LArSoft - Bug #11631

raw.cxx Uncompress gives wrong results for ZeroSuppression and Huffman together

02/04/2016 10:45 AM - David Adams

Status:	Assigned	**Start date:**	02/04/2016
Priority:	Normal	**Due date:**	
Assignee:	Jonathan Insler	**% Done:**	100%
Category:	Reconstruction	**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
Occurs In:		**Co-Assignees:**	
Experiment:	-		

Description

The utility lardata/RawData gives wrong results for the case that both zero suppression and Huffman encoding are requested. It seems the problem is this code in Uncompress:

```
else if(compress == raw::kZeroHuffman){
      UncompressHuffman(adc, uncompressed);
      ZeroUnsuppression(adc, uncompressed);
```

i.e. the uncompressed vector from the Huffman decoding is not but should be used as the input to the zero unsuppression.

When I change to this:

```
std::vector&lt;short&gt; tmp(2*adc[0]);
      UncompressHuffman(adc, tmp);
      ZeroUnsuppression(tmp, uncompressed, pedestal);
```

the problem goes away.

Note my guess for the size allocation of the tmp vector. UncompressHuffman requires the caller to preallocate enough space. I have already reported that in https://cdcvs.fnal.gov/redmine/issues/11572. In my case, it was not enough to use adc[0] because the zero suppression increases the size of the vector.

My test showing the problem can be found here:

https://github.com/dladams/art_extensions/blob/master/test/utilities/test_Compress.cxx

History

#1 - 02/04/2016 04:18 PM - Thomas Junk

- Assignee changed from Gianluca Petrillo to Jonathan Insler

#2 - 02/04/2016 05:32 PM - Jonathan Insler

- Assignee changed from Jonathan Insler to Gianluca Petrillo

- % Done changed from 0 to 100

David Adams wrote:

> The utility lardata/RawData gives wrong results for the case that both zero suppression and Huffman encoding are requested. It seems the problem is this code in Uncompress:
>
> else if(compress == raw::kZeroHuffman){
> UncompressHuffman(adc, uncompressed);
> ZeroUnsuppression(adc, uncompressed);
>
> i.e. the uncompressed vector from the Huffman decoding is not but should be used as the input to the zero unsuppression.
>
> When I change to this:
>
> std::vector<short> tmp(2*adc[0]);
> UncompressHuffman(adc, tmp);

> ZeroUnsuppression(tmp, uncompressed, pedestal);
>
> the problem goes away.
>
> Note my guess for the size allocation of the tmp vector. UncompressHuffman requires the caller to preallocate enough space. I have already reported that in https://cdcvs.fnal.gov/redmine/issues/11572. In my case, it was not enough to use adc[0] because the zero suppression increases the size of the vector.
>
> My test showing the problem can be found here:
>
> https://github.com/dladams/art_extensions/blob/master/test/utilities/test_Compress.cxx

Thanks to David for finding this! I have implemented his fix. This functionality was never properly tested.

#3 - 02/04/2016 05:33 PM - Jonathan Insler

- Assignee changed from Gianluca Petrillo to Jonathan Insler

#4 - 07/25/2016 10:40 AM - Gianluca Petrillo

- Status changed from New to Assigned