Abstract— MicroBooNE is a liquid Argon time projection chamber to be built at Fermilab for an accelerator-based neutrino physics experiment and as part of the R&D strategy for a large liquid argon detector at DUSEL. The waveforms of the ~9000 sense wires in the chamber are continuously digitized at 2 M samples/s - which results in a large volume of data coming off the TPC. We have developed a lossless data reduction scheme based on Huffman Coding and have tested the scheme on cosmic ray data taken from a small liquid Argon time projection chamber, the BO detector. For sense wire waveforms produced by cosmic ray tracks, the Huffman Coding scheme compresses the data by a factor of approximately 10. The compressed data can be fully recovered back to the original data since the compression is lossless. In addition to accelerator neutrino data, which comes with small duty cycle in sync with the accelerator beam spill, continuous digitized waveforms are to be temporarily stored in the MicroBooNE data-acquisition system for about an hour, long enough for an external alert from possible supernova events. Another scheme, Dynamic Decimation, has been developed to compress further the potential supernova data so that the storage can be implemented within a reasonable budget. In the Dynamic Decimation scheme, data are sampled at the full sampling rate in the regions-of-interest (ROI) containing waveforms of track-hits and are decimated down to lower sampling rate outside the ROI. Note that unlike in typical zero-suppression schemes, in Dynamic Decimation, the data in the pedestal region are not thrown away but kept at a lower sampling rate. An additional factor of 10 compression ratio is achieved using the Dynamic Decimation scheme on the BO detector data, making a total compression rate of approximate 100 when the Dynamic Decimation and the Huffman Coding functional blocks are cascaded. Both of the blocks are compiled in low-cost FPGA and their silicon resource usages are low.

Index Terms— Front End Electronics, TDC, FPGA Firmware

I. INTRODUCTION

MICROBOONE is a proposed Liquid Argon time projection chamber (TPC) detector [1-3] that will observe interactions of neutrinos from the on-axis Booster Neutrino Beam and off-axis NuMI Beam at Fermi National Accelerator Laboratory. The MicroBooNE experiment will address the low energy excess observed by the MiniBooNE experiment, measure low energy neutrino cross sections, and serve as the necessary next step in a phased program towards massive Liquid Argon TPC detectors. As such it will investigate techniques for free running readout needed for the study of proton decay and supernova.

A conceptual drawing of the MicroBooNE detector is shown in Fig. 1. The TPC has a drift length of 2.5 m in a 500 V/m electric field, which is housed in a cryostat that is 12 m long with inner diameter 3.8m. The MicroBooNE detector contains 150 tons liquid Argon with active volume of about 89 tons. The TPC has three readout wire planes, which include one collection plane with 2.5 m vertical wires (the Y-plane), and two induction planes (the U- and V-planes) with wires ±60° from vertical respectively. Thirty PMTs are installed in the cryostat, they are used to generate trigger for the DAQ system, and to help determine the time zero of the events.

The readout electronics process signals from approximately 9,000 channels. The waveform signal on each sense wire is amplified and continuously digitized with an analog-to-digit converter (ADC) at 2 MHz, which generates large volume of data. A systematic approach is necessary to control the total volume of data so that they can be transmitted and stored without pushing electronics technology to the cutting edge to ensure reliability of the system. In this document, several data reduction methods that we have studied: decimation, Huffman Coding and Dynamic Decimation are discussed.
II. THE MICROBOONE TPC DIGITAL PROCESSING FUNCTIONS

The MicroBooNE TPC readout system fulfills request of collecting both accelerator generated neutrino data in sync with the accelerator beam spill and the supernova data, which are continuously digitized waveforms to be temporarily stored in the MicroBooNE data-acquisition system for about an hour, long enough for an external alert from possible supernova events. A block diagram of the MicroBooNE data reduction functions is shown in Fig. 2.

The front-end modules digitize the waveform of the TPC wire signals with ADC and send to FPGA as serial data, which are first converted to parallel internal format. All raw ADC data are decimated down to 2 MHz and the data for accelerator neutrino events are compressed using lossless Huffman Coding scheme while the supernova data are compressed with both the Dynamic Decimation and Huffman Coding schemes.

The ADC for each channel operates at 16 MHz, which is primarily chosen due to the lower limit of operation frequency of the ADC devices. The data stream is reduced down to 2 MHz inside the FPGA through a standard decimation that is a two-step process: (1) low-pass anti-aliasing filtering and (2) down-sampling. The low-pass filter in general is a weighted average of multiple data samples and therefore the decimation process increases the measurement precision by about two bits.

The Dynamic Decimation and Huffman Coding schemes are discussed in the following sections.

III. WAVEFORM DIGITIZATION AND THE HUFFMAN CODING

With availability of fast ADC and switch capacitor array ASIC devices, waveform digitization becomes possible in a broad range of applications. The shapes of the signal pulses may carry much richer information than the traditional charge integration scheme for resolving pileup in high rate systems or for precise timing measurement. An unavoidable challenge is that waveform digitization produces large volume of data at a rate of one word per sampling point, creating difficulties for transporting and storing the data.

On the other hand, a common property of many waveforms we observe is that the voltage levels between two sampling points do not differ by a large amount. In other word, if the value of a data sample is known, the values of subsequent data samples are likely to be near the previous one. Taking advantage of this property, it is possible to reduce data volume without losing useful information. A scheme can be easily implemented in FPGA is a subset of the Huffman Coding.

For a sequence of values of data samples U(n), a difference is first produced between two adjacent values: U(n)-U(n-1). The values of the difference are concentrated primarily around 0, +1 and -1 and larger differences may exist but with less probabilities. The values with higher probabilities are assigned with shortest codes and longer codes are given to less possible values. The scheme for the Huffman Coding is shown in Table I.

<table>
<thead>
<tr>
<th>U(n+1)-U(n)</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4 and others</td>
<td>Full 16 bits word</td>
</tr>
<tr>
<td>-3</td>
<td>000001</td>
</tr>
<tr>
<td>-2</td>
<td>0001</td>
</tr>
<tr>
<td>-1</td>
<td>01</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>+1</td>
<td>001</td>
</tr>
<tr>
<td>+2</td>
<td>00001</td>
</tr>
<tr>
<td>+3</td>
<td>000001</td>
</tr>
</tbody>
</table>

If the difference i.e., U(n+1)-U(n) is >+3 or <-3, or if the data point is the first sample in a sequence, a full 16-bit data word with raw ADC value is used, which is indicated by bit(15,14) = (0,0) and the subsequent samples are represented as differences with their previous samples, i.e., U(n+1)-U(n). The Huffman codes are concatenated into the 16-bit data word with bit(15) = 1. In the example shown above, six data samples are packed in the 16-bit data word. If a sample suddenly jumps from previous one with a difference larger than +3, the raw measurement word is used. The words with bit(15,14) = (0,1) are reserved for data header words and other special control words.

The coding scheme we use here follows the spirit of Huffman Coding but it is a modified subset. The coding table is predefined here and when the probabilities of various values change, the scheme may not yield optimal compression ratio. However, the codes chosen are self-punctuated, i.e., they are all 0’s followed by a 1. The choice of the codes provides simplicity for hardware implementation inside FPGA.

Obviously, the Huffman Coding scheme given here is implemented in FPGA is a subset of the Huffman Coding.
lossless, i.e., all raw data can be restored from the compressed codes. It can be seen that in this coding scheme, the theoretical maximum compression ratio is 15, i.e., if all raw ADC values are identical so that all differences are 0’s. It is still possible to improve the coding efficiency further if the raw measurement values have very slow variation and small noise so that many differences are 0’s. For example, four adjacent 0’s, 0000 can be assigned to the shortest code with one bit and a maximum theoretical compression ratio can be 60.

IV. THE DYNAMIC DECIMATION

For accelerator neutrino events, lossless Huffman Coding compression is sufficient. But for supernova data, continuous samples up to about an hour are to be temporarily stored and therefore deeper compressions become necessary. Data loss is unavoidable for supernova data but the loss should be understandable and adjustable by the users.

A scheme called “Dynamic Decimation” is developed for further compression for the supernova data. The regions of interest (ROI) in the data stream are first recognized so that waveforms corresponding to the drift charges are identified and the regions other than ROI are considered as pedestal. Unlike in typical pedestal subtraction schemes, in the Dynamic Decimation, both pedestal and ROI data samples are used, except that the ROI data are kept at the original sampling rate while the pedestal data are decimated down to much lower (e.g. 1/16) sampling rate as shown in Fig. 4.

The final data record of a wire represents a full coverage of the entire time, with or without drift signal from a charged track. The ROI data are sampled at full rate to ensure best preservation of the charged track information and the pedestal region are sampled at lower rate to reduce total volume of data. If the threshold of ROI recognition is mistakenly set too high and some very low signals are excluded, they will still be seen in the pedestal region, although blurred due to lower sampling rate. This feature allows users to fine tune the ROI recognition algorithm. The data samples after the Dynamic Decimation are sent through another Huffman Coding stage separated from the accelerator neutrino data branch to further reduce the data volume.

For supernova data, entering or leaving the ROI is signified with a raw measurement word with bit13 = DD = either 0 or 1. DD=0 indicates that subsequent samples are in the ROI with normal sampling rate while DD=1 indicates in the pedestal region with (1/16) sampling rate.

The theoretical maximum compression ratio of the Dynamic Decimation alone is (1/16). Obviously the actual compression ratio depends on the occupancy of the ROI. In a liquid Argon TPC for neutrino physics experiment, tracks dominant in an event are from cosmic rays and therefore a relatively low occupancy of the ROI is anticipated.

V. TEST RESULTS

The Fermilab BO detector [4] is a test liquid Argon TPC with two wire planes sensing induction signals and a wire plane sensing collection signals. The wire planes stretch in three directions with 60° angles between each other. Each plane consists 32 wires and the waveforms of them are digitized with 10-bit ADC devices at 5 MHz. The Huffman Coding and the Dynamic Decimation schemes have been evaluated using the cosmic ray track data collected in the BO detector.

A. The Raw Waveform Data

A typical cosmic ray event in the BO detector is shown in Fig. 5 with electron drift time running from left to right and wire numbers showing vertically. Three different wire planes are shown in separate sections.

The theoretical maximum compression ratio is 15, i.e., if all raw ADC values are identical so that all differences are 0’s. It is still possible to improve the coding efficiency further if the raw measurement values have very slow variation and small noise so that many differences are 0’s. For example, four adjacent 0’s, 0000 can be assigned to the shortest code with one bit and a maximum theoretical compression ratio can be 60.

IV. THE DYNAMIC DECIMATION

For accelerator neutrino events, lossless Huffman Coding compression is sufficient. But for supernova data, continuous samples up to about an hour are to be temporarily stored and therefore deeper compressions become necessary. Data loss is unavoidable for supernova data but the loss should be understandable and adjustable by the users.

A scheme called “Dynamic Decimation” is developed for further compression for the supernova data. The regions of interest (ROI) in the data stream are first recognized so that waveforms corresponding to the drift charges are identified and the regions other than ROI are considered as pedestal. Unlike in typical pedestal subtraction schemes, in the Dynamic Decimation, both pedestal and ROI data samples are used, except that the ROI data are kept at the original sampling rate while the pedestal data are decimated down to much lower (e.g. 1/16) sampling rate as shown in Fig. 4.

The final data record of a wire represents a full coverage of the entire time, with or without drift signal from a charged track. The ROI data are sampled at full rate to ensure best preservation of the charged track information and the pedestal region are sampled at lower rate to reduce total volume of data. If the threshold of ROI recognition is mistakenly set too high and some very low signals are excluded, they will still be seen in the pedestal region, although blurred due to lower sampling rate. This feature allows users to fine tune the ROI recognition algorithm. The data samples after the Dynamic Decimation are sent through another Huffman Coding stage separated from the accelerator neutrino data branch to further reduce the data volume.

For supernova data, entering or leaving the ROI is signified with a raw measurement word with bit13 = DD = either 0 or 1. DD=0 indicates that subsequent samples are in the ROI with normal sampling rate while DD=1 indicates in the pedestal region with (1/16) sampling rate.

The theoretical maximum compression ratio of the Dynamic Decimation alone is (1/16). Obviously the actual compression ratio depends on the occupancy of the ROI. In a liquid Argon TPC for neutrino physics experiment, tracks dominant in an event are from cosmic rays and therefore a relatively low occupancy of the ROI is anticipated.

V. TEST RESULTS

The Fermilab BO detector [4] is a test liquid Argon TPC with two wire planes sensing induction signals and a wire plane sensing collection signals. The wire planes stretch in three directions with 60° angles between each other. Each plane consists 32 wires and the waveforms of them are digitized with 10-bit ADC devices at 5 MHz. The Huffman Coding and the Dynamic Decimation schemes have been evaluated using the cosmic ray track data collected in the BO detector.

A. The Raw Waveform Data

A typical cosmic ray event in the BO detector is shown in Fig. 5 with electron drift time running from left to right and wire numbers showing vertically. Three different wire planes are shown in separate sections.

In each event, 2048 samples are collected for each wire. The waveforms in an induction plane are shown in Fig. 6. The actual pedestals of the waveforms are around a same DC value of 400 ADC counts. The waveform of each wire is added with an offset so that waveforms are separated vertically in the graph.
In the liquid Argon TPC, the induction signals are bidirectional with electric charges the wire flows in a direction as the electrons from ionization of the charged track approach and flows back as the electrons leave. Therefore the pedestals are set around the midpoint of the ADC devices.

B. Compression Ratio and Noise Sensitivity

The test on the BO detector data shows that Dynamic Decimation reduces number of samples by about a factor of (1/8)-(1/12) on typical events and the compression ratio of Huffman Coding is about (1/11).

Since the Huffman Coding scheme described earlier is based on the differences of raw data samples, naturally it is very sensitive to high frequency noise. Care must be taken starting from outside of FPGA, before the ADC, on the analog bandwidth of the shaper. The requirement of the sampling theorem must be satisfied carefully. Inside FPGA, appropriate digital filtering must be applied, again, to fulfill the requirement of the sampling theorem.

The waveforms from the BO detector with original sampling rate of 5 MHz are decimated down to 1 MHz and the data compression ratios for different decimation filters are compared in Fig. 7.

The original 5 MHz data samples are given in the leftmost column with compression ratios of about 10. In the second column, the compression ratios drops significantly since there is no filter applied in the decimation process. In every 5 data samples, 4 are thrown away and only one is kept. The aliasing noises are so large that cause a significant degrading on the compression ratio. In the third column, data samples are not thrown away and the 5 data samples are averaged. It can be seen that the compression ratio are improved slightly. In the fourth column, the average of 10 data samples is used as the decimation filter. The compression ratio improves noticeably since the filter starts satisfying the sampling theorem and the alias noise is significantly reduces. In the rightmost column, a CIC-2 filter that satisfies the sampling theorem is used as the decimation filter. It shows even better compression ratio since the stop band suppression of the CIC-2 filter is better than the CIC-1 (the average) filter.

VI. FPGA IMPLEMENTATION

The blocks for Dynamic Decimation and Huffman Coding have been test designed, compiled and simulated with 250 MHz operating frequency in an Altera Cyclone III [5] FPGA device. The Huffman Coding block first calculates the difference between the current and the previous data points and then finds the Huffman code according to the value of the difference through a coding table. The Huffman codes of several data samples are concatenated together and a data valid signal is generated each time when a 16-bit data word is filled up. The silicon resource usage of this block is around 245 logic elements, which is a small fraction in a typical low cost device.
The Dynamic Decimation block keeps three pipelines. One of the pipelines finds region of interest (ROI) from the raw data flowing in. At current stage, the ROI is indentified as the ADC values in a wire pass above a predefined threshold for simplicity, but there are no fundamental difficulties to use other advanced ROI recognition algorithms. One of the other two pipelines stores the raw ADC data and the third pipeline creates decimated, i.e., filtered and down sampled values. At the output, either the raw data samples or the decimated data are selected depending whether the data is inside or outside of the RIO. The resource usage of the Dynamic Decimation is also very small.

VII. CONCLUSION

A lossless data compression scheme, Huffman Coding, is described in this document. The compression ratios of the Huffman Coding on typical events from a liquid Argon TPC, the Fermilab BO detector are about (1/10), which is sufficient for the accelerator neutrino events in the proposed MicroBooNE experiment.

The total data compression ratios typical events with both Dynamic Decimation and Huffman Coding are (1/60) to (1/100). This combination of the data compression scheme will be applied to the data for the supernova study.

ACKNOWLEDGEMENT

The author would wish to express thanks to colleagues and friends especially Bob DeMaat, Stephen Pordes of Fermilab, Hucheng Chen of Brookhaven National Laboratory and Kirk McDonald of Princeton University for their helpful inputs and discussions over years.

REFERENCES


<table>
<thead>
<tr>
<th>Device: EP3C40F484C6, $129, 39600 Logic Elements Available</th>
<th>Logic Element Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Decimation</td>
<td>217</td>
</tr>
<tr>
<td>Huffman Coding</td>
<td>245</td>
</tr>
</tbody>
</table>