Journal of Electrical Engineering, Vol. 75, No. 5, 2024, pp. 418-424

sciendo
PAPERS

Improved implementation of serial pseudorandom/natural code converters used in absolute position encoders

Milan R. Dinčić, Goran S. Miljković, Milica S. Stojanović, Dragan B. Denić

Pseudorandom encoders represent a significant type of absolute position encoders that can be implemented using only a single code track on a code disk and a single reading head, making them especially suitable for high-resolution position measurements. Since pseudorandom code words used for encoding positions are not compatible with other electronics, pseudorandom encoders include a converter that translates the pseudorandom code into the natural binary code. This paper proposes an improved implementation of the serial Fibonacci pseudorandom/natural code converter used in pseudorandom position encoders. The improvement is based on modifying the way bits are written into the converter's shift register, which reduces the propagation delay inside the code converter, thus enabling an increase in the clock frequency. Simulations of the proposed solution, performed in Multisim software, provide a realistic picture of its behaviour and performance. Based on the simulation results, the proposed solution enables a significant increase in the clock frequency of the code converter by 180.54% or 57.56%, depending on the encoder resolution. This significantly boosts the speed of code conversion and enhances the overall performance of the position encoder. The proposed solution is highly valuable for numerous position measurement applications, especially in the increasingly relevant high-resolution applications.

Keywords: position measurement, pseudorandom absolute position encoder, pseudorandom binary sequences, serial pseudorandom/natural code converter

1 Introduction

Absolute position encoders [1,2] are transducers for measuring linear and angular absolute position, offering small size, high resolution, and strong reliability. Their main advantages over incremental position encoders are the absence of error accumulation and the ability to provide accurate position information immediately after power restoration, unlike incremental encoders which require resynchronization time [2]. Absolute encoders are used in a wide range of applications related to position and angular velocity determination [3]. They are found in mechanical arms, machine positioning and printers [4], aerospace, automotive manufacturing and intelligent robots [1], radars, theodolites, CNC (computer numerical control) machines [5], integrated circuits manufacturing, machine health monitoring [4,6], crane positioning [7], industrial motors' shaft positioning [8], copiers, scanners, servo and stepper motor feedback systems, telescopes, process lines, elevators, and wind turbines.

In absolute encoders for measuring angular position, a key component is the code disk, which is mounted on the shaft and rotates with it. The code disk is divided into sectors, each encoded with a unique code word. Classic implementation of absolute position encoders is based on transversal position encoding using natural or Gray code [8], which requires a separate code track on the disk and a distinct reading head for each bit of the code word. This results in an increased number of code tracks on the disk and more reading heads as the resolution increases, leading to higher complexity, cost, and disk size, as the main drawback [2,9].

One of important innovations in the field of absolute position encoders is the introduction of pseudorandom encoders [5,7,10,11] that utilize pseudorandom binary sequences (PRBS) of maximum length (known as m-sequences) generated by linear feedback shift register [5,12,13]. PRBS m-sequences are widely used in other areas such as wireless communications [13], cryptography and steganography [14], fault diagnosis for high voltage distribution networks [15], radar systems [16], and pollutant source estimation [17]. Pseudorandom position encoders utilize longitudinal position encoding with a pseudorandom code, enabling the use of only one code track for position encoding regardless of the resolution. Also, as consecutive code words in the pseudorandom code differ by only one bit, reading of code words can be performed with a single reading head [7,18] which is not feasible with classic absolute encoders. Hence, the main advantage of pseudorandom encoders is that increasing resolution does not lead to increase the number of code tracks and reading heads, preventing in this way increasing of complexity, cost, and disk size and making them particularly suitable for high-resolution absolute encoders [2]. In addition, pseudorandom position encoders have other important advantages compared to classic absolute position encoders, including the ability to apply error detection

Department of Measurements, Faculty of Electronic Engineering Niš, University of Niš, Aleksandra Medvedeva 4, 18000 Niš, Serbia
milan.dincic@elfak.ni.ac.rs, goran.miljkovic@elfak.ni.ac.rs, milica.stojanovic@elfak.ni.ac.rs, dragan.denic@elfak.ni.ac.rs

https://doi.org/10.2478/jee-2024-0050, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X

methods for enhanced reliability [19] and possibility of direct zero position adjustment after mounting the code disk on the shaft [10].

However, pseudorandom code words are not directly usable in digital electronic systems and need to be converted to natural binary code. There are two main types of pseudorandom/natural code converters: serial [2,11,20] and parallel [2]. Parallel conversion uses a code conversion table stored in ROM, while serial conversion employs a shift register. This paper focuses on serial code converters, which are widely used due to their simplicity and the ease of zero position adjustment without significant hardware or software changes [10]. The standard design for serial pseudorandom/natural code converters is known as the Fibonacci code converter [11,20], based on a Fibonacci PRBS generator consisting of a shift register with feedback configuration. The code conversion process starts by loading the bits of the read pseudorandom code word into the shift register. With each clock pulse, the shift register transitions to a new state until it reaches the reference state corresponding to the zero position. The natural binary code word is obtained by counting the clock pulses needed for the shift register to reach this reference state.

The code conversion speed is a critical factor, affecting the operational speed of the entire pseudorandom encoder and the permissible movement speed of the rotating system being measured. The rotating system must not move to the next sector until the code word corresponding to the current sector is fully converted. The conversion time for the read code word depends on the number of clock pulses required for the Fibonacci converter's shift register to match the reference code word. The worst-case scenario must be considered, corresponding to the longest conversion time that occurs for the code word farthest from the reference. As resolution increases, the duration of the longest conversion also increases, making conversion speed particularly critical for higher resolutions.

The conversion time is directly dependent on the clock pulse duration of the Fibonacci converter, which in turn depends on the maximum propagation delay within the Fibonacci converter circuit. To enhance the overall operational speed of the position encoder, it is essential to minimize this maximum propagation delay within the Fibonacci converter, thereby increasing the converter’s clock frequency. This paper addresses this problem by proposing a solution that significantly decreases propagation delays within the converter and accelerates code conversion, particularly facilitating the implementation of pseudorandom encoders with higher resolutions needed in a number of modern applications.

The main contribution of this paper is an improved implementation of the Fibonacci code converter compared to the method known in literature [20]. This enhancement involves modifying the bit-writing process in the shift register's flip-flops, separating the path for writing bits of a new pseudorandom code word from the path for writing bits obtained during the conversion of the current code word. This eliminates certain input logic circuits, significantly reducing propagation delay and increasing clock frequency, thus speeding up the code conversion process.

Considering different resolutions, the maximum propagation delay of the Fibonacci code converter depends on the number of XOR logic gates in the feedback loop of the converter's shift register. It is known [12,13] that the number of XOR logic gates is 1 for most resolutions, while it is 3 for others. Therefore, this paper examines two resolution values for the Fibonacci code converter: a 6-bit converter (as a representative example with one XOR logic gate in the feedback loop) and an 8-bit converter (as a representative example with three XOR logic gates).

To evaluate the functionality and performance of both classic and improved Fibonacci code converters, simulations will be performed using NI Multisim software, an industry-standard tool for analysing the real behaviour of electronic circuits. These simulations will utilize logic circuits from the widely used 74LVC family. The improved Fibonacci code converter demonstrates the ability to operate at significantly higher clock frequencies than the classic implementation, which is the main contribution of this paper. Speci-fically, the improved design allows for more than a 180% increase in clock frequency for converters with one XOR logic gate in the feedback loop of the shift register, and over a 57% increase for converters with three XOR logic gates. This substantial rise in clock frequency not only accelerates the code conversion process but also enhances the performance of the entire position encoder, which is crucial for many applications, especially those requiring higher resolutions. The proposed solution is not only theoretically significant but also practically valuable due to the broad applicability of pseudorandom position encoders.

The paper is organized as follows. Section 2 describes the classic implementation of the Fibonacci code converter as known in the literature. Section 3 presents an improved implementation of the Fibonacci code converter, as the main contribution of this paper. Section 4 discusses numerical results, and Section 5 concludes the paper.

2 Classic implementation of the Fibonacci pseudorandom/natural code converter

A maximal length PRBS (m-sequence) of resolution n consists of $2^n - 1$ bits. Each sequence of n consecutive bits in the m-sequence forms a unique code word, with the last $n-1$ bits of one code word overlapping with the first $n-1$ bits of the next code word. There are $2^n - 1$ different n-bit code words in total. The m-sequence generator of resolution n consists of a linear feedback shift register (LFSR) containing n D flip-flops and a feedback loop with XOR logic gates. Each code word represents one of the $2^n - 1$ possible states of the shift register, starting from a predefined reference state. The all-zero state is not allowed, as the shift register would be unable to exit that state. Each m-sequence has an inverse sequence, appearing as its mirror image. Starting from the same reference state, the inverse sequence generator will pass through the same states as the generator of the original (direct) sequence but in reverse order.

In an n-bit pseudorandom absolute position encoder, a code disk is attached to the rotating system whose angular position is measured. The disk is divided into 2^n-1 angular sectors, each assigned a unique n-bit pseudorandom code word from a direct m-sequence inscribed on the disk. Angular position is measured relative to a reference (zero) sector, encoded with a specific reference code word $Y_1^0 Y_2^0 \ldots Y_n^0$. To determine the current angular position, the pseudorandom code word $Y_1 Y_2 \ldots Y_n$ corresponding to the current sector is read from the disk. Since pseudorandom code words cannot be directly processed by digital electronics, the read n-bit code word is converted into an n-bit natural binary code word.

Figure 1 shows the block diagram of the classic implementation of the Fibonacci pseudorandom/natural code converter for an arbitrary resolution n, as described in [20]. This converter transforms an input pseudo-random code word $Y_1 Y_2 \ldots Y_n$ into an output code word $P_1 P_2 \ldots P_n$ of the natural binary code. The core component of the Fibonacci code converter is the Fibonacci generator of the m-sequence inverse in relation to the m-sequence written on the code disk. Generator of the inverse m-sequence consists of a shift register with a feedback circuit, where the shift register is implemented using n D-type flip-flops $\mathrm{FF}_1, \ldots, \mathrm{FF}_n$. The structure of the feedback path, specifically the number and position of XOR logic gates within it, is determined by the generator polynomial $P_n(X) = X^n + c_{n-1}X^{n-1} + \cdots + c_1 X + 1$, where c_i ($i = 1, \ldots, n-1$) are binary coefficients that can be either 0 or 1 [12,13]. If $c_{n-i} = 1$ ($i = 1, \ldots, n-1$), the corresponding connection and XOR_i logic gate exist in the feedback circuit. Conversely, if $c_{n-i} = 0$, the corresponding connectionand XOR_i logic gate do not exist. Thus, if k denotes the number of coefficients c_{n-i} ($i = 1, \ldots, n-1$) equal to 1, then k XOR logic gates exist in the feedback path.

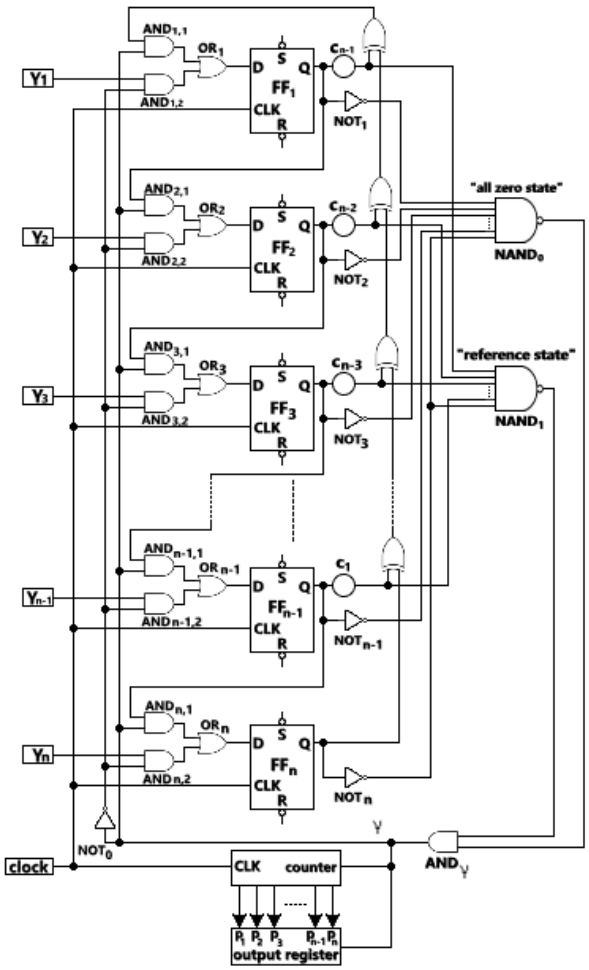

Fig. 1. Classic implementation of the Fibonacci code converter for an arbitrary resolution n

The Fibonacci pseudorandom/natural code converter operates as follows. Initially, the bits of the read pseudo-random code word $Y_1 Y_2 \ldots Y_n$ are loaded into the flip-flops of the shift register. With each clock pulse, the shift register transitions to a new state until it reaches the reference state $Y_1^0 Y_2^0 \ldots Y_n^0$. A counter counts the number of clock pulses from the start of the conversion until the shift register attains the reference state. The counter's value at this moment represents the code word $P_1 P_2 \ldots P_n$ in natural binary code.

In addition to the Fibonacci generator for the inverse m-sequence, the Fibonacci code converter includes several crucial components for proper operation. Firstly, there is a logic circuit consisting of NOT gates ($\mathrm{NOT}_1, \ldots, \mathrm{NOT}_n$) and a NAND_1 gate, designed to check if the shift register has reached the reference state. If the bit Y_i^0 ($i = 1, \ldots, n$) of the reference code word is 0, the output of the flip-flop FF_i passes through the corresponding NOT_i gate before reaching the NAND_1

gate. Conversely, if $Y_i^0 = 1$, the output of the flip-flop FF_i goes directly to the $NAND_1$ gate. Figure 1 illustrates this setup for the reference code word $Y_1^0 Y_2^0 \dots Y_n^0 = 11\dots10$.

We previously noted that a shift register state of all zeros is prohibited, as the register cannot transition out of this state. During normal operation, the shift register's design prevents this state. However, during the startup phase of the Fibonacci converter, the shift register might initially be in an all-zero state, hindering functionality. To address this, a $NAND_0$ logic gate is included in the converter circuit. The $NAND_0$ gate's inputs are connected to the inverted outputs of all the flip-flops in the shift register. If the register enters an all-zero state at the startup phase, the $NAND_0$ gate outputs a logic 0, triggering a control signal, $\gamma = 0$. This signal initiates the writing of bits from the read code word into the flip-flops, allowing the shift register to exit the all-zero state and resume normal operation. Once the startup phase is complete, the $NAND_0$ gate no longer affects the system, and only the $NAND_1$ gate controls the γ bit.

If the bit γ at the output of the ANDγ logic gate is 1, the conversion of the current pseudorandom code word is not yet complete and must continue. When the shift register reaches the reference state $Y_1^0 Y_2^0 \dots Y_n^0$, bit γ becomes 0, indicating the end of the current pseudorandom code word conversion. At this point, the current content of the counter (representing the output code word in natural binary code) is written to the output register; then, the counter is reset to zero, preparing for the conversion of the next pseudorandom code word.

In each clock pulse, a new bit is written into each flip-flop. There are two possibilities: if $\gamma = 1$, the current conversion is not finished, and bits obtained by the shifting process of the shift register are written into the flip-flops. If $\gamma = 0$, the current conversion is complete, and bits of the new pseudorandom code word are written into the flip-flops, preparing for the next conversion. In the classic implementation of the Fibonacci code converter shown in Fig. 1, bits are written into the flip-flops only through the D inputs for both cases ($\gamma = 0$ and $\gamma = 1$). Therefore, there are three logic gates ($AND_{i,1}$, $AND_{i,2}$ and OR_i) at the input of each flip-flop FF_i ($i = 1, \dots, n$), controlling the writing of bits into flip-flops. If $\gamma = 1$, bits obtained by the shift register's shifting process are written into the flip-flops through the $AND_{i,1}$ and OR_i gates. If $\gamma = 0$, bits of the next pseudorandom code word are written into the flip-flops through the $AND_{i,2}$ and OR_i gates.

To better explain the Fibonacci code converter, let's consider the conversion process for an 8-bit resolution, as illustrated in Fig. 2. This converter is based on the generator polynomial $P_8(X) = X^8 + X^6 + X^5 + X^2 + 1$ [12,13]. The coefficients are: $c_7 = 0$, $c_6 = 1$, $c_5 = 1$, $c_4 = 0$, $c_3 = 0$, $c_2 = 1$, $c_1 = 0$. Thus, there are $k = 3$ non-zero coefficients, requiring three XOR logic gates in the feedback loop. Consider $Y_1 Y_2 \dots Y_8 = 01100011$ as the pseudorandom code word to be converted to natural binary code, with $Y_1^0 Y_2^0 \dots Y_8^0 = 10100000$ as the reference pseudorandom code word (reference state). Initially, the pseudorandom code word $Y_1 Y_2 \dots Y_8$ is loaded into the shift register. With each clock pulse, the shift register transitions through the following states: 01100011 → 11000110 → 10001101 → 00011010 → 00110101 → 01101010 → 11010100 → 10101000 → 01010000 → 10100000. After the 9th clock pulse, the shift register reaches the reference state 10100000. Starting from 0, the counter counts to $p = 9$. Therefore, the result of the code conversion process is the natural binary code word $P_1 P_2 \dots P_8 = 00001001$, which is the binary representation of $p = 9$.

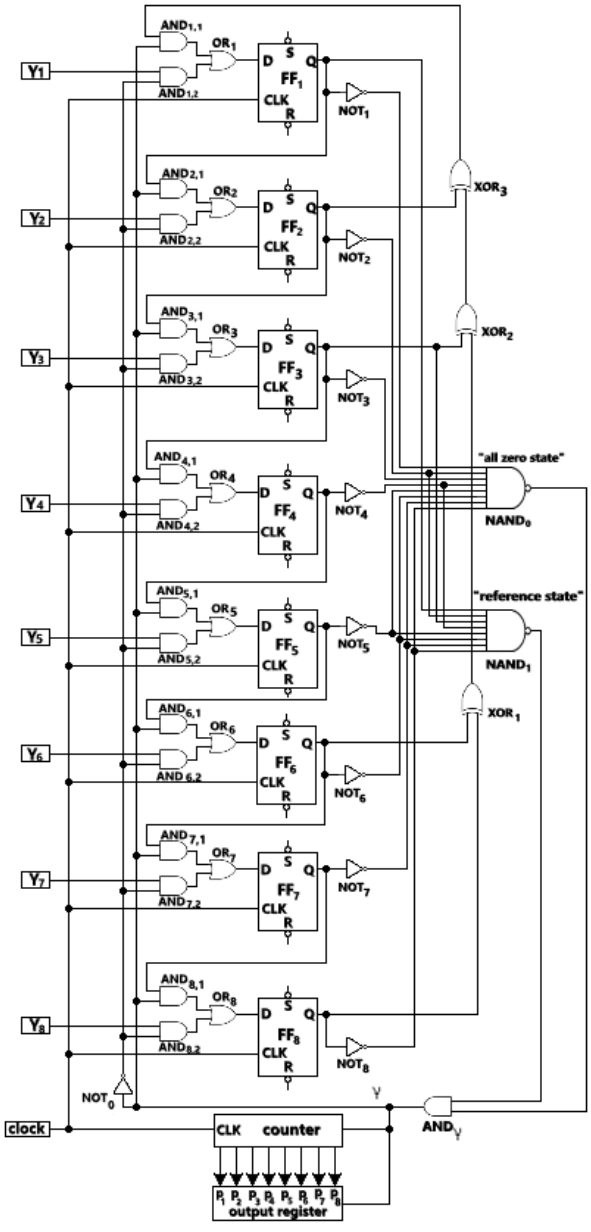

Fig. 2. Classic implementation of the Fibonacci code converter for resolution $n = 8$

3 Improved implementation of the Fibonacci pseudorandom/natural code converter

As previously described, the classic implementation of the Fibonacci pseudorandom/natural code converter involves writing bits into flip-flops solely through the D input, regardless of whether $\gamma = 0$ or $\gamma = 1$. This approach necessitates additional logic gates $\mathrm{AND}_{i,1}$, $\mathrm{AND}_{i,2}$ and OR_i at the flip-flop inputs, which increases propagation delay. In this section, we propose an improved implementation of the Fibonacci code converter by modifying the bit-writing process into the flip-flops. The new approach separates the bit-writing paths for $\gamma = 1$ and $\gamma = 0$. Specifically, when $\gamma = 1$, bits resulting from the shift register's shifting process are written into the flip-flops via the D inputs. Conversely, when $\gamma = 0$, bits of the next pseudorandom code word are written into the flip-flops using the set/reset (S/R) inputs. This improvement leverages the following rules of D flip-flop operation:

(*i*): if S = 1 and R = 1 then Q = D;

(*ii*): if S ≠ R (i.e. S = $\overline{\mathrm{R}}$) then Q = R, irrespective of D.

Using those rules, the paper proposes the modified bit-writing scheme into the flip-flops, as illustrated in Fig. 3, as its main contribution. Let us consider an arbitrary *i*-th flip-flop FF_i. The bit resulting from the shift register's shifting process is applied to the D input. The signals at the S and R inputs are defined as follows: S = $\overline{Y_i}$ OR γ and R = Y_i OR γ, where Y_i is the *i*-th bit of the next pseudorandom code word and $\overline{Y_i}$ = NOT(Y_i). Here's how the scheme functions.

- When $\gamma = 1$: S = 1 and R = 1. According to rule (*i*), Q = D, meaning the bit from the D input (obtained by shifting) is written into the flip-flop.
- When $\gamma = 0$: S = $\overline{Y_i}$ and R = Y_i, meaning that S ≠ R. According to rule (*ii*), Q = R = Y_i, so the bit Y_i of the next pseudorandom code word is written into the flip-flop.

As we can see, the proposed modification to the Fibonacci code converter ensures accurate bit-writing into the flip-flops, allowing the same functionality as the classic Fibonacci converter. However, it eliminates $\mathrm{AND}_{i,1}$, $\mathrm{AND}_{i,2}$ and OR_i gates from Fig. 1, decreasing the propagation delay in this way and significantly increasing the clock frequency and accelerating the code conversion process, as will be demonstrated in the next section.

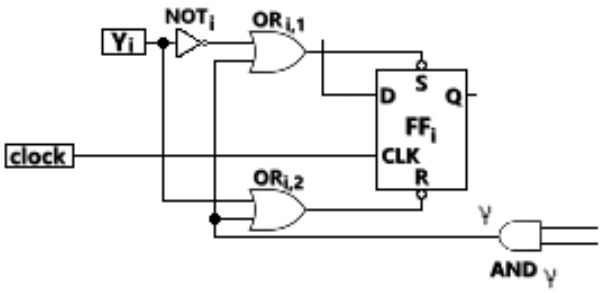

Fig. 3. New method of writing bits into the flip-flops of the shift register

The improved Fibonacci pseudorandom/natural code converter, which incorporates the modified method for writing bits into flip-flops, is illustrated in Fig. 4 for an arbitrary resolution *n*. Additionally, Fig. 5 demonstrates this converter for a resolution of $n = 8$.

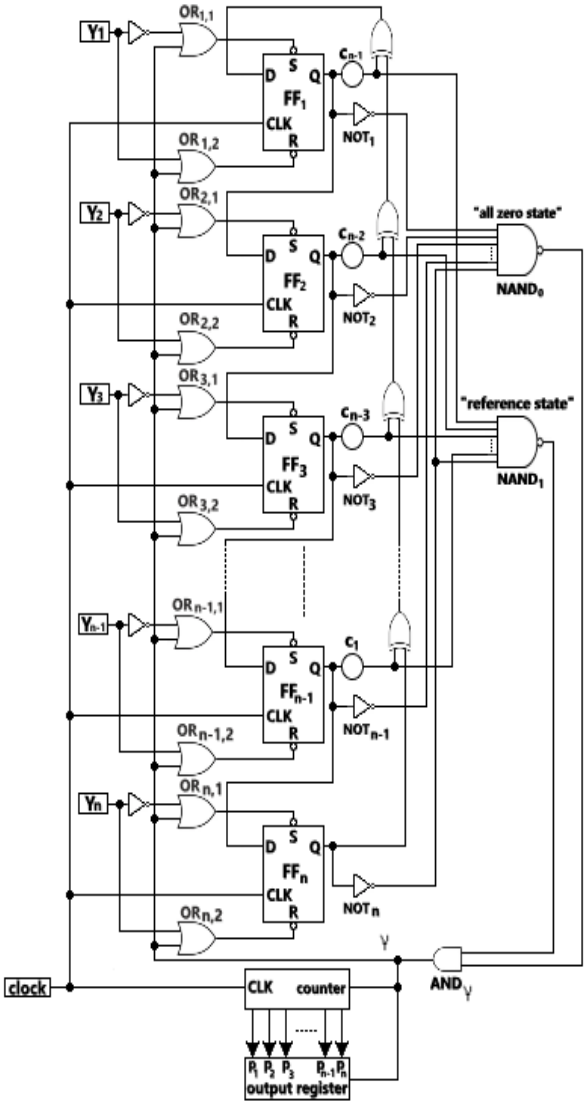

Fig. 4. Improved Fibonacci pseudorandom/natural code converter for an arbitrary resolution

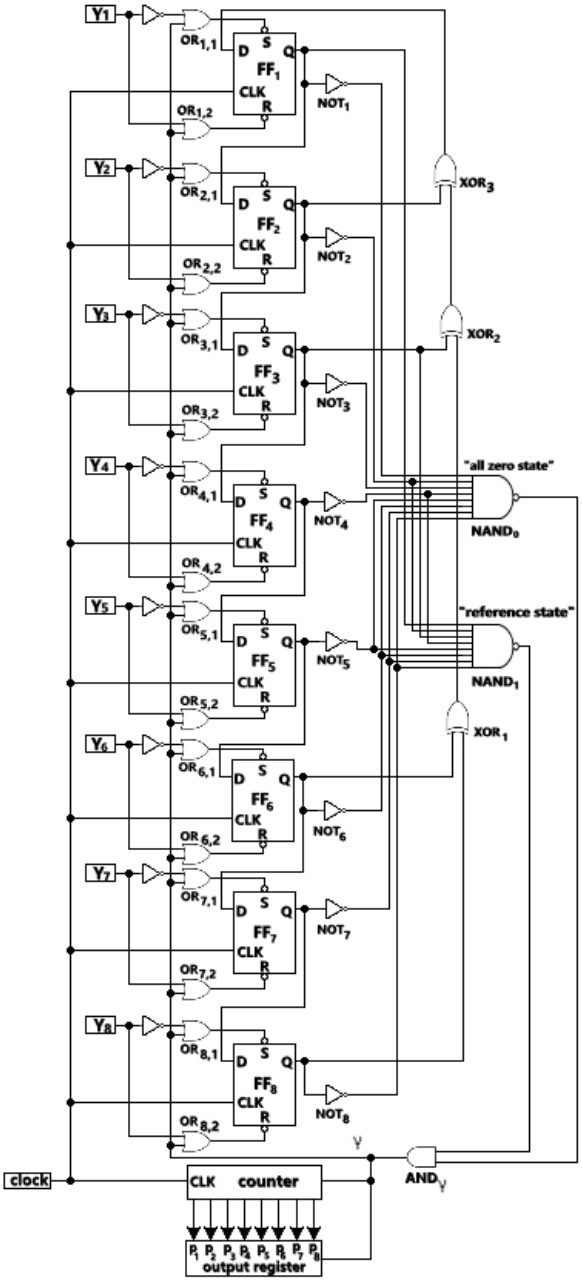

Fig. 5. Improved Fibonacci pseudorandom/natural code converter for resolution $n = 8$

4 Results and discussion

As resolution changes, the number of XOR gates in the feedback loop varies influencing the propagation delay, while the impact of other logic circuits on propagation delay remains constant. The number of XOR gates is determined by the parameter k, which indicates the number of coefficients equal to 1 in the generator polynomial $P_n(X)$. For a given resolution n, different m-sequences can be generated due to the existence of different generator polynomials $P_n(X)$ with different values of the parameter k. Since k affects the number of XOR gates, the smallest value of k is preferred in practical implementations to reduce complexity, cost and propagation delay. Examination of generator polynomials [12,13] for various resolutions n (up to several tens) reveals that the minimal values of k are 1 or 3. This means that a Fibonacci converter can be implemented with either one or three XOR gates in the feedback loop, depending on the resolution n. Therefore, the implementation of these code converters will be analysed for two specific resolution values: $n = 6$ as a representative example of the Fibonacci converter with one XOR gate, and $n = 8$ as a representative example of the Fibonacci converter with three XOR gates.

We will analyse the performance of the both types of Fibonacci code converters (classic and improved) described earlier (refer to Figs. 1 and 4) using NI Multisim, a SPICE simulation software for analog, digital, and power electronics. Multisim enables detailed analysis of circuit behaviour, including both functionality and propagation delays. For our simulations, we will use logic gates from the widely adopted 74LVC family. The software's flexibility allows us to adjust the clock frequency over a broad range. By simulating both the classic and improved Fibonacci code converters at various clock frequencies, we can determine the maximum clock frequency $f_{\max}$ for which the converters still function correctly. The results of these simulations are summarized in Table 1.

The main finding is that the improved implementation of the Fibonacci code converter operates at much higher clock frequencies than the traditional version. Specifically, the clock frequency rises from 28.98 MHz (for $k = 1$ and $k = 3$) to 81.30 MHz (for $k = 1$) and 45.66 MHz (for $k = 3$), marking increases of over 180.54% and 57.56%, respectively. This substantial improvement in clock frequency not only accelerates the code conversion process but also boosts the performance of the entire position encoder, which is crucial for many applications, particularly those requiring higher resolutions.

Table 1. Values of the maximum clock frequency $f_{\max}$ for the classical and improved Fibonacci code converters obtained by simulations

n	k	Classic implementation	Improved implementation
		$f_{\max}$ (MHz)	$f_{\max}$ (MHz)
6	1	28.98	81.30
8	3	28.98	45.66

5 Conclusion

This paper presented a modified implementation of the serial Fibonacci pseudorandom/natural code converters, allowing them to operate at significantly higher clock frequencies than the classic implementation described in the literature. The modification involved altering the process of writing bits into the flip-flops of the Fibonacci converter. This change aimed to separate the bit-writing paths for a new pseudorandom code word (at the beginning of a new code word conversion) from those used during the shifting process of the current code word conversion. By eliminating the need for certain logic circuits at the input of the flip-flops, the modification reduced propagation delays and increased the clock frequency, thereby accelerating the code conversion process. The paper detailed both the classical and modified implementations of the code converter. Simulations of both versions were conducted using Multisim software with 74LVC series logic circuits to assess their real-world behaviour. The results demonstrated that the modified implementation achieved clock frequencies by 180.54% or 57.56% higher than the classical version, depending on whether one or three XOR logic gates were used in the code converter feedback circuit. This increase in clock frequency not only significantly accelerated the code conversion process but also enhanced the performance of the entire position encoder, which is especially important for emerging high resolution applications.

Acknowledgement

This work was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia [grant number 451-03-66/2024-03/200102].

References

[1] W. Chen, W. Zhang, X. Wang, D. Hao, and Y. Liu, "A common-mode suppression structure for absolute optical encoders", *Optics Communications*, vol. 546, 129741, 2023.

[2] E. M. Petriu, "Absolute position measurement using pseudo-random binary encoding", *IEEE Transactions on Instrumentation and Measurement*, vol. 1, no. 3, pp. 19-23, 1998.

[3] R. Akkaya, and F. A. Kazan, "A new method for angular speed measurement with absolute encoder", *Elektronika ir Elektrotechnika*, vol. 26, no. 1, pp. 18-22, 2020.

[4] J. Zhao, W. Ou, N. Cai, Z. Wu, and H. Wang, "Measurement Error Analysis and Compensation for Optical Encoders: A Review", *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1-30, 1006230, 2024.

[5] H. Wang, J. Wang, B. Chen, P. Xiao, X. Chen, N. Cai, B. Wing, and K. Ling, "Absolute optical imaging position encoder", *Measurement*, vol. 67, pp. 42-50, 2015.

[6] R. N. A. Algburi, and H. Gao, "Health Assessment and Fault Detection System for an Industrial Robot Using the Rotary Encoder Signal", *Energies*, vol. 12, no. 14, 2816, 2019.

[7] I. Stojković, G. Miljković, D. Denić, and D. Živanović, "Application of pseudorandom position encoder for crane positioning", *2017 International Scientific Conference UNITECH*, Gabrovo, Bulgaria, November 2017.

[8] B. Shorkry, R. M. Daoud, and H. H. Amer, "Fault-tolerant rotary Gray encoder for industrial applications", 2023 *12th Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, 2023, pp. 1-5.

[9] J. M. Fuertes, B. Balle, and E. Ventura, "Absolute-type shaft encoding using LSFR sequences with prescribed length", *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 5, pp. 915-922, 2008.

[10] D. Denić, G. Miljković, J. Lukić, and M. Arsić, "Pseudo-random position encoder with improved zero position adjustment", *Facta universitatis - series: Electronics and Energetics*, vol. 25, no. 2, pp. 113 – 120, 2012.

[11] D. Denic, M. Dincic, G. Miljkovic, and Z. Peric, "A contribution to the design of fast code converters for position encoders", *International Journal of Electronics*, vol. 103, no. 10, pp. 1654-1664, 2016.

[12] S. Hassan, and M. U. Bokhari, "Design of Pseudo Random Number Generator using Linear Feedback Shift Register", *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 2, pp. 1956-1965, 2019.

[13] J. Ramasamy, and D. Samiappan, "Design and Analysis of Non-vulnerable PRBS Generation with Internal Shift and XOR of LFSR", *Innovations in Electrical and Electronic Engineering*, vol. 1109, pp. 573-582, 2024.

[14] K. Onuma, and S. Miyata, "An Improved Intensity Factor of Correlation-Based Steganography Using M-Sequence and DCT**",** *SN Computer Science*, vol. 5, no. 12, 2024.

[15] M. -Y. Cho, H. T. Thom and J. -F. Hsu, "Fault Diagnosis for High Voltage Distribution Networks Using Pseudorandom Binary Sequence and Cross Correlation Technique", *2016 3rd International Conference on Green Technology and Sustainable Development (GTSD)*, Kaohsiung, Taiwan, 2016, pp. 185-190.

[16] A. Srivastav, P. Nguyen, M. McConnell, K. A. Loparo, and S. Mandal, "A Highly Digital Multiantenna Ground-Penetrating Radar (GPR) System", *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 7422-7436, 2020.

[17] Z. Gu, F. Li, X. Dong, B. Zhou, and S. Fang, "Estimation of pollutant source using source-receptor response matrix from maximum-length sequence pulse technique", *Sustainable Cities and Society*, vol. 101, 105205, 2024.

[18] D. Denić, and G. Miljković, "Code reading synchronization method for pseudorandom position encoders", *Sensors and Actuators A: Physical*, vol. 150, no. 2, pp. 188-191, 2009.

[19] G. Miljković, and D. Denić, "Redundant and flexible pseudo-random optical rotary encoder", *Elektronika ir Elektro-technika*, vol. 26, no. 6, pp. 10-16, 2020.

[20] G. Miljković, D. Denić, M. Simić, and D. Živanović, "Implementation of New Serial Pseudorandom/Natural Code Converter applied to Position Encoders", in *2013 Proceedings of the 13th International Conference Research and Development in Mechanical Industry – RaDMI*, Kopaonik, Serbia, September 2013, vol. 2, pp. 869-874.

Received 23 July 2024