

Miniball User DSP Code

Nigel Warr

October 2004

Contents

1 Foreword	2
2 Miniball pulse-shape analysis	2
3 The interface to user code	3
4 Assembling the code	3
5 Meaning of terms	3
6 Notation for number representation	4
7 Input	4
7.1 Baseline - MBBASELEN	5
7.2 Energy cut off - MBENCUT	5
7.3 Trigger threshold - MBTRGTHR	6
7.4 T_{90} threshold - MBT90THR	6
7.5 PSA control status register - MBPSACSR	7
7.6 PSA length and offset - MBPSALEN, MBPSAOFF	9
7.7 TFA Energy cut off - MBTFACUT	9
8 Output	9
9 Activating the User DSP code	9
10 Data format	9
10.1 Error codes	10
11 Example	10

1 Foreword

The implementation of pulse-shape analysis code on the DSP of the DGF is entirely the work of Martin Lauer from MPI Heidelberg. This document merely attempts to describe what he did.

Martin's code was written in ADSP 2181 assembler as user code for the DSP and linked with the main code provided, in binary form, by Xia.

2 Miniball pulse-shape analysis

The rising flank of a pulse from a Ge detector depends on the interactions between the detected radiation and the material making up the detector. If an interaction takes place close to an electrode the signal on that electrode will rise very quickly, whereas if the interaction takes place far away, it will take longer to reach its maximum. We define the quantity "time to steepest slope" which is a measure of when the majority of the charge carriers reach the electrode, and depends, therefore, on the distance between the electrode and the interaction. This is a general property for all detectors, but only medium to large volume detectors have a significant variation in this time.

For segmented detectors such as Miniball, we can get an additional piece of information. Suppose an interaction takes place near the central electrode, which for Miniball detectors is positively charged. Electron-hole pairs will be produced, but the electrons will reach the core electrode much sooner than the holes can reach the outer electrode because of the different distances involved. For a time, therefore, there will be a charge imbalance, because the electrons have already been absorbed at the core electrode, but the holes haven't yet reached the outer (segment) electrode. This induces a mirror charge on the adjacent segment electrodes until the holes are absorbed. This gives rise to a transient signal on the adjacent electrodes. The magnitude of this transient signal depends on how far the holes are from the adjacent segment. So by comparing the relative amplitude between the two transient signals on the neighbouring segments on either side, we can see which segment the interaction position was nearest to.

If the interaction takes place near the edge, the opposite effect occurs, namely that the holes are quickly absorbed and it is the electrons which produce mirror charges. The sign of the induced transient signal is therefore inverted, but the ratio of the absolute value of the magnitudes still tells us where the interaction took place.

The Miniball User DSP code was written to perform an on-line analysis of the signals from the Miniball detectors to determine the time to steepest slope for each core signal and the mirror charge amplitude for each segment signal. Note that we cannot actually calculate the ratio in the DGF because not all segments are on the same DGF.

3 The interface to user code

Xia provide five entry points to user routines:

- UserBegin - called when the DGF is initialised.
- UserRunInit - called when a run is started.
- UserEvent - called for each event.
- UserChannel - called for each event for each channel.
- UserRunFinish - called when a run is ended.

Note that UserBegin is always called, but the others are only called if a bit is set in MODCSRb and UserChannel is only called if an additional bit is set in the corresponding CHANCSRbx parameter.

Except for UserBegin which is called before control is passed to the user, these routines can read data from a sixteen word block called “USERIN” and write data to a second sixteen word block “USEROUT”. The UserChannel routine can also write to a six word block called “URETVAL” which the DSP automatically copies into the list mode data stream after NWORDS, TRIG-TIME and ENERGY in the long format.

4 Assembling the code

The code has to be written in ADSP 2181 assembler and linked to the precompiled object code supplied by Xia and assembled on an assembler written for MS-DOS.

Unfortunately, the ADSP 2181 seems to be rather obsolete now and there is not a lot of support. Furthermore, it seems that the assembler produced by Analog Devices is based on the gnu assembler, but they refuse to release source code, so it violates the Gnu Public License.

There is an open source assembler for linux, developed independently of Analog Devices, but it does not seem to understand the same dialect of assembler as the Analog Devices one.

5 Meaning of terms

T_0 - the time when the signal crosses the threshold set in “USERIN03” or “USERIN04”.

T_{slope} - the time to steepest slope.

Q_{max} - the absolute value of the mirror charge amplitude.

$T(Q_{max})$ - the time to reach the maximum of the mirror charge amplitude.

T_{90} - the time for the signal to reach the T_{90} threshold which is determined by “USERIN05”.

T_{50} - the time for the signal to reach half the value of the energy.

6 Notation for number representation

The DSP uses fixed point calculations, not floating point. This means that we have to specify the format for each real number. The notation N.M means that the first N bits represent the integer part of the number, and the remaining M bits represent the fractional part. The sum of N and M must always be 16, because the DSP uses 16-bit words.

The first of the fractional bits corresponds to 2^{-1} , the next to 2^{-2} and so on. In other words, if we have an 8.8 format, the first bit is 2^7 (128), the next 2^6 (64) and so on until 2^{-8} (1/256).

Alternatively, we can view the number as an integer that we have to divide by 256 to get the result.

The 16.0 format is the same as an integer.

Sometimes a single word is used to represent more than one value. In this case, we might have a representation like 8|8, which means that the 16 bit word consists of two 8-bit parts. For example, if we take MBATFACUT01, this means that $MBATFACUT01 = (MBATFACUT1 * 256) + (MBATFACUT0)$ where MBATFACUT0 and MBATFACUT1 are 8 bit numbers (i.e. 0 to 255) for channels 0 and 1, respectively.

7 Input

The user code uses two data structures in the DSP parameters “USERIN” and “USEROUT”, each of which is 16 words long. Note that XIA only name “USERIN” and “USEROUT” and leave the other 15 values in each structure un-named, whereas Martin calls them “USERIN00”, “USERIN01” ... “USERIN15” etc.

Important: the code reads from as well as writes to “USEROUT”! This means you have to be careful, because control tasks overwrite this memory. In other words, if you perform a control task, you have to reinitialise this block of memory before performing a normal run or it won’t work properly!

Table 7 shows the sixteen different parameters in the “USERIN” block and what they do. The PSA CSR values a sets of 16-bit flags whose meanings are shown in table 7.5. The other parameters are either four 4-bit nibbles or two 8-bit bytes. For example, the least significant four bits of USERIN00 correspond to “Baseline 0” and the next four bits to “Baseline 1”. In each case, they are

four bits wide, which means they can take values from 0 to 15. Similarly, for USERIN01, the eight least significant bits are “Energy cut off 0” and the other eight bits are “Energy cut off 1” and they can have values from 0 to 255.

It is sometimes convenient to give these values in hexadecimal, since this preserves the division of bits. i.e. one hex digit corresponds to four bits.

Parameter	Name	Meaning	Format
USERIN00	MBBASELEN	Baseline 0...3	4 4 4 4
USERIN01	MBENCUT01	Energy cut off 0, 1	8 8
USERIN02	MBENCUT23	Energy cut off 2, 3	8 8
USERIN03	MBTRGTHR01	Trigger threshold 0, 1	8 8
USERIN04	MBTRGTHR23	Trigger threshold 2, 3	8 8
USERIN05	MBT90THR	T ₉₀ threshold 0...3	4 4 4 4
USERIN06	MBPSACSR0	PSA CSR 0	16-bit flags
USERIN07	MBPSACSR1	PSA CSR 1	16-bit flags
USERIN08	MBPSACSR2	PSA CSR 2	16-bit flags
USERIN09	MBPSACSR3	PSA CSR 3	16-bit flags
USERIN10	MBPSALEN01	PSA length 0, 1	8 8
USERIN11	MBPSALEN23	PSA length 2, 3	8 8
USERIN12	MBPSAOFF01	PSA offset 0, 1	8 8
USERIN13	MBPSAOFF23	PSA offset 2, 3	8 8
USERIN14	MBTFACUT01	TFA energy cut off 0, 1	8 8
USERIN15	MBTFACUT23	TFA energy cut off 2, 3	8 8

Table 1: The meaning of the 16 words in “USERIN”

7.1 Baseline - MBBASELEN

The baseline value in “USERIN00” is a four-bit value per channel (i.e. it runs from 0 to 15). The value of 0 has a special meaning, which tells the user code to use the energy filter sum preceding the signal rise. Otherwise, it is used to shift the input by N bits (i.e. to average 2^N baseline samples).

Martin tells us to always set baseline > 0 . For example, set “USERIN00” to be 0x3333 (i.e. average over $2^3 = 8$ baseline samples for each channel).

7.2 Energy cut off - MBENCUT

The energy cut off is used to decide if there is a net charge deposited in a segment or only an induced mirror charge from a neighbouring segment. We consider there to be a direct deposition of net charge if:

$$E_{total} - E_{cut\ off} > 0 \tag{1}$$

For example, “USERIN01” and “USERIN02” could be 0x0505, which sets all four channels to have a cut off value of 0x05 (i.e. 5 decimal).

Parameter	Name	Meaning	Format
USEROUT00	MBXINT0	X_int_0	0.16
USEROUT01	MBXINT1	X_int_1	0.16
USEROUT02	MBXINT2	X_int_2	0.16
USEROUT03	MBXINT3	X_int_3	0.16
USEROUT04	MBDIFF0	X_diff_0	0.16
USEROUT05	MBDIFF1	X_diff_1	0.16
USEROUT06	MBDIFF2	X_diff_2	0.16
USEROUT07	MBDIFF3	X_diff_3	0.16
USEROUT08	MBAVG0	No. average 0	16.0
USEROUT09	MBAVG1	No. average 1	16.0
USEROUT10	MBAVG2	No. average 2	16.0
USEROUT11	MBAVG3	No. average 3	16.0
USEROUT12	MBENCUT0	Energy cut 0	16.0
USEROUT13	MBENCUT1	Energy cut 1	16.0
USEROUT14	MBENCUT2	Energy cut 2	16.0
USEROUT15	MBENCUT3	Energy cut 3	16.0

Table 2: The meaning of the 16 words in “USEROUT” when it is used as an input

7.3 Trigger threshold - MBTRGTHR

The trigger threshold parameter is used by the extrapolated baseline-crossing algorithm. This algorithm starts at the middle of the signal and goes backwards until the waveform values are less than “trigger threshold” above the baseline.

However, the values are left shifted by 2 (i.e. multiplied by four) by the DSP, so in order to determine this parameter, you need to look at some signals and choose a value just above the noise, subtract the baseline level from it and divide by four.

For example, “USERIN03” and “USERIN04” could be 0x0404, which sets all four channels to have a cut off value of 0x04 (i.e. 4 decimal) which corresponds to 16 units above the threshold.

7.4 T_{90} threshold - MBT90THR

The percentage cut off is given by:

$$Threshold = (1 - 2^{value}) \times 100\% \quad (2)$$

Values of the threshold are tabulated in table 7.4

Note, however, that this is relative to the energy after ballistic deficit correction, which will generally be higher than the maximum of the peak. So very high percentages are probably not useful, since the signal probably never reaches 99% of the energy!

T ₉₀ value	T ₉₀ threshold
0	0 %
1	50 %
2	75 %
3	87.5 %
4	93.75 %
5	96.875 %
6	98.4375 %
7	99.21875 %

Table 3: Meaning of T₉₀ values

An example is 0x4444 meaning that the T₉₀ threshold is set to 93.75 % for each channel.

7.5 PSA control status register - MBPSACSR

Used by	Bit	Meaning if set	Else
MB	0	T ₀ , extrapolated baseline crossing	XIA CFD
MB	1	T ₉₀	
MB	2	T _{slope}	
MB	3	Q _{max}	
MB	4	Use energy for TFA cut off	
TFA	5	Use TFA	
TFA	6	TFA (cleared after 1 st run)	
TFA	7	Force average timing	
Averaging	8	Average pulse shape (cleared after 1 st run)	
Averaging	9	Average is reference for T ₀	Chn 0
Averaging	10	Average is reference for T ₉₀	Chn N
Averaging	11	Reference is PSA offset and length	
Switch	12	Quadratic interpolation for T ₀	
Switch	13	Interpolation T ₀ and T _{slope}	25 ns
Switch	14	Overwrite GSLT values	GSLT
Switch	15	Interpolation for T ₅₀ and T ₉₀	25 ns

Table 4: The meaning of the MBPSACSR bits

The bits of the pulse-shape analysis control-status register are flags which are tabulated in table 7.5.

If bit 0 is set we extrapolate from T₀ (the time when the signal crosses the threshold) to the level of the baseline to determine the time when the signal really starts. Note that this refers to the threshold given in “USERIN03” and “USERIN04”, not the normal threshold. If the bit is not set, then the XIA CFD value is used instead.

For some reason bit 1 is not documented!!!

If bit 2 is set, the time-to-steepest slope is written into the list mode buffer.

If bit 3 is set, the induced mirror charge amplitude is written to the list mode buffer.

If bit 4 is set, the values of “USERIN14” or “USERIN15” are used for a TFA energy cut off.

If bit 5 is set, the TFA is turned on. The first time you run, you **must** also set bit 6 and write meaningful values into the “USEROUT” structure as well.

If bit 6 is set, some values are loaded from the USEROUT block. Note that this block is overwritten by the user code itself and also by control tasks. This bit is automatically cleared when this is done, so that the block isn’t loaded again next time.

Bits 8 to 11 control the averaging. For this to work, you have to load some data from the “USEROUT” by setting bit 8. This bit is automatically cleared in the same way as bit 6. For a core signal, we can determine the reference times from the signal itself, but this won’t work for a segment unless there is a net charge. So we can use averaging and use the average times (relative to the main trigger) as a reference. Alternately, we can provide these reference times explicitly in “USERIN10” to “USERIN13”.

If bit 13 is set, we interpolate/extrapolate the start time (T_0) and time to steepest slope (T_{slope}).

If bit 15 is set, we interpolate/extrapolate the T_{50} and T_{90} times.

Option	Time [25 ns ticks]	Typical extra deadtime
Baseline	$20 + \text{MBBASELEN}$	$0.75 \mu\text{s}$
T_{50}	$45 + \text{MBPSALEN} / 2$	$1.55 \mu\text{s}$
T_{90}	$45 + \text{MBPSALEN} / 4$	$1.325 \mu\text{s}$
T_0	$45 + \text{MBPSALEN} / 4$	$1.325 \mu\text{s}$
T_{slope}	$45 + \text{MBPSALEN} * 5$	$5.5 \mu\text{s}$
Q_{max} (not hit)	$32 + \text{MBPSALEN} * 6$	$6.05 \mu\text{s}$
Q_{max} (hit)	$100 + \text{MBPSALEN} * 6$	$7.75 \mu\text{s}$

Table 5: Additional dead time for the different options

These options add extra dead time. The amount depends on the values of MBPSALEN (“USERIN10” and “USERIN11”) and MBBASELEN (“USERIN00”). These are tabulated in table 7.5.

For example, we might set MBPSACSR to be 0x6007 for a channel corresponding to a core, meaning that we have bits 0, 1, 2, 13 and 14 set, so that start of the signal is determined by the extrapolated baseline crossing method. Also, we have T_{90} and the time-to-steepest slope algorithm on, but the mirror charge amplitude algorithm is off. All the TFA and averaging stuff is off. We have selected linear interpolation for T_0 and T_{slope} (i.e. bit 12 is off and 13 is on), but we have interpolation for T_{50} and T_{90} off (bit 15 is zero) and finally we are overwriting GSLT values (bit 14).

For a segment we might set 0x660B, which has bits 0, 1, 3, 9, 10, 13, and 14 set. Compared to the core, we have turned off the time-to-steepest slope

algorithm (bit 2) and turned on the mirror charge algorithm (bit 3). We have also turned on the averaging, so we will take our time reference relative to the average. We need this for a segment because there might not be a net charge. However, it looks like we are actually using 0x600B at CERN, so the averaging is off!

7.6 PSA length and offset - MBPSALEN, MBPSAOFF

The parameters “USERIN10” to “USERIN13” are used to provide the start and length of the region over which the pulse-shape analysis is to be performed. Note that these are not the same as the PSALENGTH and PSAOFFSET parameters that Xia have defined. They are only used if bit 11 of MBPSACSR is set.

Note that the Xia parameter “TRACELENGTH” must be at least the sum of MBPSALEN and MBPSAOFF. Also there is a limit of 55 words. If you exceed this, you will corrupt data.

7.7 TFA Energy cut off - MBTFACUT

Like the trigger threshold, this value has to be a 12 bit number, which means it is shifted by four bits (i.e. multiplied by 16) with respect to the energies given by the DGF. It is only used if bit 4 of PSA CSR is set.

8 Output

As well as the data written to URETVAL, which the Xia code automatically copies into the list mode buffer, the user code writes values to USEROUT.

9 Activating the User DSP code

Set MODCSRB to 1 and CHANCSRBx (where x is the number of the channel) to 1.

Then you must set the parameters in the “USERIN” block.

10 Data format

The usual channel data is overwritten by the user code. The format is shown in table 10. The first three words are the same as the usual long Xia format.

Q_{max} is an integer, but T_0 , T_{50} , T_{90} and T_{slope} are given in “8.8 format”. This means that the most significant 8 bits are integer multiples of ticks, and the lower 8 bits are fractions of ticks. In order to get the time, you have to divide the number by 256 and multiply by 25 ns.

Parameter	Name	Meaning	Format
USEROUT00	MBUNUSED0	Unused	
USEROUT01	MBUNUSED1	Unused	
USEROUT02	MBUNUSED2	Unused	
USEROUT03	MBUNUSED3	Unused	
USEROUT04	MBPSAERR	PSA error	16.0
USEROUT05	MBERRCODE	Error code	4 4 4 4
USEROUT06	MBNGOOD	No. good events	16.0
USEROUT07	MBNBAD	No. bad events	16.0
USEROUT08	MBTFAPTR	TFA buffer pointer	16.0
USEROUT09	MBTFALEN	TFA buffer length	16.0
USEROUT10	MBAVGPTR	Average buffer pointer	16.0
USEROUT11	MBAVGLEN	Average buffer length	16.0
USEROUT12	MBQMAXPTR	Qmax buffer pointer	16.0
USEROUT13	MBQMAXLEN	Qmax buffer length	16.0
USEROUT14	MBSLPTR	Slope buffer pointer	16.0
USEROUT15	MBSLLEN	Slope buffer length	16.0

Table 6: The meaning of the 16 words in “USEROUT” when it is used as an output

Word	Meaning
0	Number of words
1	Fast trigger time
2	Energy
3	T_0 or XIA CFD (depending on bit 0 of PSACSR)
4	T_{slope}
5	GSLT A or Q_{max}
6	GSLT B or $T(Q_{max})$ (depending on bit 14 of PSACSR)
7	GSLT C or Error code (depending on bit 14 of PSACSR)
8	REALTIME HI or T_{90} (depending on bit 14 of PSACSR)

Table 7: Data output format when user DSP code is turned on

Bit	Source of error
0	T_{50}
1	T_{90}
2	T_0
3	T_{slope}

Table 8: Meaning of bits in error code

10.1 Error codes

11 Example

Baseline:48:13107:0x3333

CutOff01:49:1285:0x505
CutOff23:50:1285:0x505
T0Thresh01:51:2570:0xa0a
T0Thresh23:52:2570:0xa0a
T90Thresh:53:17476:0x4444
PSACh0:54:24583:0x6007
PSACh1:55:24587:0x600b
PSACh2:56:24587:0x600b
PSACh3:57:24587:0x600b
PSALength01:58:8995:0x2323
PSALength23:59:8995:0x2323
PSAOffset01:60:0:0x0
PSAOffset23:61:0:0x0
TFACutOff0:62:0:0x0
TFACutOff1:63:0:0x0