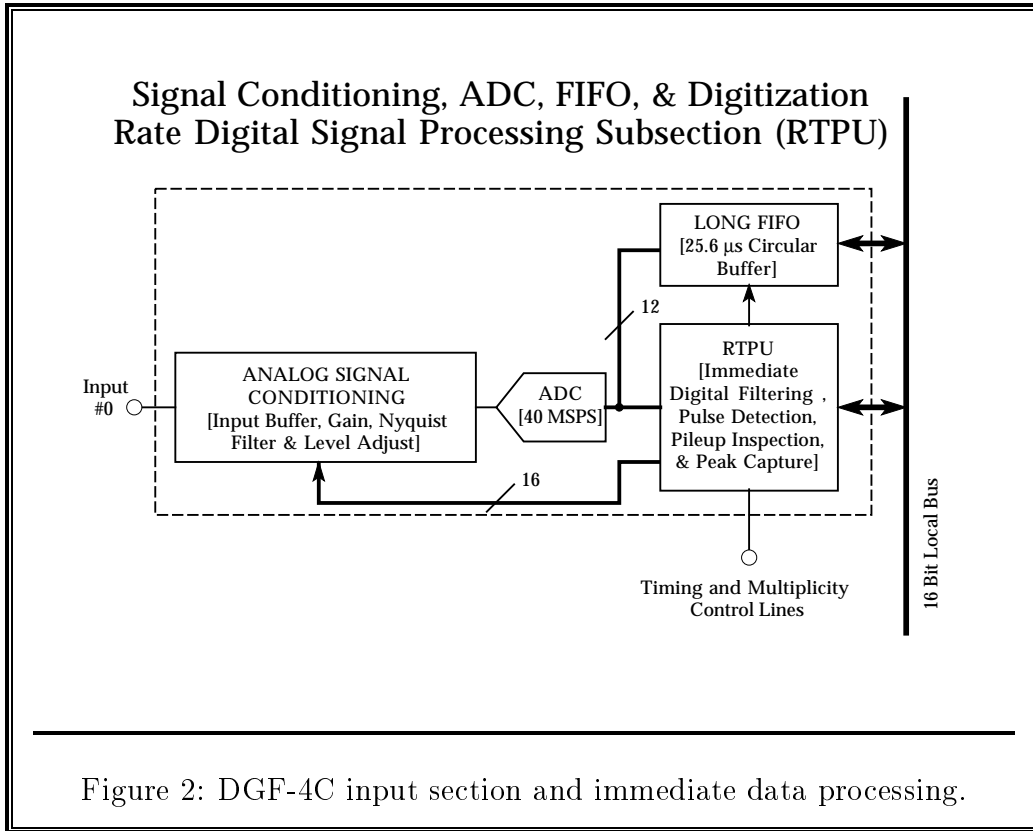
2 The DGF-4C — Functional Description

In this section we will briefly touch upon a few crucial aspects of the DGF-4C operation. They will be discussed in greater detail in sections 3 through 9.

2.1 Single Unit Functions

2.1.1 Signal inputs

As noted above, the DGF-4C was developed for use in applications where high-resolution spectroscopy is to be combined with pulse shape analysis.



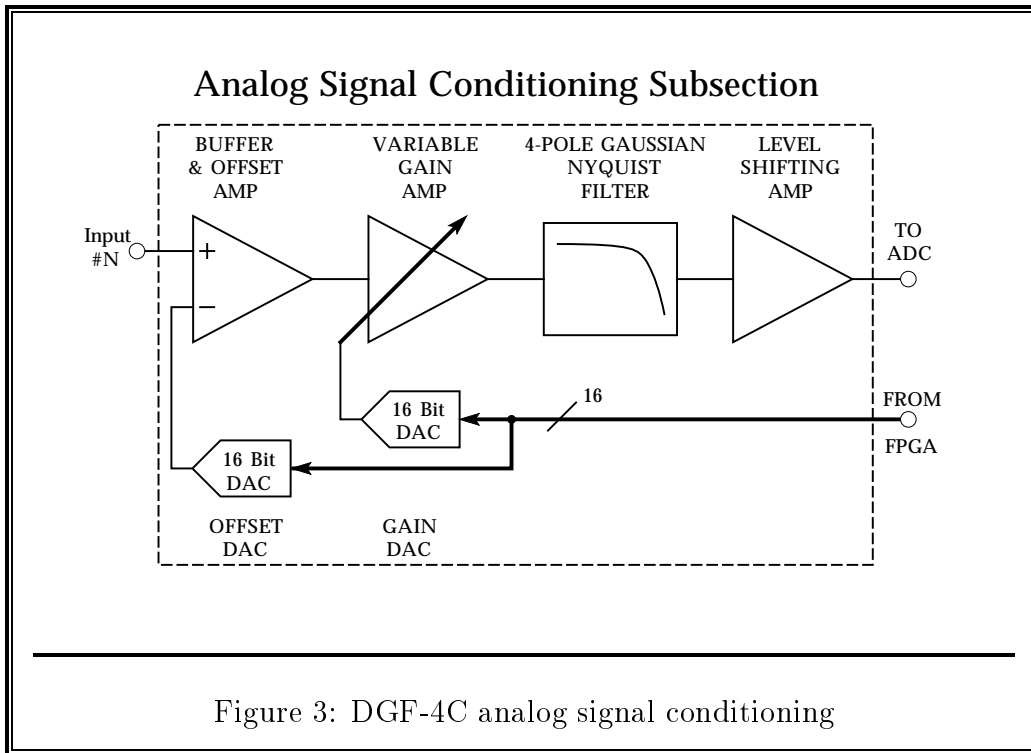
As shown in figure 1, it comprises of four equivalent input channels which are served by a common digital signal processor (DSP). Each of these sections consists of an analog signal conditioning unit (ASC), an ADC, a “long” FIFO and a real-time processing unit (RTPU), see figure 2.

The structure of the ASC is shown in figure 3. It accepts signals from conventional preamplifiers. Its task is to make the incoming signal fit into the active range of its ADC. In most cases that signal will come from an integrating preamplifier, but many other signal sources can be accommodated too, including shaping amplifiers. Offsets can be adjusted over a $\pm 3\text{V}$ range with 15 bits of precision. A variable-gain stage allows to vary the gain by 40dB in steps of 40dB/32768.

Prior to entering the ADC, the offset and gain-adjusted signal is sent through an anti-aliasing filter. This is a 4th-order Gaussian filter with a -12dB corner frequency of 20MHz. The filter is designed to respond to a step input pulse with a 10% to 90% rise time of 50ns and without any overshoot.

2.1.2 Trigger/filter processors and long FIFOs

No external shaping amplifiers are needed, as this function is carried out digitally by the DGF-4C. Each incoming preamplifier signal is digitized at a high rate and processed immediately within the Real-Time Processing Unit (RTPU). The digital wave forms are also stored continuously, just as in a



digital oscilloscope. The RTPUs replace the action of conventional shaping amplifiers using digital filters, which act on the digital data stream in real time. These RTPUs are implemented using programmable logic in the form of field programmable gate arrays (FPGAs). The FPGA of each channel performs about 500 MIPS which on a 4-channel DGF-4C adds up to an impressive computing power of 2000 MIPS.

The structure of the RTPUs is shown in figure 4. They implement the following measurement functions:

- 1) detect pulse events
- 2) extract pulse height information, ie measure energies
- 3) perform baseline measurements
- 4) implement pile-up inspection logic
- 5) accurately measure the live time and input count rate, for each channel
- 6) implement the short FIFO.

only process channels with signals and do not pass "null" data to the data collection system.

Because this processing is carried out under software control, it can be very flexible in regarding the quantities that can be computed. Not only can the waveforms in the FIFOs be analyzed, but other tests can be performed as well. For example, a software filter might be implemented in the DSP to reject unwanted events, thus reducing the computational load for the off-line data analysis.

The DGF-4C supports two modes of DSP operation: singles and list mode. In the former, the DGF-4C acts as a quad multichannel analyser, creating and storing a spectrum for each channel. In the latter, event results are tagged and loaded into an output buffer for retrieval by a host computer, which will then use data from multiple modules to reconstruct events. The two modes can be active concurrently.

The difference between this approach and that of conventional pulse height analysers is clear. The latter connect the preamplifier signal directly to pulse shaping circuitry and then proceed to measure the pulse height above the baseline. In this approach any information about the original signal pulse shape coming from an integrating preamplifier is lost entirely. In the DGF-4C approach, the preamplifier signal is digitized and copies of the data are passed into two parallel branches.

One branch stores the signal so that it can be analyzed later, if appropriate, while the other branch analyzes it immediately, in real time to detect events, extract energy values, etc. This approach is clearly superior in any case where the details of pulse shape, as opposed merely to pulse amplitude, carry information about the detected event.

2.2 System Level Functions

Using the on-board acquisition control logic, see figure 1, multiple DGF-4C modules can be configured to perform in unison, sharing clocks and triggers. This feature allows synchronous waveform acquisition for multi-element detectors. In this case, output from the list mode can be assembled offline, or in an event builder, into large event data structures by using the event tags as identifiers. The DGF-4C communicates with a central data acquisition system via 7 front panel connections. These include multiplicity and trigger I/O as well as a synchronization input. These connections are described below.

Within a larger system, DGF-4C modules can be organized into groups, with, for example, each group servicing a single detector. Triggers and clocks can be shared within a group, while readout can be controlled at the group level, e.g. to only read groups that have reported a trigger (detected a gamma-ray).

To simplify the task of monitoring the performance of each channel in a system with a large number of channels, several monitoring tasks have been

programmed into the DSP. Among these are waveform capture and presentation, automatic offset adjusts, gain measurements, and spectrum monitoring. With these tasks present, it is no longer necessary to use an oscilloscope to examine each channel and verify its performance or adjustments. All monitoring and adjusting can be done remotely from the host computer.

2.3 Embedding DGF-4C's into a data acquisition system

Multiple DGF-4C modules can operate as a stand alone system, communicating clocks and triggers amongst each other. Alternatively, they can be embedded into an existing data acquisition system via a fairly general mechanism, using up to three triggering levels.

In the most simple case, each channel can recognize an input signal and issue a trigger to all modules in the local group. A programmable 4-bit trigger mask is used to select which channels can contribute to a trigger.

Secondly, each module sends out analog multiplicity information. At the multiplicity output an analog signal is formed which has a pulse height of 50mV per channel that reported a hit. A multiplicity input allows to daisy chain several modules. They can be programmed to generate a trigger on a minimum multiplicity.

Alternatively, the modules can be programmed to wait for a more sophisticated external trigger processor to return a global first level trigger (GFLT). The GFLT has to arrive within the peaking time of the pulse height measuring filter in the FPGAs. The GFLT signal does not by itself generate a trigger, but its timely arrival allows the internal FPGA event trigger to be propagated to the DSP, when this mode is chosen.

If a trigger decision cannot be arrived at within the selected filter peaking time, a global second level trigger (GSLT) can be used. On an EventTrigger, whether or not qualified by a GFLT, the DSP transfers data to a first level buffer. These include a time stamp marking the arrival time of the event. It is possible to program the DGF-4C to finally accept and process only those events for which a GSLT is received within a programmable time window after the event, This allows to include slower secondary detectors into making trigger decisions.

The DGF-4C does not include a FERA-style interface. Interfacing to that bus standard must be done via an auxiliary CAMAC crate controller with a FERA interface. Such devices are commercially available.

2.4 Software extensions

One of the strengths of the DGF-4C is that it can be reprogrammed. As is, the module will satisfy the needs of most users. However, particular experiments may want to make use of the fact that the on-board DSP can

Name	Function	Type	Z
Front Panel			
Input #0	Analog input from preamplifier #0	Analog	Choice
Input #1	Analog input from preamplifier #1	Analog	Choice
Input #2	Analog input from preamplifier #2	Analog	Choice
Input #3	Analog input from preamplifier #3	Analog	Choice
M _{in} (in)	Multiplicity, daisy chain input	Analog	50 Ω
M _{out} (out)	Multiplicity, sum output	Analog	50 Ω
Trigger(out)	Event trigger out	NIM	50 Ω
Busy(out)	Dead time indicator	NIM	50 Ω
GFLT(in)	Global First Level (fast) Trigger	NIM	50 Ω
GSLT(in)	Global Second Level (slow) Trigger	NIM	50 Ω
Synch(in)	Wall clock reset	NIM	50 Ω
Rear Panel			
Clock(in)	External clock input	Diff. PECL	100 Ω
Rear Panel Bus connecting local group of modules			
Clock	System clock	Diff. PECL	100 Ω
Fast Trigger	Leading-edge trigger	PECL	100 Ω
DSP Trigger	Event trigger	PECL	100 Ω

Table 1: Front and rear panel connectors of the DGF-4C

be freely programmed. Since the DSP program is not resident in a PROM but is downloaded from a file by the host on start-up, existing DSP code can be enhanced by a sophisticated user.

We plan to make available a software developer's kit which contains the software core in precompiled form plus assembler and linker tools, and some general purpose libraries. Users will then be free to add special data processing algorithms.

2.5 Front/rear panel inputs and outputs

There are 11 front panel inputs and outputs, one rear input and an auxiliary bus connector in the rear. They are summarized in table 2.5 and their functions are as follows:

Inputs 0–3: These analog inputs accept signals from the detector preamplifiers. There is a choice of three jumper-selectable combinations of input impedance and input attenuation. The 3 choices are 50 Ω or 600 Ω without attenuation, and 1kΩ with 5-fold attenuation.

M_{in}: Input for multiplicity daisy chain.

M_{out}: Multiplicity output. Each recognized and unqualified fast trigger contributes 50mV when the M_{out}-signal is terminated into 50Ω. Each of the four channels in the DGF-4C can contribute to M_{out}, while any input to M_{in} is added to M_{out}. Thus, a multiplicity sum can be built across a number of DGF-4C modules.

Trigger: This signal indicates acceptance of an event by a local group of DGF-4C modules, see section 5.

Busy: This signal is set to logic true during any time when the DGF-4C is not ready to accept a new event. Typically this is the case when the DSP is reading data from the trigger/filter FPGAs.

GFLT: Global First Level Trigger

The DGF-4C can be configured to check for the arrival of a GFLT within the energy filter peaking time. If the GFLT does not arrive in time, no trigger is issued to the DSP, see section 4.2. This test is implemented in the trigger/filter FPGAs

GSLT: Global Second Level Trigger

To accommodate longer decision-making times, a GSLT may be required. When using this feature, it is assumed that the GSLT arrives within a preset time window after the event, see section 4.3. This test is performed by a DSP interrupt routine.

Synch: A logic 0 → 1 transition resets the wall clock of the module. The wall clock reading is used to tag events as well as to measure the time in between events.

Clock(in): This differential PECL input is used to feed an external system clock to a group of DGF-4C modules.

Clock(bus): System clock distributed group wide.

Fast trigger(bus): Fast trigger, derived from leading edge of input signals and synchronized with the system clock. All channels in the group are OR-ed together.

DSP-trigger: Event trigger of local group. All card triggers from the group are OR-ed together.

3 Hardware description and specifications

The DGF-4C has 4 independent channels each of which consists of an analog conditioning circuit, an ADC, a long FIFO, and a trigger/filter FPGA, see figure 3. There is one DSP to serve all four channels, and one interface FPGA which handles the I/O tasks. The various components and front panel inputs/outputs are described below.

3.1 Analog input and signal chain

The input impedance and attenuation for each channel can be selected by placing a jumper pair in one of three positions. The standard factory options are 50Ω and 604Ω without attenuation, and $1.0\text{k}\Omega$ with an attenuation factor of 5.

The signal is routed into a buffer amplifier which adds a programmable offset. The offset can vary within -3V and $+3\text{V}$. It is controlled by a 16-bit DAC with a guaranteed monotonicity to 15-bit. Thus, the practical precision achieved in adjusting the offset is $183\mu\text{V}$.

The buffer amplifier is followed by a variable gain amplifier. Its gain range spans a factor of 100 (ie 40dB). It is set by the same type of DAC as above. If the quantity $nDAC$ denotes the number dialed into the DAC, the gain of the amplifier follows a ‘linear-in-dB’ law:

$$20\text{dB} \log_{10}(\textit{gain}) = -11\text{dB} + \frac{nDAC}{65535}40\text{dB}$$

As the DAC in use guarantees 15-bit precision, the gain can be adjusted in multiplicative steps of 1.0001405. Phrased alternatively, the gain between channels can be equalized to a precision of 0.014%.

The output from the variable-gain amplifier is fed into a Gaussian anti-aliasing low-pass filter. This is a passive (RC-LC-LR) filter with four identical poles, which ensures maximum signal fidelity. It responds to a unit step input pulse with the shortest possible rise time (10% to 90% in 60ns) without any overshoot or ringing.

Downstream of the filter there is a level shifter to move the ground-referenced signal into the active ADC range which spans the interval $[2.0\text{V}, 3.0\text{V}]$.

The entire design of the analog conditioning circuitry has been driven by the requirement to maintain maximum signal fidelity. At 20 MHz the total harmonic distortion is less than -70 dB by design.

The ADC of each channel is a 40 MSPS 12-bit device, ie every 25ns the incoming waveform is digitized to 12 bits precision.

The data stream emerging from the ADC is fed into the trigger/filter FPGA, which processes the data without delay and at the ADC sampling rate. It also sends the ADC data immediately to a FIFO memory with a depth of 1024. The FIFO acts as a circular buffer with a length of $25.6\mu\text{s}$, see figure 4.

3.2 Digital filtering and triggering

The data stream emerging from the ADC of each channel is routed into the channel’s trigger/filter FPGA. There, two digital filters are applied. Both are trapezoidal filters, so called because their response to a step function, when plotted against time, looks like a trapezoid. The time it takes for the leading flank to rise is called the peaking time. The following period of time

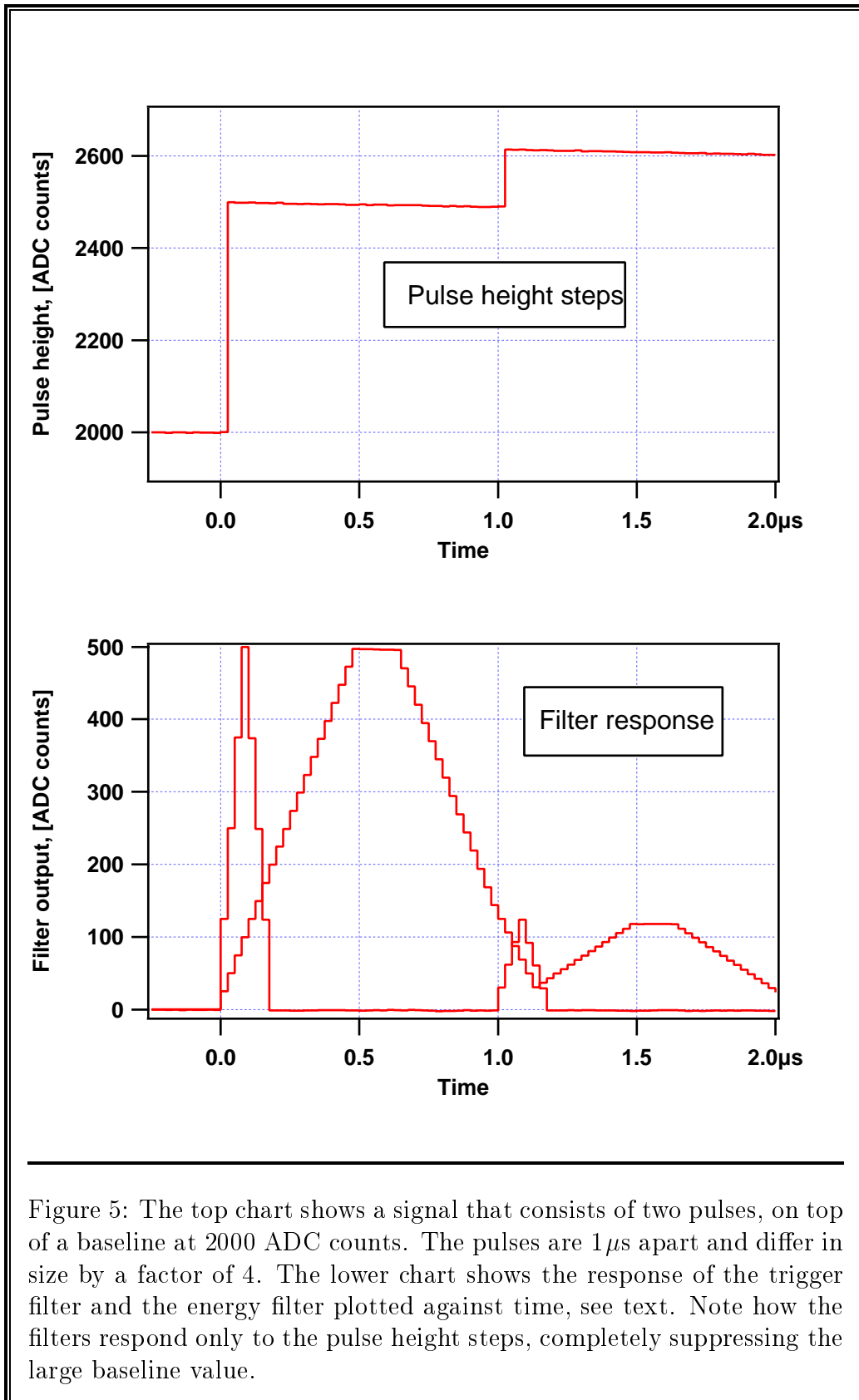


Figure 5: The top chart shows a signal that consists of two pulses, on top of a baseline at 2000 ADC counts. The pulses are $1\mu\text{s}$ apart and differ in size by a factor of 4. The lower chart shows the response of the trigger filter and the energy filter plotted against time, see text. Note how the filters respond only to the pulse height steps, completely suppressing the large baseline value.

for which the output is flat is the flat-top or gap time. Figure 3.1 depicts the response of the two filters to a signal consisting of two steps.

The fast filter is used for generating a fast trigger. The trigger filter in this example uses a peaking time of 100ns and a gap time of zero, resulting in a flat top lasting for just one ADC sampling time of 25ns. As soon as the trigger filter output exceeds a programmable threshold, a fast trigger signal is generated.

The slow filter, or energy filter, is used to measure the pulse height. In this example the energy filter peaking time is 500ns and the flat top lasts for 175ns. The pulse height of the step pulse in the signal is determined by sampling the output of the slow filter when it has reached the flat top.

The two pulses are well separated by the fast filter, but the responses from the slow filter do overlap. However, by the time the slow filter resulting from the second pulse reaches the flat top, the response due to the first pulse has subsided to zero and does not interfere with measurement of the pulse height of the second pulse.

In general, if two step pulses are separated by more than the sum of the slow filter peaking and flat-top time, they are considered to be well separated and their pulse heights can be measured separately. If they occur closer together, the pileup inspection logic will recognize this, and no trigger will be issued for such a pileup event.

However, the two fast filter peaks will, in such a case, still be used to increment a local counter for accurate input count rate measurements.

While the fast trigger filter will usually have a very short peaking time, the peaking and flat top time for the energy filter can be varied over a wide range, between 25ns and 50 μ s.

The filter functions are computed continuously and are updated at the ADC-sampling rate of 40MHz. In between events the energy filter response is used to acquire baseline information. While a perfectly flat baseline elicits a zero response from the filter, and, thus, is of no concern, a sloped baseline, eg from a previous exponentially decaying pulse, will affect the pulse height measurement of a true event. Continuous baseline measurements taken at a high rate in between events allow to correct for the influence of sloped baselines. These corrections are applied by the DSP on an event-by-event basis.

The only caveat is, that the single-pulse waveform from the preamplifier needs to be known accurately. The current DSP code assumes a perfectly exponential behavior of the preamplifier, at least a flat-top or gap time after the pulse arrival. If this is not the case, and excessive ringing at short times or signal undershoots at long times are present, this will adversely affect the energy resolution.

3.3 Multiplicity signals

There are two front panel connectors, (Min , $Mout$) for event multiplicity signals. The inbuilt multiplicity circuitry consists of a four-fold inverting adder followed by a second inverter at which a fifth, external signal can be added. The output of the unit is led to $Mout$. Min is the external signal to the second inverter.

The inputs to the four-fold inverting adder can be chosen by setting jumpers. They are either the four TTL-level signals of the fast triggers from each channel, or they are the analog signals from the outputs of the variable-gain amplifiers in the analog conditioning sections. This way, one can daisy chain across modules either the logical multiplicity signal or the analog sum of many channels.

The logical multiplicity gain is 50mV per fast trigger when terminated into 50Ω , and twice the value with high impedance termination.

The $Mout$ signal is internally routed to a discriminator whose threshold can be varied between -3V and +3V by a 16-bit DAC of the kind mentioned above. The discriminator output is OR-ed together with the external GFLT (see section 4.2) to yield an acceptance trigger. The $Mout$ path to the discriminator can be interrupted by pulling a jumper.

Alternatively to summing logic fast trigger signals in the multiplicity unit, one can set the jumpers to summing the analog outputs from the variable-gain amplifiers. With the programmable threshold of the discriminator it thus possible to trigger on analog signal sums,

3.4 Logical signals

There are two NIM-level front panel outputs, Trigger and Busy, and three NIM-level inputs, GFLT, GSLT, and Synch. On the backside of the module there is a differential PECL-clock input and an auxiliary PECL-bus connector carrying the system clock, the fast trigger and the DSP trigger lines. We will discuss each in turn below.

The trigger output is mainly meant to assist other devices in an extended system. By setting a bit in a certain register, it can be tied to the fast triggers, which are issued as the input signal steps occur. Such a setting would be useful to start a gate for conventional peak-height sensing ADC units. Alternatively, it can be tied to the DSP-trigger, ie actual event acceptance, eg to trigger an external digital oscilloscope.

The Busy signal provides information on dead time within the DGF-4C. It is set to logical 1 whenever the DGF-4C cannot accept new events. Typically this is the case when the DSP is reading event information from the trigger/filter FPGAs. Though dead time and input and throughput count rate are accurately measured within the DGF-4C, the external trigger logic might benefit from knowing exactly when the DGF-4C can not accept new events.

The external trigger signals GFLT and GSLT will be discussed in section 4.

The Synch input zeroes the “wall clock” of the DGF-4C unit. This is a 16-bit counter, which either operates at 40MHz or can be prescaled by the contents of a 16-bit register. Thus the wall clock measures time in increments varying from 25ns to 1.6384ms. It rolls over after 1.6384ms to 107s. The main use of the wall clock is to provide an event identification tag, as its value is logged with every accepted event. At the same time, it can be used to measure the time in between events, or to measure the time between the last Synch pulse and the event.

On the backside of the DGF-4C there is a connector for a PECL clock input. Its purpose is described in section 5.

Through the auxiliary PECL bus on the backside two kinds of trigger signals are communicated between modules belonging to a group. There is a fast trigger (*FT*) signal used to simultaneously halt all FIFOs and an event trigger (*DSP – Trigger*) to notify all modules that an event has been accepted.

The *FT* signal on the bus comprises of an OR of all fast triggers from all channels in a group, see section 5. The fast triggers are generated on the leading edge of the input signals. Their main purpose is to provide a time-zero for stopping data acquisition into the FIFOs. Thus, wave form acquisition can be halted synchronously across modules.

The distributed *DSP – Trigger* signal ensures that all modules within a group recognize and store an accepted event.

4 Operating modes

Each channel of the DGF-4C can essentially operate on its own. On the leading edge of the input signal a fast trigger is generated. It is used to halt, after a programmable delay, the FIFO used for pulse shape acquisition. For a given peaking (T_p) and flat-top time (T_f) of the digital pulse processing filter, the energy information will be available at a time $T = T_p + T_f/2$ after the occurrence of the fast trigger. If in that interval no other fast trigger has been generated, the event will be tagged as good by the pile-up inspection logic. Whether or not the event will be accepted, ie generate a DSP trigger, depends on the operating mode chosen.

The computations indicated above are entirely performed in the trigger/filter FPGA of that channel, without support from the DSP, and at the ADC sampling rate.

4.1 Self-triggered operation

When there is no external trigger processor present, a group of DGF-4C modules can perform all the necessary tasks. Each channel can generate a

DSP trigger and cause the event to be acquired in all modules of that group. Alternatively, the inbuilt multiplicity unit can be used to set a minimum threshold on the multiplicity required to accept a given event.

4.2 Fast external trigger

In a system in which there are other modules besides the DGF-4C, some external trigger logic may be required. If the trigger decision can be reached within the chosen filter peaking time, then a one-level external trigger will suffice. This global first level trigger (GFLT) has to be received before a DSP trigger would be generated, see section 4. It can be received any time between event arrival and $T = T_p + T_f/2$ afterwards. The timely arrival of the GFLT will allow a DSP trigger to be generated.

4.3 Validated external trigger

If a final trigger decision cannot be made within the time frame discussed above, the DGF-4C can employ a two-tiered triggering scheme. The arrival of a GFLT will ensure that the event is accepted and stored in a first-level buffer, together with a time stamp indicating the arrival time.

A validation pulse, or global second level trigger, may arrive later. The condition is that it should arrive in a programmable time window after the event. When it arrives, the DSP looks into the first-level buffer and determines if there is an event that satisfies the required time relation between event arrival time and the time the GFLT was recognized. If such an event is found, it is earmarked for processing in the main program loop.

5 Grouping modules

Within a large system it may be natural to operate DGF-4C modules in groups. Consider the following setup. Two Ge-detectors with segmented cathodes and contiguous anodes (core contacts) are placed close to each other. The experimenter wants to study position resolution and plans to read the cathode wave forms from a detector whenever its anode lights up. If one of the anodes does not report a hit, no energy was deposited in that crystal, and there is no point reading it out.

Thus, the experimenter would form two groups of DGF-4C's each serving one Ge-detector. Using the backside auxiliary bus, the system clock, fast triggers and the DSP trigger would be distributed within each group. An external differential PECL clock would be provided synchronously to both groups via a backside connector. A NIM-level synch signal applied to the front panels would reset the wall clocks in all modules, making sure they all start counting at the same time zero.

The GFLT and GSLT are supplied to all modules, regardless of their grouping. If a group did generate an internal DSP trigger the GFLT and GSLT work as described above. If it did not, the external triggers have no consequence.

Since each event is uniquely identified by its arrival time, it is perfectly ok if not all groups in a system report a hit in each event. In fact, not having to read all groups every time will reduce the load on the readout.

6 DSP software

At its very core, the DSP program consists of a main loop which continuously polls all relevant flags and user switches. From it branch the slow control tasks which help to operate the module, perform calibrations, and monitor its performance.

Event acquisition and periodic housekeeping chores are dealt with in interrupt routines that are called with three different priority levels.

Event processing, a branch off the main loop, is the most extensive task, and is potentially extendable by a sophisticated user.

6.1 Main loop

Figure 6.1 shows, somewhat simplified, the structure of the main loop which is executed indefinitely by the DSP. Asynchronous communication with the host occurs via two flags that are accessible to both. The Run_Enable flag is set by the host and read by the DSP. When set, a run is initiated and continues for as long as the host does not clear Run_Enable.

In response the DSP sets the Run_Active flag once a run has started and clears it when it has ended. The run ends either as programmed or when the host clears Run_Enable.

Both flags are accessible to the host via a CAMAC command to either read from or write to the control and status register.

6.2 Memory organization

All of the DSP data and program memory are accessible by the host computer. They are accessed via direct memory access (DMA) and without interrupting or halting the DSP.

The program memory is 16384 locations deep and 24-bit wide. Of these 8192 have been reserved as spectrum memory, 2048 as baseline spectrum memory, while the rest serves to store the DSP program.

The data memory is 16-bit wide and has a depth of 16350 words. The largest chunk, 8192 words have been reserved for the data output buffer. Various smaller chunks carry the first level buffer and numerous small buffers and local variables.

```

1  forever {
2      while (not Run_Enabled){
3          call Idle_task()
4      }
5      Set_Flag Run_Active
6      if (is_Slow_Control_Run) {
7          call Slow_Control()
8      }
9      else {
10         call Run_Start()
11         while (Run_Enabled){
12             call Baseline_Measurement()
13             call Event_Processing()
14         }
15         call Run_Finish()
16     }
17     Clear_Flag Run_Active
18 }

```

Figure 6: Simplified structure of the main loop of the DSP program in pseudo-code.

The first 256 words of data memory have been reserved for communication with the host computer. They contain information about all data buffers of interest to a user (pointers and lengths) plus all variables used to set the hardware (eg gains and offsets) and the trigger/filter FPGA parameters. All variables in this part of data memory can be addressed by name from the host computer. This not only simplifies the design of the host control software, it also makes it tolerant to relocating the named variables. The full set of these control variables constitutes a complete run profile, which can aid off-line data analysis. It can also be stored on file, be retrieved, downloaded and activated in order to recreate the exact same conditions of an earlier run.

6.3 Slow control tasks

A number of non-repetitive tasks have been grouped together under this heading. They include:

- Open or close all input relays.
- Program FPGAs:

Write all control values to the four control registers of each trigger/filter FPGA.

- Set DACs:
Adjust each DAC to its requested voltage output.
- Ramp offset DACs;
Ramps all offset DACs over their entire range, in 2048 equidistant steps, and measures the average dc-value out of ADC, even when inputs are connected to a detector and signals are present. The 4 times 2048 measurements are stored and used for gain measurement.
- Acquire ADC-trace:
Reads and stores ADC-values for 1 to 4 channels into an 8192 words deep buffer. Time between samples is programmable anywhere between $2 * 25\text{ns} = 50\text{ns}$ and $65535 * 25\text{ns} = 1.64\text{ms}$. Used for untriggered trace capture for monitoring purposes.

6.4 Interrupt tasks

There are four interrupt tasks. They control, in sequence of decreasing priority, the power-up reset, the response to an event trigger, the response to a GSLT, and to a periodic timer.

1) The power-up reset is executed automatically, when the DSP code is downloaded. The routine ensures that reasonable parameter values are set after the DSP program has started and entered the main loop.

2) The event interrupt has to gather as much data as are required (and no more) from the FPGAs. It is guided in this task by numerous masks and switches which will be summarized in section 8. If waveform data are to be exported to the host, the data are immediately written into the output buffer and are processed in place. Otherwise, they are transferred to a first-level buffer.

3) The global second level trigger (GSLT) interrupt looks for an event (in the correct buffer) that arrived at the correct time with respect to the GSLT arrival time. If it finds one, it adjusts pointers used by the Event-Processing() routine, to tell it eg where to find the next event. Note that events are processed only in non-interrupt routines branching off the main loop.

4) The periodic timer interrupt is used for recurrent housekeeping chores, such as keeping track of input counts, live time and the like.

6.5 Event processing

In this routine, and its many subroutines, the real work is done. The raw data from the trigger/filter FPGAs are linearly combined using precomputed

coefficients to reconstruct the pulse heights (ie energies) in the face of changing baselines and exponentially falling signals and backgrounds. In addition, corrections based on pulse shape analysis are also possible.

If a user wanted to enhance the DSP software, this is the most likely place. Here, one might wish add further algorithms involving pulse shape analysis, eg a software constant fraction discriminator, interaction radius determination, pulse shape descriptors (times and amplitudes of extrema and zero-crossings), etc.

6.6 DSP libraries

The only true multipurpose library in use at this time is a double precision floating point library, in which numbers are represented with a signed 15-bit exponent, and a signed 31-bit mantissa. The library has been developed by Analog Devices. A corresponding single precision floating point library will be added later.

7 Host control software

The host communicates with the DGF-4C via CAMAC commands which are summarized below. To ease the task of operating the DGF-4C a number of recurrent functions have been bundled together into a C-library. Higher level functions, such as peak fitting are not included, as users will most likely have their own priorities here, or will prefer to write their own code which will make better use of their data structures.

7.1 CAMAC commands

The CAMAC commands serve a few basic needs, mainly reading and writing of data. All tasks more complicated than that are controlled by the content of the data written.

The CAMAC commands, in brief, are these:

- Read/Write Control_Status_Register (CSR)
This register contains error flags, status bits, and the two run control flags appearing in figure 6.1, Run_enable and Run_active.
- Read/Write Transfer_Start_Address_Register (TSAR)
Subsequent reads/writes to DSP memory start at the address written to the TSAR. The DSP-IDMA controller updates this address (inside the DSP) thus allowing for block transfer. When writing to or reading from other hardware sections (trigger/filter FPGA, FIFO, interface FPGA) the address updating has to be done by repeated calls to Write_TSAR().

- Read/Write data
After the start address has been written to the TSAR, single word or block transfer proceeds via one of these two commands.
- Fast data read
Same read as above but implemented following the level-1 fast CAMAC standard, ie five times faster than normal CAMAC. This function obviously requires the presence of a CAMAC controller that supports level-1 fast CAMAC.

7.2 C-library

The C-library serves the purpose of simplifying integration of DGF-4C's into a larger system. The lower level functions are designed to hide dependencies on hardware and DSP-code as much as is reasonable. Higher level functions help with calibration tasks.

The library is designed such that a user will not have to explicitly make a call to a CAMAC command. This gives XIA the freedom to make slight adjustments to the actual CAMAC commands, without breaking user code. But the library functions are more than just wrappers. They take into account a number of hardware details, which may very well change in the future. Those details, however, will always remain hidden when using the library.

The more important functions include:

- `DGF_Read(slot,address,num_words,buffer)`
`DGF_Write(slot,address,num_words,buffer)`
The CAMAC slot is 'slot', address is the start address, num_words is the number of words (16-bit or 24-bit), and buffer is a pointer to an int32 buffer local to the C-program. All the intricacies of how 16-bit and 24-bit data are physically transferred via CAMAC, and how the TSAR has to be programmed are taken care of by these two routines.
- `DGF_Check_CSR(slot)`
`DGF_RunStart(slot)`
`DGF_RunEnd(slot)`
The first function returns the value of the CSR; the other two manipulate the CSR to start or stop a run. The module affected resides in the designated CAMAC slot.
- `DGF_Measure_Gain(slot)`
Calls the slow control function Ramp_DACs for the module in the designated slot, and analyses the acquired data to yield the gain for each channel in units of ADC-counts per volt input voltage.
- `DGF_Compute_Coeff(slot)`
Quite a number of floating coefficients are required for reconstructing

pulse heights from exponentially decaying preamplifiers. This function computes those, taking the values for the decay constant of each channel from the user profile, stored in the first 256 words of DSP data memory.

8 Software parameters and switches

There is of course a fair number of masks, parameters and switches to control operation of the DGF-4C. Only the most important shall be described here.

Masks are 4-bit numbers where the LSB corresponds to channel 0 and the MSB corresponds to channel 3.

- Triggering and readout

A `trigger_mask` governs which channels may issue an event trigger. A zero-bit inhibits that channel from sending a trigger. (To accumulate unbiased wave forms from the cathodes of a segmented Ge-detector, you may choose a trigger mask that allows only the center contact (anode) to issue a trigger.)

A `channel_mask` determines which channels shall be read upon a trigger.

A `pattern_mask` implements a software acceptance trigger, operative during the event interrupt. It operates on the trigger pattern seen in a single module. It is used, for instance, to acquire coincidence spectra. A `pattern_mask` of 0000 switches off this software filter.

- Trigger switches

The reliance on a global first and/or second level trigger is software controllable.

- Wave form acquisition

Waveforms are accessible through a short FIFO, inside the trigger/filter FPGA, through a long FIFO in an external chip, and through reading the ADC-register of the FPGA. The FIFO lengths are 32 and 1024 time slices, spanning 800ns and 25.6 μ s, respectively. They are halted by a fast trigger, after a programmable delay. As in a digital oscilloscope the trigger point can be put anywhere between the beginning and the end of the recorded/displayed trace.

For waveform acquisition, the user can choose to read all or part of any FIFO, or acquire a waveform through the `Acquire_ADC_Trace` slow control function.

- Operation control

For optimum performance, users can customize the data taking and processing. User settable software switches and masks allow to choose between histogramming modes, choose what sort of data to acquire, and choose whether or not to call a number of data postprocessing functions.

9 User-supplied DSP functions

The factory-supplied programming of the DSP and the FPGAs will allow users to meet a wide range of experiment demands. However, it is one of the strengths of the DGF-4C that its functionality and performance in a particular application may be enhanced by adding to the DSP code. To facilitate such enhancements we plan to produce a developers' kit. It would include the DSP-code in precompiled form and be bundled together with the Analog Devices assembler and linker package. At appropriate places in the DSP code, external functions are called, for which we deliver templates. Those could then be fleshed out by a sophisticated user willing to learn how to program the ADSP-2181 DSP. On the other hand, those parts of the DSP code which must not be changed would be safe from accidental interference.