

ANL Digitizer Firmware User's Manual ANL firmware for the LBNL digitizer module This document is for general users, not experts.

-- PRELIMINARY --

November 3, 2022 Version 1.0

Originator: J. Anderson Document #TBA

Table of Contents

TABLE OF FIGURES	2
INTRODUCTION	3
A SHORT INTRODUCTION TO HARDWARE Clocking and Timestamps Connection to external trigger system	3 3 4
GENERAL DESIGN OF THE FIRMWARE	4
Event DATA NOMENCLATURE Discriminator modes Pileup and discriminator hold-off Diagnostic Counters CHANNEL DESIGN OVERVIEW CHANNEL INTERACTIONS CHANNEL INTERACTIONS CHANNEL PIPELINE STRUCTURE READOUT DATA FORMAT Data Header – leading-edge discriminator mode. Data Header – constant fraction discriminator mode. WAVEFORM DATA FORMAT DETAILS	
ENERGY SUMMATION	13
Comparison of delays and sums versus the standard trapezoidal filter	13
PEAK DETECTION AND PILEUP REJECTION	14
Pileup Detection Logic Peak Detector Pileup and Event Readout Readout Interference	15 16 16 16
INTERFACE TO THE EXTERNAL TRIGGER	17
TIMESTAMP SYNCHRONIZATION EVENT VETO (TYPICALLY SPECIFIC TO GAMMASPHERE) How Triggers Work at Gammasphere and other ATLAS detector setups Details of the trigger window calculation Information Sent to the Trigger System by the Digitizer Data Elow Control (Tupotter)	
DATATLOW CONTROL (THROTTLE)	20

Table of Figures

FIGURE 1 - OVERALL BLOCK DIAGRAM OF DIGITIZER MODULE.	3
FIGURE 2 - 10 CHANNEL DATA FLOW DIAGRAM.	7
FIGURE 3 : PIPELINE STRUCTURE OF CHANNEL SHOWING PROGRESSION OF WAVEFORM WITH TIME.	8
FIGURE 4 - SUMMARY DESCRIPTION OF ALL FLAG BITS WITHIN THE HEADER.	9
FIGURE 5 - SIMULATION SHOWING OPERATION OF PILEUP COUNTER.	15
FIGURE 6 - EXAMPLE WAVEFORM FOR DESCRIBING PILEUP READOUT MODES	17
FIGURE 7 - TRIGGER WINDOWS WITHIN THE DIGITIZER	20

Introduction

This document is intended for users of the ANL firmware as implemented in the 10-channel LBNL digitizer module who will be working with the firmware and adjusting parameters. This document glosses over many details of the firmware to present a view sufficient for non-expert users. Experts are referred to the document *ANL Digitizer Firmware for Experts* for details.

A short introduction to hardware

To fully understand how the firmware works the hardware environment must be explained. The digitizer module contains two FPGAs, one for VME interface and firmware maintenance purposes, and the other for ADC data processing, as shown in Figure 1.



Figure 1 - overall block diagram of digitizer module.

Clocking and Timestamps

The ADC chips run at a frequency of 100MHz (100 MSamples/sec). The device may run from its own internal oscillator or may receive a clock from an external trigger system and synchronize to that clock. A clock distribution from the trigger allows many digitizer modules to run synchronous to each other.

The digitizer module uses a 48-bit timestamp counter run from the 100MHz clock. When connected to the trigger system, the trigger may issue a command to synchronously reset the timestamp in all digitizers. Events read out of the digitizer are tagged with a timestamp for sorting and analysis.

Connection to external trigger system

Digitizer modules may connect to a trigger system for synchronization and event selection purposes through the use of a fiber optic interface. The digitizer can run standalone without being connected to a trigger system. In this configuration every digitizer board has independent and unsynchronized clocks. Timestamps for event sorting are not useful across multiple modules as there is no synchronism, but within one board timestamps are still sensible.

General Design of the Firmware

The ANL version of digitizer firmware implements ten independent data acquisition pipelines in ten channels. Pileup detection & rejection is performed locally within the digitizer. Selective readout of events is achieved using an external trigger system. Waveform readout of all events is possible, with up to 1024 ADC samples per channel per event. A down-sampling mode allows readout of averaged samples where each waveform sample in the readout may be the average of 2ⁿ ADC samples, from 2 to 128. Increased event rates are obtained by limiting the amount of waveform data read out per event.

Each channel pipeline consists of a series of memory buffers used for delay. Discriminator logic recognizes edges within the input signal and causes data values to be sampled. When sampling occurs, the timestamp value is saved. The discriminator logic may be configured as either leading-edge (slope) or constant-fraction architecture, sensitive to either positive-only, negative-only or both edges. The discriminator implements a programmable hold-off time to ensure that the discriminator fires only once per edge. A discriminator firing captures timing and data sums simultaneously. This data is buffered so that discriminator rearms very quickly. Discriminator firing rates over 1MHz can be supported. No information reduction occurs in pileup conditions due to the fast discriminator recovery time.

All sampled values and sums are formatted into an event header that is followed by a programmable number of waveform samples. The header contains various fields, including the timestamp of the event, the timestamp of the last time the discriminator of that channel fired, the timestamp of the peak, energy information, plus some energy/time information carried over from the previous discriminator firing. Flag bits provide useful diagnostics. The waveform data contains the raw ADC samples of the event, plus serialized timing mark bits that indicate when specific actions (discriminator, peak, timeouts) occurred and a 2nd bit that indicates whether samples are down-sampled (rescaled average of 2**n samples) or full-speed.

Energy measurement is performed in a double-correlated method timed relative to the moment the discriminator fires. Programmable delay buffers positioned to measure ranges of time before (pre-rise) and after (post-rise) the discriminator logic have accumulators continuously calculating the sum of all the ADC samples within them. When the discriminator fires these sums are saved and reported in the header. These sums may then be used by software to calculate the energy (amplitude) of the input signal, with all necessary baseline and pole-zero correction information obtained from the other information in the header. This

method is arithmetically identical to the traditional "trapezoid" method but is optimized for high rates of discriminator operation.

Hits captured by the discriminator may optionally be rejected if pileup occurs. If piledup events are allowed the piling-on events may be read out in a variety of ways including extended and offset waveforms or just additional headers. The reverse logic is also supported, in which *only* piled-up events are available for readout. The firmware interfaces with an FPGAbased trigger system, also designed at ANL, to provide event selection based upon programmable trigger conditions. Specific triggering modes appropriate to the different detector systems at ANL have been developed.

Event data nomenclature

Signal edges marked by discriminator firings are named discriminator hits.

- *Discriminator hits* become *accepted hits* after passing through pileup rejection logic.
- The firmware may run by itself ("internal accept all" mode) or with the trigger system ("TTCL mode").
 - When running in "internal" mode *accepted hits* immediately become *accepted events* that are later read out.
 - When the trigger system is used *accepted hits* wait in a queue for a limited time. The *accepted hits* in this queue only become *accepted events* if selected by a *trigger accept message* from the trigger. Accepted hits not marked as accepted events fall off the end of the pipeline and are discarded.
- The expectation is that all *Accepted Events* will be read out. *Accepted Events* that were unable to be copied into the output buffer due to FIFO backup or other readout interference are named *dropped events*.
 - When using the trigger system, the digitizer may assert a *Throttle Request* to the trigger if its FIFO is getting too full. This request, if honored, will result in the trigger suspending the issuance of trigger accept messages so that the readout system may drain the FIFO forming an automatic flow control mechanism to avoid *dropped events*.

Discriminator modes

The firmware supports either *leading-edge* or *constant-fraction* discriminator logic, independently selected per channel. In *leading-edge* mode the operation is controlled by a delay value 'd' and a threshold such that the discriminator fires if the difference between sample X(n) and sample X(n-d), after some filtering, is greater than the threshold. In *constant-fraction* mode, the user specifies a fraction value along with the same timing parameter 'd'; the discriminator logic continuously calculates [X(n)*fraction] - X(d), and fires when this signal recrosses the initial value it had before the start of the edge.

A 2nd, separate copy of the *leading-edge* discriminator, with its own threshold, called the *coarse* discriminator is also provided. The *coarse* discriminator fires well before the main

discriminator does as it sees the edge earlier. The output of this device is used to capture early energy sum values well before the edge occurs, for use in pole-zero and/or baseline correction software after data is collected. Figure 3, later in this document, will explain these terms more fully.

Pileup and discriminator hold-off

- When pileup rejection is **on**, discriminator hits become accepted hits only if there is no pileup.
- When pileup rejection is *off, discriminator hits* always become *accepted hits,* but differentiation between the *first* accepted hit and all that pile up upon it (a "pileup train") is needed.
 - The *first* hit of a pileup train is still called the *accepted hit* but all *subsequent* hits within a pileup train are called *extended events*.
- The *pileup inspection time* is the minimum separation between *discriminator hits* before pileup is declared to exist.
 - The firmware can withstand a maximum of 16 events piled up on top of each other before the pileup inspection time of the first event expires. Should this be exceeded the digitizer enters the *General Error* state and all data processing within the channel stops until the digitizer is reset.
- Each channel implements a *discriminator hold-off* time that is the amount of time after a *discriminator hit* that the discriminator is blocked from marking another hit. This is related to the rise time of the signal so that only one *discriminator hit* is marked per rise.
 - If the user sets the *discriminator hold-off* time to a value less than the *pileup inspection time*, the digitizer will enter the *Pileup Too Short* error state and must be manually reset with adjusted parameters.

Diagnostic Counters

A primary diagnostic by which operation of the firmware may be monitored is by perchannel counters that monitor *discriminator hits, accepted hits, accepted events* and *dropped events.* Discriminator setup errors or analog input issues often result in no *discriminator hits.* If pileup rejection is enabled the ratio of *accepted hits* to *discriminator hits* provides immediate feedback regarding the percentage of events that are piled up. When using the "TTCL" mode (selection of events by the trigger) the ratio of *accepted events* to *accepted hits* may be used to determine if the trigger settings and/or digitizer trigger time windows are set appropriately. Recording any *dropped events* at all typically indicates that the amount of waveform data requested per event is too large for the event rate and that the readout of the digitizer can't keep up with the input data.

Channel design overview

The 10 channels of the firmware are implemented as continuously running data pipelines. Data enters the pipeline every tick of the clock from the ADC, proceeds through the pipeline and eventually falls off the back end. If conditions are satisfied data samples at the back end are copied into a buffer known as the *Channel FIFO*. In the middle of the pipeline

multiple delay buffers with accumulators provide running sums of the data spanning programmable ranges of time. Other delay buffers in the pipeline provide delays for calculation of discriminator functions. When a *discriminator hit* occurs, various time values and data sums are stored as a *header*. *Headers* are stored in an *Event Header FIFO* that holds the headers so long as the associated waveform data is still in pipeline.



Figure 2 - 10 channel data flow diagram.

Channel interactions

When an event is selected for readout, the *header* from the *Event Header FIFO* is combined with the matching set of ADC data samples from the *Channel FIFO*, and this package of data is sent to the board-wide *Accepted Event FIFO* that services all 10 channels. The *Accepted Event FIFO* is drained by a state machine that copies the data to the FIFO memory external to the FPGA unless the external FIFO cannot be written to because it is too full; in that case the data is lost and counted as a *dropped event*. Because there are 10 *Channel FIFOs* that fill simultaneously from all 10 channels but only one *Accepted Event FIFO*, large event sizes combined with sufficient rate may result in *dropped events* because the channels block each other from access to the *Accepted Event FIFO*.

Channel Pipeline Structure

The pipeline structure found in every channel is shown as Figure 3. The signal enters at the left and propagates to the right with time, backwards of a traditional oscilloscope image. When the edge of the signal reaches the middle, where the discriminator logic lies, the

discriminator fires, capturing all the sum values shown plus timestamp and status information. This is all read out in the header of the data as explained in the next section.



Note the discriminator in the middle of the design across buffer 'd', and the coarse discriminator placed well earlier in the pipeline.

Figure 3 : Pipeline structure of channel showing progression of waveform with time.

Readout Data Format

The channel pipelines have two basic modes of operation, Leading-

Edge Discriminator and **Constant-Fraction Discriminator.** The latter is often abbreviated "CFD", but the similar contraction "LED" is easily confused with the identical abbreviation for *light emitting diode*. Because of this the abbreviations shall not be used. Data in both modes reads out as a packet consisting of *header* followed by *waveform*. VME data readout is always 32-bit words. A *header type* value in the *header* indicates which mode the channel was in when the data was taken. A *header length* field indicates how many 32-bit words are in the *header*, and a separate *packet length* field states the total length of *header* plus *waveform*.

Data Header – leading-edge discriminator mode

The header format for the ANL digitizer, for the leading-edge discriminator mode, is shown in Table 1. This table reflects the format for the August 2021 release. All data read from the digitizer is 32 bits wide, and the header is 14 words long.

																						<u> </u>										
										L	EADI	NG ED	GE D	ISCRI	MINA	TOR	DATA	FORN	1AT - 1	HEAD	DER TY	PE IS	7									
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															FIXE	D 0xA	AAA	AAAA														
1		Ge	eo A	ddr						PACK	ET LE	NGTH									USE	R PAC	KET C	ATA)				CHAN	NELI	D
2												LEA	DING	EDG	E DISC	RIMI	NATO	R TIN	ESTA	MP[3	1:0]											
3		HE	ADE	R LENG	GTH		EV	ENT T	YPE	CEM	TTS	PBYP	Н	EADE	R TYP	Έ				LEA	DING	6 EDGE	DISC	RIMIN	IATO	R TIM	ESTAI	MP[4	7:32]			
4			TIM	ESTAN	1P OF	PREV	IOUS	leadi	NG E	DGE D	SCRI	MINA	TOR[1	5:00]			PF	PO	GE	SE	CV	OF	PV	ED	TSM	VF	WF	PTE	0	0	0	0
5											TIME	STAM	P OF I	PREV	OUS	LEADI	NG E	DGE D	ISCRI	MINA	TOR[47:16]										
6	0	0	0	0	Р	ILEUP	COUI	NΤ										SA	MPLE	D BA	SELIN	IE[23:	00]									
7				D	ETECT	OR D	ATA F	ROM	TRIGO	GER (Ta	arget	Whee	el)								EXTR	A DAT	A FR	DM TF	RIGGE	R (Mu	ltipli	cities)			
8			POS	T RISE	SUM[07:00]												PRE	RISE	SUM[23:0]										
9						TIN	NESTA	MP C	F PEA	\K[15:	00]												POST	RISE	SUM[:	23:08]						
10						TR	IGGE	RTIM	STAN	/IP DA	TA						CPTS	P2M	_					P2	2_SUN	/[13:0	00]					
11		PREVI	IOUS	_HIT_	POST	RISE[23:16]										MUL	TIPLE	X_DA	TA_F	IELD[2	3:00]									
12		PREVI	IOUS	_HIT_	POST	RISE[15:08]										EARL	Y_PRE	_RISI	E_EN	ERGY[23:00]									
13		PREVI	IOUS	HIT	POST	RISE[07:00]			TIME	STAM	P_OF	COA	RSE[0	9:00]			CF	PCV	PTSN	2DF				P2	2 SUM	v[23:	14]			

Table 1 - format of channel header, leading-edge discriminator mode

Many items are clear from the names, but a few entities should be explained.

- The *Header Type* field is a four-bit code indicative of which format of header this is. The firmware of August 2021 uses header type 7 for LED information and header type 8 for CFD information. Values of 0 through 6 were used by earlier, now-deprecated versions of the firmware.
- The *Event Type* field is a three-bit field that indicates which trigger system algorithm selected this event if the digitizer is being run in TTCL mode. If the digitizer is being run in Internal mode, the *Event Type* is "000".

Many flag bits are found within the header. A summary description of the flag bits is provided in Figure 4.

FLAG NAME	MEANING	USAGE
TTS	G_TS_MODE. if 0, the TRIGGER_TIMESTAMP_DATA field is the timestamp when the trigger message arrived. If 1, it is the timestamp of the messa	
PBYP	If 0, the digitizer is in TTCL mode. If 1, the PEQ is bypassed and the digitizer is in Internal-Accept-All mode.	
PF	Pileup Flag; event was piled up.	
PO	Pileup Waveform Only. User has only allowed readout of piled-up waveforms.	
GE	General Error. A general internal error has occurred and the firmware should be reloaded.	
SE	Sync Error. The digitizer reports a timestamp synchronization error.	
OF	Offset Flag. This data is an Extended Event whose readout if offset due to readout interference.	
PV	Peak Valid. The peak-sensing logic has found a peak in this event, so the peak timestamp is valid.	
ED	External Discriminator Flag. This event was caused by external discriminator, not internal leading edge discriminator logic.	
VF	Veto Flag. If digitizer were enabled to process router vetoes this event would have been vetoed.	
WF	Write Flags. 0:ADC data is 14 bit with flags. 1: ADC data is 14.2 format, no flag bits.	
PTE	Pileup Time Error flag. User has set illegal combination of holdoff/pileup values Digitizer must be completely reset.	
P2M	P2 Buffer Mode. 0:P2 length set by reg_p_window. 1:Length of (P2+Post) set by M, P2 set by reg_p_window, Post length is 'm'-'p'.	
CF	Coarse Fired. Bit is set if the coarse discriminator fired for this event.	
PTSM	Preamp Reset TS Match. If set bits 47:28 of timestamp of last preamp reset match bits 47:28 of current timestamp.	
2DF	2nd threshold discriminator flag. Set if 2nd threshold was crossed before holdoff time elapsed.	
CPTS	Inidicates mode of the multiplex field. 0: field is 2nd early pre-rise sum. 1: field is timestamp of last preamp reset.	
PCV	Previous CFD Valid. Carry-forward of CV bit from the previous discriminator firing of this channel.	CFD ONLY
CEM	CFD Esum Mode. 0: capture pre- and post-rise energy when the CFD fires. 1: Capture energy sums using delayed copy of LED instead.	CFD ONLY
CV	FD Valid Flag. 1: CFD OK. 0: CFD samples are invalid and timestamp is timestamp of pre-arming leading edge discriminator. Always 0 in LED mode	CFD ONLY
TSM	Time Stamp Match. 1: bits 37:30 of previous CFD match bits 47:30 of this event. 0: upper TS bits do not match. Always 0 in LED mode.	CFD ONLY

Figure 4 - Summary description of all flag bits within the header.

In addition to the *header type, event type* and flags, many different data fields provide sufficient summary data for each event to perform timing, energy analysis, pole-zero correction, and many other functions without need to read any waveform data whatsoever. These can be grouped into categories:

- **Event/device identifier** fields such as the User Packet Data, the channel ID and the Geo Addr; these are used for data sorting.
- Multiple *sums* captured from the accumulators that span the P2, Post-rise and Prerise buffers are provided for energy measurement.
- **Timestamps** allow for correlation of this event to other events, both with respect to this channel and all other channels in a system of multiple digitizers synchronized by the trigger. The timestamp of the peak detector is also recorded to simplify analysis of events with variable rise time.
- **Trigger Information** tags events with system information known only to the trigger at the time of the event such as target wheel rotational position or system-wide multiplicity sums.

Event/device identifier data

- The **Geo Addr** field provides the slot number of the VME crate the digitizer is in.
- The **Packet Length** field lists the size of the total event, inclusive of the header.
- The **User Packet Data** field is free for the user to fill in with whatever data is desired, such as a detector identifier.
- The **Channel ID** field identifies which channel the data came from.
- The **Header Length** field states how many words of the packet are "header", with the remainder of the **Packet Length** defined as "waveform".
- The **Event Type** field identifies which trigger system algorithm selected this event for readout, if the digitizer is set to use trigger accepts. If the digitizer is set to accept all hits as events ("Internal Accept All" mode), the **Event Type** is always 0.
- The **Header Type** field is the version number of the header. For August 2021 firmware this will always be 7 (LED mode data) or 8 (CFD mode data).

Sum data

The various sum fields contain the sum of a contiguous set of ADC samples captured at specific timing relative to the discriminator hit and are used for energy spectra. Please refer to Figure 3 for the names of the buffers when reading this list.

- The **Post-rise sum** is the sum of the data across the post-rise "M" buffer, captured at the time the discriminator fires.
- The **Pre-rise sum** is the sum of the data across the pre-rise "M" buffer, captured at the time the discriminator fires.
- The **P2 sum** is the sum of the data across the "P2" buffer, captured at the time the discriminator fires. This can be considered an extended post-rise sum that may be used if event separation is large enough.
- The **Early Pre-rise sum** is the sum of the data across the pre-rise "M" buffer, captured at a time prior to the edge reaching the discriminator area, and thus is an extension in the prior-to-rise akin to the P2 sum but in the other direction.
 - Some configuration modes allow for a 3rd sample of the pre-rise buffer even earlier than the **Early Pre-rise sum**. See the expert's manual for details.

• The **Sampled Baseline** is the sum of the data across a 10.24us long buffer captured at the time the discriminator fires. This buffer samples the data in the 10.24us prior to the **Pre-rise sum**.

Timestamp data

The various timestamp fields contain timestamp counts at which important items occurred. Please refer to Figure 4 for the names of the fields and flag bits when reading this list.

- The **Timestamp of Discriminator** is the 48-bit timestamp count when the main discriminator fires. This is the timestamp of the event.
- The **Timestamp of Previous Discriminator** is the timestamp of the last time the main discriminator of the channel fired, prior to the current event. This may be used along with Sum information for baseline extrapolation and/or pole-zero correction. Note that this timestamp is the timestamp of the most recent *discriminator* operation; in a system using triggers to select events that previous discriminator firing may not have been selected for readout.
- The **Timestamp of Peak** is the lower 16 bits of the 48-bit timestamp when the peak detector logic found a peak. The difference between the discriminator timestamp and the peak timestamp may be used by sorting software to separate events by rise time.
 - If the PV (Peak Valid) flag bit is not set in the header, the Timestamp of Peak is invalid as the peak detector did not find a peak before the discriminator holdoff time elapsed.
- The **Timestamp of Coarse** is the lower 10 bits of the 48-bit timestamp latched when the coarse discriminator fired. This is provided to allow correct interpretation of the Early Pre-Rise sum(s).
- The **Multiplex Data Field** <u>may</u> contain bits 27:4 of the system timestamp, captured when the preamp reset logic last detected a preamp reset condition.
 - The 24-bit value spans 2²⁸ clock ticks at 10ns per tick, or 2.68 seconds, with a resolution of 160ns.
 - The **PTSM** bit, if set, indicates that bits 47:28 of the preamp reset timestamp also match bits 47:28 of the discriminator timestamp. This ensures that the full 48-bit timestamp span is used.
 - The **Multiplex Data Field** should be interpreted as the timestamp of last preamp reset *only if the CPTS flag bit is 1.*
- The **Trigger Timestamp Data** field provides the lower 16 bits of a timestamp associated with a trigger acceptance message that selected the event for readout.
 - If the **TTS** flag bit is 0, the Trigger Timestamp is the timestamp within the digitizer at the time the acceptance message *arrived; this is only used by engineers.*
 - If the TTS flag bit is 1, the Trigger Timestamp is the lower 16 bits of the 48-bit timestamp that was contained within the trigger accept message, not the timestamp at which the message arrived.
 - The timestamp within the message is always the timestamp used for event acceptance.

Trigger information

The trigger information fields contain information from the trigger system that provide additional event characterization. Please refer to Figure 4 for the names of the fields and flag bits when reading this list.

- The **Detector Data From Trigger** contains a data value indicating the address source for each of the three lookup tables (*Sweep_RAM, Trig_RAM and Veto_RAM*) within the trigger, plus the current address of the *Sweep_RAM*. This is usually indicative of the rotational position of the rotating target wheel.
- The Extra Data From Trigger contains the 8-bit multiplicity totals from trigger virtual planes X and Y at the time the trigger message was sent. Typically, in Gammasphere these would be the "clean" and "dirty" multiplicity sums, providing an indication of how many detectors participated in this event. This value can be used to sort events by multiplicity at the time of the event.

Data Header – constant fraction discriminator mode

The header format for the ANL digitizer firmware when operated in constant-fraction mode is as shown in Table 2.

										CON	STAN	T FRA	стю	N DIS	CRIM	NATO	OR DA	TA FC	RMA	T - HE	ADEF	TYPE	IS 8									
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															FIXE	D 0xA	AAAA	AAAA														
1		Ge	eo Ad	ldr						PACK	ET LE	NGTH									USE	R PAC	KET D	ATA					C	HAN	NEL IC)
2											С	ONST	ANTE	RAC	TION	DISCR	IMIN/	ATOR	TIME	STAM	P[31:	0]										
3		HE	ADER	LENG	STH		EV	ENT T	YPE	CEM	TTS	PBYP	Н	EAD	ER TYP	E			C	ONST/	ANT F	RACT		DISCRI	MINA	TOR	TIMES	TAMP	[47:3	2]		
4		TIN	NESTA	AMP C)F PRE	EVIOU	JS CO	NSTA	NT FR	ACTIC	DN DIS	CRIM	INAT	OR[1	5:0]		PF	PO	GE	SE	CV	OF	PV	ED	TSM	VF	WF	PTE	TRIG	_DET_D	ATA[15	:12]
5	TDD[:	11:10]						C	FD SA	MPLE	0						TDD	[9:8]	TIN	1ESTA	MP O	F PRE	VIOU	S CON	ISTAN	IT FRA	ACTIO	N DISC	CRIMI	NATO	R[29	:16]
6		٦	RIG_	DET_I	DATA	[07:00)]											SA	MPLE	D BA	SELIN	E[23:0	00]									
7	PILE_C	NT[32]						C	FD SA	MPLE	2						PILE_C	NT[3:2]						С	FD SA	MPLE	1					
8			POST	RISE	SUM[07:00]]												PRE I	RISE S	UM[2	3:00]										
9						TIN	NEST	AMP C	OF PEA	٨K[15:	:00]												POST	RISE	SUM[23:08]]					
10						TR	IGGEI	R TIM	ESTAN	VP DA	TA						CPTS	P2M						P2	2_SUN	/[13:0	00]					
11	I	PREVI	OUS_	HIT_P	POST_	RISE[23:16]										MUL	TIPLE	X_DA [.]	TA_FI	ELD[2	3:00]									
12	1	PREVI	OUS_	HIT_P	POST	RISE[15:08			_								EARL	/_PRE	_RISE	E_ENE	RGY[2	23:00]				_					
13		PREVI	OUS	HIT_P	POST_	RISE[07:00]			TIME	STAM	P_OF	_COA	ARSE[0	9:00]			CF	PCV	PTSN	2DF				P2	2_SUN	/[23:1	4]			

Table 2 - format of channel header, CFD build

The constant-fraction header has a few differences with respect to the leading-edge header:

- The constant-fraction header provides three CFD Sample values. These are signed objects that are the result of the subtraction of the digital constant-fraction equation calculated by the firmware. The constant-fraction equation is essentially a differentiation of the input pulse, so the logic looks for the crossover point from one sign to the other. CFD Sample 0 is the sample associated with the sign crossing, whereas CFD Samples 1 and 2 are the two previous samples of this subtraction taken 10ns and 20ns before CFD Sample 0.
 - a. These new fields take up space, requiring sacrifice of other information. The **Extra Data From Trigger** field of the LED header is sacrificed to make room.
 - b. Similarly the size of the **Timestamp of Previous Discriminator** field is cut back from a 48-bit number to a 30-bit number.
- 2. The **PCV**, **CEM**, **CV** and **TSM** bits, always zero in the LED header, have meaning in the constant-fraction header.

Waveform Data Format Details

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
Μ	D					AD	C D	AT/	A SA	٩M	PLE					М	D					AD	C D	AT/	A SA	٩M	PLE				

Tuble 5 Torniat of Waverorni data

When reading waveform data, the data sample in bits 13:0 is the earlier sample, with the data sample found in bits 29:16 having been captured one 10ns clock tick later.

The "M" bits are *timing mark* bits. These bits are set to indicate the samples associated with discriminator or peak detector firing. Timing marks are short serial sequences embedded within the data that mark when things of interest occurred. The "D" bits are *down-sample status change* bits. When reading down-sampled data, a "D" bit of '1' will indicate a change between down-sampled data and full speed data. Most users of down-sampling will choose to down-sample the entire readout, in which case the "D" bit may be ignored.

Energy Summation

The ANL digitizer firmware has no preconceptions regarding waveform shaping, polezero correction or baseline restoration. The two sums provided (Pre-Rise and Post-Rise), along with the other sums and timestamp information in the header of the data, is considered sufficient for external software to perform whatever system-specific corrections are needed. The raw, uncorrected energy of the pulse as shown in Figure 3 would simply be **(Post-Rise Sum)** – **(Pre-Rise Sum)**.

Comparison of delays and sums versus the standard trapezoidal filter

An unfortunate naming convention has reversed the names of the delay buffers in the firmware relative to the naming convention used in seminal documents such as *Digital synthesis of pulse shapes in real time for high resolution radiation spectroscopy* (Jordanov and Knoll, Nuclear Instruments and Methods in Physics Research A 345 (1994) 337-345). The integrating buffers in the firmware have been inadvertently named as M1 and M2 rather than K1 and K2. The delay amount between the integrating buffers of the firmware is the *sum* of multiple delay buffers named "k0", "k", "d", "d2" and "d3".

For comparison purposes <u>in this section alone</u>, the naming convention used in Jordanov and Knoll will be preserved. That is, the integration time ("m" in the firmware) will be called "k" for easier comparison with the language of Jordanov & Knoll, and the sum of all the intervening

delays between integration buffers ("k0" + "k" + "d" + "d" + "d" + "d" in the firmware) will be called "m".

Referencing figure 5 of Jordanov and Knoll, the accumulator output is given as

$$\sum_{0}^{k} \left[\left(X(n) - X(n-k) \right) \right] - \left[\left(X(n-m-k) - X(n-m-2k) \right) \right]$$

The ANL digitizer firmware continuously calculates both parts of this as separate sums

$$\sum_{0}^{k} [(X(n) - X(n-k))] \qquad \sum_{0}^{k} [(X(n-m-k) - X(n-m-2k))]$$

Both sums are sampled at the time the discriminator fires. Thus, instead of continuously calculating the Jordanov/Knoll equation and attempting to determine a peak or otherwise seemingly optimal value, instead the firmware samples the two halves simultaneously at the time of the discriminator firing with the "m" term of Jordanov and Knoll's equations defined by the sum of the firmware parameters "k0", "'k", "d", "d2" and "d3". Pole-zero compensation and calculation of the net energy is left to analysis software, using the parameters that are reported by the firmware in the event header in addition to these two partial sums. If waveform data is saved in addition to header data, the timing marks and full knowledge of all delay parameters allows the user to perform any other analysis desired in software after the event is collected.

Reporting the two separate partial sums allows for higher accuracy corrections in those cases where the pulse of interest has occurred very quickly (i.e. within the first time constant) after a previous pulse. The multiple sums taken prior to the rise of the signal along with the potential of using different sums (e.g. Post-Rise and/or P2) and the carry-forward of the Post-Rise from the previous discriminator firing provides the opportunity to calculate the different slope of the exponential decay each of the two parts are riding upon and perform both pole-zero and baseline corrections.

Peak Detection and Pileup Rejection

The firmware may either accept or reject pileup. Pileup is defined as a *discriminator hit* that occurs closer to another *discriminator hit* than the time spanned by the **Pre-Rise Sum, K and KO** buffers (pileup inspection window). This handles both piling-upon and piled-upon timing cases. The firmware can withstand up to 16 *discriminator hits* all occurring within the pileup inspection window. If pileup is allowed, many different readout options are available that handle pileup in different ways; these are discussed in detail in a later section of this document.

The peak detector and the pileup rejection logic are solely related to the operation of the leading-edge discriminator, irrespective of whether the firmware is being used in the leading-edge or constant-fraction mode. Since the constant-fraction discriminator cannot fire without first being pre-armed by the leading-edge discriminator, controlling the peak and pileup based only upon the leading-edge is sufficient.

Pileup Detection Logic

Pileup recognition is implemented by use of a delay equal to the settings ("m" + "k") that can hold multiple discriminator firing marks and a counter in a small state machine. The counter is incremented each time the discriminator fires. When the delayed copy of a discriminator firing falls out of the delay, the counter is decremented. Thus, the counter continuously tracks the number of pulses stacked up on top of each other over the pileup time. When the counter performs a decrement that makes it fall back to zero, this is indicative of either the end of a pileup train or a non-piled-upon pulse. Pileup is determined by whether the counter value exceeded 1 before falling back to zero. Figure 5 shows how this works.



The pileup counter in each channel is a four-bit counter. For exceptionally odd choices of the various delay settings (e.g. a value of "m" > 16 times the total pulse rise time, coupled with a low discriminator threshold) it is possible that more than 16 pulses may pile up across the summation buffer. In this exceptional case, since the pileup counter has overflowed, the channel pileup logic will seize up into a *pileup overflow* state. This state is detectable by reading status bits in the master status register, requiring software or user intervention to reset the channel logic and the pileup counter.

Peak Detector

The firmware implements a filtering value named *peak sensitivity* that is used by the peak detection state machine. The peak detection state machine is held idle until the leadingedge discriminator fires. After this, during the rise (fall) of the trace, the peak detection state machine continuously hunts for a new potential maximum (minimum) as the signal rises (falls). At some point the trace rolls over and the time between new potential peak detections increases. The *peak sensitivity* value is *the number of clocks after a potential peak without a new potential peak* before the peak is finally marked. The sensitivity setting is typically a small, but non-zero, number (most users choose 4). Due to this, the timestamp saved for the peak is typically offset from the first sample at the peak value by a few 10s of nanoseconds. The peak finding algorithm is not guaranteed to find a peak under all combinations of noise, pileup, holdoff and sensitivity setting. If no peak has been found when the hold-off timeout elapses, the header of the event is stored with no peak recorded. A flag bit in the header indicates whether the peak has been recorded and thus the timestamp of the peak is valid. If no peak is found, the peak timestamp reported is zero.

Pileup and Event Readout

The ANL digitizer handles pulse pileup at three levels. The first level of logic is at the channel level where events are accepted or rejected. If a channel is set to reject piled-up events neither the readout logic nor the trigger system knows that these pileup events occurred. However, the coarse discriminator will always report found edges irrespective of pileup settings. A second level of pileup control occurs at the readout machine level. This logic necessitates a bit of terminology. We define events leaving the channel logic and entering the readout logic as being either *accepted* or *extended*.

- When pileup rejection is on, all events proceeding to the readout logic are, by definition, not piled up. Such events are all flagged as "*accepted*" events.
- When pileup rejection is off, only the *first* event in a string of piled-up events, or any isolated and non-piled event, is *accepted;* all subsequent events in a piled-up string are marked as *extended* events.
- By definition *extended* events can only occur following an *accepted* event. The ACCEPTED_HIT signal is the source of the delayed signal EVENT_EXPIRED that removes events from being accepted for readout by an external trigger system. Thus the PEQ only contains the timestamps of *accepted* hits.

The third level of pileup control occurs based upon a pileup flag stored in the header of all *accepted* events. The readout machine may be set to only read out those *accepted* events that have the pileup bit set, resulting in *only piled up events showing up in the readout and all other events suppressed.*

Readout Interference

When the readout length of one event *overlaps* the data from the next event there is risk of *readout interference*. This situation usually arises when the selected readout length is

longer than the accumulation buffer depth. Figure 6 shows an example waveform highlighting the effects of pileup and selected waveform readout length. In Figure 6, four pulses cause four discriminator firings and, thus, four events. The timing between the firings, when compared to the depth of the summation buffer "m", causes events #2 and #3 to be piled upon one another. However, the waveform readout depth selected by the user causes the **readout** of event #1 to interfere with that of event #2. The lower portion of Figure 6 shows the events on a timeline showing when each event would ideally read out.

The firmware offers a variety of readout modes, described in the *Expert* manual, to cope with *Readout Interference*, but the best strategy of all is to understand that the header provides all the data needed by most experiments and that long waveform readout hinders high data rate. Thus, don't read out long waveforms except during initial setup!



Figure 6 - Example waveform for describing pileup readout modes

Interface to the external trigger

The ANL firmware implements a high-speed serial link between digitizer and trigger system. On the receive side, the digitizer receives the control stream from the trigger and synchronizes itself to the clock from the trigger and to the timestamp as broadcast by the trigger. The digitizer firmware responds to the command frames embedded within the data stream from the trigger to provide event selection mechanisms. Specific features of the digitizer firmware related to the triggering system are discussed in this section.

Timestamp Synchronization

Every two microseconds the trigger system broadcasts the system timestamp to all digitizers. If properly linked up, all digitizer timestamp counters will be synchronized to this value. A command to the trigger called *Imperative Sync* orders all digitizers to reset their timestamp counters to zero and resume counting synchronously with the release of the *Imperative Sync* command.

Diagnostic counters in the master trigger, the router trigger and the digitizers should be monitored to ensure the system is fully locked and synchronized. Errors are counted at all interconnections.

Event Veto (typically specific to Gammasphere)

Provision is made for an Event Veto response from the trigger for each channel, calculated by the trigger based upon the timing relationship between reported discriminator bits from different channels. Response to the Event Veto is enabled by register bits within the digitizer. If this function is enabled the *most recently entered value* in the Pending Event Queue is erased should the trigger assert the Event Veto bit for a given channel. This feature requires relatively tight timing, on the order of microseconds.

In the specific case of Digital Gammasphere, two channels of each digitizer are associated with the Ge Center and BGO Sum signals of a given detector. Digital Gammasphere detectors are intended to be run in "clean" (Ge hit without corresponding BGO), "dirty" (Ge and BGO) or "module" modes (Ge or BGO), but the hits are not coincident in time. By use of the Event Veto function, the Router modules of the external trigger may monitor the channel pairs, implement a coincidence time window, and issue commands to Veto the events that are "dirty". This erases the events from the digitizer's Pending Event Queue so that "dirty" events are no longer available to be selected for readout, resulting in only the "clean" Ge/BGO events surviving to be potentially selected.

How Triggers Work at Gammasphere and other ATLAS detector setups

The trigger collects discriminator bits from the digitizers and maps them as the user desires into two "virtual planes" named X and Y. These planes may be used to map channels to two planes of a dual-sided silicon strip detector or may be used to collect "clean" multiplicity and "dirty" multiplicity (Gammasphere). The trigger system may also receive inputs from external sources such as a target wheel, a NIM input from some auxiliary detector or even the trigger message stream from another digital data acquisition system. In these cases, various forms of coincidence logic may be applied to form triggers. NIM inputs and target wheel position may be used to create either trigger requests or vetoes that inhibit triggers.

In all cases, when a trigger condition is satisfied, the trigger sends out a message containing the system timestamp when the trigger condition was satisfied. This message is received by all the digitizers who may then use the timestamp value *contained within the*

message – not the timestamp when received – to select events for readout. When using signals or data from external systems the trigger provides timestamp offset calculations and delay buffers to compensate for time-of-flight differences or different pileup inspection times between detectors.

The trigger system supports multiple triggering algorithms and multiple algorithms may be simultaneously selected. Events within the Pending Event Queue of the digitizer selected by any trigger algorithm are read out; analysis of the trigger type code stored within the data header may then be used to sort events by which trigger selected them.

Details of the trigger window calculation

When a trigger acceptance message is received, all event timestamps currently valid in the digitizer are compared against the timestamp value contained within the trigger message. A bit of arithmetic is required. The timestamp reported by the trigger is the time at which the trigger module determined that a trigger algorithm was satisfied. If an event occurs at time T₀, the trigger will respond at a variable time T_{tf} later. When processing the events in the pending event queue the timestamps in that queue are the time of the leading edge (T₀).

Upon receipt of the trigger message, the timestamp of the *trigger message* is subtracted from the timestamp of each *discriminator hit* still available for readout within the digitizer. Under normal conditions this will always yield a negative number. The subtraction answer is compared against a pair of time window registers (upper and lower) and if both comparisons are valid, the event is accepted for readout.

The method of subtraction used means that the window is defined *with respect to the time the Master trigger detected the appropriate condition* and *not* the time of the discriminator firing. That is, if there were an oscilloscope connected to the discriminator signal and also to the trigger signal and the oscilloscope were triggering on the signal from the trigger system, discriminator firings would be seen in a range of times prior to the trigger signal. Thus, the correct settings for the two window registers would also be *negative*, with the '*min* window' register set to the relative time of the earliest (farthest back from the trigger) discriminator firings. This apparent reversal of "min" and "max" is because the logic always requires that the "max" window edge be *later* in absolute time than the "min" position. See Figure 7.



Figure 7 - Trigger windows within the digitizer

The appropriate values for the time window registers are set based upon the time values programmed in the channel pipeline for energy integration and signal rise time. A typical calculation is the timestamp in the trigger acceptance message will be approximately "m"+"k"+"k0"+ 0.75us after the timestamp of the actual discriminator firing. The acceptance windows should then be set to -(m+k+k0) and -(m+k+K0+1.5us) to insure collection of the appropriate events.

Information Sent to the Trigger System by the Digitizer

In the ANL firmware each word every 20ns is a unique message. The data sent every 20ns is simply a snapshot of the state of every channel's discriminator bit plus the fast, or coarse, discriminator bits from channels 5-9. The selection of channels 5-9 for the coarse bits is driven by the specific cable plant of Digital Gammasphere, where channels 5-9 are connected to the germanium center contacts and channels 0-4 are connected to the BGO sum signals. By having the fast, coarse, Ge discriminator bits sent to the trigger system prompt multiplicity-based pre-triggers may be formed system-wide and distributed to auxiliary detectors promptly. As all channels are reported the Ge/BGO nomenclature is only a convenience for Digital Gammasphere, and any other detector system still has all the information from all channels.

The amount of time each discriminator bit is asserted is programmable to ensure multiplicity sums are properly calculated. Normally these discriminator one-shorts are set to ~120-150ns to ensure all detectors participate in every multiplicity trigger.

Data Flow Control (Throttle)

The digitizer hardware supports two non-serialized LVDS control signals that run from the digitizer to the router trigger board. One of these is defined as the Throttle bit. This bit is asserted by the firmware whenever the board-wide FIFO becomes sufficiently full. The trigger routers collect the Throttle bits from each digitizer, provide some time filtering to ensure very short duration requests are suppressed, and the Master Trigger may suspend the issuance of trigger accept messages while any Throttle bit from any Router is asserted. While in this state, readout software will drain the FIFOs, but no new data is added, so after the FIFOs are sufficiently drained triggers resume automatically.