GRIFFIN Data Format Specification

Compiled November 2015

The GRIFFIN data acquisition system runs under the MIDAS Data Acquisition system in order to record the data from the GRIFFIN HPGe spectrometer for decay spectroscopy experiments at TRIUMF-ISAC.

Chapter 1

Common event information

GRIFFIN data contains a variety of types of events. An event here is defined as a group of 32-bit words beginning with a header word (identified by a first hex letter of 0x8) and ending with a trailer word (identified by a first hex letter of (0xe). The words in each event are numbered in this document with Roman numerals, beginning with I (the header word). Words that are not included in normal running conditions are indicated with the Roman numeral of the previous word and a lowercase letter (i.e. IVa).

For various reasons, certain pieces of information are found in every event. The following sections describe these pieces of information.

1.1 Module Type

The Module Type indicates where the data was generated (GRIF-16, GRIF-4G, GRIF-C etc.) and is provided in word I as bits 25-27 (3 bits). The packet type is identified in the 4 upper bits of word I as type 8 (header).

Example of identifying a header word type:

Word I		Mask		Shift		Hex result		Decimal result
0x8074A998	&	0xF0000000	>>	28	=	0x8	=	8

Example unpacking of module type:

Word I		Mask		Shift		Hex result		Decimal result
0x82800001	&	0x0E000000	>>	25	=	0x1	=	1 (GRIF-16)
0x8424A998	&	0x0E000000	>>	25	=	0x2	=	2 (GRIF-4G)
0x86A4A998	&	0x0E000000	>>	25	=	0x3	=	3 (GRIF-C Slave)
0x8834A998	&	0x0E000000	>>	25	=	0x4	=	4 (GRIF-C Master)

1.2 Word Count

The word count reports how many non-waveform words are expected to be in the event. This includes the header and the trailer. The word count is located in word I (the header), in bits 20-24.

Example of identifying a header word type:

Word I		Mask		Shift		Hex result		Decimal result
0x82A28280	&	0xF0000000	>>	28	=	0x8	=	8

Example of unpacking the address from the header word:

Word I		Mask		Shift		Hex result		Decimal result
0x82A28280	&	0x01F00000	>>	20	=	0x0A	=	10

1.3 Address

The Address is provided in word I as bits 4 to 19 (16 bits). This indicates which channel on which module has provided this event. Addresses are determined by the path each signal takes through the electronics. More specifically, the address can be broken down into the following properties:

Property	Size	Values	Description
M	4 bits	0 to 15	Input channel to Master or ID number of Slave
S	4 bits	0 to 15	Input channel to Slave or ID number of Digitizer
C	8 bits	0 to 255	Input channel to Digitizer

The one exception is that an address of 0xFFFF identifies the event as a PPG event.

Example of identifying a header word type:

Word I		Mask		Shift		Hex result		Decimal result
0x8074A998	&	0xF0000000	>>	28	=	0x8	=	8

Example of unpacking the address from the header word:

Word I		Mask		Shift		Hex result		Decimal result
0x82228280	&	0x000FFFF0	>>	4	=	0x0505	=	1285

Examples of unpacking address properties:

Address		Mask		Shift		Hex result		Decimal result
0x1405	&	0x0000F000	>>	12	=	0x01	=	1 (M)
0x1405	&	0x00000F00	>>	8	=	0x04	=	4 (S)
0x1405	&	0x000000FF	>>	0	=	0x05	=	5 (C)

1.4 Detector Type

The Detector Type is provided in word I as bits 0 to 3 (4 bits). This further specifies the type of data that follows. For data events originating in the GRIF-16 or GRIF-4G, this indicates which detector type has generated this event fragment and is used primarily in master filter decisions. For other types of events (Scaler or PPG), it identifies the type of information given in the rest of the event. A full list of detector type flags can be found in Tables 5.4 and 5.5.

Example of identifying a Header word type:

Word I		Mask		Shift		Hex result		Decimal result
0x8074A928	&	0xF0000000	>>	28	=	0x8	=	8

The Detector Type value can be extracted as:

Word I		Mask		Shift		Hex result		Decimal result
0x8074A900	&	0x0000000F	>>	0	=	0x00	=	00 (HPGe Low Gain)
0x8074A902	&	0x0000000F	>>	0	=	0x02	=	02 (SCEPTAR)
0x8074A905	&	0x0000000F	>>	0	=	0x05	=	05 (Si(Li))

1.5 Network Packet Counter Value

The network packet counter value is provided in word II as the 28 lower bits. The network packet counter value is an event counter that increments by 1 each time a new event is dispatched by the GRIF-C master. The word type is identified in the 4 upper bits as type 0xD.

The network packet counter value is assigned as a group of GRIFFIN events leave the GRIF-C. It is only assigned to the first event in that group; the network packet counter value for all other events will be zero. The network packet counter value is shared by all events: GRIF-16 events (data and scaler), GRIF-C events (PPG), and GRIF-4G events. The network packet counter value number can wrap around if it becomes too large. The user should monitor the network packet counter value numbers which should increase throughout the run. The maximum value of network packet counter value is 268435455. If the number changes from very large to small then the counter has wrapped around, this should be tested at every event.

Example of identifying the network packet counter value word type:

Word II		Mask		Shift		Hex result		Decimal result
0xD00461FF	&	0xF0000000	>>	28	=	0xD	=	13

Example of unpacking the network packet counter value:

Word VIa		Mask		Shift		Hex result		Decimal result
0xD00461FF	&	0x0FFFFFFF	>>	0	=	0x000461FF	=	287231

Chapter 2

Fragment events

Events containing information on interactions in detectors are called fragments events. These events originate in one of two electronic modules: a GRIF-16 or GRIF-4G. The chapter details the information available in all fragment events, whether it is present in all fragment events, only in GRIF-16 fragment events, or only in GRIF-4G fragment events.

2.1 Common fields

The fields described in this section are present in all fragment events.

2.1.1 Filter Patterns

There are up to eight (more in the future) Filter Patterns which can be set and tested for every fragment. The Identifier(s) (1-8) of the Filter Pattern which was passed is provided in word III as bits 16 to 29 (14 bits). If multiple Filter conditions were passed by this fragment then there will be multiple Filter Patterns indicated in this word.

Example of extracting the Filter Pattern(s):

Word III		Mask		Shift		Hex result		Decimal result
0x00010F81	&	0x3FFF0000	>>	16	=	0x0001	=	1 (Only Filter 1 passed)
0x00050F81	&	0x3FFF0000	>>	16	=	0x0005	=	5 (Filters 1 and 3 passed)
0x00850F81	&	0x3FFF0000	>>	16	=	0x0085	=	133 (Filters 1, 3 and 8 passed)

2.1.2 Waveform Indicator

The waveform indicator is provided in word III as a single bit (bit 15). It indicates the expected presence of a waveform in the data. If this value is 0, there is no waveform present. If the value is 1, a waveform sample is expected.

Example of unpacking the waveform indicator word:

Word III	Mask	Shift	Hex result	Decimal result
0x00508001	0x00008000	15	0x1	1

2.1.3 Filter Condition Counter Value

The filter condition counter value are provided in word IV as 31 bits. The filter condition counter is an event counter which is reset at the start of a run and increments by 1 each time a new event master filter condition is passed in the master filter algorithm in the Master GRIF-C. The maximum value is 2147483649 above which the counter will wrap around. If a fragment satisfies multiple filters, it will have multiple filter condition counter value words. The order of the counter values will mirror the order of the satisfied conditions in the filter pattern.

Example of extracting the filter condition counter value:

Word IV		Mask		Shift		Hex result		Decimal result
0x56007C84	&	0x7FFFFFFF	>>	0	=	0x56007C84	=	1442872452

2.1.4 Channel Hit Detection Counter Value

The Channel Hit Detection Counter Value is provided in word V as the 28 lower bits. The Channel Trigger counter is an event counter which is reset at the start of a run and increments by 1 each time a new threshold crossing occurs for this channel on the GRIF-16 or GRIF-4G. The word type is identified in the 4 upper bits as type 9. The Channel Trigger counter value can wrap around if it becomes too large. The user should monitor the channel hit detection counter values which should increase throughout the run. The maximum value of channel hit detection counter value is 268435455. If the number changes from very large to small then the counter has wrapped around, this should be tested at every event.

Example of identifying the Channel Trigger counter value word type:

Word V		Mask		Shift		Hex result		Decimal result
0x900461FF	&	0xF0000000	>>	28	=	0x9	=	9

Example of unpacking the Channel Trigger counter value:

Word IV		Mask		Shift		Hex result		Decimal result
0x900461FF	&	0x0FFFFFFF	>>	0	=	0x000461FF	=	287231

2.1.5 Timestamp

The timestamp is stored in two separate words. The lower 28 bits of word VI contain the low bits of the timestamp and the lowest 14 bits of word VII contain the high bits of the timestamp. The type is identified in the 4 upper bits of both words as type 10 (0xA) for the lower bits and 11 (0xB) for the upper bits. The upper and lower bits of the Timestamp can then be combined to make the full 42-bit timestamp. This is done in two steps by shifting the upper bits and then adding the lower bits using a bitwise or (||), as shown below. The timestamp is in units of 10 nanoseconds and has a maximum value of 4,398,046,511,103. The timestamp will therefore reset either after 12 hours or when the run is restarted.

Note: The timestamps in the fragment events are 42 bits. The timestamps in the scaler events are 44 bits. The timestamps in the PPG events are 56 bits. For accurate comparisons between the different event types, use only the lowest 42 bits.

Example of identifying a Timestamp word type:

Word VI/VII		Mask		Shift		Hex result		Decimal result
0xA074A998	&	0xF0000000	>>	28	=	0xA	=	10 (lower bits)
0xB074A998	&	0xF0000000	>>	28	=	0xB	=	11 (upper bits)

Example of unpacking the Timestamp value:

Word VI/VII		Mask		Shift		Hex result		Decimal result
0xA074A998	&	0x0FFFFFFF	>>	0	=	0x0074A998	=	7645592
0xB0000005	&	0x00003FFF	>>	0	=	0x00000005	=	5

Example of combining the two timestamp sub-words:

Build Timestamp				Hex result		Decimal result
0x00000005	<<	28	=	0x05000000		
0x05000000	||	0x0074A998	=	0x0574A998	=	91531672

2.1.6 Deadtime

There are two methods of determining the event-by-event deadtime of the system for precision measurements; firstly to operate at a fixed deadtime for every event, or secondly to measure and record the deadtime of each event. Both methods will be trialled and evaluated as part of the GRIFFIN DAQ development.

The deadtime for each event is provided as bits 14-27 (14 bits) of word VII. The type is identified in the 4 upper bits as type 0xB. The deadtime is in units of 10 ns and the maximum value is 16,383 representing 162.383 microseconds of deadtime for this event readout.

Example of identifying a deadtime word type:

Word VII		Mask		Shift		Hex result		Decimal result
0xB074A998	&	0xF0000000	>>	28	=	0xB	=	11

An example of unpacking the deadtime value:

Word VII		Mask		Shift		Hex result		Decimal result
0xC074A998	&	0x0FFFC000	>>	14	=	0x001D2	=	466

2.1.7 Channel Accepted Hit Counter Value

The Channel Accepted Hit counter value is provided in word XIII (the trailer) as bits 14-27. The Channel Accepted Hit counter is a 14-bit event counter which is reset at the start of a run and increments by 1 each time a new trigger is accepted for this channel on the GRIF-16 or GRIF-4G. The word type is identified in the 4 upper bits as type 0xE. The Channel Accepted Hit counter value number can wrap around if it becomes too large. The user should monitor the Channel Accepted Hit counter values which should increase throughout the run. The maximum value of the channel accepted hit counter value is 16383. If the number changes from very large to small then the counter has wrapped around, this should be tested at every event.

Example of identifying the Channel Trigger counter value word type:

Word IV		Mask		Shift		Hex result		Decimal result
0xE074A998	&	0xF0000000	>>	28	=	0xE	=	14

Example of unpacking the Channel Accepted Hit counter value:

Word IV		Mask		Shift		Hex result		Decimal result
0xE074A998	&	0x0FFFC000	>>	14	=	0x1D2	=	466

2.2 GRIF-16

This section details sub-words that are particular to fragment events that originate in the GRIF-16 modules.

Table 2.1: GRIFFIN GRIF-16 fragment event format, bank GRF3. Format is based on 4-byte (32-bit) words. The number of bits for each sub-word is indicated in brackets. Words with an asterisk might be repeated within a single fragment event.

I	Packet Type (4) **0x8**	Module Type (3)	Word Count (5)	Address (16)			Det. Type (4)	4 bytes
II	Packet Type (4) **0xd**	Network Packet Counter Value (28)						4 bytes
III	Fixed 00 (2)	Filter Patterns (14)			Waveform Ind. (1)	reserved (10)	Pile-up type (5)	4 bytes
IV*	Fixed 0 (1)	Filter Condition Counter Value (31)						4 bytes
V	Packet Type (4) **0x9**	Channel Hit Detection Counter Value (28)						4 bytes
VI	Packet Type (4) **0xa**	Time Stamp low bits (28)						4 bytes
VII	Packet Type (4) **0xb**	Deadtime (14)			Time Stamp high bits (14)			4 bytes
VIIa*	Packet Type (4) **0xc**	Waveform Sample (28)						4 bytes
VIII	Fixed 0 (1)	Int. Len. high bits (5)	Pulse Height (26)					4 bytes
IX	Fixed 0 (1)	Int. Length low bits (9)		CFD (22)				4 bytes
X	Packet Type (4) **0xe**	Channel Accepted Hit Counter Value (14)			Event Trailer - repeated Channel Hit Detection Counter Value low bits (14)			4 bytes
							Total one hit:	36 bytes

2.2.1 Pile-up evaluation

Chris, please review this section. It could use some expansion.

Detectors counting at a high rate will register multiple hits during a fixed evaluation period. In order to take advantage of higher counting rates, it is therefore essential to handle these pile-up events and recover the energies and times for every interaction in the detector. A method for extracting these energies and times is shown in Figure 2.1 which involves splitting the integration window of size, K, into multiple windows, K1, K2 etc. The event can also be flagged and the waveform output for evaluation offline.

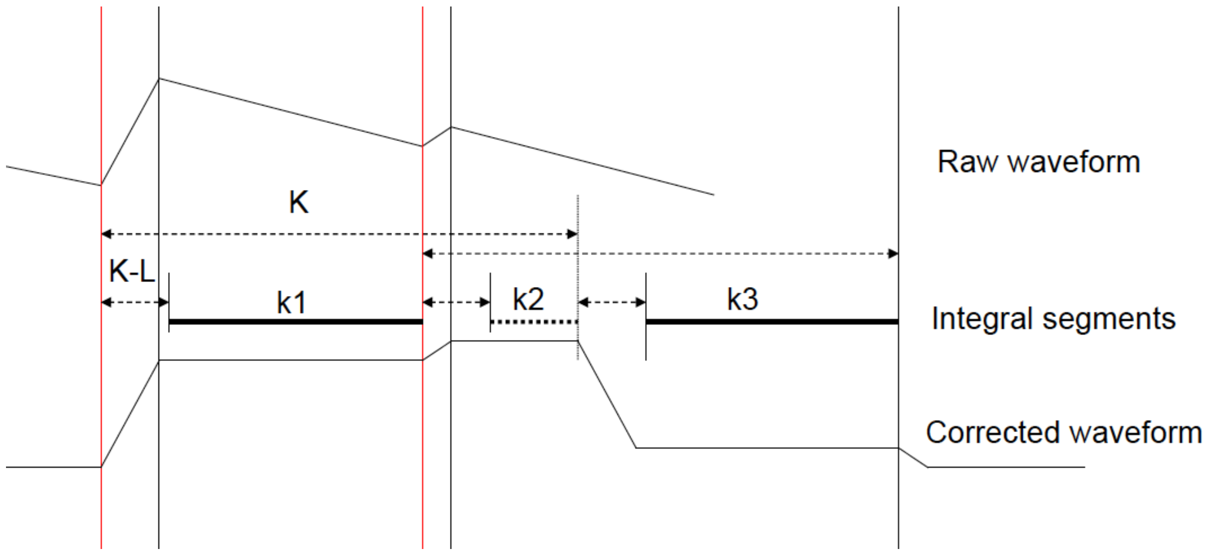

Figure 2.1: Method for extracting energy and time for pile-up events in the firmware.

Energy and time can only be evaluated on the first three pile-up interactions. These interactions will be written to disk as individual fragment events. After those three hits, the channel will be dead until the signal returns to baseline.

2.2.2 Pile-Up Type

The Pile-Up Type indicates how many hits were recorded within the evaluation window, with the maximum being 31 hits in one window, though only the first three will be included in the event with pulse height and CFD words. The Pile-Up Type is provided in word III as bits 4-0 (5 bits).

Example of extracting the Pile-Up Type:

Word I		Mask		Shift		Hex result		Decimal result
0x02000000	&	0x0000001F	>>	0	=	0x0	=	0 (1 Hit)
0x02200002	&	0x0000001F	>>	0	=	0x2	=	2 (3 Hits)
0x02400004	&	0x0000001F	>>	0	=	0x4	=	4 (5 Hits)
0x00F0001F	&	0x0000001F	>>	0	=	0x1F	=	31 (>31 Hits = Bad)

2.2.3 Waveform Data

The Waveform Data is provided in multiple words, the number of which depends on the waveform length. These words are inserted into the event between words VII and VIII. Waveform Data words can be identified by their type which is in the 4 upper bits of each words as type C. The first waveform data word contains a counter of how many waveform data words to be expected (including this word). After that, each Waveform Data word contains two signed 14 bit values representing two samples of the waveform in ADC units. The first sample is in the lower 14 bits of the 28 bit value and the second sample is in the higher 14 bits of the 28 bit value. The most significant bit in each sample indicates After extraction, both samples should be checked for negative values: if the most significant bit in a sample is 1, then it is negative.

Example of identifying a Waveform Data word type:

Word		Mask		Shift		Hex result		Decimal result
0xC57715CC	&	0xF0000000	>>	28	=	0xC	=	12

Example of identifying the first sample from a Waveform Data word:

Word		Mask		Shift		Hex result		Decimal result
0xC57715CC	&	0x00003FFF	>>	0	=	0x15CC	=	5580

Example of identifying the second sample from a Waveform Data word:

Word		Mask		Shift		Hex result		Decimal result
0xC57715CC	&	0x0FFFC000	>>	14	=	0x15DC	=	5596

Example of checking 14-bit samples for negative values:

Sample		Mask		Hex result		Decimal result	Negative?
0x15CC	&	0x2000	=	0x1	=	1	yes

2.2.4 Integration Length

The 14 bits of the integration length are provided in words VIII as bits 26-30 and IX as bits 22-30. The integration length is the number of samples used in the integration step of the pulse height algorithm. The upper and lower bits of the integration length can then be combined to make the full integration length. This is done in two steps by shifting the upper bits and adding the lower bits as shown below.

Example of extracting the integration length:

Word VIII/IX		Mask		Shift		Hex result		Decimal result
0x0474A998	&	0x7C000000	>>	26	=	0x01	=	1
0x4584A998	&	0x7FC00000	>>	22	=	0x116	=	278

Example of building the integration length:

Build value				Hex result		Decimal result
0x01	<<	9	=	0x200		
0x200	||	0x116	=	0x316	=	790

2.2.5 Pulse Height Signal

The pulse height signal is provided in word VIII as the 26 lower bits. The maximum value is 67108863 and is in units of ADC channels.

Two examples of unpacking the Pulse Height value:

Word VIII		Mask		Shift		Hex result		Decimal result
0x680001FF	&	0x03FFFFFF	>>	0	=	0x000001FF	=	511
0x680005B5	&	0x03FFFFFF	>>	0	=	0x000005B5	=	1461

2.2.6 CFD Signal

The timing from a CFD algorithm is provided as a signed integer in word VIII as the 22 lowest bits. The value given in the CFD word is the corrected lowest 18 bits of the time stamp and an additional four bits to specify the sub-timestamp timing. The sub-timestamp timing is given in sixteenths of a timestamp, or 10/16ths of a ns. The value ranges from 0 to 4194303 and is in units of 10/16 nanoseconds. To see the CFD value in timestamp units, divide it by 16.

Two examples of unpacking the CFD value:

Word IX		Mask		Shift		Hex result		Decimal result		Decimal value (ti
0x78000052	&	0x003FFFFF	>>	0	=	0x00000052	=	82	=	5.12
0x780000F6	&	0x003FFFFF	>>	0	=	0x000000F6	=	246	=	15.3

2.2.7 Event Trailer

The Event Trailer is provided in word XI and it indicates the end of the data event fragment. The type is identified in the 4 upper bits as type 14 (0xE). The 14 lower bits of the Channel Trigger counter value are repeated in the data event trailer. This can be used as an additional check to ensure the event is not corrupted.

Example of identifying the Event Trailer word type:

Word XIII		Mask		Shift		Hex result		Decimal result
0xEEEEEEEE	&	0xF0000000	>>	28	=	0xE	=	14

Example of unpacking the lower bits of the Channel Trigger counter value:

Word XIII		Mask		Shift		Hex result		Decimal result
0xE074A998	&	0x0003FFFF	>>	0	=	0x0000A998	=	43416

2.3 GRIF-4G

The format from this digitizer will be similar to that of the GRIF-16 but additional information may be provided associated with DESCANT detectors. This information will include output from additional signal-processing modules such as a neutron-gamma discrimination algorithm.

Table 2.2: GRIFFIN GRIF-4G fragment event format, bank GRF3. Format is based on 4-byte (32-bit) words. The number of bits for each sub-word is indicated in brackets. Words with an asterisk might be repeated within a single fragment event.

I	Packet Type (4) **0x8**	Module Type (3)	Word Count (5)	Address (16)	Det. Type (4)	4 bytes
II	Packet Type (4) **0xd**	Network Packet Counter Value (28)				4 bytes
II	Fixed 00 (2)	Filter Patterns (14)	Waveform Ind. (1)	reserved (15)		4 bytes
IV*	Fixed 0 (1)	Filter Condition Counter Value (31)				4 bytes
V	Packet Type (4) **0x9**	Channel Hit Detection Counter Value (28)				4 bytes
VI	Packet Type (4) **0xa**	Time Stamp low bits (28)				4 bytes
VII	Packet Type (4) **0xb**	Deadtime (14)	Time Stamp high bits (14)			4 bytes
VIIa*	Packet Type (4) **0xc**	Waveform Sample (28)				4 bytes
VIII	Fixed 0 (1)	Int. Len. high bits (5)	Pulse Height (26)			4 bytes
IX	Fixed 0 (1)	Int. Length low bits (5)	CFD Timing (26)			4 bytes
X	Fixed 0 (1)	Long Int. high bits (10)	Short Charge Integration (21)			4 bytes
XI	Fixed 0 (1)	Long Int. low bits (10)	Zero-Crossing Time (21)			4 bytes
XII	Packet Type (4) **0xe**	Channel Accepted Hit Counter Value (14)	Event Trailer - Repeated Channel Hit Detection Counter Value low bits (14)			4 bytes
	Total one hit		10 words I-IX, XII:			40 bytes
	Total one hit (DESCANT Detector Type only)		12 words I-XII:			48 bytes

2.3.1 Waveform Data

The Waveform Data is provided in multiple words, the number of which depends on the waveform length. These words are inserted into the event between words VII and VIII. Waveform Data words can be identified by their type which is in the 4 upper bits of each words as type C. Each Waveform Data word contains two signed 14 bit values representing two samples of the waveform in ADC units. The first sample is in the lower 14 bits of the 28 bit value and the second sample is in the higher 14 bits of the 28 bit value. The most significant bit in each sample indicates After extraction, both samples should be checked for negative values: if the most significant bit in a sample is 1, then it is negative.

Example of identifying a Waveform Data word type:

Word		Mask		Shift		Hex result		Decimal result
0xC57715CC	&	0xF0000000	>>	28	=	0xC	=	12

Example of identifying the first sample from a Waveform Data word:

Word		Mask		Shift		Hex result		Decimal result
0xC57715CC	&	0x00003FFF	>>	0	=	0x15CC	=	5580

Example of identifying the second sample from a Waveform Data word:

Word		Mask		Shift		Hex result		Decimal result
0xC57715CC	&	0x0FFFC000	>>	14	=	0x15DC	=	5596

Example of checking 14-bit samples for negative values:

Sample		Mask		Hex result		Decimal result	Negative?
0x15CC	&	0x2000	=	0x1	=	1	yes

2.3.2 Integration Length

The 10 bits of the integration length are provided in words VIII and IX as bits 26-30. The upper and lower bits of the integration length can then be combined to make the full integration length. This is done in two steps by shifting the upper bits and adding the lower bits, as shown below.

Example of extracting the integration length:

Word VIII/IX		Mask		Shift		Hex result		Decimal result
0x2074A998	&	0x7C000000	>>	26	=	0x08	=	8
0x5874A998	&	0x7C000000	>>	26	=	0x16	=	22

Example of building the integration length:

Build value				Hex result		Decimal result
0x08	<<	5	=	0x100		
0x100	||	0x16	=	0x116	=	278

2.3.3 Pulse Height Signal

The pulse height signal is provided in word VIII as the 26 lower bits, which represent Pulse height 1. The maximum value is 67108863 and is in units of ADC channels. There is no handling of pile-up in the GRIF-4G module. Pile-up events will cause occasional bad data.

Two examples of unpacking the Pulse Height value:

Word VII		Mask		Shift		Hex result		Decimal result
0x680001FF	&	0x03FFFFFF	>>	0	=	0x000001FF	=	511
0x680005B5	&	0x03FFFFFF	>>	0	=	0x000005B5	=	1461

2.3.4 CFD Timing Signal

Vinzenz, this could use some expansion.

The timing from a CFD algorithm is provided in two parts word IX as the 21 lower bits. There is both a CFD time itself (bits 0-20) and the "remainder" (bits 21-24) which is generated because of the difference between the 100 MHz timestamp and the 125 MHz clock that the FPGA is running on. The CFD time is in units of 1/256 ns.

A time in units of 1 ns can be constructed as $[(\text{Timestamp} \times 10) + \text{Remainder}]$.

A time in steps of 1/256 ns (~ 3.9 ps) can be constructed as $[(\text{Timestamp}\times 10)+\text{Remainder}+\frac{\text{CFD}}{256}]$.

2.3.5 Short Charge Integration

Note: Words X and XI are only read out for the DESCANT detector type.

The 19 bits of the Short Charge Integration are provided in word X as bits 0-18. This short integration of the charge signal is used in combination with the Long Charge Integration and the CFD information to provide neutron-gamma discrimination. The word type is identified in the 4 upper bits as type 0xF.

Example of identifying the Short Charge Integration word type:

Word X		Mask		Shift		Hex result		Decimal result
0xF00461FF	&	0xF0000000	>>	28	=	0xF	=	15

Two examples of unpacking the Short Charge Integration value:

Word X		Mask		Shift		Hex result		Decimal result
0xF00461FF	&	0x001FFFFF	>>	0	=	0x000001FF	=	511
0xF00465B5	&	0x001FFFFF	>>	0	=	0x000005B5	=	1461

2.3.6 Long Charge Integration

Note: Words X and XI are only read out for the DESCANT detector type.

The 19 bits of the Long Charge Integration are provided in words X as bits 19-27 and XI as bits 22-31. The upper and lower bits of the Long Charge Integration can then be combined to make the full Long Charge Integration. This is done in two steps by shifting the upper bits and adding the lower bits, as shown below.

Example of extracting the Long Charge Integration:

Word X/XI		Mask		Shift		Hex result		Decimal result
0xF044A5B5	&	0x7FE00000	>>	21	=	0x382	=	898
0x02C4A998	&	0x7FE00000	>>	21	=	0x16	=	22

Example of building the long charge integration:

Build Long Integration				Hex result		Decimal result
0x0016	<<	9	=	0x2000		
0x2000	||	0x382	=	0x2382	=	9090

2.3.7 Zero-Crossing Time

Note: Words X and XI are only read out for the DESCANT detector type.

The Zero-Crossing Time is provided in word XI as the 21 lower bits. This is the result of a CFD algorithm processing the signal after it has been integrated.

Two examples of unpacking the Zero-Crossing Time value:

Word XI		Mask		Shift		Hex result		Decimal result
0x78000052	&	0x001FFFFF	>>	0	=	0x00000052	=	82
0x780000F6	&	0x001FFFFF	>>	0	=	0x000000F6	=	246

2.3.8 Event Trailer

The Event Trailer is provided in word XII and it indicates the end of the event fragment. The type is identified in the 4 upper bits as type 14. The 14 lower bits of the timestamp is repeated in the event trailer. This can be used as an additional check to ensure the event is not corrupted.

Example of identifying the Event Trailer word type:

Word XII		Mask		Shift		Hex result		Decimal result
0xEEEEEEEE	&	0xF0000000	>>	28	=	0xE	=	14

Example of unpacking the lower bits of the Timestamp value:

Word XI		Mask		Shift		Hex result		Decimal result
0xE074A998	&	0x0003FFFF	>>	0	=	0x0000A998	=	43416

Chapter 3

Scaler events

3.1 Event Format and Size

Table 3.1: GRIFFIN GRIF-16 Scaler Event Format written to disk, bank GRF3. Words with asterisks may be repeated.

I	Packet Type (4) **0x8**	Module Type (3)	Word Count (5)	Address (16)	Det. Type (4) **0xF**	4 bytes
II	Packet Type (4) **0xd**	Network Packet Counter Value (28)				4 bytes
III	Packet Type (4) **0xa**	Time Stamp low bits (28)				4 bytes
IV*	Scaler value (32)					4 bytes
V	Packet Type (4) **0xe**	Scaler Type (4)	Time Stamp high bits (16)	Repeated Time Stamp low bits (8)		4 bytes

3.1.1 Timestamp

The 46-bit timestamp is stored in two separate words as the lower 28 bits and bits 8-23 of two words, word III and word V respectively. Word III is the lower bits of the Timestamp and Word V (trailer) contains the upper bits of the Timestamp. The type is identified in the 4 upper bits of both words as type 10 (0xa) for the lower bits and 15 (0xe) for the upper bits. The upper and lower bits of the Timestamp can then be combined to make the full 46-bit timestamp. This is done in two steps by shifting the upper bits and then adding the lower bits using a bitwise or (||), as shown below.

Note: The timestamps in the fragment events are 42 bits. The timestamps in the scaler events are 44 bits. The timestamps in the PPG events are 56 bits. For accurate comparisons between the different event types, use only the lowest 42 bits.

Example of identifying a Timestamp word type:

Word III/V		Mask		Shift		Hex result		Decimal result
0xA074A998	&	0xF0000000	>>	28	=	0xA	=	10 (lower bits)
0xE0000190	&	0xF0000000	>>	28	=	0xE	=	15 (upper bits)

Example of unpacking the Timestamp value:

Word III/V		Mask		Shift		Hex result		Decimal result
0xA074A998	&	0x0FFFFFFF	>>	0	=	0x0074A998	=	7645592
0xE0000190	&	0x00FFFF00	>>	8	=	0x00000001	=	1

Example of building the timestamp:

Build Timestamp				Hex result		Decimal result
0x00000005	<<	28	=	0x05000000		
0x05000000	||	0x0074A998	=	0x0574A998	=	91531672

The timestamp is in units of 10 nanoseconds and has a maximum value of 4,398,046,511,103. The timestamp will therefore reset either after 12 hours or when the run is restarted.

3.1.2 Scaler Values

Scaler values are provided by the GRIF-16 modules and are particular to each channel. Scaler events include either a series of deadtime counters or a series of rate counters, depending on the scaler type. Scaler values have no identifying first bits and are simply identified as the words between word III (identifier of 0xa) and word V (identifier of 0xe) in a scaler event. The number of scaler values will vary with the different scaler types.

Two examples of unpacking the scaler value:

Word IV		Mask		Shift		Hex result		Decimal result
0x85756ab4	&	0xFFFFFFFF	>>	0	=	0x85756ab4	=	2239064756
0x64bd760e	&	0xFFFFFFFF	>>	0	=	0x64bd760e	=	1690138126

3.1.3 Scaler Type

The Scaler Type is provided in word V (trailer) of GRIF-16 Scaler Events, in bits 24-27. It identifies whether a Scaler Event lists the deadtime scalers or the rate scalers. A type of 0 indicates a deadtime scaler, while a type of 1 indicates a rate scaler.

Example of identifying the Scaler Type word type (trailer):

Word XIII		Mask		Shift		Hex result		Decimal result
0xE0000190	&	0xF0000000	>>	28	=	0xE	=	14

Example of unpacking the Scaler Type:

Word XIII		Mask		Shift		Hex result		Decimal result
0xE0000190	&	0x0F000000	>>	24	=	0x1	=	1

3.1.4 Event Trailer

The Scaler Event Trailer is provided in word V and it indicates the end of the Scaler Event fragment. The first 8 bits of the Time Stamp Low value are repeated in the scaler event trailer. This can be used as an additional check to ensure the event is not corrupted.

Example of identifying the Scaler Event Trailer word type:

Word V		Mask		Shift		Hex result		Decimal result
0xE0000190	&	0xF0000000	>>	28	=	0xE	=	14

Example of unpacking the first 8 bits of the Time Stamp Low value from the Scaler Event Trailer word:

Word V		Mask		Shift		Hex result		Decimal result
0xE0000190	&	0x000000FF	>>	0	=	0x90	=	144

Chapter 4

PPG events

Additional information will be produced from the GRIF-C such as statistics and information on any events which have been dropped and did not make it to the disk file. The PPG event described in this chapter is one such piece of additional information.

Table 4.1: GRIFFIN GRIF-C PPG event format, bank GRF3. Format is based on 4-byte (32-bit) words. The number of bits for each sub-word is indicated in brackets.

I	Packet Type (4) **0x8**	Module Type (3) **0x4**	Word Count (5)	Address (16) **0xFFFF**	Det. Type (4) **0x0**	4 bytes
II	Packet Type (4) **0xd**	Network Packet Counter Value (28)				4 bytes
III	Packet Type (4) **0x0**	Expected GRIF-C PPG pattern (28)				4 bytes
IV	Packet Type (4) **0x9**	Confirmed PPG pattern (28)				4 bytes
V	Packet Type (4) **0xa**	Time Stamp low bits (28)				4 bytes
VI	Packet Type (4) **0xb**	Time Stamp high bits (28)				4 bytes
VIII	Packet Type (4) **0xe**	Trailer: previous PPG pattern (28)				4 bytes

4.1 Expected GRIF-C PPG pattern

The expected GRIF-C PPG pattern is provided in the last 28 bits of word III of GRIF-C PPG events. It is identified with a packet type of 0x0. The PPG pattern itself is 16 bits long. The 28 bit value contains the lower 12 bits of the PPG pattern followed by the full 16-bit pattern.

Example of identifying an expected GRIF-C PPG word:

Word III		Mask		Shift		Hex result		Decimal result
0x0008c008	&	0xF0000000	>>	28	=	0x0	=	0

Example of unpacking an expected GRIF-C PPG word:

Word III		Mask		Shift		Hex result		Decimal result
0x0008c008	&	0x0FFFFFFF	>>	0	=	0x008c008	=	573448

4.2 Confirmed PPG pattern

The confirmed PPG pattern that is read back from the PPG module is provided in the last 28 bits of word IV of GRIF-C PPG events. This should have the same value as the expected GRIF-C PPG pattern - any difference indicates a PPG failure or event corruption. It is identified with a packet type of 0x9. The PPG pattern itself is 16 bits long. The 28 bit value contains the lower 12 bits of the PPG pattern followed by the full 16-bit pattern.

Example of identifying a confirmed PPG word:

Word IV		Mask		Shift		Hex result		Decimal result
0x9008c008	&	0xF0000000	>>	28	=	0x9	=	9

Example of unpacking a confirmed PPG word:

Word IV		Mask		Shift		Hex result		Decimal result
0x9008c008	&	0x0FFFFFFF	>>	0	=	0x008c008	=	573448

4.3 Timestamp

4.3.1 Data Event Timestamp

The timestamp is stored in two separate words as the lower 28 bits of words V and VI. Word V is the lower bits of the Timestamp and Word VI is the upper bits of the Timestamp. The type is identified in the 4 upper bits of both words as type 10 (0xA) for the lower bits and 11 (0xB) for the upper bits. The upper and lower bits of the Timestamp can then be combined to make the full 56-bit timestamp. This is done in two steps by shifting the upper bits and then adding the lower bits using a bitwise or (||), as shown below. The timestamp is in units of 10 nanoseconds.

Note: The timestamps in the fragment events are 42 bits. The timestamps in the scaler events are 44 bits. The timestamps in the PPG events are 56 bits. For accurate comparisons between the different event types, use only the lowest 42 bits.

Example of identifying a Timestamp word type:

Word V/VI		Mask		Shift		Hex result		Decimal result
0xA074A998	&	0xF0000000	>>	28	=	0xA	=	10 (lower bits)
0xB074A998	&	0xF0000000	>>	28	=	0xB	=	11 (upper bits)

Example of unpacking the Timestamp value:

Word V/VI		Mask		Shift		Hex result		Decimal result
0xA074A998	&	0x0FFFFFFF	>>	0	=	0x0074A998	=	7645592
0xB0000005	&	0x0FFFFFFF	>>	0	=	0x00000005	=	5

Example of building the timestamp:

Build Timestamp				Hex result		Decimal result
0x00000005	<<	28	=	0x05000000		
0x05000000	||	0x0074A998	=	0x0574A998	=	91531672

4.4 Previous PPG pattern

The previous PPG pattern is provided in the last 28 bits of word VIII (trailer) of GRIF-C PPG events. This should have the same value as the expected GRIF-C PPG pattern from the previous PPG event - any difference indicates a PPG failure or event corruption. It is identified with a packet type of 0xe. The PPG pattern itself is 16 bits long. The 28 bit value contains the lower 12 bits of the PPG pattern followed by the full 16-bit pattern.

Example of identifying a previous PPG word:

Word VIII		Mask		Shift		Hex result		Decimal result
0xe008c008	&	0xF0000000	>>	28	=	0xe	=	15

Example of unpacking a previous PPG word:

Word VIII		Mask		Shift		Hex result		Decimal result
0xe008c008	&	0x0FFFFFFF	>>	0	=	0x008c008	=	573448

Chapter 5

Reference information

5.1 Bank version information

Over the lifetime of the data acquisition system, the data format is expected to change. Care will be taken to make these changes backward compatible, but occasionally, this will not be possible. In that case, the bank name will be used to track the data format used. This document will be maintained with the current data format. Differences between the current data format and the older data formats will be detailed in this section. The current bank name is GRF3.

Old GRIF-16 data format tables are included in section 5.3.

- GRF3: current bank name
- GRF2:
  - Different header format than GRF3. Header word contains the following information (bit size): Packet Type (4), Pile-Up Type (2), Module Type (3), Number Filter Patterns (3), Address (16), Detector Type (4).
  - Different GRIF-16 / GRIF-4G word III format. Word III contains the following information (bit size): Fixed 00 (2), Filter Patterns (14), PPG pattern (16).
  - Different GRIF-16 word IX format. Word IX contains the following information (bit size): Integration length low bits (5), CFD (26).
  - Different trailer format than GRF3. Trailer word contains the following information (bit size): Packet Type (4), Event Trailer (14), Event Trailer - Repeated Time Stamp low bits (14).
  - This was the first bank to include scaler events.
  - This bank did not have PPG events.
- GRF1:

- Different header format than GRF2. Header word contains the following information (bit size): Packet Type (4), Number Filter Patterns (4), Module Type (3), Pile-Up Type (3), Address (14), Detector Type (4).
- This bank did not have PPG events.

5.2 Binary Counting Definition

The bits in the words in this document are counted from 0 to 7 in each byte, using LSB bit numbering where bits are counting right to left, from the least to most significant. Therefore a 4-byte, 32 bit word contains bits from 0 to 31.

An example of how the bits in a 16-bit, 2-byte word are counted in this document:

Bit Number:	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Bit Weight:	32,768				2,048				128				8			

5.3 Reference tables

Packet Type Hexadecimal	 Decimal	 Word description
0x8	8	Header (start of event)
0x9	9	Channel Hit Detection Counter Value
0xA	10	Timestamp Lower bits word
0xB	11	Deadtime and Timestamp high bits
0xC	12	Waveform sample
0xD	13	Network Packet Counter Value
0xE	14	Trailer (end of event)
0xF	15	Long Charge Integration and Short Charge Integration (GRIF-4G only)

Table 5.1: Data event packet types.

Packet Type Hexadecimal	 Decimal	 Word description
0x8	8	Header (start of event)
0x9	9	Confirmed PPG pattern (PPG only)
0xA	10	Timestamp Lower bits word
0xB	11	not used
0xC	12	not used
0xD	13	Network Packet Counter Value
0xE	14	Trailer (end of event)
0xF	15	not used

Table 5.2: Packet types for scaler and PPG events.

Mask	Shift	Name
0x0E000000	25	Module Type
0x01F00000	20	Word count
0x000FFFF0	4	Address
0x0000000F	0	Detector Type
0x0FFFFFFF	0	Network Packet Counter Value
All fragment events		
0x3FFF0000	16	Filter Patterns
0x00080000	15	Waveform Indicator
0x7FFFFFFF	0	Filter Condition Counter Value
0x0FFFFFFF	0	Channel Hit Detection Counter Value
0x0FFFFFFF	0	Timestamp Lower Bits
0x00003FFF	0	Timestamp Upper Bits
0x0FFFC000	14	Deadtime
0x0FFFC000	14	Channel Accepted Hit Counter Value
GRIF-16 data		
0x0000001F	0	Pile-Up Type
0x00003FFF	0	Waveform sample 1
0x0FFFC000	14	Waveform sample 2
0x7C000000	26	Integration Length Lower Bits
0x7CC00000	22	Integration Length Upper Bits
0x03FFFFFF	0	Pulse Height Signal
0x003FFFFF	0	CFD Timing Signal
GRIF-4G data		
0x00003FFF	0	Waveform sample 1
0x0FFFC000	14	Waveform sample 2
0x7C000000	26	Integration Length Lower Bits
0x7C000000	26	Integration Length Upper Bits
0x03FFFFFF	0	Pulse Height Signal
0x03FFFFFF	0	CFD Timing Signal
0x001FFFFF	0	Zero-Crossing Time
0x001FFFFF	0	Short Charge Integration
0x7FE00000	21	Long Charge Integration Upper Bits
0x7FE00000	21	Long Charge Integration Lower Bits
Scaler		
0x00FFFF00	8	Timestamp high bits
0x0FFFFFFF	0	Timestamp low bits
0xFFFFFFFF	0	Scaler values
0x0F000F00	24	Scaler type
PPG		
0x0FFFFFFF	0	PPG patterns
0x0FFFFFFF	0	Timestamp low bits
0x0FFFFFFF	0	Timestamp high bits

Table 5.3: Bit masks and bit shifts used to unpack GRIFFIN event data.

ID	Short Name	Full Name	Description
00	GRGa	Ge	16 HPGe Clovers Low Gain
01	GRGb	Ge	16 HPGe Clovers High Gain
02	SEP	Beta	20 plastic scintillators (SCEPTAR)
03	DAN	DANTE	8 LaBr3 scintillators Energy signals
04	DAT	DANTE	8 LaBr3 scintillators TAC signals
05	PAC	PACES	5 SiLi for ICE
06	DSC	DESCANT	70 Scintillators for neutrons
07	GRS	Suppressors	16 sets of BGO for clovers
08	DAS	Suppressors	8 sets of BGO for DANTE
09	SET	Beta	Zero-Degree scintillator (ZDS)
10	DSC	DESCANT (Gamma)	Gamma identified by pulse shape
11	DSC	DESCANT (Neutron)	Neutron identified by pulse shape
12			Undefined
13			Undefined
14			Undefined
15			Scaler

Table 5.4: Detector types used in GRIFFIN data events originating in the GRIF-16 or GRIF-4G digitizers.

ID	Description
00	PPG event
01	Undefined
02	Undefined
03	Undefined
04	Undefined
05	Undefined
06	Undefined
07	Undefined
08	Undefined
09	Undefined
10	Undefined
11	Undefined
12	Undefined
13	Undefined
14	Undefined
15	Undefined

Table 5.5: Detector types used in GRIFFIN data events originating in the GRIF-C digitizers.

PPG Pattern	Description
0xc000	This one doesn't turn the beam on.

Table 5.6: PPG patterns used in GRIFFIN PPG events.

Data item name	GRSISort data member name
Module Type	`DataType`
Address	`ChannelAddress`
Detector Type	`DetectorType`
Network Packet Counter Value	`NetworkPacketNumber`
Filter Patterns	`TriggerBitPattern`
Filter Condition Counter Value	`TriggerId`
Channel Trigger Counter Value	`ChannelId`
Timestamp low bits	`TimeStampLow`
Timestamp high bits	`TimeStampHigh`
Deadtime	`DeadTime`
Channel Accepted Counter Value	`AcceptedChannelId`
Pile-Up Type	`NumberOfPileups`
Waveform data	`wavebuffer`
Integration Length	`KValue`
Pulse Height Signal	`Charge`
CFD Signal	`Cfd`
Short Charge Integration	`ccShort`
Long Charge Integration	`ccLong`
Zero-Crossing Time	`Zc`

Table 5.7: GRSISort data member names associated with data items.

Table 5.8: GRIFFIN GRIF-16 fragment event format, bank GRF4. Format is based on 4-byte (32-bit) words. The number of bits for each sub-word is indicated in brackets. Words with an asterisk might be repeated within a single fragment event.

I	Packet Type (4) **0x8**	Module Type (3)	Word Count (5)	Address (16)			Det. Type (4)	4 bytes
II	Packet Type (4) **0xd**	Network Packet Counter Value (28)						4 bytes
III	Fixed 00 (2)	Filter Patterns (14)			Waveform Ind. (1)	reserved (10)	Pile-up type (5)	4 bytes
IV*	Fixed 0 (1)	Filter Condition Counter Value (31)						4 bytes
V	Packet Type (4) **0x9**	Channel Hit Detection Counter Value (28)						4 bytes
VI	Packet Type (4) **0xa**	Time Stamp low bits (28)						4 bytes
VII	Packet Type (4) **0xb**	Deadtime (14)			Time Stamp high bits (14)			4 bytes
VIIa*	Packet Type (4) **0xc**	Waveform Sample (28)						4 bytes
VIII	Fixed 0 (1)	Int. Len. high bits (5)	Pulse Height (26)					4 bytes
IX	Fixed 0 (1)	Int. Length low bits (9)		CFD (22)				4 bytes
X	Fixed 0 (1)	reserved (7)	Num. hits (8)		reserved (2)	Int. Len. 2 (14)		4 bytes
XI	Fixed 0 (1)	Pulse Height 2 (31)						4 bytes
XII	Fixed 0 (1)	reserved (1)	Int. Len. 4 (14)		reserved (2)	Int. Len. 3 (14)		4 bytes
XIII	Fixed 0 (1)	Pulse Height 3 (31)						4 bytes
XIV	Fixed 0 (1)	Pulse Height 4 (31)						4 bytes
XV	Packet Type (4) **0xe**	Channel Accepted Hit Counter Value (14)			Event Trailer - repeated Channel Hit Detection Counter Value low bits (14)			4 bytes
							Total one hit:	36 bytes

Table 5.9: GRIFFIN GRIF-16 fragment event format, bank GRF3. Format is based on 4-byte (32-bit) words. The number of bits for each sub-word is indicated in brackets. Words with an asterisk might be repeated within a single fragment event.

I	Packet Type (4) **0x8**	Module Type (3)	Word Count (5)	Address (16)			Det. Type (4)	4 bytes
II	Packet Type (4) **0xd**	Network Packet Counter Value (28)						4 bytes
III	Fixed 00 (2)	Filter Patterns (14)			Waveform Ind. (1)	reserved (10)	Pile-up type (5)	4 bytes
IV*	Fixed 0 (1)	Filter Condition Counter Value (31)						4 bytes
V	Packet Type (4) **0x9**	Channel Hit Detection Counter Value (28)						4 bytes
VI	Packet Type (4) **0xa**	Time Stamp low bits (28)						4 bytes
VII	Packet Type (4) **0xb**	Deadtime (14)			Time Stamp high bits (14)			4 bytes
VIIa*	Packet Type (4) **0xc**	Waveform Sample (28)						4 bytes
VIII	Fixed 0 (1)	Int. Len. high bits (5)	Pulse Height (26)					4 bytes
IX	Fixed 0 (1)	Int. Length low bits (9)		CFD (22)				4 bytes
X	Packet Type (4) **0xe**	Channel Accepted Hit Counter Value (14)			Event Trailer - repeated Channel Hit Detection Counter Value low bits (14)			4 bytes
							Total one hit:	36 bytes

Table 5.10: Old GRIFFIN GRIF-16 fragment event format, bank GRF2. Format is based on 4-byte (32-bit) words. The number of bits for each sub-word is indicated in brackets. Words with an asterisk might be repeated within a single fragment event.

I	Packet Type (4) **0x8**	Pile-Up Type (2)	Module Type (3)	Num. Filter Patterns (3)	Address (16)	Det. Type (4)	4 bytes
II	Packet Type (4) **0xd**	Network Packet Counter Value (28)					4 bytes
III	Fixed 00 (2)	Filter Patterns (14)			PPG pattern (10)		4 bytes
IV*	Fixed 0 (1)	Filter Condition Counter Value (31)					4 bytes
V	Packet Type (4) **0x9**	Channel Hit Detection Counter Value (28)					4 bytes
VI	Packet Type (4) **0xa**	Time Stamp low bits (28)					4 bytes
VII	Packet Type (4) **0xb**	Deadtime (14)			Time Stamp high bits (14)		4 bytes
VIIa*	Packet Type (4) **0xc**	Waveform Sample (28)					4 bytes
VIII	Fixed 0 (1)	Int. Len. 1 high bits (5)	Pulse Height 1 (26)				4 bytes
IX	Fixed 0 (1)	Int. Len. 1 low bits (5)	CFD 1 (26)				4 bytes
X	Fixed 0 (1)	Int. Len. 2 high bits (5)	Pulse Height 2 (26)				4 bytes
XI	Fixed 0 (1)	Int. Len. 2 low bits (5)	CFD 2 (26)				4 bytes
XII	Fixed 0 (1)	Int. Len. 3 high bits (5)	Pulse Height 3 (26)				4 bytes
XIII	Fixed 0 (1)	Int. Len. 3 low bits (5)	CFD 3 (26)				4 bytes
XIV	Packet Type (4) **0xe**	reserved (14)			Event Trailer - repeated Channel Hit Detection Counter Value low bits (14)		4 bytes

Total one hit (words I-IX,XIV): 36 bytes
Total two hits (words I-XI,XIV): 44 bytes
Total three hits (words I-XIV): 52 bytes

Table 5.11: Old GRIFFIN GRIF-16 fragment event format, bank GRF1. Format is based on 4-byte (32-bit) words. The number of bits for each sub-word is indicated in brackets. Words with an asterisk might be repeated within a single fragment event.

I	Packet Type (4) **0x8**	Num. Filter Patterns (3)	Module Type (3)	Pile-Up Type (3)	Address (16)	Det. Type (4)	4 bytes
II	Packet Type (4) **0xd**	Network Packet Counter Value (28)					4 bytes
III	Fixed 00 (2)	Filter Patterns (14)			PPG pattern (10)		4 bytes
IV*	Fixed 0 (1)	Filter Condition Counter Value (31)					4 bytes
V	Packet Type (4) **0x9**	Channel Hit Detection Counter Value (28)					4 bytes
VI	Packet Type (4) **0xa**	Time Stamp low bits (28)					4 bytes
VII	Packet Type (4) **0xb**	Deadtime (14)			Time Stamp high bits (14)		4 bytes
VIIa*	Packet Type (4) **0xc**	Waveform Sample (28)					4 bytes
VIII	Fixed 0 (1)	Int. Len. high bits (5)	Pulse Height 1 (26)				4 bytes
IX	Fixed 0 (1)	Int. Len. low bits (5)	CFD 1 (26)				4 bytes
X	Fixed 0 (1)	Int. Len. high bits (5)	Pulse Height 2 (26)				4 bytes
XI	Fixed 0 (1)	Int. Len. low bits (5)	CFD 2 (26)				4 bytes
XII	Fixed 0 (1)	Int. Len. high bits (5)	Pulse Height 3 (26)				4 bytes
XIII	Fixed 0 (1)	Int. Len. low bits (5)	CFD 3 (26)				4 bytes
XIV	Packet Type (4) **0xe**	reserved (14)			Event Trailer - repeated Channel Hit Detection Counter Value low bits (14)		4 bytes

Total one hit (words I-IX,XIV): 36 bytes
Total two hits (words I-XI,XIV): 44 bytes
Total three hits (words I-XIV): 52 bytes