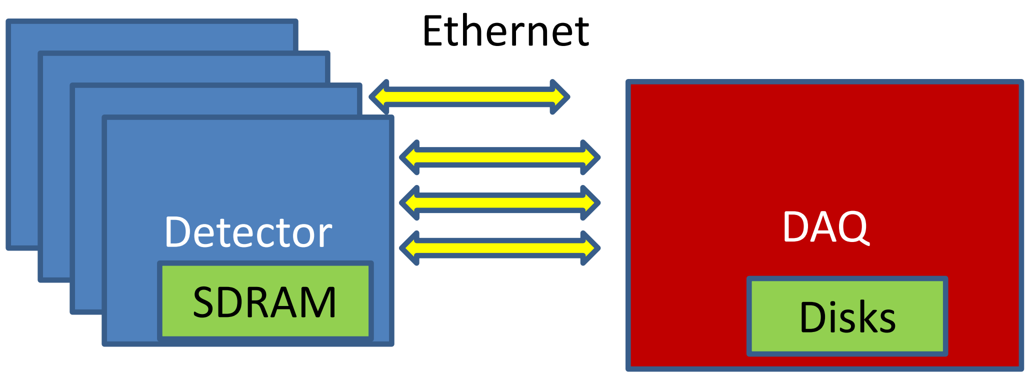
ECAL Compression Techniques

Grzegorz Pastuszak
Warsaw University of Technology

DAQFEET 2021

Need for compression

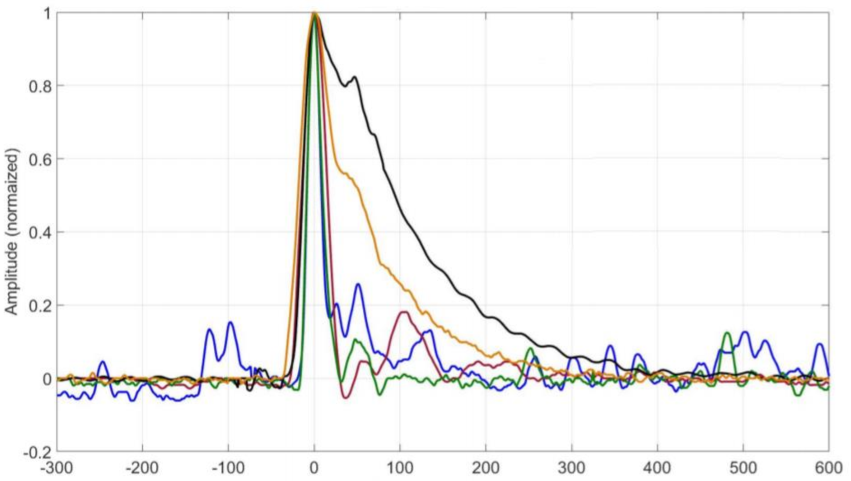
- Saving disk space for the archiving
- Limited bandwidth between detectors and the data acquisition system (DAQ)
- Saving RAM capacity in detector modules

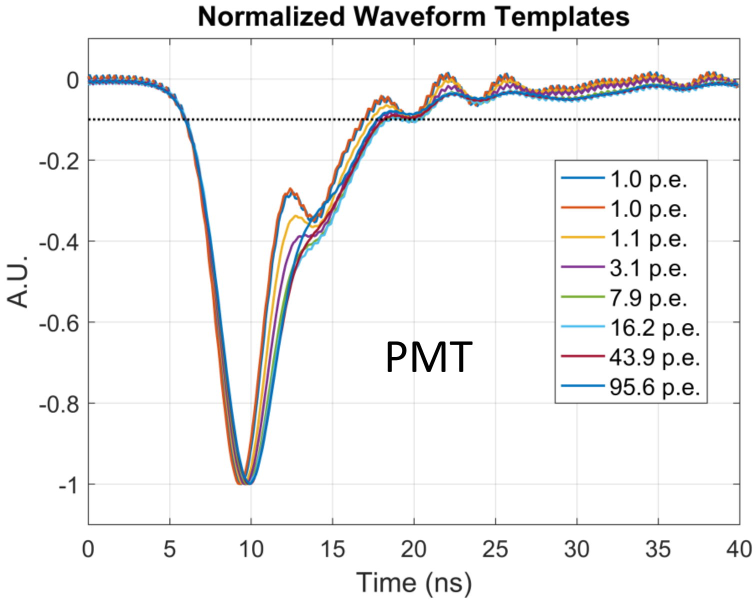


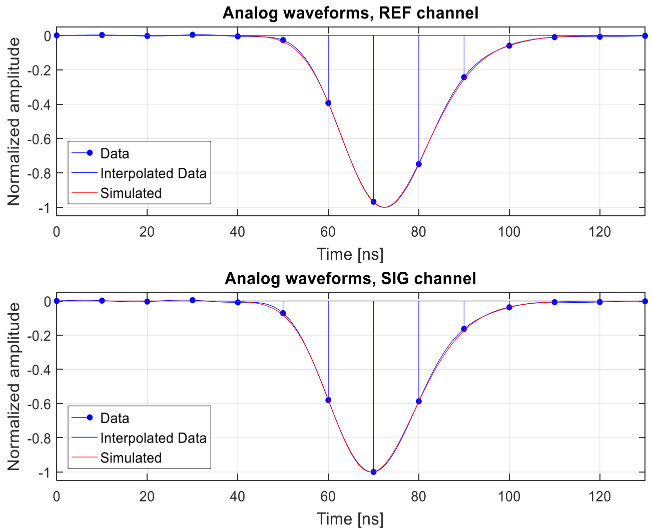
- Constraints on resources and power

Input Signals

- Acquired PMT and SIPM signals:
  - Similar to some extent,
  - Stability is limited,
  - Shaping changes original signal.

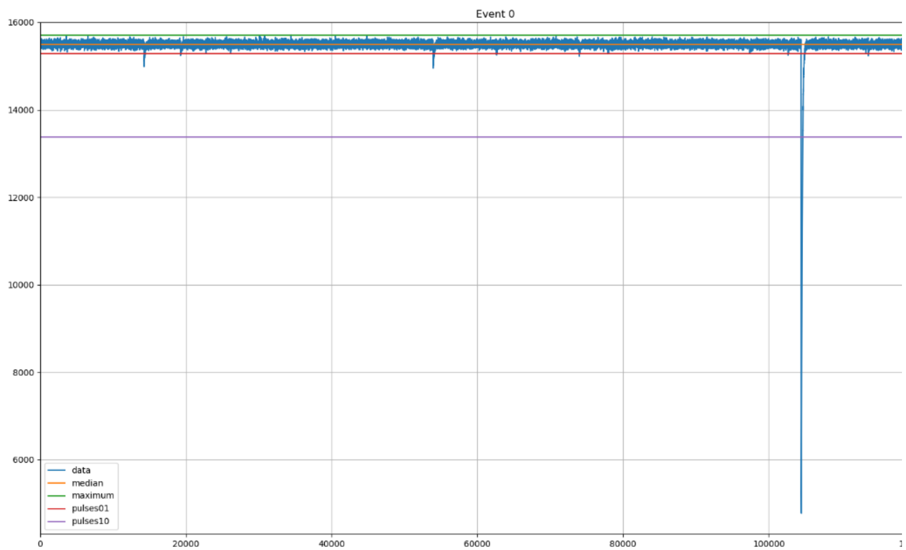


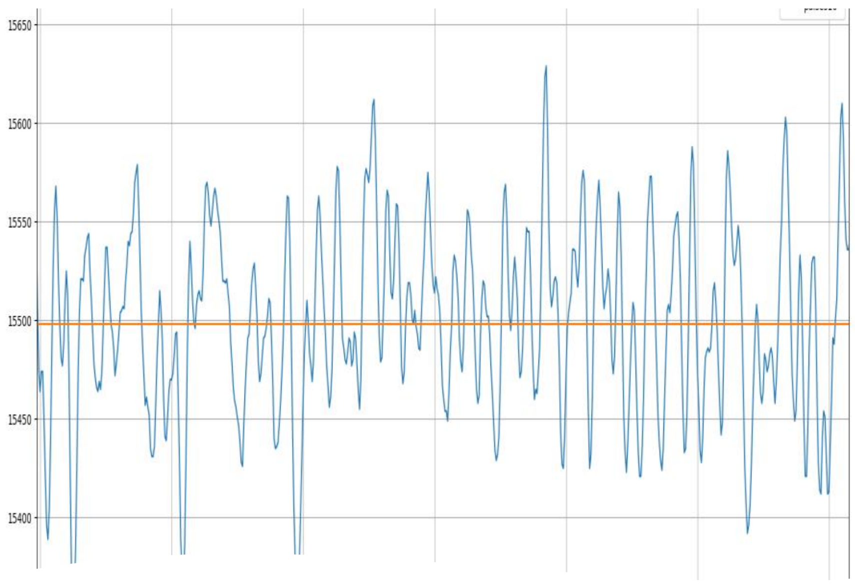




Compression scenarios

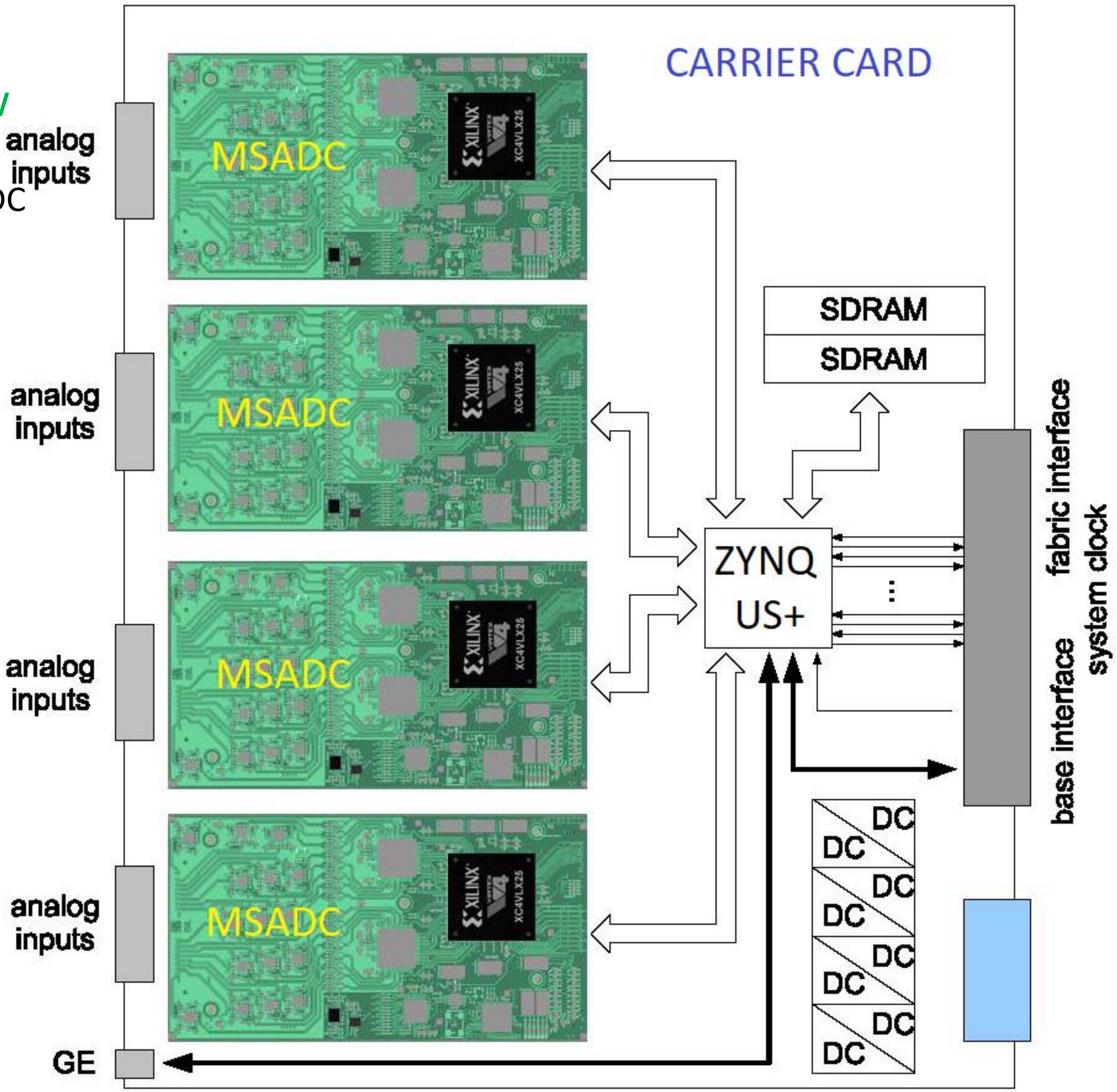
- Lossless or lossy compression:
  - Allowable losses in processing should be small to preserve key signal features
- Compression of all samples or pulse selection:
  - signals without selection
  - Compression of pulses with high sensitivity
  - Compression of pulses with low sensitivity (large pulses)
  - Number of selected samples around pulses
- Sampling frequency vs. bandwidth





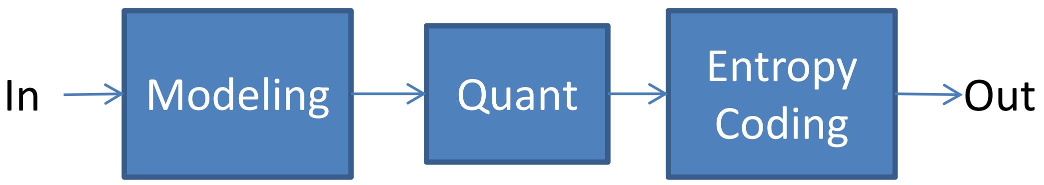
Selected compression scenario

- In the AMBER experiment, available MSADC boards feed new Carrier Cards
  - FPGA Virtex-4 devices in available MSADC boards have limited amount of free resources
  - New Carrier Card is equipped with Zynq UltraScale+ FPGA to extract time and charge
  - a limited throughput on interfaces between FPGAs
- Low-complexity compression indispensable
  - lossless
  - Selection with conservative selection – thresholds close to signal noise
  - Wide selection window, e.g. 4k samples
- Development conducted for available SiPM waveforms
  - Waiting for decoded data from COMPASS experiment
  - Methodology is the same - only pulse shapes and SNRs change to some extend



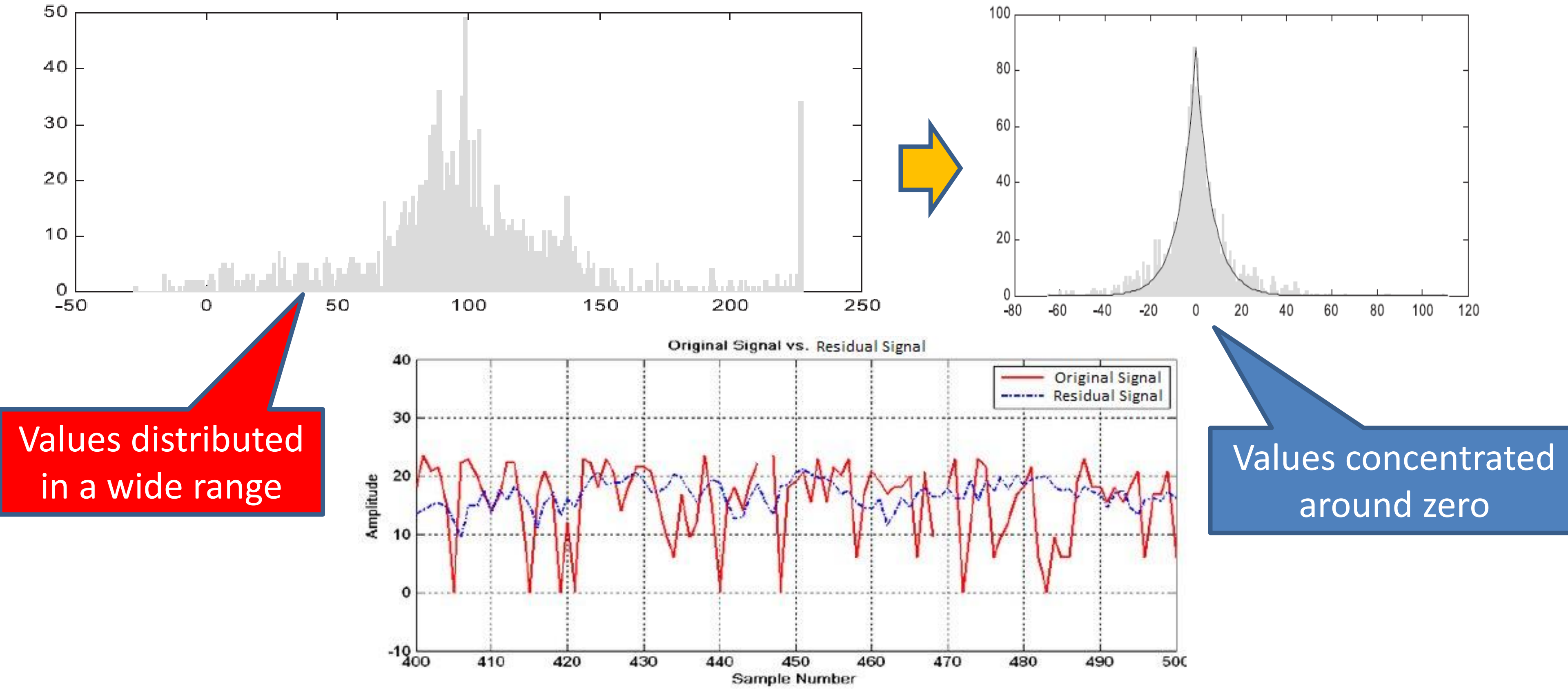
Compression Methods

- Modeling
  - Linear Prediction LPC
  - Signal Models
  - Transforms



- Quantization (lossy compression)
  - Scalar quantization
  - Vector quantization – using signal models
- Entropy Coding
  - Variable length coding
  - Arithmetic coding – more complex and better compression

Signal Modelling



- Predictions and/or Transformations decrease the dynamics
- Distributions of residual signal concentrated around zero
- Signal reconstruction using reverse operations

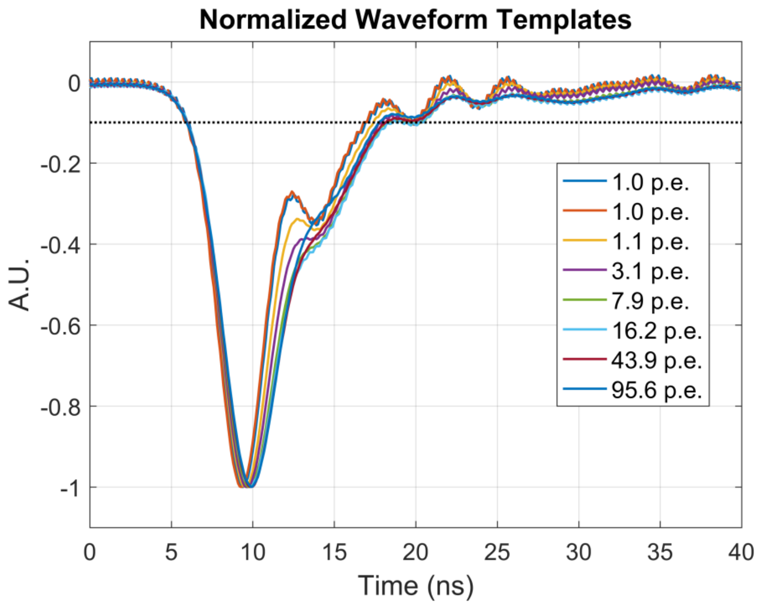
Signal Models

- Set of representative sample sequences are compared with acquired samples to find the best matching in terms of SAD or MSE

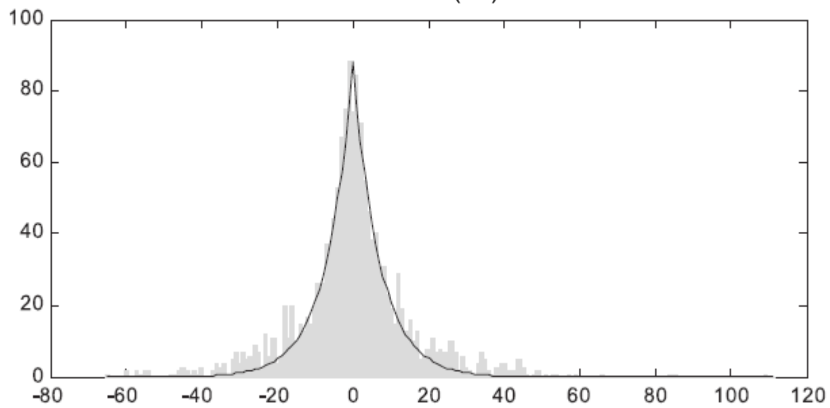
SAD $$i = \arg\min\left(\sum_t |\, x[t,i] - x[t] \,|\right)$$

MSE $$i = \arg\min\left(\sum_t \left(x[t,i] - x[t]\right)^2\right)$$

$$x_{predicted}[t] = x[t,i]$$



- Residuals (equal to difference between input samples and their predictions) have much lower values and energy



Compression efficiency similar to LPC

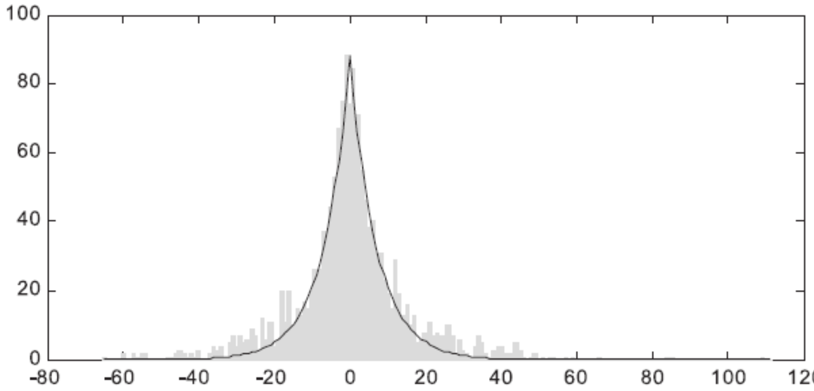
Higher computational cost

Linear Predictive Coding (LPC)

- Prediction as a sum of previous samples multiplied by coefficients

$$x_{predicted}[t] = \sum_{i=1}^{N} a_i x[t-i]$$

- Residuals (equal to difference between input samples and their predictions) have much lower values and energy



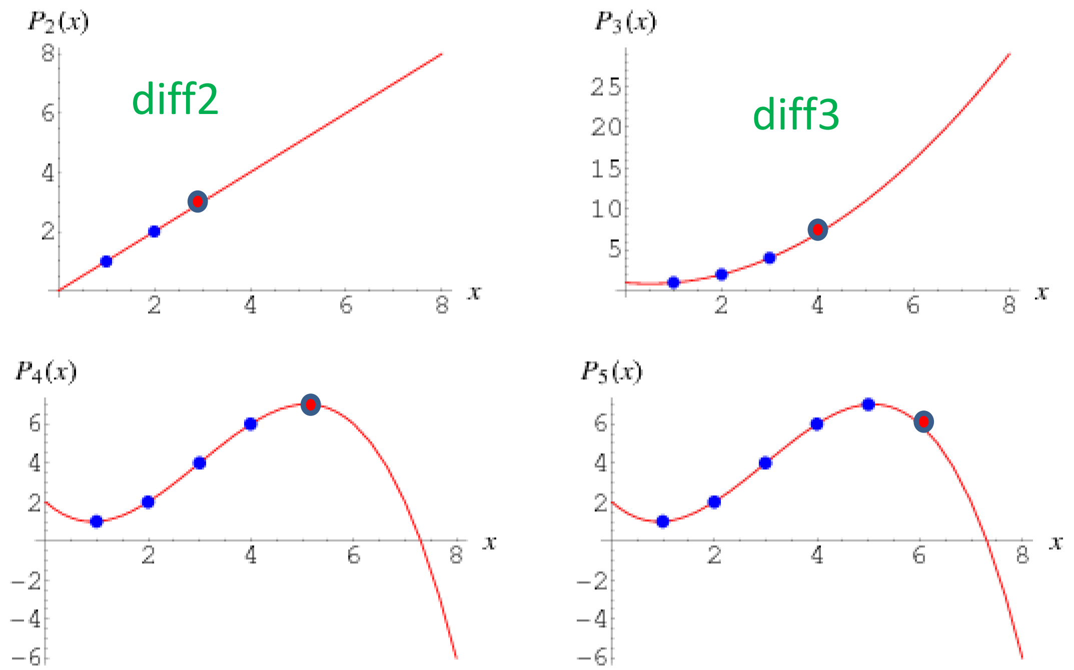
$$\Delta x[t] = x[t] - \sum_{i=1}^{N} a_i x[t-i]$$

- Coefficients a_i must be known at the decoder
  - precomputed or sent with residuals
- Error energy:

$$E = \sum_{t=0}^{T} (\varepsilon[t])^2 = \sum_{t=0}^{T} \left(x[t] - \sum_{i=1}^{N} a_i x[t-i] \right)^2$$

Prediction based on derivatives

- Polynomial model based on directly preceding samples
- Applied separately or jointly with LPC
- Diff1 Improves slightly LPC by up to 0.05 bps

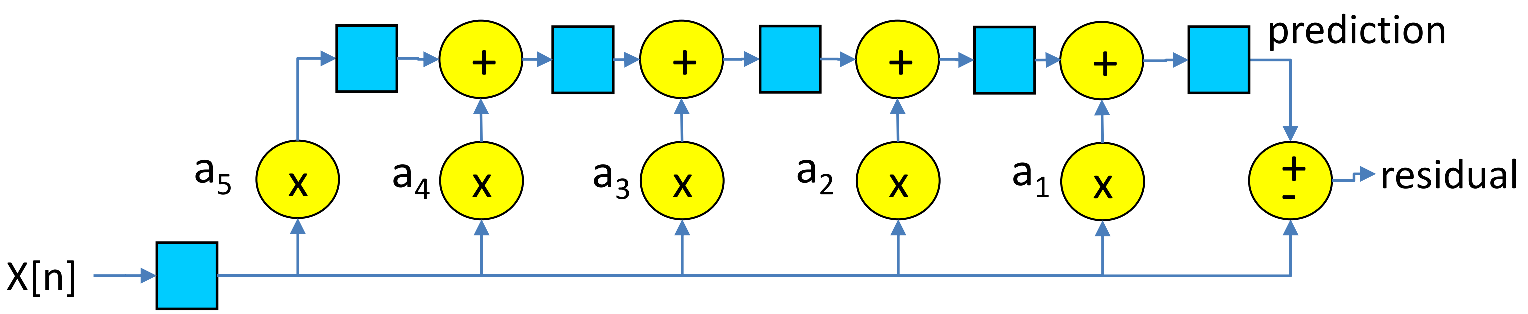


- diff0: signal: x[i]
- diff1: x[i] - x[i-1]
- diff2: x[i] - 2*x[i-1] + x[i-2]
- diff3: x[i] - 3*x[i-1] + 3*x[i-2] - x[i-3]

Assure unbiased predictor for LPC

LPC architecture

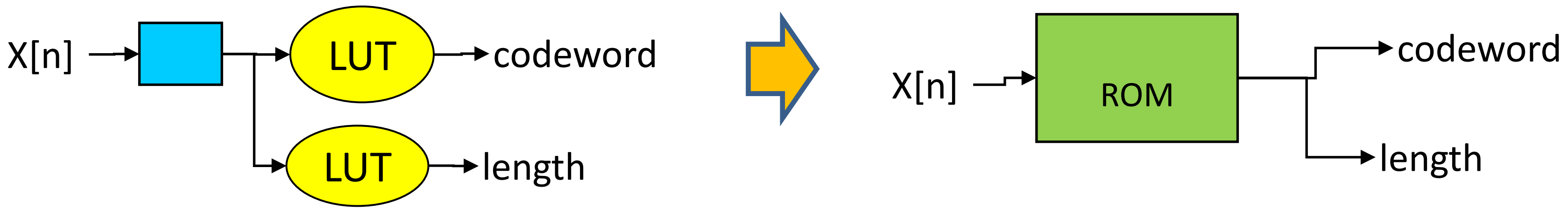
- LPC is n as a sum of previous samples multiplied by coefficients
- Filter architecture suitable for LPC



- Coefficients a_i are successive values of the covariance function
- Increasing filter order improves compression efficiency
  - Four-tap filter is usually sufficient to get bit-rates about 5.5 bps
  - Changing order from 4 to 16 reduces bit-rate by up to 0.2 bps

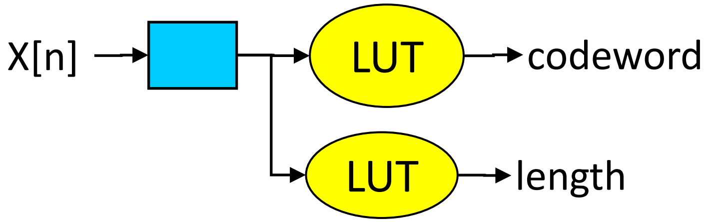
Variable Length Coding (1)

- Assignment of input values to codewords
- Codewords with variable lengths inversely proportional to probabilities
- Bit rate greater than the information entropy by a small fraction of bit per sample (~0.03-0.09 bps)
- Variable Length Coding is simple in implementation

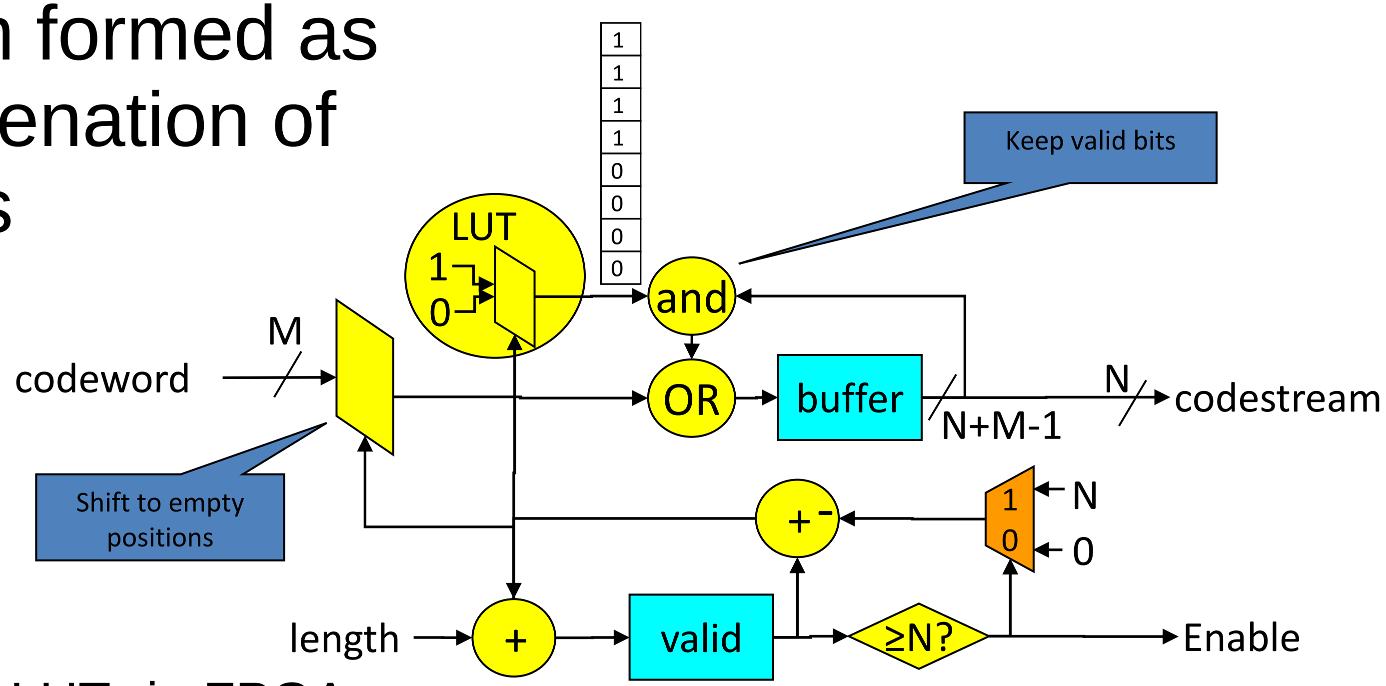


Variable Length Coding (2)

- Input value directly mapped to codewords and their lengths

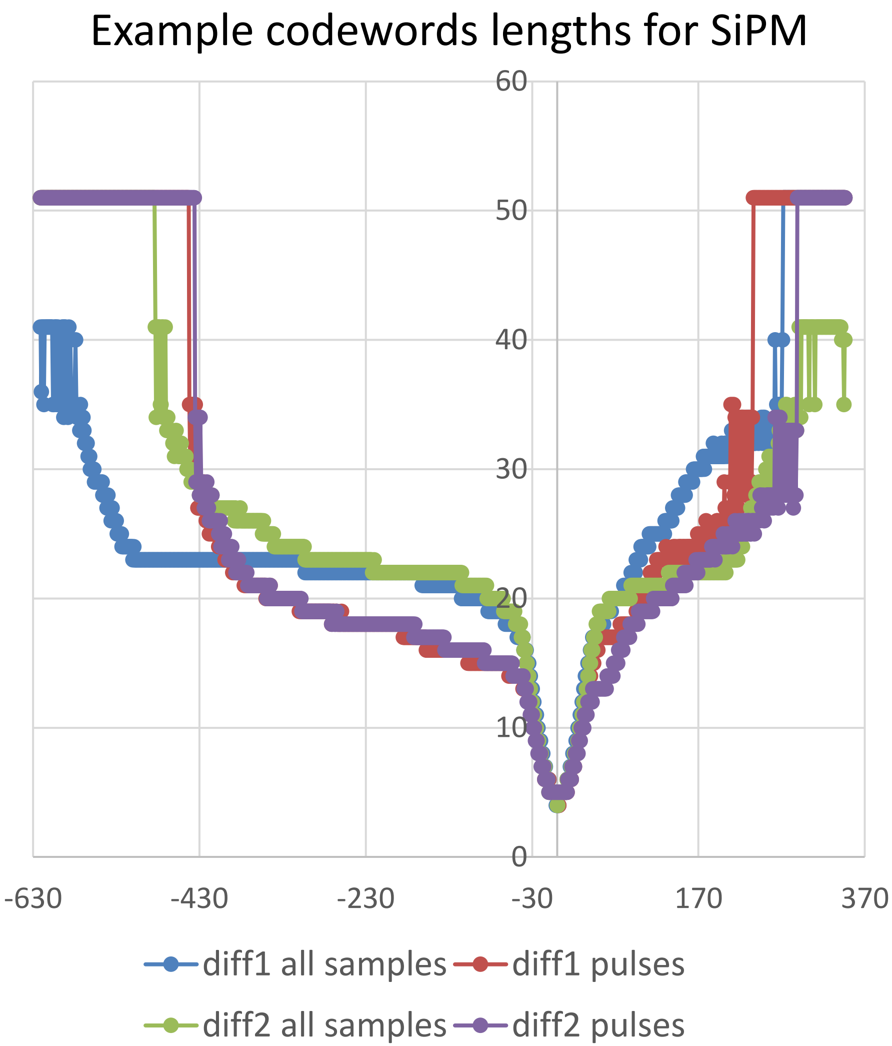


Code-stream formed as the concatenation of codewords



Hardware cost: ~500 LUTs in FPGA

Variable Length Coding (3)



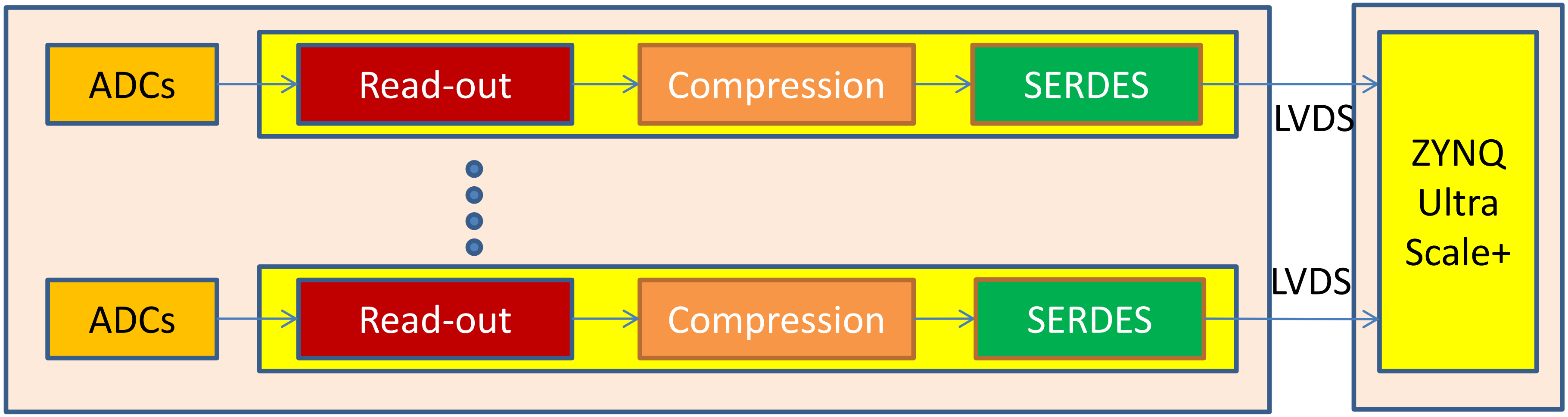
- Short codewords for most probable values close to 0
- Very small probable values can be coded on the fixed number of bits
- Possible division into subranges with separate Golomb codes – suitable for logic
- 1k RAM/ROM sufficient to map all samples

Compression Efficiency

- Lossless Coding of waveforms
  – Compression ratio: 2 to 3
  – About 5.3 bits per sample (bps) for 14-bit ADC for SiPM
  – It corresponds to input noise std. deviation ~50
  – Expected smaller values for 12-bit ADC for ECAL
- Selection of pulses strongly limits the amount of data to compress
  – Compression ratio depends on selection rate

FPGA – architecture

- Compression in basic version is not complex
  - LPC filter order determines the number of utilized multipliers in DSP units
  - Variable length coding is implemented as LUTs mapped to logic resources or on-chip RAM
  - Estimated resources are below 1k 6-bit-input LUTs per channel
- Parallel processing for 16 PMTs proportionally increases requirements on resources and power



Summary

- Waveform compression utilizes Linear Prediction Coding (LPC) and Huffman coding
- Lossless compression allows about 5.3 bps for SiPM
- Similar compression ratios for scenarios with all samples and pulses
  - Ratio depends on noise
  - Slightly higher bit-rates for pulses
- Amount of resources is moderate for multichannel implementations
  - Limitations results from multipliers in LPC filters
- Future works:
  - Verify the compression for PMTs/ECALs
  - Hardware design and implementation
  - Optimization of variable length coding by contexts