*Proceedings of the 14th International Conference "Reliability and Statistics in Transportation and Communication" (RelStat'14), 15–18 October 2014, Riga, Latvia, p. 53. ISBN 978-9984-818-70-2*
*Transport and Telecommunication Institute, Lomonosova 1, LV-1019, Riga, Latvia*

# EFFICIENT MEASURING OF COMPLEX PROGRAMMABLE LOGIC DEVICE (CPLD) IMPLEMENTED ANALOG-TO-DIGITAL CONVERTERS STATIC PARAMETERS

***Sergejs Šarkovskis[1], Sergejs Horošilovs[2]***

*[1] The Faculty of Computer Science and Telecommunication, Transport and Telecommunication Institute*
*Lomonosova street 1/4, LV-1019, Riga, Latvia, e-mail: sergejs.sarkovskis@gmail.com*
*[2]ALFA RPAP AS*
Ropazu street 140, LV-1006, Riga, Latvia, e-mail: *quazi_modo@yahoo.com*

The precision of analog-to-digital signal conversion plays an important role in digital communication systems. At the present moment, the histogram method is the method mainly used for measuring analog-to-digital converter (ADC) static parameters, such as differential nonlinearity (DNL) and integral nonlinearity (INL). In such case the ADC input signal is being changed linearly and output data is used to construct a histogram, thus calculating the number of specific codes occurrences. The main problem of such method implementation is necessity to store a large volume of data – ADC output code words during the collection of information. It is usual to use memory modules for such purpose. This article focuses on implementation of histogram method of ADC static parameters measuring based on inexpensive complex programmable logic device (CPLD) arrays. The article proposes histogram real-time calculation algorithm and it's realization on CPLD, which allows reducing volume of required memory or stopping using it entirely.

**Keywords:** analog-to-digital converter, differential nonlinearity, integral nonlinearity, histogram method.

## 1. Introduction

At the present moment, analog-to-digital converters (ADC) are widely used in the fields of electronics and communications. Digital signal processing is base tool in measuring devices; digital telephony and digital television systems; data collecting and such processing is impossible without high-quality ADC.

In ideal case ADC converts continuous analog signal into digital, usually binary coded. The minimal change of input signal available for conversion is quant, which is related to ADC bitness. This quant is least significant bit (LSB) and it determines resolution of ideal ADC. The LSB value can be estimated according to (Kester, 2005)

$$LSB = \frac{\Delta_{FS}}{2^N}, \tag{1}$$

where $\Delta_{FS}$ – is range of input signal, $N$ – ADC bitness. $\Delta_{FS}$ defines unipolar or bipolar ADC operation mode. In the first case input signal is in range between 0 and *FS*, the *FS* denotes full scale, which is often related to reference voltage of ADC. In case of bipolar operation mode, $\Delta_{FS}$ is in range between – *FS* and *FS*.

Depending on the ADC bitness *N* the output code varies in range between 0 and $2^N - 1$. Thus, for 3 bit ADC there are 8 possible levels, beginning with code 0 ($000_2$) and ending with code 7 ($111_2$). Transfer function (TF) of such ideal ADC in unipolar mode is shown in Figure 1a. The analog input is given as fraction of FS. Note, that all ones code corresponds to analog value *FS* – 1*LSB*.

In Figure 1a dots denote code transitions, which are separated by 1LSB beginning with 0.5LSB for ideal ADC. For real ADC it is impossible to define unambiguous level of input signal for each digital code. For the same digital code there is certain range of input signal, which doesn't exceed 1LSB in ideal case (Figure 1b). Such phenomenon is named a quantization uncertainty and it leads to difficulties in definition of code transitions (Kester, 2005).

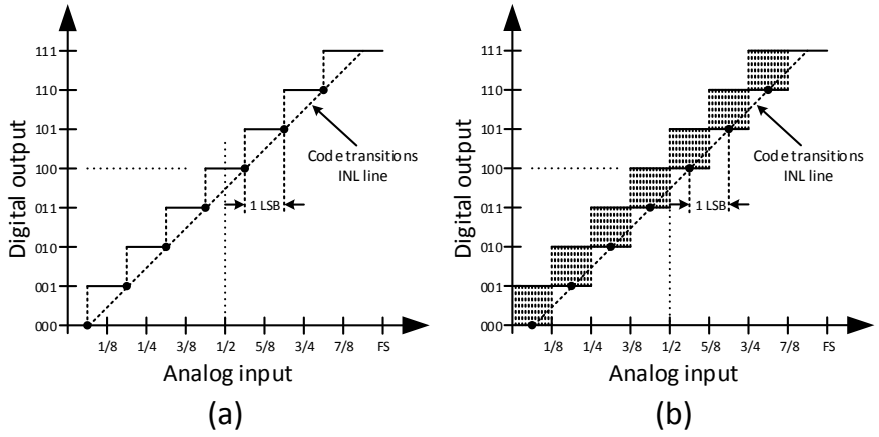


*Figure 1.* Ideal transfer function of 3 bit ADC

At the present moment converters in market vary in pricing, resolution, performance and quality. The last is defined by set of parameters, which can be divided into static and dynamic.

## 2. Histogram method

There are different methods for evaluation of ADC static parameters which are based on finding points of code transitions. The main problem related to defining these points is the noise of code transitions, especially in the presence of negative differential non-linearity (DNL). The possibility of such methods applicability is limited by a few tenths of an LSB peak-to-peak noise. Histogram method (code density method) can be used for evaluation of any ADC static parameters for any level of noise. In such case for input signal of specified form it is necessary to accumulate ADC output data and calculate occurrences of each code word. Input signal should slightly overload (recommended value if 10%) ADC. It is possible to use ramp signal for this purpose (Kester, 2005), the sequence of triangular pulses (Alegria, Arpaia etc, 2002) or harmonic function (Blair, 1994). For code words $k$ from 1 to $2^N - 2$ there are $M$ samples and each code word occurrence is being calculated. Note, that "overload" codes, such as all zero and all ones are not included in calculations. Theoretical number of occurrences for each code word $k$ can be evaluated according to (Kester, 2005)

$$n(k)_{\text{theoretical}} = \frac{M}{2^N - 2}. \tag{2}$$

If $n(k)_{\text{actual}}$ – is real number of occurrences for code word with number $k$ it is simple to calculate $\text{DNL}_k$ for this code word (Kester, 2005)

$$\text{DNL}_k = -\frac{n(k)_{\text{actual}}}{n(k)_{\text{theoretical}}} - 1. \tag{3}$$

DNL for full ADC can be calculated according to (Kester, 2005)

$$\text{DNL} = \max\left(\left|\text{DNL}_k\right|\right). \tag{4}$$

Figure 2 shows typical histogram with wide, narrow and missing code words.

For given $\text{DNL}_k$ it is possible to calculate integral non-linearity (INL) by usual addition (Kester, 2005)

$$\text{INL}_m = \sum_{k=1}^{m} \text{DNL}_k. \tag{5}$$

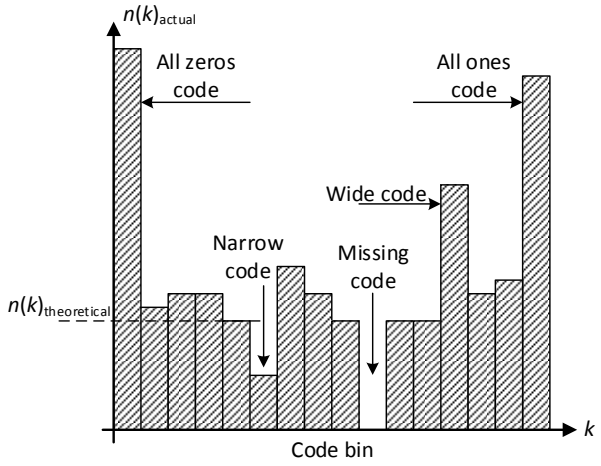


*Figure 2.* Typical histogram for ramp input signal

Histogram method removes the effect of ADC noise by calculating average value over entire code intervals and is ideally suited for testing of any ADC. At the present moment histogram method is standard method for static parameters measuring.

### 2.1. Implementation of histogram method

Figure 3 shows typical histogram method setup.

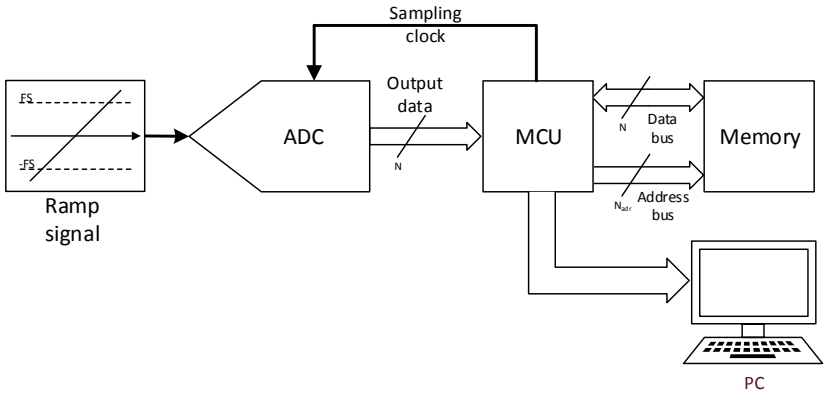


*Figure 3.* Histogram method setup

The ADC for testing purpose is driven by linear positive slope signal generator. A microcontroller provides clock for ADC, collects output data and copies them into memory. When testing is complete, the data is being sent to PC for further processing and required parameters calculation.

This article proposes to use complex programmable logic device (CPLD) arrays for histogram calculation and control process. The use of CPLD arrays provides different benefits, such as parallel and independent process control, high data processing rate and so on. The proposed histogram calculation algorithm applied to CPLD arrays reduces the volume of required memory or makes it possible to stop using it entirely.

The simplest way to calculate histogram by means of programmable logics uses the output code word as an address for corresponding counter. Every time the code word is being read the value of respective counter is increased by one unit, thus it is possible to form required histogram. When using such approach the functional scheme consists of one $N$-to-$2^N$ decoder and $2^N$ counters, where $N$ is the bitness. The problems of such approach are obvious – 10 bit ADC requires 10-to-1024 decoder and 1024 counters, at the same time 12 bit ADC requires 4096 counters and 12-to-4096 decoder. Such increase of scheme components requires CPLD arrays of higher capacity or conversion to FPGA arrays with high cost. It would be beneficial to implement universal scheme, which can be used for most ADC available on the market.

## 3. Windowed Histogram Method - WHM

The improved histogram calculation scheme proposed in this article is based in the first place on knowledge of the analog signal form. If the signal increases linearly it's logical to assume that ADC output code words will also monotonically advance from 0 to $2^N - 1$. At the beginning of measurements output codes will be localized at zero level and probability of high values is low. In such case it's possible to define the window with certain width and calculate histogram within its bounds while shifting the window if necessary in a direction of the code words increments. The resulting set of the histograms at the output should be summed as is shown in Figure 4.

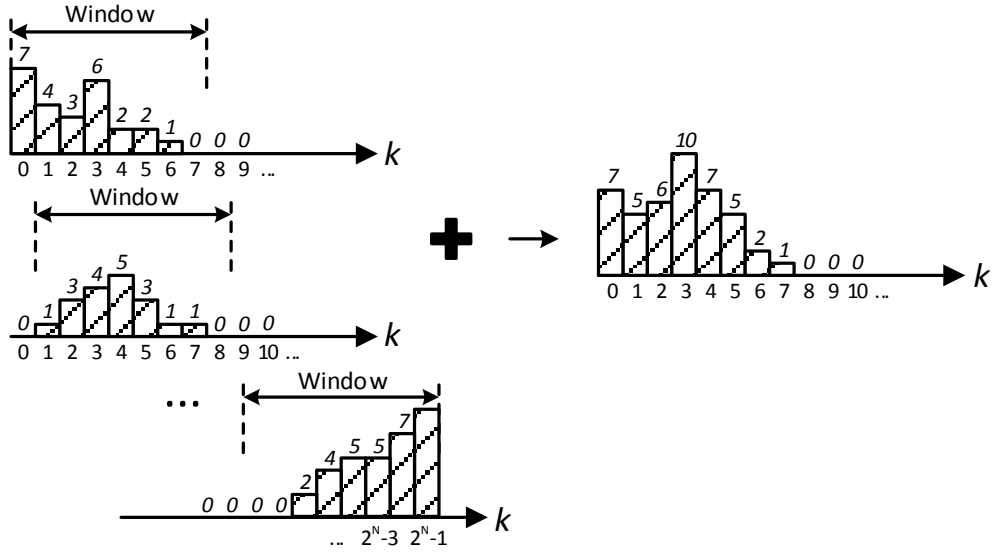


*Figure 4.* Windowed histogram method

Such approach, which is called the windowed histogram method, leads to two main questions:

1. what width of the window should be chosen;
2. when the window should be shifted.

The answer on the first question is directly connected with ADC code transition noise. For stronger noise the wider window is necessary. It is proposed to shift histogram in case, when during certain time interval there were no code words in the beginning of window.

When implementing WHM via CPLD array it was decided to choose the width of window equal to 8, i.e. the one, which corresponds to 3 bits. In fact, this means that ADC noise may not be stronger than 4-5 LSB. The choice of specific window width equal to 8 dictates the necessity to use 3-to-8 decoder and 8 counters, regardless of the ADC bitness. Decoder is driven by three low-order bits – those are numbers from 0 to 7, so that the value of the first counter increases by 1 when the input is 0, the number of second counter increases by 1 when the input is 1, and so on.

It has been decided to shift histogram at the moment, when there have been no code words from the first two bins during 10-15 clocks of the histogram window. The smaller is number of clocks impacting the shift the more often it should be shifted, and hence the more often output data should be saved. The higher number of clocks can force code words to leave the right border of the window, which is unacceptable.

When shifting window, it is necessary to edit input data, since only three low-order bits are being used. When the window shifts to the right, the bin of code word 0 departs and the bin of code word 8 arrives. Now three zeroes in ADC output low-order bits denote the code word 8 and further the value of the last counter is increased, instead of the value of the first counter. Such correction can be implemented by subtraction of the number of shifts from the input code and further addition on the modulo 8. The block diagram of such WHM algorithm is shown in Figure 5.

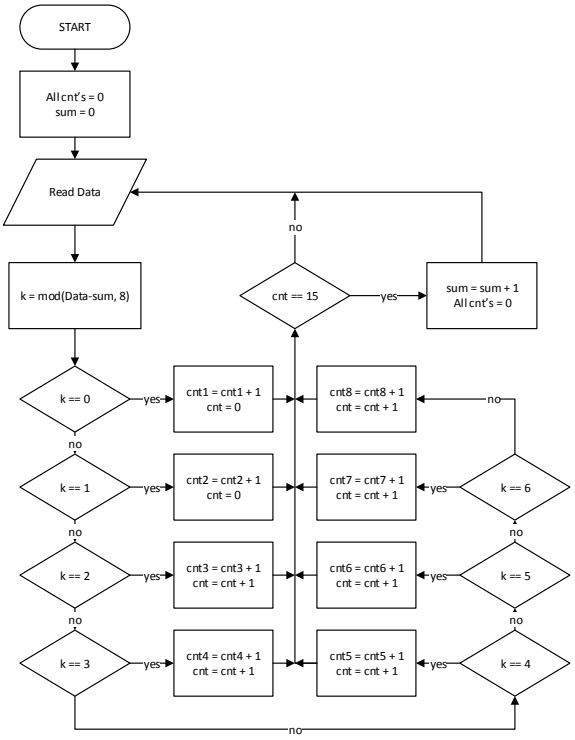


*Figure 5.* The block diagram of WHM algorithm

Here Data – is ADC data, cnt1..8 – 8 counters for window bins, cnt – clock counter for the 1-st and the 2-nd bins when there are no code words present, sum – addition of the shifts.

### 3.1. WHM implementation via ALTERA company CPLD family MAX3000A

It is expected to implement designed scheme in existing project with purpose of testing ADC parameters. This project includes signal generator based on 16 bit digital-to-analog converter (DAC), CPLD EPM3256ATC144-10 and coupling board for PC with 16 digital inputs/outputs. The scheme with WHM implementation was designed in Quartus II and its schematic is shown in Figure 6.

This scheme consists of one 3-to-8 decoder implemented in block lpm_decode0 and 8 counters implemented in blocks lpm_counter0. Counters length is set to 5 bits. This means these counters can be used to count up to 31 before reset occurs. The choice of such counters length isn't accidental and it's related to theoretical number of code word occurrence. The input of decoder is driven by three low-order bits of ADC – ADC[2..0]. The indication scheme for histogram shifting is implemented with counter lpm_counter3 and comparator lpm_compare0. If the upper two counters (representing two left bins of the window) have data, then OR2 element resets counter lpm_counter3, otherwise this counter is incremented by 1 after a new code word. When this value will reach the value of comparator, counter data from lpm_counter0 is sent to output register lpm_shiftreg0 with length of 40 bits (8 counters with 5 bit length) and the specified counters are then reset. In such case there is an output signal 'Shift'. Further to this, it is necessary to prevent the loss of data for overflow of these counters. This can be achieved by 'cout' outputs, which signal counters overflow and OR8 element. The signal of this element launches the similar to window shifting mechanism excepting the signal 'Shift'.

The code correction scheme for shifting is implemented by another counter lpm_counter2 and subtracter lpm_add_sub0. The counter lpm_counter2 is cyclic one, so modulo 8 adding is achieved automatically. Also, note that when overflow of counters lpm_counter0 occurs the window isn't shifted, so lpm_counter2 counter value doesn't increment.

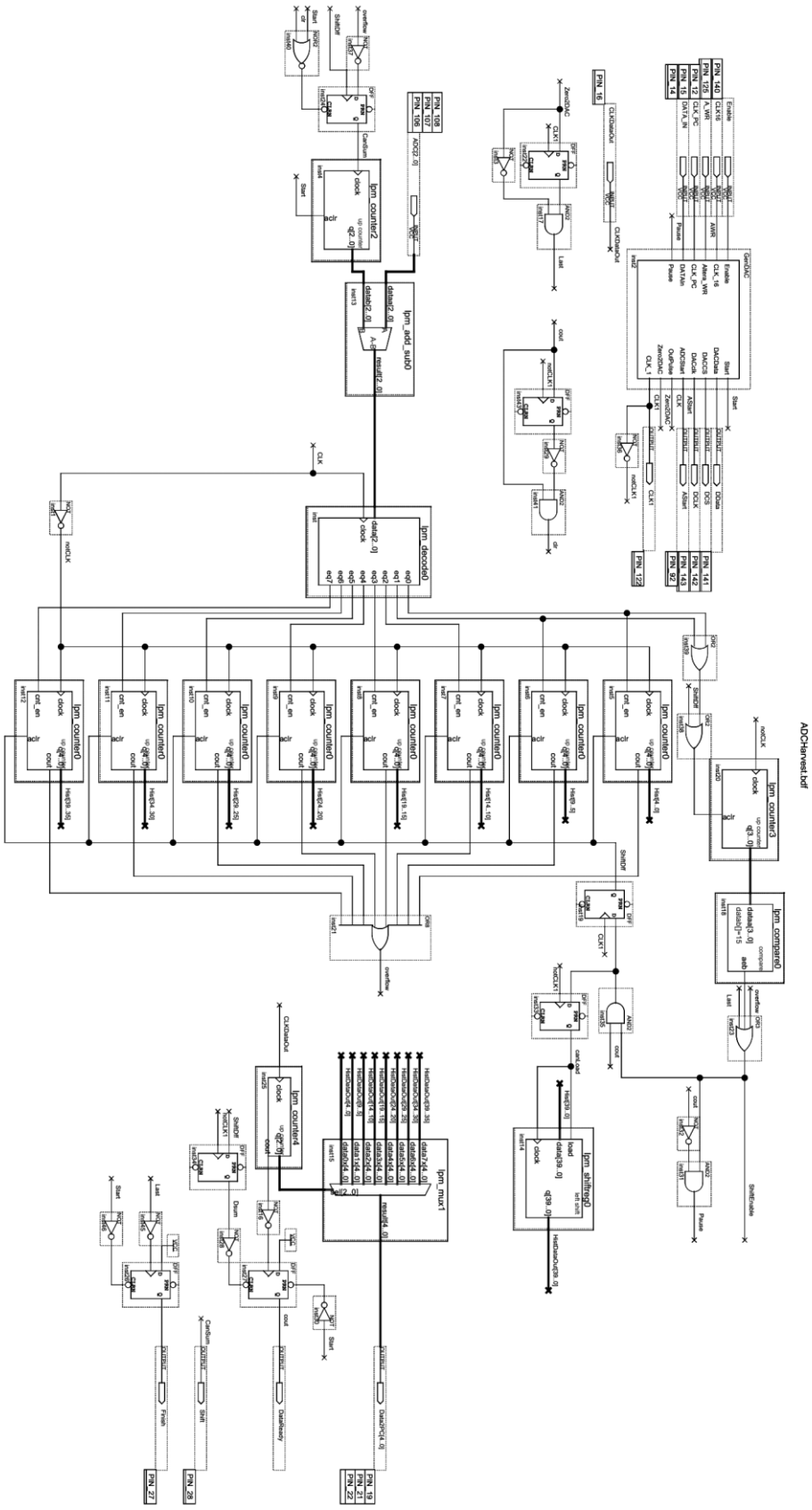


*Figure 6.* Functional schematic of WHM implementation in Quartus II

The data transferring to PC is realized with multiplexer lpm_mux2 and 'DataReady' signal, which indicates the status of collected data. With external clocking the data of output register is serially transferred to outputs 'Data2PC'. If the new shift of the window is necessary before the data has been transferred to PC, the scheme includes 'Pause' signal, which suspends execution of the algorithm. Such approach makes it possible to completely avoid the use of memory and send data directly to PC.

The scheme also includes block 'GenDac', which has been written in Verilog HDL language to form control signals for external DAC and clocks ADC in testing. In scheme also defines internal signal generation logics.

The full project volume is 205 macrocells of 256 available (80%) and used 26 ports of 116 available (22%).

## 4. Simulation

In order to test functionality of designed scheme it was necessary to create Test Bench using Verilog HDL language and execute simulation using ModelSim Altera Starter Edition software. In order to bring simulation to reality as close, as it is possible there is MATLAB program for input data simulation. The project uses analog signal generator based on 16 bit DAC and output voltage in range ±12.5V. Thus, the program simulated analog signal and relative ADC with noise was driven by this signal. Three low-order bits of ADC output code word were recorded to the text file, which was used in ModelSim simulation.

During the simulation output data (the window of histogram) was also recorded into text file, which has been used in MATLAB program to plot resulting histogram. Finally, this histogram is being compared with standard template histogram which has also been built in MATLAB for source data. Then it is possible to conclude, whether WHM algorithm works correctly.

ADC noise has been generated by random numbers generator with uniform distribution an expectation of E = 0. Probability density function (PDF) width can be chosen arbitrarily.

Below there are results of simulation for the following parameters:

- the relative 10 bits ADC in testing has reference voltage 10.24V, thus 1LSB = 1mV;
- the input signal varies in range from -1.024V to 11.264V (with 10% overload);
- the width of ADC noise PDF is 5LSB;
- artificially created missing codes – the 15-th and the 40-th.

The generated voltage and respective ADC code is shown in Figure 7.

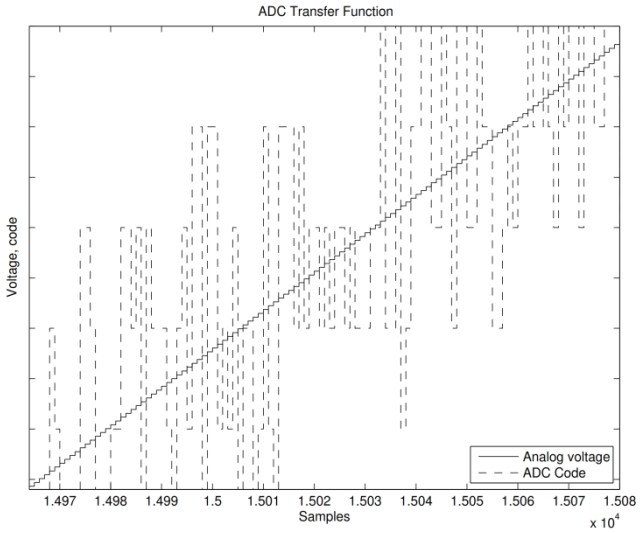


*Figure 7.* The input analog voltage and output ADC code

ModelSim output text file is formatted as following:

```
1   0   4 10   4   2   3   1   0
1   0   4   6   8   5   3   1   0
1   0   0   4   5   1   5   0   0
```

The first column indicates shifting signal, the rest of 8 columns are bins of the histogram. After simulation was complete, this data has been processed and used to plot resulting histogram. The example of such histogram for the first 50 bins is shown in Figure 8.

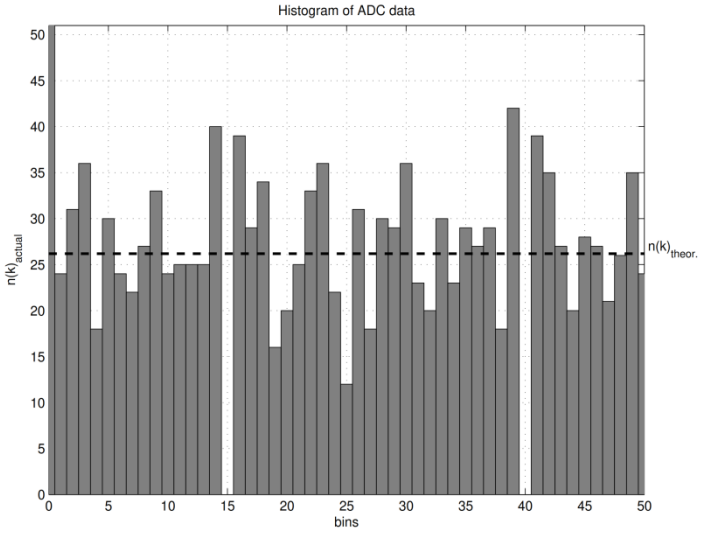


*Figure 8.* The part of resulting histogram

The histogram clearly shows the missing codes in the 15-th and the 40-th bins of histogram. The comparison of resulting histogram with standard template histogram shows no errors.

## 5. Conclusion

The article proposes new software implementation method for testing of ADC static parameters. This method is based on knowledge of input analog signal – it is ramp function. It is proposed to create histogram only for certain window with specified width, which is being shifted from low-order code words to high-order code words, rather than generate entire histogram at once. This method is named Windowed Histogram Method (WHM) and it has been implemented via ALTERA company CPLD array of the family MAX3000A with 256 macrocells. The designed scheme volume takes up 80% of the full CPLD array capacity. In order to test functionality of the WHM the simulation has been executed in ModelSim software, and the data for this simulation has been simulated in MATLAB. The input data has been intentionally noised with uniformly distributed noise with PDF width of 5LSB. As a result of this simulation there are histograms before shifting, which have been used to construct resulting ADC histogram in MATLAB. The WHM designed scheme has been implemented via cheap CPLD array (the cost at mouser.com is approx. 17.7€ when the article has been written) and it allows to avoid use of additional memory, which significantly decrease costs of the project.

## References

1. Alegria F., Arpaia P., da Cruz Serra A.M, Daponte P. (2002) Performance Analysis of an ADC Histogram Test Using Small Triangular Waves. *IEEE Transaction on Instrimentation and Measurement*, 51(4), 723-729.
2. Blair J. (1994) Histogram Measurement of ADC Nonlinearities Using Sine Waves. *IEEE Transaction on Instrimentation and Measurement*, 43(3), 373-383.
3. Kester W.A. (2005) *The Data Conversion Handbook.* Oxford: Elsevier. 976 p.