http://doi.org/10.35784/iapgos.2225

# BIT ERROR NOTIFICATION AND ESTIMATION IN REDUNDANT SUCCESSIVE-APPROXIMATION ADC

**Serhii Zakharchenko, Roman Humeniuk**
Vinnytsia National Technical University, Department of Computer Facilities, Vinnytsia, Ukraine

***Abstract.*** *The article is devoted to research on the possibilities to use redundant number systems for bit error notification in a successive-approximation ADC during the main conversion mode. The transfer function of a successive-approximation ADC with a non-binary radix is analyzed. If the radix is less than 2, not all possible code combinations appear on the converter output. The process of formation of unused combinations is investigated. The relationship between the bit's deviations and the list of unused combinations is established. The possibilities of estimating the bit error value without interrupting the process of analog-to-digital conversion is considered.*

**Keywords**: successive-approximation ADC, redundant number systems, ADC transfer function

# POWIADOMIENIE O BŁĘDZIE BITOWYM I OCENA W STOPNIOWEJ REDUNDANTNEJ APROKSYMACJI ACP

***Streszczenie.*** *Artykuł jest poświęcony badaniu możliwości wykorzystania redundantnych systemów liczbowych do powiadamiania o błędach bitowych w stopniowej aproksymacji ACP podczas konwersji głównej. Analizowana jest funkcja transferu stopniowej aproksymacji ACP z niebinarną podstawą. Jeśli podstawa jest mniejsza niż 2, nie wszystkie możliwe kombinacje kodów pojawią się na wyjściu konwertera. Badany jest proces tworzenia nieużywanych kombinacji, i ustalane są relacje między odchyleniami bitu a listą nieużywanych kombinacji. Autorzy również przeanalizowali możliwości oceny wartości błędu bitowego bez przerywania procesu konwersji analogowo-cyfrowej.*

**Słowa kluczowe**: stopniowa aproksymacja ACP, redundantne systemy liczbowe, funkcja transferu ACP

## Introduction

Successive-approximation ADCs are very popular now due to their high resolution at the level of 14–18 binary digits, and relatively high sampling rate in the range from 50 kHz to 50 MHz. However, if the number of bits exceeds 12–14, we have the influence of external factors, such as temperature change, which leads to bit errors. The maximum absolute errors will be in the most significant bits (MSB) [1]. The result of this is the increasing of differential and integral nonlinearities. There are two main ways to resolve this problem: technological and algorithmic. The technological methods are time- and cost-consuming, and provide the ability to improve the linearity by several bits. The universal method to overcome this problem is to use a calibration procedure for the MSBs [2, 3, 6]. The traditional calibration procedure is performed after the device is turned on and is periodically repeated during operation. The ADC can operate in either the main conversion mode or calibration. New calibration technologies allow the general and calibration modes to be combined [4, 5]. Using a non-binary radix provides the opportunity for notification of MSB deviations [7].

## 1. Transfer function analysis for successive-approximation ADC with non-binary radix

The ADC transfer function (TF) determines the relationship between the input analog signals with the output code combination. When we are using a binary numeral system, each value of the input analog signal corresponds to one relevant code combination. At the same time, when using a redundant positional numeral system (radix less than 2), there are zones of TF, where one value of the input signal corresponds to several output code combinations, as shown in Fig. 1a. However due to the successive-approximation algorithm in the output code, we will have only one of the possible output combinations (Fig. 1b), which we will call "used" (UC). Accordingly, those combinations that do not occur in the output code will be called "unused" (UnC). For example, in Fig. 1, the TF for radix 1.618 does not include combinations 0011, 0110, 0111 and 1011. These combinations will be UnC. The quantity and location of the UnC are determined by the radix, ADC resolution and bit errors.

The combination location on the TF is defined by the equation:

$$A(K^s) = \sum_{i=0}^{n-1} a_i \cdot Q_i \,, \tag{1}$$

where $K$ – code combination, $s$ – number of code combinations (decimal notation of binary combinations), $n$ – ADC resolution, $Q_i = \alpha^i (1+\delta_i)$ – bit value with number $i$, where $\alpha$ – radix, $\delta_i$ – $i$-bit deviation. $a_i \in \{0,1\}$ – bit values of $K$.

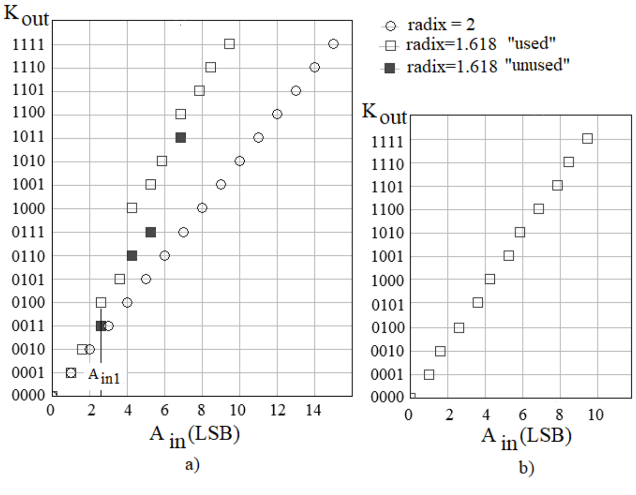

*Fig. 1. ADC transfer function 4-bit ADC: a) for radix 2 and 1.618, b) for radix 1.618 without UnC*

The combination will be "unused" if there is a "used" combination of output code with a larger code combination number, s, and a smaller value of the input analog signal:

$$A(K_{UC}^{s1}) \le A(K_{UnC}^{s2}), \tag{2}$$

where the value of the input analog signal corresponding to the UnC with the number s2 and the UC with the number s1, respectively, and s1 > s2. For example, UnC number 6 (0110) and UC number 8 (1000) in Fig. 1 form a pair of code combinations for which condition (2) is satisfied. Similar pairs form combinations with numbers 3 (0011) and 4 (0100), 7 (0111) and 8 (1000), 11 (1011) and 12 (1100).

UnCs form the groups with one and more successive code combinations. For example, for the TF on Fig. 1, there are three zones of unused combinations. The central zone, which we will call the (n-1)-level zone, has two successive combinations: 0110 and 0111. The (n-2)-level zone has two subzones, which contain the 0011 and 1011 combinations. The difference between combinations is in the first, most significant bit (MSB). This bit identifies the subzone number: 0 – for 0011 and 1 – for 1011.

The value of $K_{UC}^{s1}$ for any zone or subzone is explicitly defined, it follows the largest UnC and we will call it the border following combination (BFC). It is very important that the values of BFC do not depend on the radix, no ADC resolution and they can be defined by the next rule:

- for the (n-1)-level zone, $BFC_{n-1}^{0} = 100...$ – the MSB = 1, then all following "0"s;
- for the (n-2)-level zone, $BFC_{n-2}^{0} = 0100...$ , $BFC_{n-2}^{1} = 1100...$ – the first MSB indicates the number of the subzone, the next bit "1", then all following "0"s;
- For the (n-k)-level zone – k-1, the first MSB indicates the number of the subzone, the next bit "1", then all following "0"s.

From equation (2) follows the condition for the existence of zone unused combinations:

$$A(K_{UC}^{s1}) \le A(K_{UnC}^{s1-1}), \qquad (3)$$

In other words, the value of the input analog signal, which converts into the BFC, must be less than or equal to the input analog signal, which converts into the code combination that immediately precedes the BFC. From (3), we derive the existence of the the inequality for zone (n-k)-level:

$$Q_{n-k} \le \sum_{i=0}^{n-k-1} Q_i. \qquad (4)$$

Because the smallest level zone has only one UnC and the other zones have more than one, it is reasonable to find only the smallest level zone. For example, we have an ideal n-bit (without bit errors) redundant ADC with a radix of 1.7, then in equation (4) will become true beginning from (n-k) = 3. In fact, the number of the smallest level zone of the ideal redundant ADC is defined only by the radix. The relationship between the smallest level zone number (SLZN) and the radix is shown in Table 1.

*Table 1. Radix and smallest level zone number relations*

| The radix | 1.618 | 1.84 | 1.93 | 1.96 |
|---|---|---|---|---|
| Smallest level zone number | 2 | 3 | 4 | 5 |

Therefore, the quantity of zones of unused combinations may be calculated as n – SLZN.

Because the radix and ADC resolution are constant, transition from unused combination to used ones and vice-versa is forced only by a change in $\delta_i$ .

## 2. Influence of single bit deviation on the transition of UC to UnC and vice-versa

The deviation of the MSB value is not equal to zero, while the other bits are ideal. From equation (1), we derive:

$$A(K^s) = a_{n-1} \cdot \alpha^{n-1} \cdot (1+\delta_{n-1}) + \sum_{i=0}^{n-2} a_i \cdot \alpha^i . \qquad (5)$$

As a result, the first part of the TF will not change because $a_{n-1} = 0$, while the second part of the TF will be shifted left if $\delta_i < 0$ , or right if $\delta_i > 0$ . Fig. 2 shows the reaction of the TF of the successive-approximation ADC with a radix of 1.8 on the deviation of the MSB value: in Fig. 2a, the value of the deviation of the MSB is equal to zero, in Fig. 2b, it is equal to +5%, and in Fig. 2c, it is equal -5%.

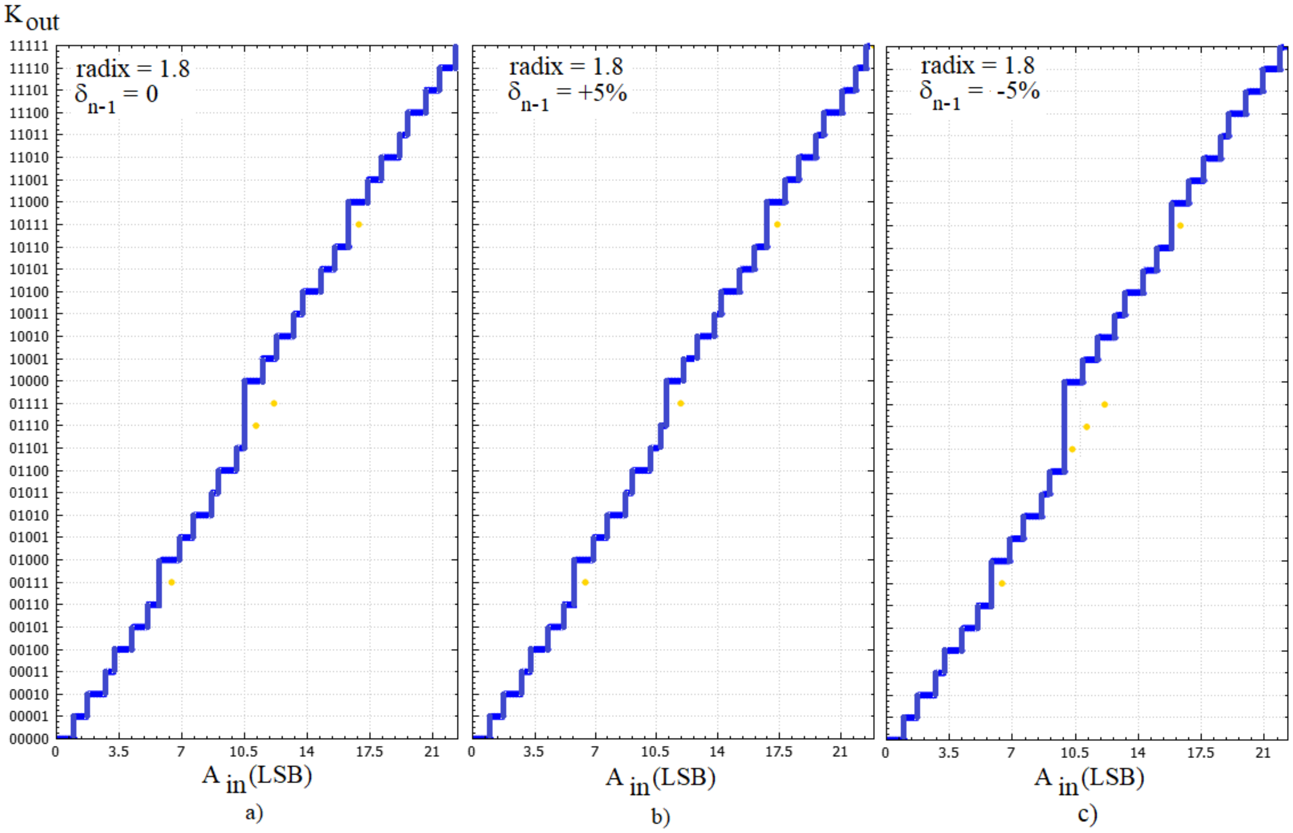

*Fig. 2. ADC transfer function 5-bit ADC for radix 1.8: a) without MSB value deviation, b) with positive MSB value deviation, c) with negative MSB value deviation*

Hence, the consequence of positive MSB values shifting is the 01110-combination transition from "unused" to "used". On the other hand, the consequence of negative MSB values shifting is the 01101-combination transition from "used" to "unused". The border MSB deviation value for the 01110-combination transition may be calculated from:

$$\alpha^{n-1}\cdot(1+\delta_{n-1})=\alpha^{n-2}+\alpha^{n-3}+\alpha^{n-4}, \qquad (6)$$

and after transformation:

$$\delta_{n-1}=\frac{\alpha^{n-2}+\alpha^{n-3}+\alpha^{n-4}-\alpha^{n-1}}{\alpha^{n-1}}. \qquad (7)$$

For a radix of 1.8 and n = 5, the border MSB shifting value according to (7) will be 0.036, or 3.6%. In the same way, the border MSB deviation value for the 01101-combination will be:

$$\delta_{n-1}=\frac{\alpha^{n-2}+\alpha^{n-3}+\alpha^{n-5}-\alpha^{n-1}}{\alpha^{n-1}}. \qquad (8)$$

For a radix of 1.8 and n = 5, the border MSB shifting value will be calculated according to (8), it will be -0.04, or -4%. In other words, for a 5-bit successive-approximation ADC with a radix of 1.8, the presence of 2 "unused" combinations in the central (n-1)-level zone (Fig. 2a) guarantees that the MSB value shifting borders are from -4% to 3.6%. It is important that the MSB value shifting does not influence the other zones of "unused" combinations, for example the (n-2)-level zone in Fig. 2.

The value deviation is not equal to zero for the (n-2) bit, but all other bits including the MSB are ideal. From equation (1), we derive:

$$A(K^s)=a_{n-1}\cdot\alpha^{n-1}+a_{n-1}\cdot\alpha^{n-2}(1+\delta_{n-2})+\sum_{i=0}^{n-3}a_i\cdot\alpha^i. \qquad (9)$$

The border (n-2) deviation value for the 01110-combination transition may be calculated from:

$$\alpha^{n-1}=\alpha^{n-2}(1+\delta_{n-2})+\alpha^{n-3}+\alpha^{n-4}. \qquad (10)$$

and after transformation:

$$\delta_{n-2}=\frac{\alpha^{n-1}-\alpha^{n-2}-\alpha^{n-3}-\alpha^{n-4}}{\alpha^{n-2}}. \qquad (11)$$

For a radix of 1.8 and n=5 (n-2)-bit let us shift values accordingly (11), they will be -0.06, or -6%. In a similar way, the border (n-2)-bit deviation value for the 01101-combination will be:

$$\delta_{n-2}=\frac{\alpha^{n-1}-\alpha^{n-2}-\alpha^{n-3}-\alpha^{n-5}}{\alpha^{n-2}}. \qquad (12)$$

For a radix of 1.8 and n = 5, the border (n-2)-bit shifting value, calculated according to (12), will be 0.07, or 7%.

The (n-2)-bit deviation will influence not only the (n-1)-level zone, but also the (n-2)-level zone. For ideal bit values, it is only one "unused" combination in every subzone of the (n-2)-level zone (Fig. 2): X0111, where X equals 0 for the first subzone and 1 for the second. The result of the (n-2)-bit deviation will be the transmission X0111 combination to the "used" category or the transmission X0110 combination to UnC. To calculate the condition for the first transmission, we will use the next equation:

$$\alpha^{n-2}(1+\delta_{n-2})=\alpha^{n-3}+\alpha^{n-4}+\alpha^{n-5}. \qquad (13)$$

Or after transformation:

$$\delta_{n-2}=\frac{\alpha^{n-3}+\alpha^{n-4}+\alpha^{n-5}-\alpha^{n-2}}{\alpha^{n-2}}. \qquad (14)$$

In fact, equation (14) is the same as (7), which means that the (n-2) bit shifting value for a radix of 1.8 and n = 5 will be 0.036, or 3.6%. In a similar way, the border (n-2)-bit deviation value for the X0110-combination will be:

$$\delta_{n-2}=\frac{\alpha^{n-3}+\alpha^{n-4}-\alpha^{n-2}}{\alpha^{n-2}}. \qquad (15)$$

For a radix of 1.8 and n=5, the border (n-2)-bit shifting value according to (15) will be -0.14, or -14%.

The (n-i)-bit value shifting will influence the UnC quantity in the (n-i)-level zone and other zones with numbers less than (n-i).

## 3. Influence of multiple-bit deviation on the transition of UC to UnC and vice-versa

Let the MSB and (n-2) bit values deviations be not equal to zero, while the other bits are ideal. From equation (1), we derive:

$$A(K^s)=a_{n-1}\cdot\alpha^{n-1}\cdot(1+\delta_{n-1})+ \\ +a_{n-2}\cdot\alpha^{n-2}\cdot(1+\delta_{n-2})+\sum_{i=0}^{n-3}a_i\cdot\alpha^i. \qquad (16)$$

The border MSB and (n-2) bit deviation values for the 01110 and 01101 combinations can be calculated from (17) and (18) accordingly:

$$\alpha^{n-1}\cdot(1+\delta_{n-1})=\alpha^{n-2}\cdot(1+\delta_{n-2})+\alpha^{n-3}+\alpha^{n-4}. \qquad (17)$$

$$\alpha^{n-1}\cdot(1+\delta_{n-1})=\alpha^{n-2}\cdot(1+\delta_{n-2})+\alpha^{n-3}+\alpha^{n-5}. \qquad (18)$$

The graphical interpretation of equations (14), (15), (17) and (18) for a radix of 1.8 and n=5 are shown in Fig. 3.

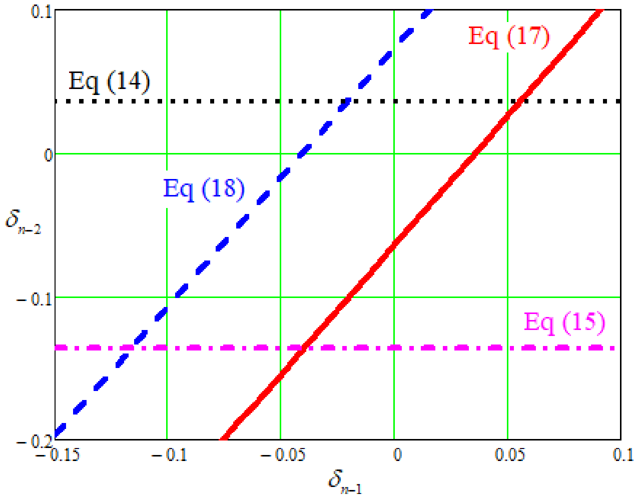

*Fig. 3. Graphical interpretation of equations (14), (15), (17) and (18)*

Fig. 3 demonstrates the opportunities to control two MSB deviations. If the bit value deviations $\delta_{n-1}$ and $\delta_{n-2}$ are inside the parallelogram created by equations (14), (15), (17) and (18), the quantity of "unused" combinations in the (n-1) and (n-2)-level zones will not change, and vice-versa – if the quantity of "unused" combinations has changed, it means that the bit value deviations exceeded certain thresholds. To control the "unused" combinations, it is not necessary to interrupt the main conversion if the input analog signal captures the main zones of "unused" combinations.

## 4. Bit deviation estimation based on UnC analysis

Control of the quantity of "Unused" combinations not only identifies the fact of bit deviation, but estimates this deviation. The relationship between the quantity of UnCs in a certain zone of "unused" combinations and bit deviations is shown above. The reverse task is to estimate the bit deviation if the quantity of UnCs in a certain zone is known. For a start description of the simplest situation, when only one bit has deviated and its number is known.

For example, the ADC transfer function looks like Fig. 2c. It is known that the bit deviation is only the MSB and the quantity of UnCs is equal to three (Fig. 2c).

To calculate the upper bound of the MSB deviation, it is necessary to equate the analog signal for combination 10000 (BFC for (n-1)-level zone of UnCs) with the analog signal for the 01101 combination – the smallest UnC:

$$\alpha^{n-1}\cdot(1+\delta_{n-1}^{\max 3})=\alpha^{n-2}+\alpha^{n-3}+\alpha^{n-5}. \tag{19}$$

The lower bound of $\delta_{n-1}$ can be calculated from:

$$\alpha^{n-1}\cdot(1+\delta_{n-1}^{\min 3})=\alpha^{n-2}+\alpha^{n-3}, \tag{20}$$

which corresponds to the 01100 combination – the last UC between the series UnC. The average value of $\delta_{n-1}$ can be calculated as:

$$\delta_{n-1}^{avr3}=\frac{\delta_{n-1}^{\max 3}+\delta_{n-1}^{\min 3}}{2}. \tag{21}$$

For the sample in Fig. 2c, the values of $\delta_{n-1}^{\max 3}$, $\delta_{n-1}^{\min 3}$ and $\delta_{n-1}^{avr3}$ will be: -0.04, -0.14, and -0.09.

It is obvious that $\delta_{n-1}^{\max 3}=\delta_{n-1}^{\min 2}$, and $\delta_{n-1}^{\min 3}=\delta_{n-1}^{\max 4}$.

The second situation is where only two bits have deviations and their numbers are known. Fig. 4 shows the graphical diagram that can be used to estimate the deviations of two MSBs for a 5-bit ADC with a radix of 1.8.

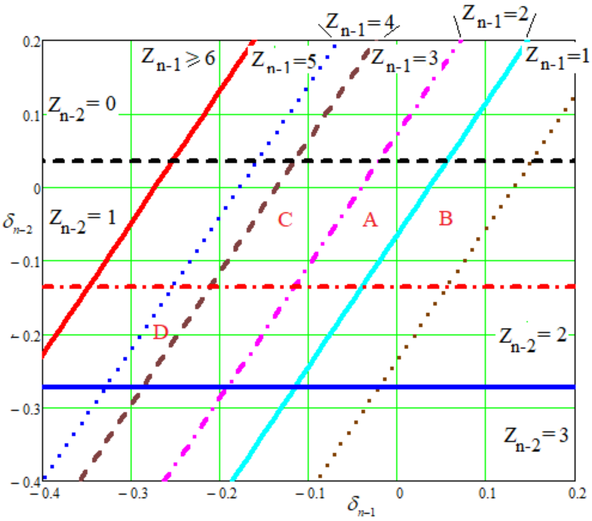

*Fig. 4. Graphical diagram for estimation of two-MSB deviations*

$Z_{n-1}$ and $Z_{n-2}$ are the quantities of "unused" combinations in the (n-1) and (n-2)-level zones, respectively. For example, parallelogram A corresponds to two UnCs in the (n-1)-level zones and one UnC in the (n-2)-level zone as in Fig. 2a. The point is that $\delta_{n-1}$ =0, $\delta_{n-2}$ =0 is inside parallelogram A. Parallelogram B corresponds to one UnC in the (n-1)-level zone and one UnC in the (n-2)-level zone as in Fig. 2b. Parallelogram C corresponds to three UnCs in the (n-1)-level zone and one UnC in the (n-2)-level zone as in Fig. 2c. Parallelogram D corresponds to four UnCs in the (n-1)-level zone and two UnCs in the (n-2)-level zone.

It is important that, to estimate the deviations of two bits, it is necessary to have the information about the "unused" combinations in two correspondent UnC zones.

To calculate the deviation values, it is necessary to define the parallelogram center in coordinates $\delta_{n-1}$, and $\delta_{n-2}$. For example, in Fig. 3, the first step is to calculate $\delta_{n-2}^{avr}$ by means of averaging $\delta_{n-2}$, received from (14) and (15):

$$\delta_{n-2}^{avr}=\frac{\delta_{n-2}^{\max}+\delta_{n-2}^{\min}}{2}. \tag{22}$$

The next step is to substitute $\delta_{n-2}^{avr}$ into (17) and (18), and calculate $\delta_{n-1}^{\max}(\delta_{n-2}^{avr})$ and $\delta_{n-1}^{\min}(\delta_{n-2}^{avr})$. The last step is to average the received values:

$$\delta_{n-1}^{avr}=\frac{\delta_{n-1}^{\max}(\delta_{n-2}^{avr})+\delta_{n-1}^{\min}(\delta_{n-2}^{avr})}{2} \tag{23}$$

Based on (22–23), the values of $\delta_{n-2}^{avr}$ and $\delta_{n-1}^{avr}$ for regions A, B, C and D in Fig. 4 shown in Table 2.

*Table 2. Estimated values of deviations of two MSB for for radix 1.8 and n = 5*

| Region | A | B | C | D |
|---|---|---|---|---|
| $\delta_{n-2}^{avr}$ | -0.05 | -0.05 | -0.05 | -0.20 |
| $\delta_{n-1}^{avr}$ | -0.03 | 0.05 | -0.12 | -0.27 |

## 5. Conclusion

The article shows the opportunity to analytically identifiy the "unused" combinations in the transfer function of a redundant ADC. The simple way to calculate the list of "unused combinations" allows the bit error notification and bit deviation to be estimated for successive-approximation ADCs during the main conversion without using external units and procedures. The relationships between the different zones of "unused" combinations allow the time and computing resources to implement this method to be significantly reduced.

## References

[1] Chakradhar A., Rajesh Kumar Srivastava, Sreenivasa Rao Ijjada: Calibration Techniques of Analog to Digital Converters (ADCs). International Journal of Innovative Technology and Exploring Engineering 8, 2019, 415–419.

[2] Hae-Seung Lee.: A Self-calibrating 15-bit CMOS A/D Converter. IEEE J. Solid-State Circuits 19(6), 1984, 813–817.

[3] Hae-Seung Lee, Hodges D. A.: Self-calibration technique for A/D converters. IEEE Transactions on circuits and systems 30(3), 1983, 188–190.

[4] Khen-Sang Tan, Kiriaki S., de Wit M.: Error correction techniques for high-performance differential A/D Converters. IEEE J. Solid-State Circuits 25(6), 1990, 1318–1327.

[5] McCreary J. L.: Matching properties, and voltage and temperature dependence of MOS capacitors. IEEE J. Solid-State Circuits 16, 1981, 608–616.

[6] McNeill J., Coln M. C. W., Larivee B. J.: "Split ADC" Architecture for Deterministic Digital Background Calibration of a 16-bit 1-MS/s ADC. IEEE J. Solid-State Circuits 40(12), 2005, 2437–2445.

[7] Zakharchenko S., Zakharchenko M., Humeniuk R.: Method of determining the unused combinations in the ADC of successive approximation with weight redundancy. International Conference Methods and Means of Encoding, Protection and Compression of Information (MMEPCI 2017), 114–117.

**Ph.D. Serhii Zakharchenko**
e-mail: zahar@vntu.net

Serhii Zakharchenko is Associate Professor of the Department of Computer Engineering at the Vinnytsia National Technical University. His main scientific interests include analog-to-digital conversion and redundant number systems.

http://orcid.org/0000-0003-3977-2908

**M.Sc. Roman Humeniuk**
e-mail: romchik003@gmail.com

Roman Humeniuk is a Ph.D. Student in the Department of Computer Engineering at the Vinnytsia National Technical University. His main scientific interests include programming, and the software testing process.

http://orcid.org/0000-0001-9986-8894

*otrzymano/received: 4.09.2020* *przyjęto do druku/accepted: 10.12.2020*