

Article

Development of Recurrent Method of Logarithmic Analogue-to-Digital Conversion on Switched Capacitors with Charge Redistribution for Increased Speed

Zynoviy Mychuda ¹, Mykola Mykyichuk ², Lesia Mychuda ³, Adam Szczesniak ^{4,*} and Zbigniew Szczesniak ⁵

¹ Department of the Computer-Assisted Systems of Automation, Lviv Polytechnic National University, 79-000 Lviv, Ukraine; zynovii.r.mychuda@lpnu.ua

² Faculty of Computer Technology, Automation and Metrology, Lviv Polytechnic National University, 79-000 Lviv, Ukraine; mykola.m.mykyichuk@lpnu.ua

³ Department of Information Technology Security, Lviv Polytechnic National University, 79-000 Lviv, Ukraine; lesia.z.mychuda@lpnu.ua

⁴ Faculty of Mechatronics and Mechanical Engineering, Kielce University of Technology, 25-314 Kielce, Poland

⁵ Faculty of Electrical Engineering, Automation and Computer Science, Kielce University of Technology, 25-314 Kielce, Poland; z.szczesniak@tu.kielce.pl

* Correspondence: adam_szczesniak@o2.pl

Abstract

This article is devoted to the development of a recurrent method of logarithmic analogue–digital conversion on switched capacitors with charge redistribution, which differs from known methods by the fact that multiple returns will only affect the older bit. The essence of the method and its implementation are presented. Mathematical models of recurrent logarithmic ADCs (LADCs) based on this method have been developed, analysis has been carried out, and an assessment of errors and speed is given. It is shown that such recurrent LADCs have four or more times higher speed than known analogues. For example, 10-bit LADCs in the range of input voltages from 1 mV to 10 V provide 12-bit and 14-bit accuracy, the conversion time is less than 203 μ s and 748 μ s with the number of conversion tacts being 81 and 299, and the error of the output code is no more than 0.003% and 0.0007%, respectively.

Keywords: logarithmic analogue to digital conversion; recurrent method; switched capacitors; speed; accuracy



Academic Editors: Salvador Alepuz and Nilanjan Ray Chaudhuri

Received: 26 February 2025

Revised: 14 May 2025

Accepted: 24 June 2025

Published: 27 June 2025

Citation: Mychuda, Z.; Mykyichuk, M.; Mychuda, L.; Szczesniak, A.; Szczesniak, Z. Development of Recurrent Method of Logarithmic Analogue-to-Digital Conversion on Switched Capacitors with Charge Redistribution for Increased Speed. *Electronics* **2025**, *14*, 2590. <https://doi.org/10.3390/electronics14132590>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Analogue-to-digital converters (ADCs) are essential functional units of modern computer systems with a wide range of applications. The characteristics of ADCs largely determine the properties of such systems. ADCs are divided into linear and nonlinear types based on their processing characteristics. The vast majority of nonlinear ADCs are logarithmic. Logarithmic analogue-to-digital converters (LADCs) attract particular attention from designers and engineers. In addition to converting analogue values into digital values, LADCs provide several additional features, including the following: the compression of the input signal range, a constant value of relative processing error, the linearisation of processing characteristics, and the ability to perform logarithmic arithmetic operations. The use of logarithmic arithmetic significantly increases processing speed, since time-consuming operations of conventional arithmetic, such as multiplication and division in computer systems, are executed in logarithmic arithmetic as quick addition

and subtraction operations. This is particularly important for real-time systems, such as those used in process control, robotics, telecommunications, aviation, space exploration, and other fields where minimising information processing delays is critical.

The initial solutions for logarithmic ADCs exhibited significantly inferior characteristics compared to linear ADCs. However, with the development of LADCs using switched capacitors at Lviv Polytechnic University, the situation has changed drastically: the characteristics of LADCs and linear ADCs have become nearly equivalent, resulting in the emergence of a new class of LADCs.

Pipeline LADCs with switched capacitors have proven convenient in integrated circuit technology as they consist of a series of identical functional units [1–3].

In [1], a pipeline logarithmic analogue-to-digital converter with switched capacitors is described, which does not require quadrature or any other complex function in analogue representation. This approach is attractive in scenarios where a wide dynamic range of input signals is required but with a low peak signal-to-noise ratio. The architecture of the developed pipeline LADC and the functional diagrams of its main nodes are presented. The main nodes are implemented using switched capacitors, while the reference voltage setting unit is realised with resistors. The designed and fabricated pipeline LADC, based on 0.18 μm CMOS technology, features an 8-bit output signal, a dynamic range of input signals of 80 dB (absolute input signal values ranging from 0.4 mV to 1 V), and a signal-to-noise ratio of 36 dB, and occupies an area of 0.56 mm^2 with a power consumption of 2.54 mW.

In [2], an ADC with a truly logarithmic pipeline architecture is proposed. This converter processes the input signal in the same way as a linear converter, except in the logarithmic domain. The article describes the construction of a logarithmic pipeline ADC, implemented as an integrated circuit in 0.25 μm CMOS technology. The converter is characterised by a 9-bit output code, an input signal dynamic range of 80 dB, a signal-to-noise ratio of 44.3 dB, and a power dissipation of 478 mW.

In [3], the development and fabrication of a low-power pipeline logarithmic analogue-to-digital converter in an integrated design using 0.35 μm CMOS technology are described. The LADC is designed for processing sub-microampere analogue output currents from pH sensors.

The exceptionally high resistance of MOS transistors and the translinear conversion principle enable the simple design and low power consumption of the LADC. The LADC was modelled with a supply voltage of 1V. It dissipates 3.3 μW of static power at a clock frequency of 1 kHz. The LADC's output code is 8-bit. The range of input current signals is from 0.7 nA to 100 nA.

The flash LADCs proposed in Patent [4], with resistive cells, are also promising for fabrication in integrated circuit technology. Their relative processing error is less than 0.4%, with a processing time of 10–20 ns.

Recurrent LADCs proposed in [5] were studied in Work [6]. They enable the achievement of an accuracy 3–4 bits higher than the nominal bit count of the LADC. The desired accuracy in recurrent LADCs is achieved with fewer sample values compared to SAR ADCs.

SAR LADCs on switched capacitors are described in Work [7], where the principles of operation, modelling, and error analysis of these converters were developed. In the designed SAR ADC, 16-bit accuracy was achieved in the input voltage range from 1 mV to 10 V.

To enhance speed and accuracy, a new principle for logarithmic analogue-to-digital conversion on switched capacitors with a variable logarithmic base was proposed in [8]. Unlike known methods, the proposed approach compensates for the absolute processing error during each processing cycle until a specified error value is reached. In LADCs

based on the proposed method, with an input voltage range of 1 mV to 10 V, the relative processing error decreases to 0.01% with a processing time of no more than 105 μ s.

In Patent [9], ADC error correction is implemented using an error signal obtained by integrating the difference between the input charge and feedback pulses.

A comparison of logarithmic and linear ADCs is provided in [10] based on biomedical applications, where signals have a wide range. It has been demonstrated that logarithmic ADCs perform better in the region of smaller signals. However, for large amplitudes, linear ADCs exhibit a smaller absolute error.

Paper [11] presented research on a logarithmic voltage-to-time converter for use in novel logarithmic analogue-to-digital converter architectures that do not require analogue blocks, such as amplifiers, with signal processing being carried out in the time domain to the maximum possible extent. The time domain resolution increases.

Paper [12] describes a logarithmic compression ADC using a subranging TDC and the transient response of a comparator. Comparator settling time is used here for logarithmic compression instead of a logarithmic amplifier. The settling time of the comparator is inversely proportional to the logarithm of input voltage. In the proposed ADC, input voltage is converted into a pulse whose width represents the settling time of the comparator. Subsequently, the TDC converts the pulse width into a binary code.

The results of the ADC tests on switched capacitors with charge redistribution with successive approximations are provided in Paper [13], in which the principles of the construction and error analysis of these converters were developed.

Ref. [14] is a manual on charge pumps for ultra-low-power applications. It presents the most modern integrated topologies. Recommendations are given for the selection of the optimal solution for given design requirements.

In [15], a compact and energy-efficient SAR ADC is described, where noise is reduced through error-based feedback (EF). The prototype ADC was fabricated using 65 nm 1P9M CMOS technology.

In [16], a 14-bit 20 MS/s energy-efficient SAR ADC using 65 nm CMOS technology is described for portable medical ultrasound systems. To improve the linearity of the SAR ADC, a digital background mismatch calibration technique was employed. Additionally, a compact noise reduction technique is provided.

In [17], a 7-bit dual-channel time-interleaved ADC and a two-clock-cycle flash ADC with a speed of 3 Gb/s are presented. The ADC incorporates a built-in reference voltage source based on a capacitive DAC and an advanced offset calibration technique. The prototype ADC was implemented in 40 nm CMOS technology.

A continuous-time pipeline ADC was proposed in [18]. This is a new architecture that implements the equivalent of a smoothing filter followed by an ADC. It has been shown that the SNDR of the proposed ADC is significantly higher than that of traditional ADCs. This theory was tested using a macro model and transistor-level simulations.

Research into additive successive approximation methods has yielded a novel approach: bitwise balancing [19]. This innovative method, supported by a robust mathematical model, promises to advance analogue-to-digital conversion by significantly reducing the conversion time. Empirical analysis suggests that this reduction could range from 6% to 25%.

In [20] a discretisation of the direction of arrival of signals in phase and in quadrature by different ADCs was proposed using the Cramer–Rao lower bound (CRB), which, unlike existing schemes, allowed for the optimisation of ADC placement.

In [21] a single-bit delta–sigma modulator ($\Delta\Sigma$) based on a switched-capacitor inverter for ADCs was presented. The modulator is characterised by its power consumption in the range of nanowatts, which is particularly important for portable devices. The

experimental results of nine prototypes implemented in a standard 180 nm CMOS process were presented: the peak SINAD value was 77.8 dB and SNR was 79.4 dB, with a power consumption of 71.5 nW and a supply voltage of 900 mV.

In this article, we propose an improvement to the recurrent method of logarithmic analogue-to-digital conversion on switched capacitors with charge redistribution to increase speed while maintaining the required accuracy.

The essence of recurrent algorithms lies in storing the last value and using it for further calculations to enhance the accuracy of the result. In the case of analogue-to-digital conversion, the repeating algorithm operates slightly differently. Recurrent ADCs are similar to SAR ADCs. Thus, a series of etalons (standards)—sample reference voltages that are multiples of each other—is needed.

In recurrent LADCs, processing consists of several subbands in which the input voltage U_{IN} is compared with the compensation voltage U_k in each processing cycle. In turn, the compensating voltage is equal to the product of the reference voltages of the individual bit values, which are alternately requested from the most significant to the least significant. After querying the least significant bit of the i -th subrange, the value of the compensating voltage (U_{ki}) corresponding to that subrange is stored and used for the next $(i + 1)$ -th subrange as the initial value U_k . When, at a certain stage of processing, the compensating voltage $U_{k(n+1)}$ passes through the U_{IN} level, further processing proceeds using the SAR encoding method (bit by bit) with the initial value of the compensating voltage U_{kn} . Ultimately, the compensating voltage will be equal to the product of the reference voltage (U_O) and the weights of all the bits where it was greater than the input voltage U_{IN} .

The output code is generated by comparing the input voltage U_{IN} with the compensating voltage U_k , and the current values of U_k smaller than U_{IN} are discarded, with zeros inserted in the corresponding output code bits.

A characteristic feature of the proposed improved recurrent method for logarithmic analogue-to-digital conversion is the presence of only two processing subranges. In the first subrange, the compensating voltage is generated by the product of the reference voltage of only the first bit, followed by multiple returns. In the second subrange, the compensating voltage is generated as the product of the sample voltages of each successive bit, but only once.

The advantage of recurrent LADCs on switched capacitors is their greater productivity and the direct dependence of the type of output code on the method of shaping the weighting coefficients of the bits and reference voltages. However, their speed and accuracy are typical for SAR ADCs.

The novelty of this article, compared to Publications [5,6], is that

- (1) In order to increase speed while maintaining the required accuracy, the method of recurrent analogue-to-digital conversion has been improved, differing from known methods by repeatedly referencing only the reference voltage of the most significant bit;
- (2) Electric models of the improved recurrent LADCs and mathematical models of their errors are being developed, with errors and processing speed being evaluated.

The aim of this article is to develop the electrical model of the converters and mathematical models of errors, and to evaluate the errors and speed of recurrent LADCs on switched capacitors with charge redistribution, based on the proposed method, considering the parameters of contemporary components.

In this study, we used the methods of computer modelling and a computer experiment to study the errors and conversion times and the equations that we derived from the well-known laws of electrical engineering and electrostatics.

The developed models relate to the implementation of the recurrent LADC on switched capacitors proposed by us, whose block diagram is presented in a simplified form in Figure 1 and which constitutes an improvement over the known LADC [5]. The improvement lies in increasing the speed while maintaining the required accuracy by repeatedly using the reference voltage of the most significant bit.

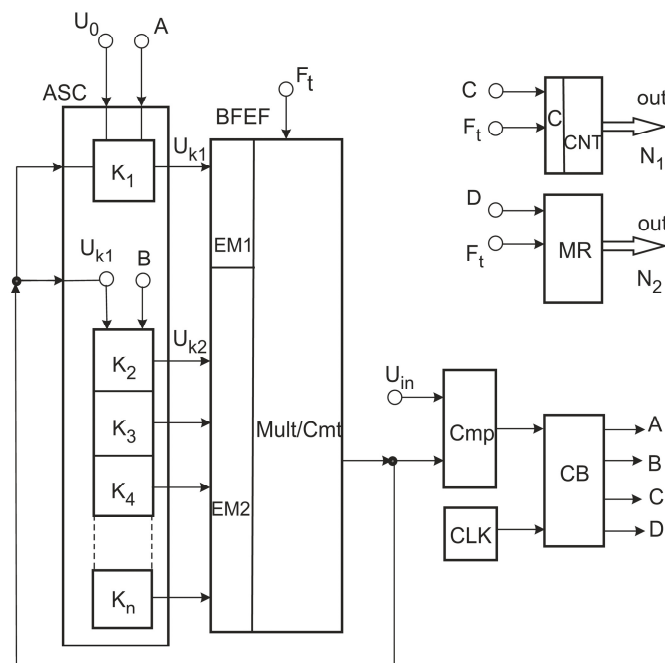


Figure 1. A simplified diagram of a logarithmic analogue-to-digital converter in which the proposed improved recurrent method is implemented.

2. Physical Model of an LADC Implementing the Improved Recurrent Method of Logarithmic Analogue-to-Digital Conversion

The proposed logarithmic analogue-to-digital conversion recurrent method is implemented using the diagrams shown in Figures 1 and 2—voltage diagrams illustrating the operation of this converter; the solid line and dots show the voltages at the input and output of the adjustable scaling converter, respectively.

Figure 1 shows the following: ASC—adjustable scaling converter; BFEF—block of formation of an exponential function; Cmp—comparator; CLK—clock; CB—control block; CNT—counter; MR—memory register; Mult/Cmt—Multiplier/Commutator; and EM1 and EM2—first and second elements of memory.

Marked in the figure are the following: U_O —reference voltage; U_{IN} —input voltage; F_t —clock frequency; N_1 and N_2 —higher and lower bits of the output code, respectively; U_{k1} and U_{k2} —compensation voltage on the first and second sub-scopes; $K_1 - K_n$ —transfer coefficients of the adjustable scaling converter, ASC, on inputs 1 – n ; and A , B , C and D —control signals.

The values of the voltage transmission coefficients (K_i) at the inputs of the ASC for any i -th input are preset according to the formula

$$K_i = \zeta \frac{N_H}{2^i}, \quad (1)$$

where N_H —the nominal value of the output code;

$\zeta = const$; moreover, $\zeta < 1$, and its value depends on the required accuracy.

The coefficient ζ is the base of the logarithm and sets the value of the discrete increments of the transmission coefficient of the ASC; the base of the logarithm is determined by the expression

$$\zeta = \frac{1}{1 + \frac{\delta}{100\%}} \quad (2)$$

where δ —the value of the desired error expressed as a percentage.

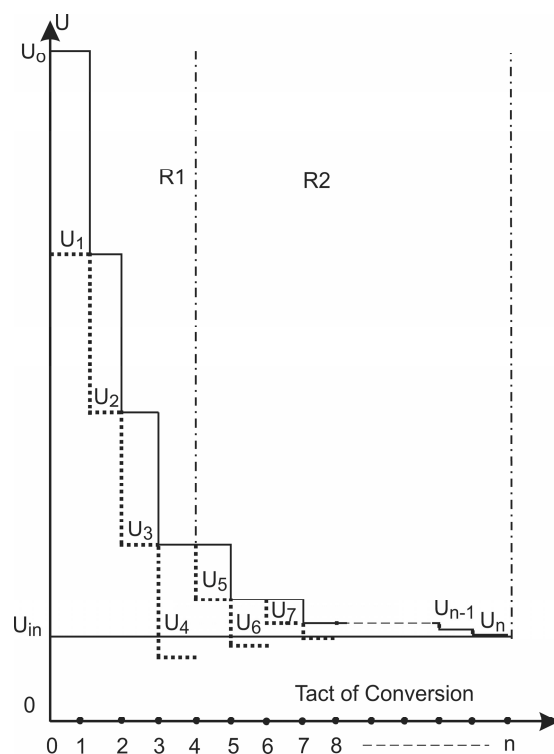


Figure 2. Voltage graphs illustrating the operation of a logarithmic analogue-to-digital converter in which the proposed improved recurrent method is implemented.

In converters with charge redistribution, ζ is defined by the ratio of the capacitances of the storage and dosing capacitors:

$$\zeta = \frac{C_D}{C_D + C_A} < 1, \text{ where } C_D \ll C_A \quad (3)$$

and the nominal output code (N_H) can be determined from the selected logarithm base:

$$N_H = \frac{1}{\ln \zeta} \cdot \ln \frac{U_{inmin}}{U_{inmax}} \quad (4)$$

The operation of the logarithmic analogue-to-digital converter shown in Figure 1 proceeds as follows:

The control unit (CB), using an initial setup signal, resets the result counter (CNT) and the result register (MR), stores the reference voltage level U_O in the first memory element (EM1), and generates control signals A and C, which enable the converter to operate in the first subrange. The signal A, applied to the control input of the scaling converter (ASC), activates the first input of this converter, while the signal C enables the operation of the result counter (CNT).

In the first subrange of processing, when the output of the comparator, Cmp, is at a logical high level, the control unit retains the signals A and C, thereby maintaining the converter's operation within the first subrange. Upon the first activation of the comparator,

Cmp, indicated by its output switching to a logical low level, the control unit, CB, disables the signals A and C and transitions the converter to the second subrange.

The maximum transfer coefficient of the scaling converter, ASC, will be applied to the first input, that is, when $i = 1$,

$$K_1 = \zeta^{\frac{N_H}{2}} \quad (5)$$

After the initial setup, the processing begins.

In the first subrange of processing, during the duration of the first clock pulse, the voltage at the output of the ASC assumes the value

$$U_1 = U_0 \cdot K_1 = U_0 \cdot \zeta^{\frac{N_H}{2}} \quad (6)$$

This voltage U_1 is recorded in the EM2 memory element and fed to the output of the BFEF indicator function generation block, from which it is fed to the Cmp, the comparator, and compared with the input signal U_{IN} .

When the voltage at the output of the exponential function generation block, BFEF, is greater than the input signal, the comparator, Cmp, sets its output to a logical one. Otherwise, it switches to a logical zero.

In Figure 2, the first case is illustrated, where the voltage is greater, and the comparator, Cmp, output is at a logical one level.

Therefore, the first clock pulse is recorded in the result counter, CNT.

The second clock pulse connects the output voltage of EM2 to the first input of the scaling converter, ASC, and the first memory element, EM1, to the output of the scaling converter, ASC, where the voltage assumes the value

$$U_2 = U_1 \cdot \zeta^{\frac{N_H}{2}} = U_0 \cdot \zeta^{2 \cdot \frac{N_H}{2}} \quad (7)$$

The voltage U_2 (Figure 2) exceeds the input signal level, and the comparator maintains a logical one level at its output.

Since the state of the comparator, Cmp, has not changed, the operation of the second clock pulse is similar to that of the first, meaning that the second pulse is recorded in the result counter, CNT.

The third clock pulse connects the first memory element, EM1, to the input of the scaling converter, ASC, and the second memory element, EM2, to the output of the scaling converter, ASC, where the voltage assumes the value

$$U_3 = U_2 \cdot \zeta^{\frac{N_H}{2}} = U_0 \cdot \zeta^{3 \cdot \frac{N_H}{2}} \quad (8)$$

The voltage U_3 (Figure 2) exceeds the input signal level, and the comparator maintains a logical one level at its output.

Since the state of the comparator, Cmp, has not changed, the operation of the third clock pulse is similar to the previous one, meaning that the third pulse is recorded in the result counter, CNT.

The fourth clock pulse connects the second memory element, EM2, to the input of the scaling converter, ASC, and the first memory element, EM1, to the output of the scaling converter, ASC, where the voltage assumes the value

$$U_4 = U_3 \cdot \zeta^{\frac{N_H}{2}} = U_0 \cdot \zeta^{4 \cdot \frac{N_H}{2}} \quad (9)$$

Since the voltage U_4 is lower than the input voltage, the output of the comparator, Cmp, transitions to a logical zero state. In this state of the comparator, the control unit, CB, prevents the clock pulses from being passed to the result counter, CNT, and to the clock

input of the BFEF. This means that the memory elements EM1 and EM2 remain connected to the regulated scaling converter, ASC, as they were after the third pulse.

The number of clock pulses n_1 recorded in the result counter, CNT, determines the value of the higher bits (N_1) of the output code of the logarithmic analogue-to-digital converter (Figure 1) according to the formula

$$N_1 = n_1 \cdot w_1 \quad (10)$$

where w_1 —the weight of the first (highest) bit of the output code of the converter, which is equal to $w_1 = \frac{N}{2}$.

Thus, the value of the higher bits of the output code of the logarithmic analogue-to-digital converter shown in Figure 2 is equal to

$$N_1 = 3 \cdot \frac{N}{2} \quad (11)$$

After the first transition of the comparator from logical state 1 to logical state 0 (in the example shown in Figure 2, after the fourth clock pulse), the processing in the second subrange begins as soon as the converter is activated.

In the second processing subrange, the control unit, CB, generates control signals B and D. Signal B enables the operation of the ASC at inputs 2 through n , while signal D allows the operation of the result register, MR.

The fifth clock pulse writes a logical zero to the first bit of the result register, MR, and activates the second input of the regulated scaling converter, ASC, thereby setting the second value for the transfer coefficient:

$$K_2 = \zeta^{\frac{N_H}{2^2}} \quad (12)$$

Therefore, the voltage at the output of the ASC, and thus at the output of the BFEF for generating the indicator function, becomes equal to

$$U_5 = U_3 \cdot K_2 = U_0 \cdot \zeta^{3 \cdot \frac{N_H}{2}} \cdot \zeta^{\frac{N_H}{2^2}} \quad (13)$$

The voltage U_5 is recorded in the first memory element, EM1, connected to the output of the scaling converter, ASC.

Since the voltage U_5 (Figure 2) exceeds the input signal level, the comparator, Cmp, switches to a logic high state at its output, and the control unit, CB, enables the clock pulses to pass to the clock input of the BFEF.

The sixth clock pulse writes a logic high to the second bit of the result register, MR, and sets the third transfer coefficient value of the ASC. In the same clock pulse, the voltage from the first memory element, EM1, is applied to the input of the ASC, and the output voltage of the ASC becomes equal to

$$U_6 = U_5 \cdot K_3 = U_0 \cdot \zeta^{3 \cdot \frac{N_H}{2}} \cdot \zeta^{\frac{N_H}{2^2}} \cdot \zeta^{\frac{N_H}{2^3}} \quad (14)$$

Since this voltage U_6 is lower than the input voltage (Figure 2), the comparator, Cmp, switches to a logic zero state at its output. After this logic low state, the control unit, CB, prevents the clock pulses from passing to the clock input of the BFEF, and the memory elements EM1 and EM2 remain connected to the regulated scaling converter, ASC, as in the state after the fifth clock pulse.

The seventh clock pulse has the same effect as the fifth (it only sets the fourth transfer coefficient value of the ASC), and so on.

After n clock pulses have passed in the second processing subrange, the voltage at the output of the BFEF will have the value

$$U_{n_2} = U_{n_1} \cdot \prod_{i=2}^{i=n} \zeta^{A_i \cdot \frac{N}{2^i}} \quad (15)$$

where U_{n_1} —the voltage value of the first subrange after n_1 clock pulses;

A_i —the coefficient that in each i -th processing cycle takes the value of 1 or 0, depending on the state of the comparator (logic “1” or logic “0”, respectively).

The voltage value U_{n_2} in the second subrange after (n) clock pulses, i.e., after connecting the last n -th input of the scaling converter, will be equal to the input voltage

$$U_{n_2} = U_{IN} \quad (16)$$

with an error not exceeding the weight of the last lower bit.

The $n + 1$ -th clock pulse in the result register, MR, will store the value of the lower bits of the output code (N_2) of the logarithmic analogue-to-digital converter (Figure 1).

$$N_2 = \sum_{i=2}^n A_i \cdot \frac{N}{2^i} \quad (17)$$

Thus, the output code (N) of the logarithmic analogue-to-digital converter, in which the proposed recurrent method is implemented (Figure 1), is equal to the sum of the codes in the first and second subranges:

$$N = N_1 + N_2 \text{ or } N = \frac{N_H}{2} \cdot \left(n_1 + \sum_{i=2}^n A_i \cdot \frac{1}{2^{i-1}} \right) \quad (18)$$

which means it is proportional to the logarithm of the ratio of the input voltage U_{IN} to the reference voltage U_O :

$$N = \frac{1}{\log \zeta} \cdot \log \frac{U_{IN}}{U_O} \quad (19)$$

Therefore, the converter developed based on the proposed improved recurrent method belongs to logarithmic ADCs and is characterised by increased speed compared to classical recurrent LADCs.

3. Mathematical Modelling of the Improved Recurrent Method of Logarithmic Analogue-to-Digital Conversion

Similarly to the classical recurrent LADC [6], the proposed improved recurrent LADC with switched capacitors is a charge-redistribution converter. Its logarithmic base is linked to the processing error according to Expression (2). Therefore, the user can select the desired accuracy before starting the conversion process. The maximum number of reference voltages corresponds, as in the classical recurrent method, to the specified number of output code bits.

According to the proposed improvements to the classical recurrent analogue-to-digital conversion method, multiple returns will only affect the older bit, as shown in Figure 2. This method significantly increases the processing speed.

3.1. Development of the Algorithm for the Improved Recurrent Method

The block diagram of the algorithm developed for the improved recurrent LADC is shown in Figure 3.

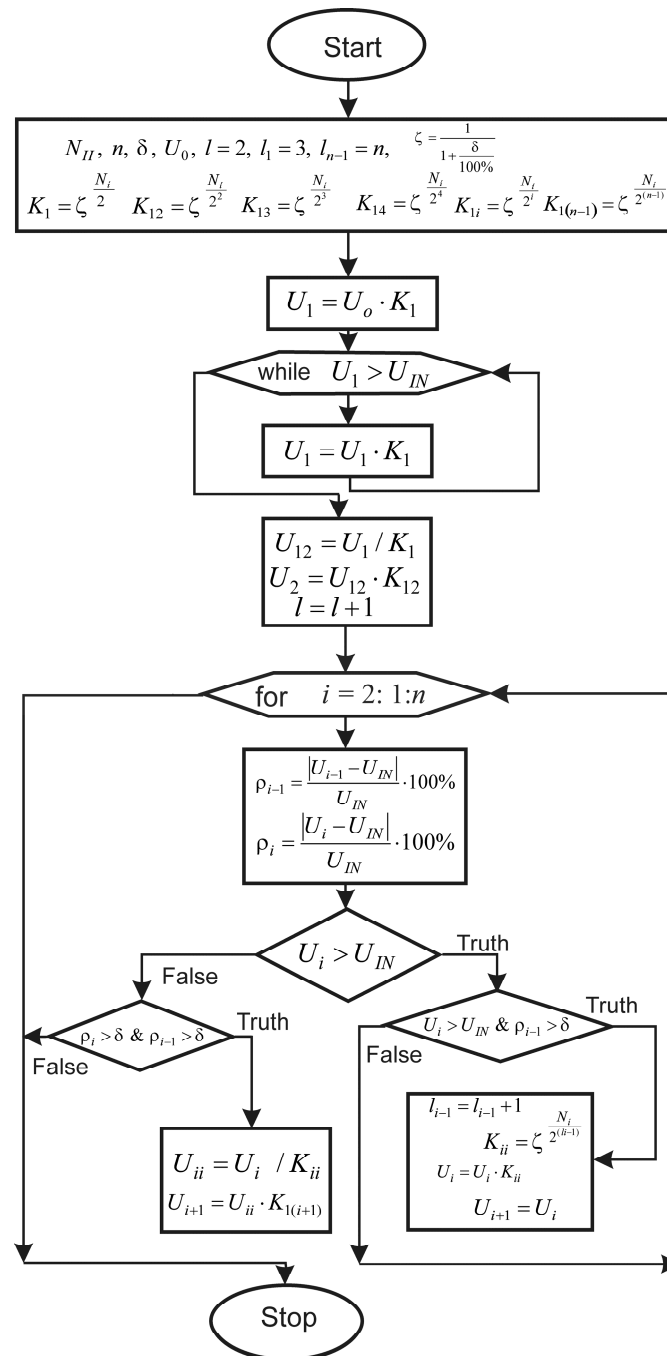


Figure 3. Block diagram of the algorithm of the improved recurrent method of logarithmic analogue-to-digital conversion.

The initial values for the processing are as follows: the recurrent voltage U_0 , a specified error value δ , which is set by the user, and the number of coefficients K_i .

Using the scaling converter, ASC, a certain number of coefficients K_i are generated. It should be noted that the proposed algorithm allows the use of only some of the standards to achieve the desired accuracy. The value of the coefficients K_i is influenced by the logarithmic base ζ and the specified number of bits for the converter through the corresponding nominal output code N_H ; for example, $N_H = 256$ for 8 bits, and $N_H = 1024$ for 10 bits. In accordance with (5), (12), and (14),

$$K_1 = \zeta^{\frac{N_H}{2}}, K_2 = \zeta^{\frac{N_H}{2^2}}, K_3 = \zeta^{\frac{N_H}{2^3}}, \dots, K_i = \zeta^{\frac{N_H}{2^i}} \quad (20)$$

In the first subrange, the compensating voltage is compared with the input voltage. If it is greater, each subsequent value is obtained as the product of the previous one and the coefficient K_1 . This means that we have a classic recurrent algorithm.

The maximum error in this subrange can be determined based on Equations (7)–(9) through the increase in the compensating voltage:

$$\Delta U_n = U_4 - U_3 = U_0 \cdot \zeta^{4 \cdot \frac{N_H}{2}} - U_0 \cdot \zeta^{3 \cdot \frac{N_H}{2}} = U_0 \cdot \zeta^{\frac{N_H}{2}} \quad (21)$$

and

$$\delta = \frac{\Delta U_n}{U_0} \cdot 100\% = \frac{U_0 \cdot \zeta^{\frac{N_H}{2}}}{U_0} \cdot 100\% = \zeta^{\frac{N_H}{2}} \cdot 100\% \quad (22)$$

After registering the transition through the input voltage level, we return to the previous compensating voltage value.

Subsequent values are then obtained by multiplying the current value by the coefficient K_{ii} when the compensating voltage is greater than the input voltage.

Therefore, in the second subrange of processing, the algorithm is a combination of the processing of the classic recurrent converter and SAR. This significantly increases the speed of devices based on the proposed method.

In the second subrange, the increase in compensating voltage, and consequently the maximum error, will depend on which bits are included in the output code. As in the classical recurrent method, in the bits corresponding to compensating voltage values lower than the input voltage, zeros are placed. For the example from Figure 2, taking into account Equations (12)–(14), the increase in compensating voltage is as follows:

$$\begin{aligned} \Delta U_n &= U_n - U_{n-1} = U_1 \cdot K_2 \cdot K_3 \cdot K_4 \cdot \dots \cdot K_{n-2} \cdot K_{n-1} \cdot (1 - K_n) = \\ &= U_0 \cdot K_1^{N_1} \cdot K_2 \cdot K_3 \cdot K_4 \cdot \dots \cdot K_{n-2} \cdot K_{n-1} \cdot (1 - K_n) = \\ &= U_0 \cdot \zeta^{N_1 \cdot \frac{N_H}{2}} \cdot \zeta^{\frac{N_H}{2^2}} \cdot \zeta^{\frac{N_H}{2^3}} \cdot \zeta^{\frac{N_H}{2^4}} \cdot \dots \cdot \zeta^{\frac{N_H}{2^{n-2}}} \cdot \zeta^{\frac{N_H}{2^{n-1}}} \cdot \left(1 - \zeta^{\frac{N_H}{2^n}}\right) \end{aligned} \quad (23)$$

where n —number of bits, and N_1 —code generated at the end of the first subrange of processing.

Taking into account Equations (14) and (15), the maximum processing error is given by

$$\delta = \frac{\Delta U_n}{U_{n-1}} \cdot 100\% = (1 - K_n) \cdot 100\% = \left(1 - \zeta^{\frac{N_H}{2^n}}\right) \cdot 100\% \quad (24)$$

Starting from the second subrange of processing, an error check is introduced for the current and previous cycles. This allows the halting of processing once a specified error is reached, without utilising the entire range of coefficients.

The actual output code is generated according to Expressions (10), (17), and (18), based on the number of repetitions of the weight of the first bit and the weight of each subsequent bit:

$$N_{out} = A_1 \cdot \frac{N_H}{2} + A_2 \cdot \frac{N_H}{2^2} + A_3 \cdot \frac{N_H}{2^3} + \dots + A_i \cdot \frac{N_H}{2^i} + \dots + A_n \cdot \frac{N_H}{2^n} \quad (25)$$

Alternatively, the actual output code can be determined using the Processing Function (19) by substituting, in place of the input voltage, the compensating voltage obtained during modelling:

$$N_{out} = \frac{1}{\log \zeta} \cdot \log\left(\frac{U_k}{U_0}\right) \quad (26)$$

The ideal value of the output code is determined by Expression (19).

The actual nominal value of the output code is calculated according to Expression (4). For example, for the improved recurrent LADC in the 10-bit version, (1) the specified

processing error of 0.1% corresponds to a logarithmic base $\zeta = 0.999$ and the actual nominal value of the output code $N_H = 9214$; (2) the given processing error of $\delta = 0.00625\%$ corresponds to a logarithmic base of $\zeta = 0.99994$ and the actual nominal value of the output code $N_H = 147,370$.

The calculations and modelling carried out show that the output code values obtained using Expressions (25) and (26) are either the same or differ slightly.

For example,

- For an input voltage $U_{IN} = 8.5$ V, the ideal output code value calculated according to Expression (19) is $N_{out_{id}} = 2600.3841$, while (1) the output code calculated according to Expression (25) is $N_{out} = 2600$, and accordingly, the relative error of the output code $\delta_{N_{out}} = 0.00026066 = 0.0003\%$; (2) the output code calculated according to Expression (26) is $N_{out} = 2600$, and correspondingly, the relative error of the output code $\delta_{N_{out}} = 0.00026066 = 0.0003\%$.
- For an input voltage $U_{IN} = 0.5$ V, the ideal output code value calculated according to Expression (19) is $N_{out_{id}} = 47933.2142$, while (1) the output code calculated according to Expression (25) is $N_{out} = 47934$, and accordingly, the relative error of the output code $\delta_{N_{out}} = 0.0005332 = 0.0005\%$; (2) the output code calculated according to Expression (26) is $N_{out} = 47932$, and correspondingly, the relative error of the output code $\delta_{N_{out}} = 0.00082393 = 0.0008\%$.
- For input voltage $U_{IN} = 0.1$ V, the ideal value of the output code calculated with Formula (19) is $N_{out_{id}} = 73685.0255$, while (1) the output code calculated with Formula (25) is $N_{out} = 73686$, and accordingly, the relative error of the output code $\delta_{N_{out}} = 0.00066124 = 0.0007\%$; (2) the output code calculated according to Expression (26) is $N_{out} = 73684$, and accordingly, the relative error of the output code $\delta_{N_{out}} = 0.00069589 = 0.0007\%$.

Thus, the differences between the relative errors of the output code when calculated using (25) and (26) do not exceed the desired error, and the actual output code can be determined using any of the provided formulas.

During the research on logarithmic analogue-to-digital conversion using the improved recurrent method, the operation graphs of the developed LADC were constructed.

Modelling was carried out for various arbitrary input voltage values ranging from 0 to 10 V, with fifteen of them evenly distributed across the entire range of input voltage changes and three corresponding to values of 1 mV, 10 mV, and 100 mV to assess the possibility of extending the dynamic range of operation.

It should be noted that the reference voltage was set to 10 V.

The change in voltage in the modelling result graphic is not shown by lines but by individual stars. This method most fully reproduces the discrete nature of the change in the compensating voltage.

3.2. Results of Modelling Analogue-to-Digital Conversion Using the Improved Recurrent Method

The main task was to increase the speed of recurrent LADCs. Thus, during the modelling, the primary focus was the processing time.

Modelling was conducted with a selected nominal code of 256, corresponding to 8 bits, for the given relative error values of the output code: $\delta = 0.25\%$, $\delta = 0.1\%$, $\delta = 0.05\%$, and $\delta = 0.01\%$. Known 8-bit analogues typically provide an error of 0.4%. The error value $\delta = 0.1\%$ corresponds to a 10-bit converter and the error value $\delta = 0.01\%$ is better than that of a 13-bit converter.

Modelling was also carried out with a selected nominal code of 1024, corresponding to 10 bits, for the specified relative error values of the output code: $\delta = 0.025\%$, $\delta = 0.01\%$, $\delta = 0.00625\%$, and $\delta = 0.0015625\%$. Note that the usual known analogues at 10-bit provide

an error of 0.1%, and the $\delta = 0.00625\%$ and $\delta = 0.0015625\%$ values correspond to 14-bit and 16-bit.

Thanks to an improved recurrent analogue-to-digital conversion method, through the selection of a specific logarithm base ζ , greater accuracy can be achieved compared to known analogues. We set, for the 8-bit LADC, the logarithm base $\zeta = 0.9975$ for the desired error $\delta = 0.25\%$, the logarithm base $\zeta = 0.9990$ for the desired error $\delta = 0.1\%$, the logarithm base $\zeta = 0.9995$ for the desired error $\delta = 0.05\%$, and the logarithm base $\zeta = 0.99990$ for the desired error $\delta = 0.01\%$. And for the 10-bit LADC, we set the logarithm base $\zeta = 0.99975$ for the desired error $\delta = 0.025\%$, the logarithm base $\zeta = 0.99990$ for the desired error $\delta = 0.01\%$, the logarithm base $\zeta = 0.99994$ for the desired error $\delta = 0.00625\%$, and the logarithm base $\zeta = 0.99998$ for the desired error $\delta = 0.0015625\%$.

The results of the modelling of the improved recurrent 8-bit LADC are presented in the form of graphs and in Table 1.

Table 1. Results of modelling of LADCs implementing the proposed improved recurrent method of logarithmisation.

Input Voltage Range	Output Code	Desired Relative Error of the Output Code	The Relative Error of the Output Code Obtained	Conversion Time	Number of the Conversion Tacts
V	bit	%	%	μs	-
10–0.001	8	0.25	0.005–0.026	22.5–85	9–34
		0.1	0.002–0.01	20–195	8–78
		0.05	0.001–0.0097	22.5–377.5	9–151
		0.01	0.0002–0.0036	27.5–1820	11–727
10–0.001	10	0.025	0.0005–0.003	30–202.5	12–81
		0.01	0.0002–0.002	25–472.5	10–189
		0.00625	0.0001–0.0007	30–747.5	12–299
10–1	10	0.0015625	0.00001–0.00015	47.5–747.5	19–299
1–0.001	10	0.0015625	0.00001–0.00015	750–2910	300–1164

Figure 4 shows the graphs of the relative error of the output code, while Figure 5 illustrates the conversion time.

The relative errors of the output code not only do not exceed the specified error value but are all smaller. For most input voltage values, the errors are up to ten times smaller. Specifically, for a specified error of 0.25%, the relative output code errors range from 0.005% to 0.026%; for an error of 0.1%, they range from 0.002% to 0.01%; for an error of 0.05%, they range from 0.001% to 0.0097%; and for an error of 0.005%, they range from 0.0002% to 0.0036%.

Conversion time doubles for each halving of the desired relative error. For example, if at $\delta = 0.25\%$ the conversion took place up to 34 doses (less than 85 μs), then at $\delta = 0.1\%$, it takes place up to 78 doses (198 μs) and for $\delta = 0.05\%$ up to 151 doses (less than 378 μs). But when the maximum possible accuracy is achieved, there is a significant, 4-fold increase in the conversion time. In the 8-bit LADC, for a given error $\delta = 0.01\%$, the conversion duration increased to 728 doses or 1.82 ms. The clock frequency was 400 kHz (or a period of 2.5 μs).

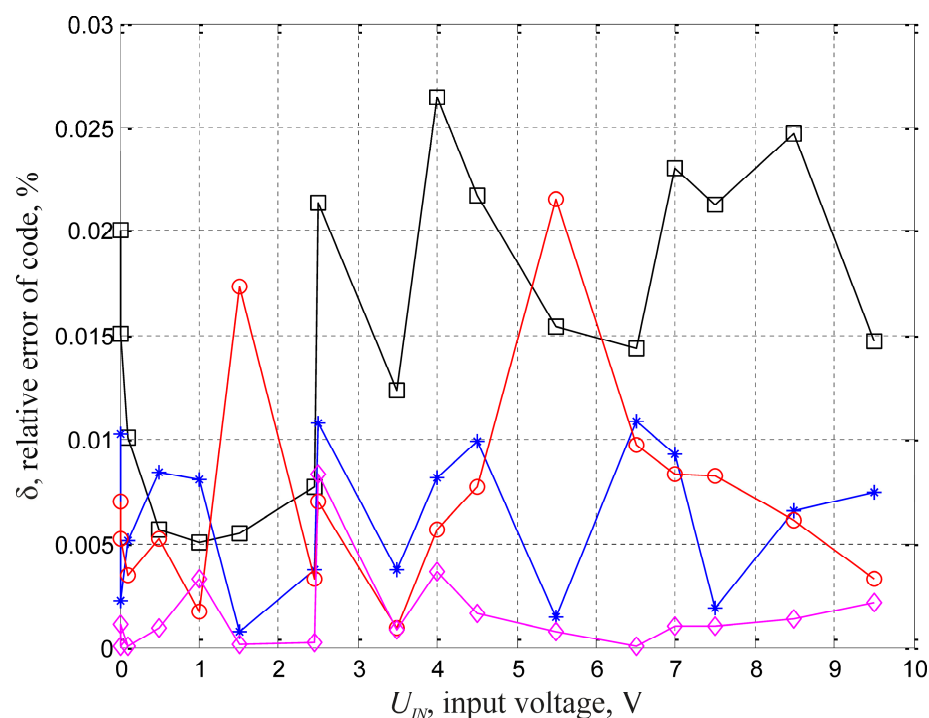


Figure 4. Graphs of the relative error of the output code of the improved recurrent LADC with a bit rate of 8 and a logarithmic base $\zeta = 0.9975$ (black square), $\zeta = 0.9990$ (blue *), $\zeta = 0.9995$ (red o), or $\zeta = 0.9999$ (pink rhombus).

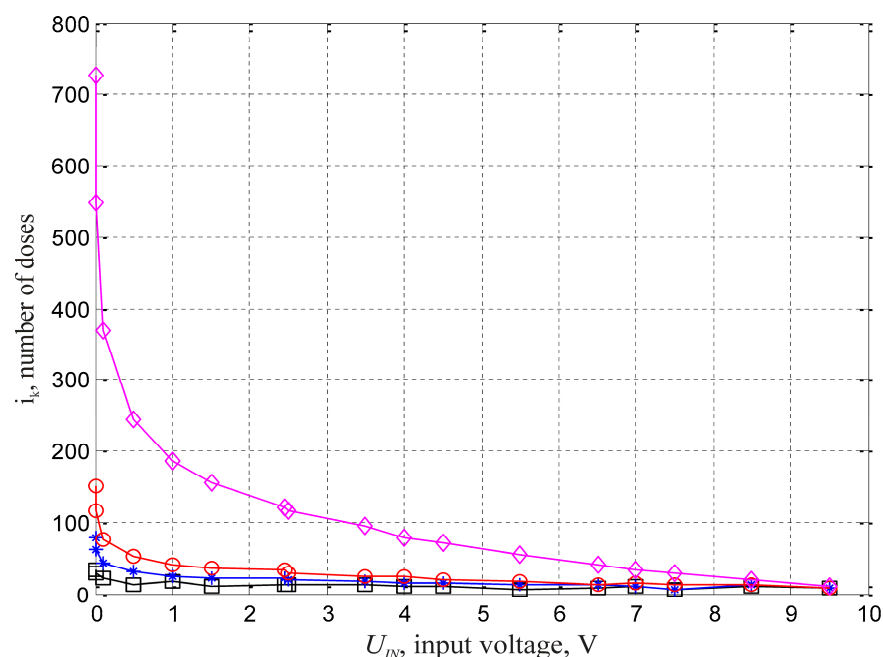


Figure 5. Graphs of the duration of the conversion of the improved recurrent LADC with a bit rate of 8 and a logarithmic base $\zeta = 0.9975$ (black square), $\zeta = 0.9990$ (blue *), $\zeta = 0.9995$ (red o), or $\zeta = 0.9999$ (pink rhombus).

It should be noted that when the input voltage range is limited, specifically from 1 V to 10 V, the processing time is significantly reduced. Even for the highest accuracy (error $\delta = 0.01\%$), it is less than 200 doses, or 500 μs . For lower accuracy, the time is even shorter: for $\delta = 0.25\%$, it does not exceed 17 doses (42.5 μs); for $\delta = 0.1\%$, it is 25 doses (62.5 μs); and for $\delta = 0.05\%$, it is 42 doses (105 μs).

This behaviour was confirmed in the simulation of the improved, recurrent 10-bit LADC.

Figure 6 shows the processing graph for $U_{IN} = 2.444$ V, where the error $\delta = 0.025\%$ corresponds to the graph created by the “black squares”; the error $\delta = 0.01\%$ corresponds to the graph created by the “blue *”; the error $\delta = 0.00625\%$ corresponds to the graph created by the “red o”; and the error $\delta = 0.0015625\%$ corresponds to the graph created by the “pink rhombuses”.

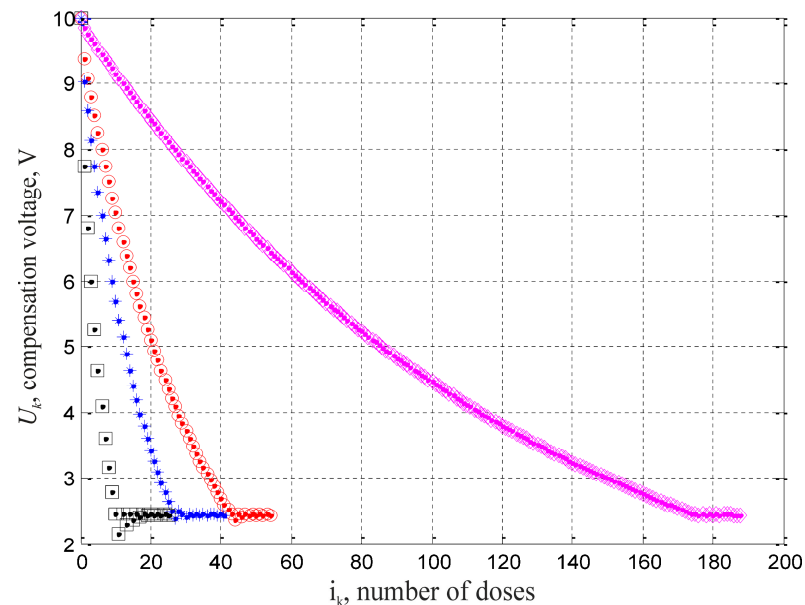


Figure 6. Graphs of the conversion of $U_{IN} = 2.444$ in the improved recurrent LADC with a bit rate of 10 and a logarithmic base $\zeta = 0.99975$ (black square), $\zeta = 0.99990$ (blue *), $\zeta = 0.99994$ (red o), or $\zeta = 0.99998$ (pink rhombus).

The error value $\delta = 0.0015625\%$ corresponds to the actual error level of 16-bit LADCs and represents the upper performance limit for the developed improved 10-bit LADC. With this specified error, the processing time significantly increased, nearly fourfold. Therefore, the specified error value $\delta = 0.00625\%$ can be recommended as optimal for the 10-bit LADC.

Graphs of the performance of the proposed recurrent 10-bit LADC for a specified error of 0.00625% across various input voltage values are shown in Figure 7.

They show that the closer the input voltage value is to the reference voltage, the fewer steps are required in the first subrange with a return to the initial weight. For example, for $U_{IN} = 9.5$ V, it is 1 step (Figure 7a), for $U_{IN} = 5.5$ V, there are 18 steps (Figure 7b), and for $U_{IN} = 1$ V and $U_{IN} = 1$ mV, respectively, there are 71 steps (Figure 7c) and 287 steps (Figure 7d).

In the second subrange of processing, the coefficient K_i changes accordingly with the progression toward the input voltage value. The processing procedure becomes analogous to SAR encoding (bit by bit).

The change in the relative output code error over time for each of the studied input voltage values reflects the essence of the processing method. In the first subrange, the errors are larger, but subsequently, with each successive step, they decrease.

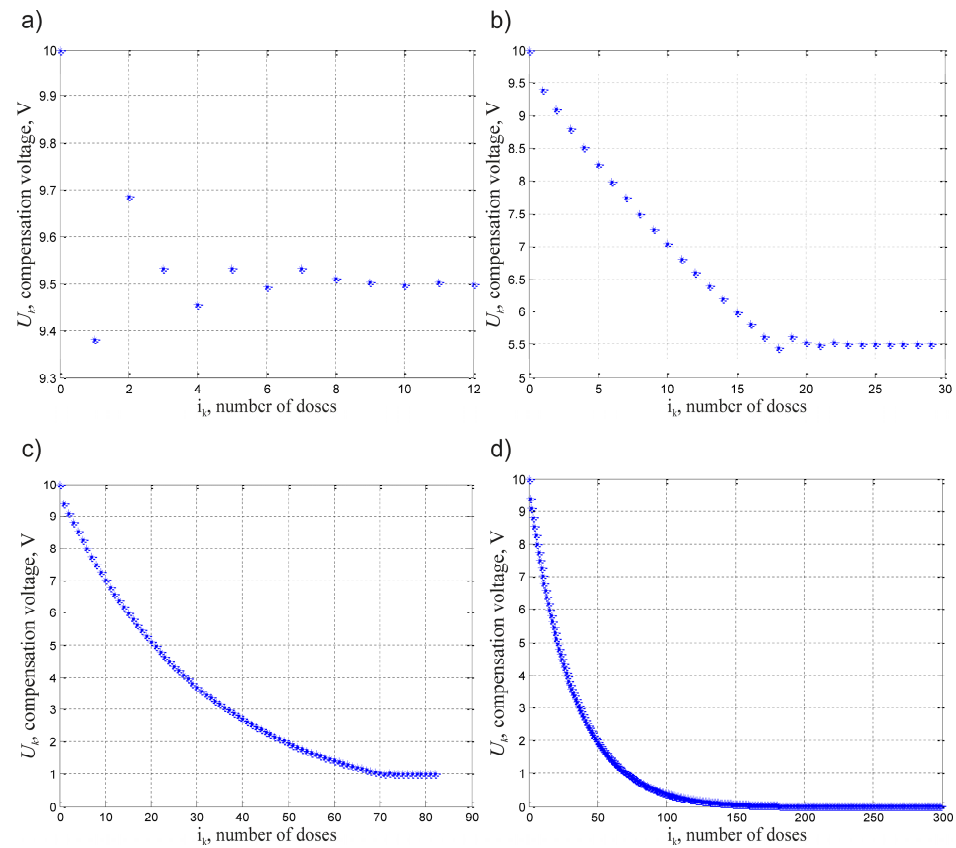


Figure 7. Graphs of the operation of the LADC according to the improved recurrent method for a desired error of 0.00625% at the input voltage values (a)— $U_{IN} = 9.5$ V; (b)— $U_{IN} = 5.5$ V; (c)— $U_{IN} = 1$ V; (d)— $U_{IN} = 0.001$ V.

3.3. The Results of the Simulation of the Improved 10-Bit Recurrent LADC

Figure 8 shows graphs of the relative error of the output code, and Figure 9 shows graphs of the conversion time.

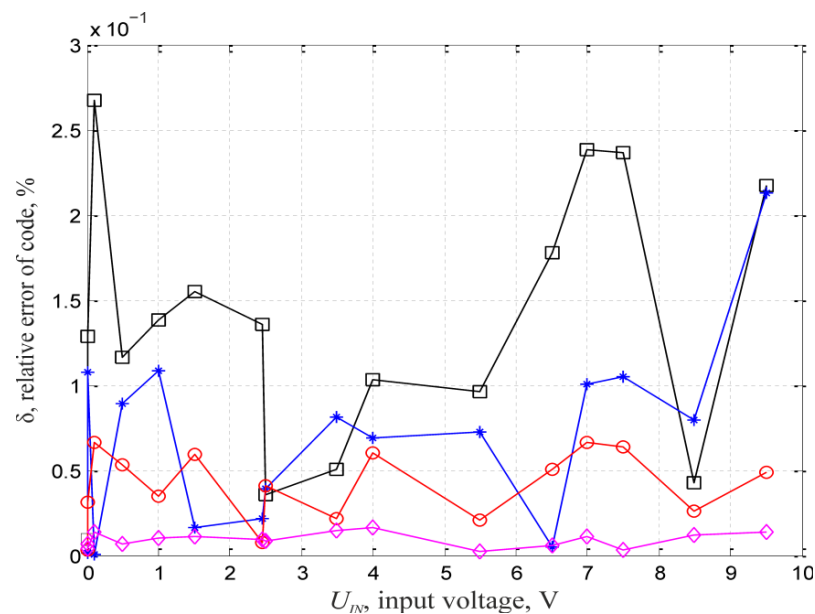


Figure 8. Graphs of the relative error of the output code of the improved recurrent LADC with a bit rate of 10 and a logarithmic base $\zeta = 0.99975$ (black square), $\zeta = 0.99990$ (blue *), $\zeta = 0.99994$ (red o), or $\zeta = 0.99998$ (pink rhombus).

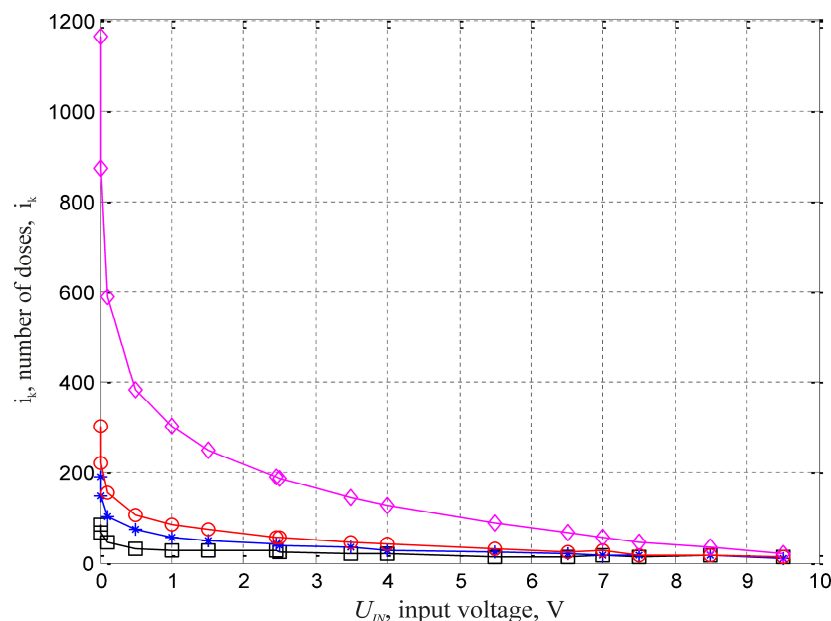


Figure 9. Graphs of the duration of the conversion of the improved recurrent LADC with a bit rate of 10 and a logarithmic base $\zeta = 0.99975$ (black square), $\zeta = 0.99990$ (blue *), $\zeta = 0.99994$ (red o), or $\zeta = 0.99998$ (pink rhombus).

The results of LADC modelling based on the proposed improved recurrent logarithmisation method are shown in Table 1.

During the simulation of the 10-bit LADC, significantly higher accuracy was achieved compared to similar 10-bit analogues. None of the relative output code errors exceeded the specified value, and they were nearly ten times smaller.

For example, within the input voltage range of 1 mV to 10 V, with a given error of $\delta = 0.00625\%$, the relative output code errors ranged from 0.0001% to 0.0007%, with a processing time of 12–299 doses (30 μs –747.5 μs). When limiting the input voltage range from 1 V to 10 V, the processing time was reduced to 12–82 doses (30 μs –205 μs). For processing with a specified error of $\delta = 0.01\%$, the processing times ranged from 10 to 189 doses (25 μs –472.5 μs) for the input range of 1 mV to 10 V and from 10 to 53 doses (25 μs –132.5 μs) when limiting the range below 1 V.

It can thus be observed that doubling the accuracy results in a doubling of the processing duration. At the maximum achievable accuracy—corresponding to 16-bit accuracy for the 10-bit recurrent LADC with an error of 0.0015625%—the processing time increased fourfold. It reached 299 doses (747.5 μs) within the 1 V–10 V range and 1164 doses (2.91 ms) for input voltages ranging from 1 mV to 1 V.

4. Discussion

Table 2 provides a comparison between known LADCs and new LADCs in which the proposed recurrent logarithmic analogue-to-digital conversion method was implemented using switched capacitors with charge redistribution.

Table 2. Comparison of known and new recurrent LADCs.

LADC	Input Voltage Range	Output Code/Number of Sub-Scopes	Desired Relative Error of the Output Code	The Relative Error of the Output Code Obtained	Number of the Conversion Tacts/Conversion Time
-	V	bit	%	%	-/ μ s
Known [4] Flash LADC	10–0.001	8/1	-	0.4	1/0.010
Known [17] Flash LADC	-	7/2	-		2/0.003
Known [8] LADC with a Variable Log Base	10–0.001	14/4	-	0.01	26/65
	10–0.001	17/5	-	≤ 0.001	42/105
Known [5] Recurrent LADC	10–0.001	8/1	0.25	≤ 0.03	138/345
		10/1	0.025	≤ 0.0032	406/1015
Proposed Recurrent LADC	10–0.001	8/1	0.25	0.005–0.026	9–34/22.5–85
			0.1	0.002–0.01	8–78/20–195
			0.05	0.001–0.0097	9–151/22.5–377.5
			0.01	0.0002–0.0036	11–727/27.5–1820
	10–0.001	10/1	0.025	0.0005–0.003	12–81/30–202.5
			0.01	0.0002–0.002	10–189/25–472.5
			0.00625	0.0001–0.0007	12–299/30–747.5
	10–1	10/1	0.0015625	0.00001–0.00015	19–299/47.5–747.5
	1–0.001	10/1	0.0015625	0.00001–0.00015	300–1164/750–2910

In summary, we can conclude that recurrent LADCs, compared to flash LADCs and LADCs with successive approximation, (1) have a significantly simpler structure; (2) have higher accuracy with a smaller number of source code bits, but are inferior in speed. Compared to the classic recurrent LADC, multiple access to the most significant bit significantly increases the conversion speed by 4–5 times while maintaining the specified accuracy. In particular, for the new recurrent LADC in the entire input voltage range from 1 mV to 10 V, the relative error of the output code does not exceed (1) 0.026% with a conversion time of no more than 85 μ s for an 8-bit converter; (2) 0.003% (better than 15-bit accuracy) with a conversion time of less than 203 μ s for a 10-bit converter.

The research confirms that the proposed recurrent LADC on switched capacitors with charge redistribution, compared to known analogues, increases the speed of processing while maintaining the specified accuracy and gives the user the ability to select these parameters in advance.

5. Conclusions

Based on the conducted modelling of the improved recurrent method of logarithmic analogue-to-digital conversion on switched capacitors with charge redistribution, we claim the following:

1. Recurrent LADCs based on the improved method have a processing speed four or more times higher compared to analogues, with their speed increasing as the nominal bit count of the LADC grows.

The following are the technical parameters of such converters:

- Recurrent LADC with 8 bits operating with accuracy (1) equivalent to a 10-bit converter (desired error 0.1%)—output code error less than 0.01% with a pro-

- cessing time under 195 μ s; (2) equivalent to an 11-bit converter (desired error 0.05%)—output code error less than 0.0097% with a processing time under 378 μ s; equivalent to a 13-bit converter (desired error 0.01%)—output code error less than 0.0036% with a processing time under 1820 μ s.
- Recurrent LADC with 10 bits operating with accuracy (1) equivalent to a 12-bit converter (desired error 0.025%)—output code error less than 0.003% with a processing time under 203 μ s; (2) equivalent to an 14-bit converter (desired error 0.00625%)—output code error less than 0.0007% with a processing time under 748 μ s; equivalent to a 16-bit converter (desired error 0.0015625%)—output code error less than 0.00015% with a processing time under 2910 μ s.
2. In recurrent LADCs based on the improved method, the desired accuracy is determined by the logarithm and can be specified by the user during the design stage of the converters. These converters achieve an increase in accuracy of 2 to 6 bits relative to the nominal value, with the optimal accuracy-to-speed ratio corresponding to an accuracy improvement of 4 bits.
 3. Recurrent LADCs based on the improved method outperform known converters in terms of increased speed while maintaining the specified accuracy and are more flexible in providing the required characteristics in the LADC design process.

Author Contributions: Conceptualisation, Z.M., M.M., L.M., A.S. and Z.S.; methodology, Z.M., M.M., L.M., A.S. and Z.S.; writing—original draft preparation, Z.M., M.M., L.M., A.S. and Z.S.; writing—review and editing, Z.M., M.M., L.M., A.S. and Z.S.; visualisation, Z.M., M.M., L.M., A.S. and Z.S.; supervision, Z.M., M.M., L.M., A.S. and Z.S.; funding acquisition, A.S. and Z.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lee, J.; Kang, J.; Park, S.; Seo, J.; Anders, J.; Guilherme, J.; Flynn, M.P. A 2.5 MW 80 DB DR 36 DB SNDR 22 MS/s Logarithmic Pipeline ADC. *IEEE J. Solid State Circuits* **2009**, *44*, 2755–2765. [\[CrossRef\]](#)
2. Guilherme, J.; Vital, J.; Franca, J. A CMOS Logarithmic Pipeline A/D Converter with a Dynamic Range of 80 dB; IEEE: New York, NY, USA, 2002; Volume 1, pp. 193–196.
3. Sirimasakul, S.; Thanachayanont, A.; Jeamsaksiri, W. Low-Power Current-Mode Logarithmic Pipeline Analog-to-Digital Converter for ISFET Based PH Sensor. In Proceedings of the 2009 9th International Symposium on Communications and Information Technology, Incheon, Korea, 28–30 September 2009; pp. 1340–1343.
4. Mychuda, Z.; Kuzemko, O.; Kostruba, O. Paralelny Analoho-Tsyfrovyy Peretvoriuvach. UA Patent 80,691, 25 October 2007.
5. Mychuda, Z. Logarytmiczny Przetwornik Analogowo-Cyfrowy. PL Patent 232,384, 6 June 2019.
6. Mychuda, Z.; Yelisieieva, H.; Szcześniak, Z. Features of implementation of recurrent logarithmic ADCs / Міжвідомчий науково-технічний збірник. Вимірювальна техніка та метрологія *Meas. Equip. Metrol.* **2022**, *83*, 5–10. [\[CrossRef\]](#)
7. Mychuda, Z.; Zhuravel, I.; Mychuda, L.; Szcześniak, A.; Szcześniak, Z.; Yelisieieva, H. Mathematical modelling of the influence of parasitic capacitances of the components of the logarithmic analogue-to-digital converter (LADC) with a successive approximation on switched capacitors for increasing accuracy of conversion. *Electronics* **2022**, *11*, 1485. [\[CrossRef\]](#)
8. Mychuda, Z.; Mykyichuk, M.; Zhuravel, I.; Mychuda, L.; Szcześniak, A.; Szcześniak, Z. New Method for Logarithmic Analogue-to-Digital Conversion Using Switched Capacitors with a Variable Logarithmic Base. *Electronics* **2024**, *13*, 29. [\[CrossRef\]](#)
9. Petilli, E.M. Logarithmic Analog to Digital Converter Devices and Methods Thereof. U.S. Patent 20170179972, 22 June 2017.
10. Pagnin, M.; Ortmanns, M. Evaluation of logarithmic vs. linear ADCs for neural signal acquisition and reconstruction. In Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Jeju, Republic of Korea, 11–15 July 2017; pp. 4387–4390.

11. Santos, M.; Guilherme, J.; Horta, N. *Logarithmic Voltage-to-Time Converter for Analog-to-Digital Signal Conversion*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019. Available online: <https://www.springerprofessional.de/en/logarithmic-adc/16620222> (accessed on 15 September 2023).
12. Inagaki, Y.; Sugimori, Y.; Ioka, E.; Matsuya, Y. Logarithmic Compression ADC Using Transient Response of a Comparator/IEICE TRANSACTIONS on Electronics. *IEICE Trans. Electron.* **2017**, *E100-C*, 359–362. [[CrossRef](#)]
13. Zakharchenko, S.M. Structures of Integral Capacitors as a Basis for Construction of Capacitor Matrices for ADC with Charge Redistribution with Weight Redundancy. *Sci. Work. VNTU* **2009**. Available online: <http://praci.vntu.edu.ua/index.php/praci/article/view/163> (accessed on 15 September 2023).
14. Ballo, A.; Grasso, A.D.; Palumbo, G.; Tanzawa, T. Charge Pumps for Ultra-Low-Power Applications: Analysis, Design, and New Solutions. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 2895–2901. [[CrossRef](#)]
15. Yi, P.; Liang, Y.; Liu, S.; Xu, N.; Fang, L.; Hao, Y. A 625kHz-BW, 79.3dB-SNDR Second-Order Noise-Shaping SAR ADC Using High-Efficiency Error-Feedback Structure. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *69*, 859–863. [[CrossRef](#)]
16. Liang, Y.; Li, C.; Liu, S.; Zhu, Z. A 14-b 20-MS/s 78.8 dB-SNDR Energy-Efficient SAR ADC With Background Mismatch Calibration and Noise-Reduction Techniques for Portable Medical Ultrasound Systems, in IEEE TBioCAS. *IEEE Trans. Biomed. Circuits Syst.* **2022**, *16*, 200–210. [[CrossRef](#)] [[PubMed](#)]
17. Oh, D.R.; Seo, M.J.; Ryu, S.T. A 7-Bit Two-Step Flash ADC With Sample-and-Hold Sharing Technique. *IEEE J. Solid State Circuits* **2022**, *57*, 2791–2801. [[CrossRef](#)]
18. Kumar, C.; Pavan, S. Power-Noise Trade-Offs in Continuous-Time Pipelined ADCs and Active Filters. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2023**, *70*, 3832–3842. [[CrossRef](#)]
19. Petryshyn, L.; Petryshyn, M. Method and Algorithm of Successive Approximation Analog-to-Digital Conversion of Information in Management Systems. In Proceedings of the 2024 14th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, 19–21 September 2024; pp. 622–625. [[CrossRef](#)]
20. Zhang, X.; Cheng, Y.; So, H.C.; Li, J. A Novel Mixed-ADC Architecture for DOA Estimation. *IEEE Signal Process. Lett.* **2024**, *31*, 611–615. [[CrossRef](#)]
21. Catania, A.; Gagliardi, F.; Piotto, M.; Bruschi, P.; Dei, M. Ultralow-Power Inverter-Based Delta-Sigma Modulator for Wearable Applications. *IEEE Access* **2024**, *12*, 80009–80019. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.