ICECUBE

DELTA COMPRESSOR DATA FORMAT AND PROCESSES

Version1.0

Joshua Sopher
jesopher@lbl.gov
Lawrence Berkeley National Laboratory

July 26, 2005

modified by Dawn Williams
dawnwill@phys.psu.edu
Penn State University

Dec. 12, 2005

1 Introduction

The delta compressor reads uncompressed data from a buffer containing fADC and ATWD data. It processes and outputs a header and compressed data as long words for recording in the Look Back Memory of the DOM Main Board. Compressed data are packed as long words, and unpacked as long words for outputting.
The document describes these processes, with examples at each stage of processing. Examples of how data are decompressed are also shown.
The document provides the necessary information for writing the de-compression code, software for the DOM CPU to generate the full time stamp, and for testing compressed data in the Look Back Memory of the DOM Main Board.

Data is recorded in DOM memory at 2k boundaries as follows:

2 Description Of The Header

Compressed data begins with a header, followed by compressed data.

2.1 Header Word Used By "Domapp" Only

WORD0:	Compr flag	D31	1 bit
	Compression method	D30..D28	3 bits
	Unused	D27...D16	12 bits
	Time Stamp (16m.s. bits)	D15..0, D15 = msb	16 bits

2.2 Compression Header

WORD1:	Compr flag	D31	1 bit
	Trigger Word	D30..D18, D30 = msb	13 bits
	LC	D17..D16, D17 = msb	2 bits
	fADC Avail	D15	1 bit
	Atwd Avail	D14	1 bit
	Atwd Size	D13..D12, D13 = msb	2 bits
	ATWD_AB	D11	1 bit
	Hit Size	D10..D0, D10 = msb	11 bits
WORD2:	Time Stamp (32 l.s. bits)	D31..D0, D31 = msb,	32 bits
WORD3:	Peak Range	D31	1 bit
	Peak Sample	D30..D27, D30 = msb,	4 bits
	Pre-peak Count	D26..D18, D26 = msb,	9 bits
	Peak Count	D17..D9, D17 = msb,	9 bits
	PostPeak Count	D8..D0, D8 = msb,	9 bits

Word0 is not part of the compressor header transmitted to the DAQ. It is described here for use by DOM software. Compressed data transmitted to the DAQ begins with Word1.

2.3 Meaning Of Header Bits

Bits for word0.... word3 are described in this section.

2.3.1 Word0 (not transmitted to the DAQ with compressed data)

D31: Compressed/Uncompressed flag.
1 = compressed data
0 = uncompressed data.
This bit tells DOM software that the most significant 16 bits of the Time Stamp are contained in Word0. The lower bits of the Time Stamp are contained in Word2. Software monitors when the lower 32 bits roll over, and then transmits the full time stamp to the surface. (For more information, refer to domapp API documentation at
http://www.npxdesigns.com/domapp/api.html).
D30...D28: Indicates which method of compression is used.
000 = road grader
001 = delta compression
010...111 = TBD
D27..D16: Unused bits
D15..D0: Time Stamp(D47..D32). D15 is the msb. The highest 16 bits of the 48 bit "raw data" Time
Stamp are recorded.

2.3.2 Word1

D31: Compressed/Uncompressed flag.
1 = compressed data
0 = uncompressed data
This bit is used by the de-compressor on the surface to identify compressed data, so that it is ALWAYS equal to 1 for compressed data.
D30 to D18 Compression Trigger Word.
The lower 13 bits of the 16 bit raw data Trigger Word.
D17, D16: LC (Local Coincidence)
D[17..16] = 01 LC tag came from below
D[17..16] = 10 LC tag came from above
D[17..16] = 11 LC tag came from below and above
D15: fADC Available
1 = true, 0 = false. If false, Atwd Available will = 0.
D14: Atwd Available
1 = true, 0 = false
D13, D12: Atwd Size: Number of Atwd channels.
D[13..12] = 00 ch0 only
D[13..12] = 01 ch0 and ch1
D[13..12] = 10 ch0, ch1, and ch1
D[13..12] = 11 ch0, ch1, ch2, and ch3
D11: ATWD_AB
1 = Atwd B
0 = Atwd A
D10 to D0: Hit Size.

This is the number of bytes in the hit, and includes the header. (Numbering starts from 1 not 0). For example, if only the header (12 Bytes) is sent to the DAQ, then Hit Size in memory (Word1, D10..0) = 000 0000 1100. The average hit size for tagged data (when there is local coincidence), assuming three small pulses = 96.
It does not count the bytes of Word0, since they are not part of compression data sent to the DAQ.

2.3.3 Word2

D31 to D0 Compression Time Stamp. D31 is the msb. The lowest 32 bits of the 48 bit (full) Time Stamp are used. Word2 roll-over occurs every 1.789 minutes.

2.3.4 Word3

D31: Peak Range
1 = Higher 9 bits
0 = Lower 9 bits
fADC data is 10 bits wide. Peak range. Pre-, post-, and peak counts use either the lower or the upper 9 bits. Peak range is based on the *peak* count.
D30 to D27: Peak Sample (number)
This is the sample number of the peak count. The first sample number is 0.
D26 to D18: Pre-peak Count
This is the count of the fADC output of the sample preceding the peak sample.
D17 to D9: Peak Count
This is the count of the fADC output of the peak sample.
D8 to D0: Post-peak Count
This is the count of the fADC output of the sample following the peak sample. If the peak does not occur within the range of 0 to 15 samples, the post-peak count will exceed the peak count.

The de-compressor uses these parameters to keep track of the stream of data, and not lose its place.
The hit size tells when you get to the end of a set of hit data.
Atwd Size is used to calculate the number of 128 10-bit words recorded per channel.
fADC Available is used to calculate if 256 fADC words are recorded.
If fADC data is not recorded, then ATWD data is also not recorded.

3 Compressed Data (Word4 … Wordn)

WORDn is given by the Hit size, as described above.
Data is obtained from different data sources (fADC, AWTD channels), depending on the various flag values. For instance, if fADC Available = 0, only the header is recorded. If fADC Available = 1, and Atwd Available = 0, only the header and fADC data is recorded.
Compressed data is read out of memory in the following order.
fADC is first, AWTD Ch0 is next, followed by Ch1, Ch2, and Ch3.

4 Method Of Compression

The number of data sources such as fADC, ATWD ch0, etc, that are read out and processed is given in the raw data header. For a description, see the section called "Raw Data Format Header" in "DOMAPP firmware API document" by Thorsten Stezelberger.

Uncompressed data is a 10 bit word (count). The difference between two successive words is calculated. For example, zero is subtracted from the first uncompressed data to provide the first delta count plus the sign. The first uncompressed data is subtracted from the second count to give the second signed delta count etc. The signed subtraction is in Two's Complement (TC) format.
Read a data word from the buffer.
Subtract it from the previous data word (or zero for the first word of the fADC, Ch0, Ch1, Ch2, and Ch3).
Save the signed difference (delta).

The deltas are packed into words of length 1,2,3,6 or 11 bits depending on magnitude. The number of bits per word (bpw) is initialized to 3. Then the algorithm is as follows (all numbers are little endian):

```
if bpw==1
        if |delta| == 0
                output “0”
        else
                output “1”
                bpw=2
if bpw==2
        if |delta|<2
                output delta in 2 bits
                if |delta|<1
                        bpw=1 <--- if delta is small enough, transition to a lower bpw
        else
                output “10”  <---- this is a flag which indicates a transtion to a higher bpw for the
                                        decoder
                bpw=3
if bpw==3
        if |delta|<4
                output delta in 3 bits
                if |delta|<2
                        bpw=2
        else
                output “100”
                bpw=6
if bpw==6
        if |delta|<32
                output delta in 6 bits
                if |delta|<4
                        bpw=3
        else
                output “100000”
                bpw=11
if bpw==11
        output delta in 11 bits
        if |delta|<32
                bpw=6
```

IMPORTANT NOTE: The first delta word for the FADC is referenced to zero. The first delta word of each ATWD is also referenced to zero; it is not referenced to the last uncompressed word of the fADC or the preceding ATWD channel.

5 Method of Decompression

The decompression algorithm is as follows:

Initialize bpw=3. In general, for bpw=n (n=1,2,3,6,11) do the following:

1) Read out the next n bits = word.
2) If the MSB of the word is 1 and the other bits are all 0, then set bpw to the next higher value and go back to 1). Otherwise go on to 3)
3) Convert the word to a delta. If it is the first decompressed word, then it is equal to the first value in the waveform.
4) If |delta|< 2^(next lower bpw -1), set bpw to the next lower bpw;
   i.e., if bpw=6 and |delta|< 4 =2^(3-1), then set bpw=3.
5) Repeat until the waveform is decoded: that is, you have N decoded values, where N=256+128*(number of ATWD channels).

Decompression example:
first word: 00112304 = 0 00000000001 00010010001 100000 100 (little endian)
second word: c70671e0 = 110001110000 01 1 00 11 1 00 01 111 00000 (little endian)

Read the first three bits: 100 => MSB is 1 and all other bits zero, this is a flag, bpw=6
Read the next six bits: 100000 => MSB is 1 and all other bits zero, this is a flag, bpw=11
Read the next eleven bits: 00010010001 = 145 => first data point = 145
Read the next eleven bits: 00000000001 = 1 => second data point = 145+1 = 146
|1|<2^(6-1), so bpw=6.
Read the next six bits: 000000 = 0 => third data point = 146+0 = 146
|0|<2^(3-1), so bpw=3.
Read the next three bits: 111 = -1 => fourth data point = 146-1 = 145
|-1|<2^(2-1), so bpw=2
Read the next two bits: 01 = 1 => fifth data point = 145+1 = 146

Read the next two bits: 00 = 0 => sixth data point = 146 + 0 = 146
|0|<2^(1-1), so bpw=1
Read the next bit: 1 => this is a flag, bpw = 2.
Read the next two bits: 11 = -1 => seventh data point = 146-1 = 145
Read the next two bits: 00 = 0 => eighth data point = 145+0 = 145
|0|<2^(1-1), so bpw=1
Read the next bit: 1 => this is a flag, bpw = 2
Read the next two bits: 01 = 1 => ninth data point = 145+1 =146

Uncompressed data stream = 145, 146, 146, 145, 146, 146, 145, 145, 146, etc.

Note that if you are decompressing the first word of an FADC or ATWD channel, it is referenced to zero and not to the last word of the preceding channel.

6 Decompressing End Bits

The number of bits necessary to encode a channel is not necessarily a multiple of 32, so the last 32-bit word in the encoded stream may be padded with zeroes. These zeroes should be ignored. There are no extraneous zeroes anywhere in the data stream except at the end.

7 Formatting Data For Transmission To The Surface

Compressed data is recorded in memory as 32 bit words. The number of compressed bytes, given by the header will not necessarily be a multiple of 32 bits. There may be invalid bytes contained in the long word.
In order to get the full benefit of compression, these invalid bytes must not be sent to the surface since they are an overhead that wastes bandwidth. Accordingly, only valid bytes must be used. This will require concatenating the last byte of a hit, with the first byte of the next hit, and so on. Compressed bytes from successive Hits would be packaged into a stream of data containing valid bytes only.
(For more information regarding concatenation, refer to domapp API documentation at http://www.npxdesigns.com/domapp/api.html).
This data will then need to be wrapped into the format of the serial transmitter, which contains start and stop bits.

7 Processing Time

8 Use Of FPGA Resources