FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 38, Nº 1, March 2025, pp. 109 - 126
https://doi.org/10.2298/FUEE2501109D

Original scientific paper

PERFORMANCE ANALYSIS OF PSEUDORANDOM ABSOLUTE POSITION ENCODER WITH EMBEDDED SERIAL PSEUDORANDOM/NATURAL CODE CONVERTER

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

University of Niš, Faculty of Electronic Engineering Niš, Department of Measurements, Niš, Serbia

ORCID iDs:	Milan R. Dinčić	https://orcid.org/0000-0001-7508-0277
	Milica S. Stojanović	https://orcid.org/0009-0000-5528-3675
	Goran S. Miljković	https://orcid.org/0009-0004-1973-2911
	Dragan B. Denić	https://orcid.org/0000-0001-5582-0944

Abstract. *Pseudorandom absolute position encoders represent an advanced version of the widely used absolute position encoders, particularly suitable for high-resolution angular position measurement. This paper presents a detailed performance analysis of the pseudorandom absolute position encoder with embedded serial pseudorandom/natural code converter, focusing on the maximum operating frequency and the absolute error in angular position measurement. The generalized analysis, applicable to various resolution values, represents a significant contribution of the paper. Determining the maximum operating frequency involves a detailed analysis of propagation delays in the serial code converter circuit. It is demonstrated that encoder resolution profoundly impacts performance, with a dual effect: higher resolution decreases the absolute error but also reduces the maximum operating frequency, necessitating the determination of the most suitable resolution value for each specific application. In addition to its theoretical importance, the generalized analysis aids practical application by facilitating performance calculations and the selection of the most suitable resolution for specific applications. The performance evaluation was conducted for code converters implemented using the widely used 74LVC logic circuits, considering a 6-bit converter as a representative example of converters with a single XOR logic gate in the feedback loop of the shift register, and an 8-bit converter as a representative example of serial converters with three XOR logic gates in the feedback loop. By applying the proposed analysis, the maximum clock frequency was determined to be 29.85 MHz for both resolution values (6 and 8 bits). Simulations in NI Multisim software validate the performed analysis, showing a strong correlation between simulation and theoretical results.*

Received July 23, 2024; revised October 2, 2024; accepted October 16, 2024
Corresponding author: Milan R. Dinčić
University of Niš, Faculty of Electronic Engineering Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia.
E-mail: milan.dincic@elfak.ni.ac.rs

Key words: *angular position measurement, pseudorandom absolute position encoder, pseudorandom binary sequence, serial pseudorandom/natural code converter, propagation delay analysis*

1. INTRODUCTION

Absolute position encoders [1-14] are transducers utilized for determining the linear or angular position of a movable system, with numerous applications such as in automotive manufacturing, intelligent robots, aerospace [1], radar systems, machine positioning, printers, mechanical arms, machine health monitoring [2], CNC (computer numerical control) machines [3], integrated circuits manufacturing [2, 4], crane positioning [5], industrial motor shaft positioning [6], scanners, telescopes, process lines and elevators. When focusing on angular position measurement, absolute position encoders typically feature a code disk attached to the rotating system. This disk is divided into distinct angular sectors, each corresponding to a unique *n*-bit binary code word, where *n* represents the encoder's resolution. In commonly used optical absolute encoders, the bits of these code words are represented on the code disk by transparent and opaque (or reflective and non-reflective) fields. The current angular sector of the rotating system is identified by reading the corresponding code word from the code disk using LED diodes and phototransistors [1]. Absolute position encoders offer significant advantages over other angular position sensors, such as potentiometers, synchros, resolvers, and rotary variable differential transformers, by providing direct digital information and eliminating the need for analog-to-digital conversion [15]. Additionally, they can immediately provide angular position information upon power restoration, unlike incremental encoders that require initialization to a reference position [1, 3, 7, 8]. Classic absolute encoders typically use Gray (or natural) binary code [6], necessitating *n* concentric code tracks on the code disk and *n* optical reading heads for an *n*-bit absolute position encoder. Thus, as the resolution increases, the complexity also rises, making it challenging to apply classic absolute encoders in emerging high-resolution applications, which is their main drawback [8, 16].

Pseudorandom absolute position encoders [3, 5, 8, 17-21] are an advanced type of encoders that use pseudorandom code words inscribed along a single code track on the code disk to encode angular sectors. These code words are part of a pseudorandom binary sequence (PRBS) of maximum length, known as an m-sequence [22-25], generated using a linear feedback shift register (LFSR) [26-31]. Since two consecutive pseudorandom code words in an m-sequence differ by only one bit, detecting only that bit is required, with the rest inferred from the previous code word, allowing the use of a single reading head [5, 8, 19]. Thus, the pseudorandom encoder design, which includes a single code track and one reading head regardless of resolution, maintains low complexity even as resolution increases. This simplifies the implementation of high-resolution encoders, which is their primary advantage [8]. Additionally, pseudorandom encoders enhance reliability by enabling the implementation of error detection methods [21] and allow for direct zero position adjustment after the code disk is mounted on the shaft [17]. However, a notable challenge arises from the incompatibility of pseudorandom code words with conventional digital electronics, necessitating their conversion into natural binary code. Two primary types of pseudorandom/natural code converters are employed for this purpose: the serial converter [8, 18, 20], which utilizes a shift register, and the parallel converter [8], retrieving code words from ROM memory. This paper considers the serial code converter, widely used in practical applications due to its simplicity

and ability to facilitate direct zero position adjustment with minimal hardware or software modifications [17].

An absolute position encoder identifies the current angular sector of the rotating system but may not pinpoint the exact position within it, which can lead to measurement errors. The maximum absolute error in angular position measurement is one of the crucial performance metrics for pseudorandom absolute position encoders. Equally significant is their maximum operating frequency, which indicates the highest rotational speed (measured in rotations per second) at which the encoder can accurately determine the current angular sector.

The paper thoroughly examines the performance of the pseudorandom absolute position encoder with an embedded serial pseudorandom/natural code converter, focusing on its maximum operating frequency and maximum absolute error in angular position measurement. The study finds that the maximum operating frequency depends on the maximum clock frequency of the serial code converter, which in turn is determined by its maximum propagation delay. Starting with an analysis of the propagation delay within the serial code converter circuit, the paper derives an expression for its maximum clock frequency. It then extends to derive equations for the maximum operating frequency of the encoder and its maximum absolute error in angular position measurement. It should be emphasized that these analyses are conducted in a generalized manner to accommodate any resolution value, thereby enhancing their significance and broad applicability. The performance of the encoder is assessed for two resolution values (6 and 8 bits), assuming the implementation of the serial code converter using logic gates from the 74LVC family. To validate the analysis, simulations of the serial code converter are performed using NI Multisim software [32], standardly used in research and industry for analyzing the real behavior of electronic circuits. The simulation results closely match the theoretical outcomes, confirming the correctness of the analysis.

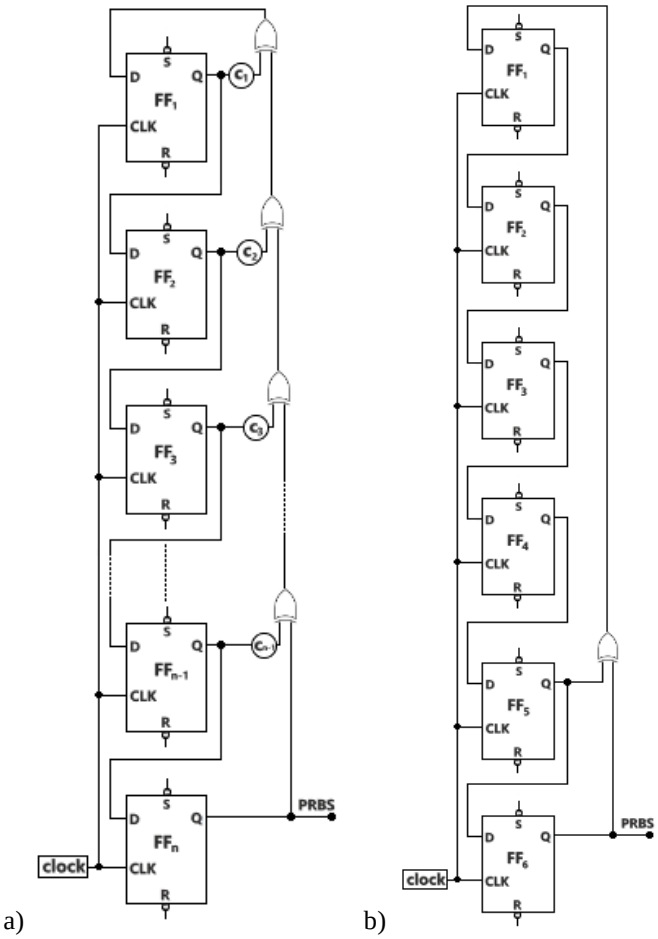
The theoretical analysis and numerical results outlined in the paper indicate that increasing the resolution affects the pseudorandom absolute position encoder in two ways. First, it lowers the maximum operating frequency, which poses a challenge. However, it also reduces the maximum absolute error in angular position measurement, which is advantageous. Hence, determining the most suitable resolution value tailored to the specific application is crucial, as it balances accuracy requirements and expected rotation speeds of the system.

Beyond its substantial theoretical contribution in formulating generalized expressions for the performance of pseudorandom absolute position encoders, the paper also carries notable practical implications, as the derived expressions enable the assessment of performance across different resolution values, facilitating the selection of the most suitable resolution value for each specific application.

The paper is organized as follows. Section 2 presents the generators of direct and inverse PRBS, while Section 3 provides a detailed description of the serial pseudorandom/natural binary code converter. Section 4 focuses on the analysis of propagation delay in the circuit of the serial pseudorandom/natural code converter, while Section 5 provides the performance analysis of the pseudorandom absolute position encoder. Numerical and simulation results for the serial code converter implemented by logic circuits from the 74LVC family are presented in Section 6, while Section 7 concludes the paper.

2. GENERATORS OF DIRECT AND INVERSE PRBS

A PRBS of maximum length, known as m-sequence, with a resolution of n, is a cyclic binary sequence with a period of $2^n - 1$ bits. It is generated by an n-bit linear shift register comprising n D-type flip-flops (FF_1, ..., FF_n), with a feedback branch [26-29], as illustrated in Fig. 1a. Bits of PRBS are obtained at the output of the last flip-flop FF_n. The number of different states of the shift register is $2^n - 1$ rather than 2^n, since the all-zero state is not allowed; if it were, the shift register would never be able to exit that state. PRBS generation starts from an initial (reference) state of the shift register, where any of the allowed $2^n - 1$ states can be selected as the reference state. With each clock cycle, the shift register moves to a new state, producing one PRBS bit until it completes $2^n - 1$ clock cycles encompassing all possible states and returns to the reference state. Each n-tuple of consecutive PRBS bits forms a unique code word, with the total of $2^n - 1$ different n-bit code words within one PRBS period. Each code word is equal to a distinct state of the shift register.



a) b)

Fig. 1 Generator of direct PRBS for a) an arbitrary resolution n; b) resolution $n = 6$

Inputs of flip-flops FF_i ($i = 2, \ldots, n$) are obtained directly from the outputs of the previous flip-flops FF_{i-1}, while the input to FF_1 is derived as a linear combination of flip-flops' outputs, generated by the feedback branch, comprising XOR logic gates implementing modulo 2 summation. The structure of the feedback branch, including the number and position of XOR logic gates, is pivotal for the design of the PRBS generator. Only with a specific structure of the feedback branch, the shift register can traverse through all $2^n - 1$ possible states, producing a PRBS of maximum length of $2^n - 1$ bits; otherwise, shorter sequences are generated. The feedback branch structure for a shift register of length n, producing the maximum length PRBS, is determined by binary coefficients c_i ($i = 1, \ldots, n$-1) of the primitive generator polynomial $P_n(X)$ of degree n, defined as [26-29]:

$$P_n(X) = X^n + c_{n-1}X^{n-1} + \cdots + c_1 X + 1. \tag{1}$$

The binary coefficient c_i ($i = 1, \ldots, n$-1) determines whether the output of the i-th flip-flop FF_i contributes to the modulo 2 summation; if $c_i = 1$, the output of FF_i participates in modulo 2 summation, otherwise it does not. With r coefficients equal to 1, there are r XOR logic gates in the feedback branch. If X_i represents the bit written in the i-th flip-flop FF_i, the state of the shift register of the direct PRBS generator is marked as $X_n X_{n-1} \ldots X_1$, implying that bits are observed from the last flip-flop (FF_n) towards the first (FF_1). The reference state is denoted as $X_n^0 X_{n-1}^0 \ldots X_1^0$.

For the resolution $n = 6$, the generator polynomial has the form $P_6(X) = X^6 + X^5 + 1$, giving the coefficients' values $c_5 = 1$, $c_4 = c_3 = c_2 = c_1 = 0$. The direct PRBS generator for resolution $n = 6$ is shown in Fig. 1b. Since only one coefficient is equal to 1, there is only one XOR logic gate in the feedback branch.

An essential aspect of implementing the serial Fibonacci code converter is the existence of an inverse sequence corresponding to each direct PRBS of maximum length, which resembles a mirror image of the direct sequence. The inverse sequence generator for arbitrary resolution n, depicted in Fig. 2a, comprises an n-bit shift register with a feedback branch, resembling the direct PRBS generator but with notable distinctions:

- the feedback branch of the inverse PRBS generator is structured similarly to the direct PRBS according to the generator polynomial $P_n(X)$ defined in (1). The difference is that the coefficient c_{n-i} ($i = 1, \ldots, n-1$) corresponds to the output of the i-th flip-flop FF_i; if $c_{n-i} = 1$, the output of FF_i contributes to the modulo-2 summation, otherwise it does not;
- bits of the inverse PRBS are taken from the input of the first flip-flop FF_1;
- if Y_i indicates the bit of the i-th flip-flop (FF_i) within the shift register of the inverse PRBS generator, the shift register's state is denoted as $Y_1 Y_2 \ldots Y_n$, with bits observed from the first flip-flop (FF_1) to the last (FF_n);
- if $Y_1^0 Y_2^0 \ldots Y_n^0$ is designated as the initial (reference) state of the shift register, serving as the starting point for generating the inverse PRBS, then $Y_1^0 Y_2^0 \ldots Y_n^0 = X_n^0 X_{n-1}^0 \ldots X_1^0$, meaning that the inverse PRBS originates from the identical reference state as the corresponding direct PRBS, with the distinction that the bits are entered into the shift register in reverse order, i.e. $Y_i^0 = X_{n+1-i}^0$, $i = 1, \ldots, n$.

Fig. 2b shows the inverse PRBS generator for a resolution of $n = 6$. When the reference state is set to 111100 for $n = 6$, the generator depicted in Fig. 1b produces the direct PRBS:

1111000001000011000101001111010001110010010110111011001101010 11, while the generator depicted in Fig. 2b generates the corresponding inverse PRBS: 1101010110011011101101001001110001011110010100011000010000001111. These sequences clearly exhibit mirror symmetry.

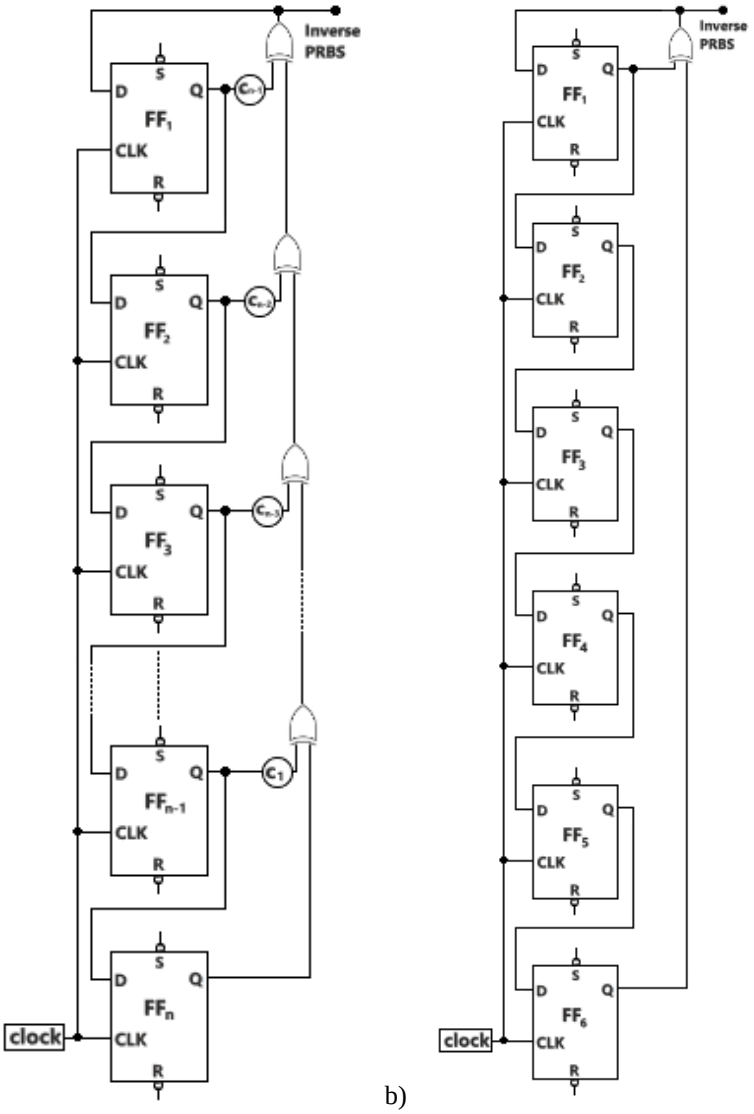


Fig. 2 Generator of inverse PRBS for a) an arbitrary resolution *n*; b) resolution *n* = 6

An important characteristic of the inverse PRBS generator is that its shift register, starting from the specified reference state, traverses all permissible states but in reverse order, compared to the shift register in the direct PRBS generator. For instance, the

following are some of the states through which the shift registers progress in both the direct and inverse PRBS generators while generating one PRBS period, considering a resolution of $n = 6$ and the reference state 111100:

- states of the shift register in the direct PRBS generator: 111100, 111000, 110000, 100000, 000001, 000010, 000100, 001000, 010000, ..., 101010, 010101, 101011, 010111, 101111, 011111, 111111, 111110, 111100;
- states of the shift register in the inverse PRBS generator: 111100, 111110, 111111, 011111, 101111, 010111, 101011, 010101, 101010, ..., 010000, 001000, 000100, 000010, 000001, 100000, 110000, 111000, 111100.

After passing through 63 different states, both the direct PRBS generator and the inverse PRBS generator return to the initial (reference) state 111100 (which is underlined to indicate the beginning of a new PRBS period).

3. Description of the Serial Pseudorandom/Natural Binary Code Converter

An important element on an *n*-bit pseudorandom absolute position encoder is the code disk, affixed to the rotating system whose angular position is being measured. The code disk is divided into $2^n - 1$ angular sectors, each corresponding to a unique *n*-bit pseudorandom code word within a direct PRBS of $2^n - 1$ bits written along the edge of the code disk. The angular position is measured relative to some zero (reference) sector, encoded with a reference code word $Y_1^0 Y_2^0 \ldots Y_n^0$, To ascertain the current angular sector of the rotating system (i.e. the distance from the reference sector), the initial step involves reading the pseudorandom code word $Y_1 Y_2 \ldots Y_n$ from the code disk corresponding to that specific sector. As digital electronics cannot process pseudorandom code words directly, the read *n*-bit code word requires conversion into an *n*-bit code word of the natural binary code.

The serial pseudorandom/natural code converter [8, 18, 20], whose block diagram is shown in Fig. 3 for an arbitrary resolution *n*, is one of the most widely utilized methods for converting pseudorandom code into natural binary code. It operates on the principle that each direct PRBS has a corresponding inverse PRBS. The inverse PRBS generator, which produces the inverse PRBS relative to the PRBS encoded on the code disk, is at the core of the serial code converter. The following is a brief overview of how the serial code converter operates. The read code word $Y_1 Y_2 \ldots Y_n$ is entered into the *n*-bit shift register of the inverse PRBS generator within the code converter. Starting from state $Y_1 Y_2 \ldots Y_n$, the shift register transitions to a new state at each clock cycle, traversing states in reverse order compared to the direct PRBS generator, thereby approaching the reference state $Y_1^0 Y_2^0 \ldots Y_n^0$. When the shift register reaches the reference state, the code conversion is completed. The serial code converter incorporates an *n*-bit counter that counts clock cycles from the start of the conversion. The output of the counter, at the moment when the shift register reaches the reference state, is an *n*-bit code word $P_1 P_2 \ldots P_n$ in natural binary code, representing the distance between the read code word $Y_1 Y_2 \ldots Y_n$ and the reference code word $Y_1^0 Y_2^0 \ldots Y_n^0$. This provides the distance between the current and reference sectors, which was intended to be measured. The code word $P_1 P_2 \ldots P_n$ is written into the output register as the result of the serial code conversion.

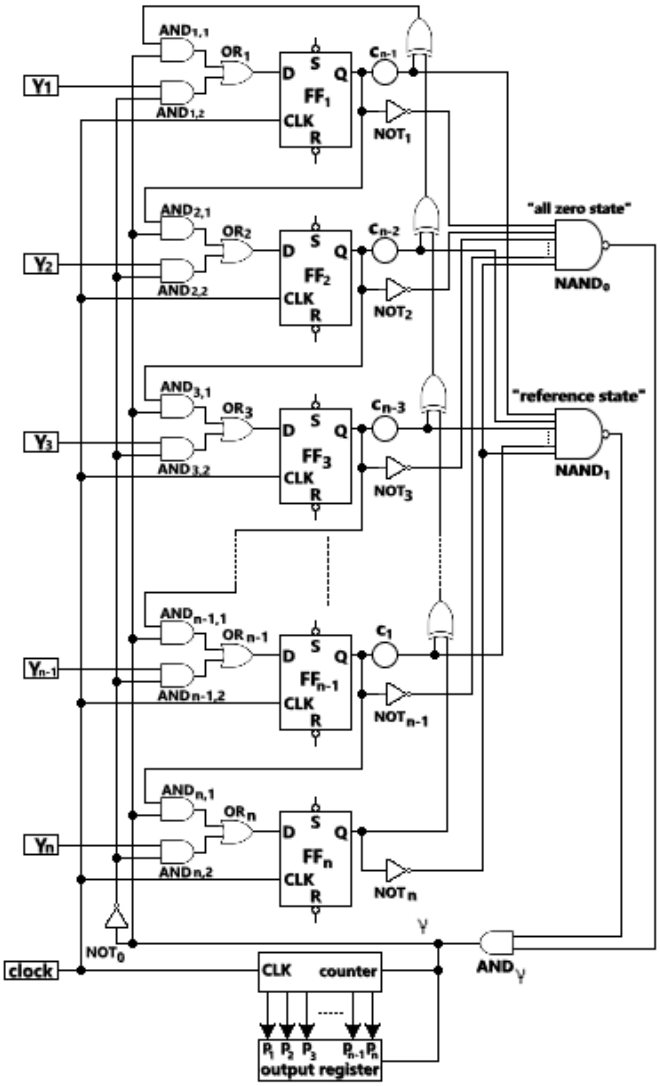


Fig. 3 The block diagram of the serial pseudorandom/natural code converter for an arbitrary resolution *n*

The serial code converter also includes several other essential elements crucial for its proper functioning. Firstly, the $NAND_1$ logic gate is used to determine whether the shift register has reached the reference state. The inputs to the $NAND_1$ gate come from the outputs of flip-flops within the shift register. If the *i*-th bit of the reference code word is equal to 1, the output of the corresponding flip-flop is directly linked to the $NAND_1$ gate. Conversely, if it is 0, the output of the *i*-th flip-flop is inverted by a NOT_i inverter before reaching the $NAND_1$ gate. Consequently, the $NAND_1$ gate outputs a logic 0 only when the shift register reaches the reference state; otherwise, it outputs logic 1. Fig. 3 is drawn

for the case where the reference code word is 111...10, composed of all ones except for the last bit, which is 0, resulting in a direct connection between the outputs of all flip-flops and the $NAND_1$ gate, except for the last flip-flop (FF_n), whose output is inverted before reaching the $NAND_1$ gate. If the output of the $NAND_1$ gate is 0, the corresponding γ bit at the output of the AND_γ gate is also set to 0, indicating the completion of the current pseudorandom code word conversion. At this point, the counter's current content is transferred to the output register as the resultant natural binary code word, and the counter resets to 0, preparing it for the next conversion cycle. In the next clock cycle, conversion of a new pseudorandom code word read from the code disk begins by writing its bits into the flip-flops of the shift register.

It was previously mentioned that a shift register state consisting of all zeros is prohibited since the shift register cannot exit that state. In normal operation, the shift register avoids entering such a state. However, there is a possibility that during the startup of the serial converter the shift register may initially find itself in an all-zero state, hindering proper operation. To address this, a $NAND_0$ logic gate is integrated into the converter circuit, with its inputs linked to the inverted outputs of all flip-flops. If the shift register encounters an all-zero state during startup, the $NAND_0$ gate outputs logic 0, triggering $\gamma = 0$, which commences writing bits of the read code word into the flip-flops. This action enables the shift register to exit the all-zero state and operate correctly. After startup, the $NAND_0$ no longer has any influence, with only the $NAND_1$ gate influencing the γ bit value.

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, …, n) in the shift register, responsible for managing the bit-writing process. Specifically, when $\gamma = 1$, the conversion of the current pseudorandom code word is not finished yet and bits obtained by the shifting process of the shift register are written into the flip-flops through $AND_{i,1}$ and OR_i gates. Conversely, when $\gamma = 0$, the conversion of the next pseudorandom code word begins by writing its bits into the flip-flops through $AND_{i,2}$ and OR_i gates.

To better explain the Fibonacci code converter, its operation will be considered for a specific 6-bit resolution, as depicted in Fig. 4. Its core component is the inverse PRBS generator from Fig. 2b. For the sake of illustration, let $Y_1Y_2...Y_6$ = 001000 represent the read pseudorandom code word to be converted into natural binary code, and let $Y_1^0Y_2^0\ldots Y_6^0$ = 111100 serve as the reference pseudorandom code word. Initially, the read pseudorandom code word is loaded into the shift register. Subsequently, with each clock pulse, the shift register transitions to a new state, progressing through the following states: 001000 → 000100 → 000010 → 000001 → 100000 → 110000 → 111000 → 111100. By the 8th clock pulse, the shift register reaches the reference state 111100. Counting from 0, after 8 clock pulses the counter will increment up to 7; its content will be $P_1P_2\ldots P_6$ = 000111, which is the resulting code word of the natural binary code, representing the value of position p = 7.

A pseudorandom/natural code converter for a resolution of n = 8 is shown in Fig. 5. Compared to Fig. 4, the difference lies in the feedback circuit. For a resolution of n = 8, the generator polynomial has the form $P_8(X) = X^8 + X^6 + X^5 + X^2 + 1$. In this case, three coefficients are equal to 1, resulting in 3 XOR logic gates in the feedback branch.

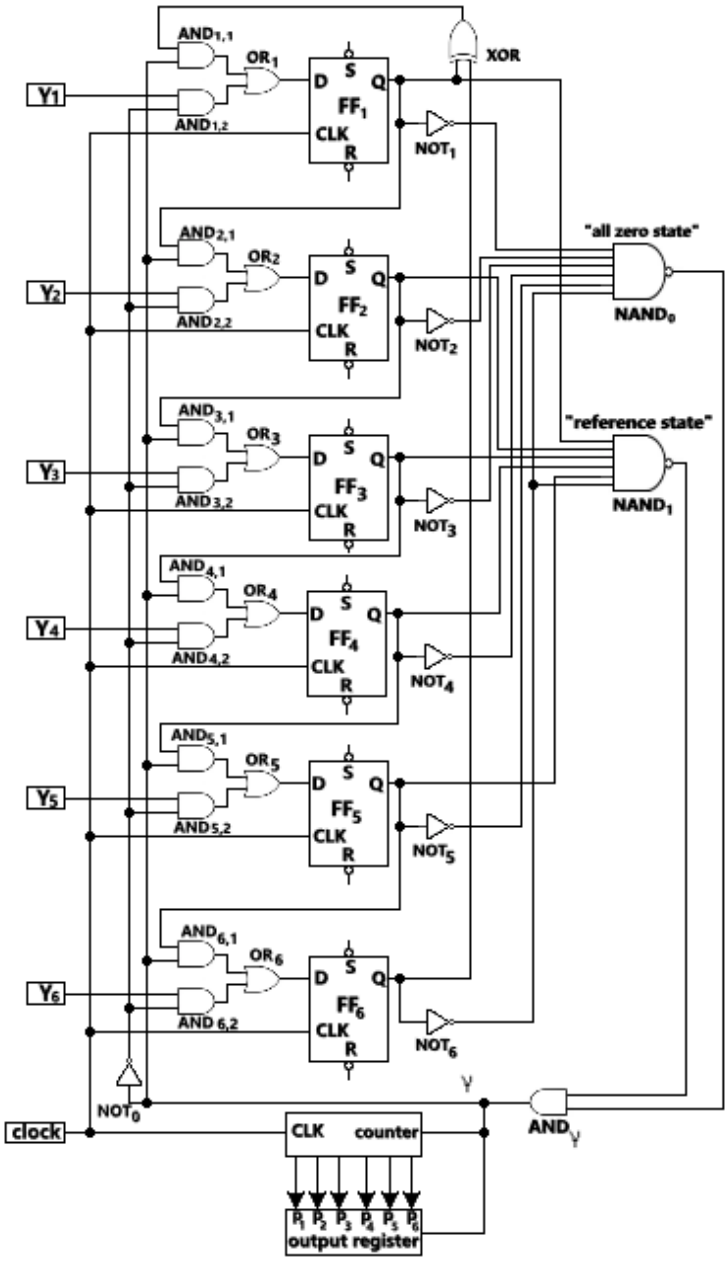


Fig. 4 The block diagram of the serial pseudorandom/natural code converter for resolution $n = 6$

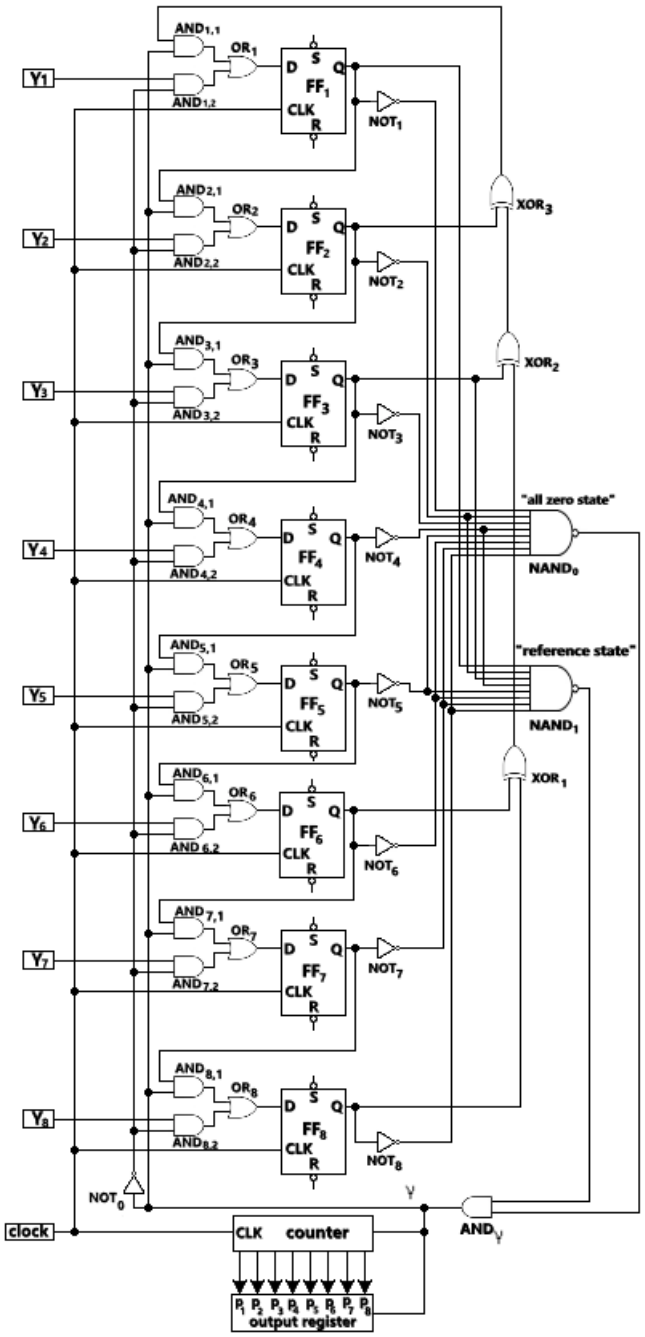


Fig. 5 The block diagram of the serial pseudorandom/natural code converter for resolution $n = 8$

4. Analysis of Propagation Delay in the Circuit of the Serial Pseudorandom/Natural Code Converter

Let τ_{AND}, τ_{OR}, τ_{NAND} and τ_{NOT} denote propagation delays of AND, OR, NAND and NOT gates, respectively. Considering the propagation delays of the D flip-flop, let $\tau_{\text{setup}}^{\text{FF}}$ denote the amount of time before the rising edge of the clock pulse during which the signal at the D input of the flip-flop must remain stable, while $\tau_{\text{CLK}\to\text{Q}}^{\text{FF}}$ denote the amount of time after the rising edge of the clock pulse required for the flip-flop to produce a change at the output Q.

Bits are written into the flip-flops at the rising edge of the clock signal. Therefore, all propagation delays in the circuit are measured relative to the rising edge of the clock.

The data path to obtain the bit γ is: flip-flops → NOT_i gates → NAND_1 gate → AND_γ gate. Hence, the propagation delay for obtaining the bit γ is:

$$\tau_\gamma = \tau_{\text{CLK}\to\text{Q}}^{\text{FF}} + \tau_{\text{NOT}} + \tau_{\text{NAND}} + \tau_{\text{AND}}. \tag{2}$$

If γ = 1, the conversion of the current pseudorandom code word continues by writing bits obtained through the shifting process of the shift register into the flip-flops using $\text{AND}_{i,1}$ and OR_i logic gates. Let a_i denote one input bit of the $\text{AND}_{i,1}$ gate (i = 1,...,n) obtained through the shifting process of the shift register, while the second input bit of the $\text{AND}_{i,1}$ gate is the bit γ. Let τ_{a_i} denote propagation delay to obtain the bit a_i. The data path to obtain the bit a_1 (which is one input of the $\text{AND}_{1,1}$ gate) is: flip-flops → feedback circuit, producing the propagation delay $\tau_{a_1} = \tau_{\text{CLK}\to\text{Q}}^{\text{FF}} + k\cdot\tau_{\text{XOR}}$, since the feedback circuit consists of k XOR logic gates connected in series. Bits a_i (i = 2,...,n) are obtained directly from the output of the previous flip-flop FF_{i-1}, producing the propagation delay $\tau_{a_i} = \tau_{\text{CLK}\to\text{Q}}^{\text{FF}}$. If τ_a denotes the maximum propagation delay for obtaining all the bits a_i (i = 1,...,n), according to the previous analysis, it follows that:

$$\tau_a = \max_{1\le i\le n}\{\tau_{a_i}\} = \tau_{a_1} = \tau_{\text{CLK}\to\text{Q}}^{\text{FF}} + k\cdot\tau_{\text{XOR}}. \tag{3}$$

Hence, for γ = 1, the delay required for all the bits obtained by the shifting process to be ready for writing into the D inputs of the flip-flops through $\text{AND}_{i,1}$ and OR_i gates is:

$$\begin{aligned}\tau_D^{\gamma=1} &= \max\{\tau_a, \tau_\gamma\} + \tau_{\text{AND}} + \tau_{\text{OR}} + \tau_{\text{setup}}^{\text{FF}} \\ &= \tau_{\text{CLK}\to\text{Q}}^{\text{FF}} + \tau_{\text{AND}} + \tau_{\text{OR}} + \max\{k\cdot\tau_{\text{XOR}}, \tau_{\text{NOT}} + \tau_{\text{NAND}} + \tau_{\text{AND}}\} + \tau_{\text{setup}}^{\text{FF}}\end{aligned}. \tag{4}$$

If γ = 0, a new conversion starts by writing bits of a new pseudorandom code word into the flip-flops through $\text{AND}_{i,2}$ and OR_i gates. The bits of the new pseudorandom code word $Y_1Y_2...Y_n$ are ready at the inputs of the $\text{AND}_{i,2}$ gates, but the formation of the bit $\bar{\gamma}$ (inverted γ bit) needs to be awaited, serving as the second input of the $\text{AND}_{i,2}$ gates. The propagation delay to obtain the bit $\bar{\gamma}$ is $\tau_{\bar{\gamma}} = \tau_\gamma + \tau_{\text{NOT}}$. Let $\tau_D^{\gamma=0}$ represent the time delay measured from the beginning of the current clock pulse, required for the bits of the new code word to be written into the D inputs of the flip-flops, for γ = 0:

$$\tau_D^{\gamma=0} = \tau_{\bar{\gamma}} + \tau_{\text{AND}} + \tau_{\text{OR}} + \tau_{\text{setup}}^{\text{FF}} = \tau_{\text{CLK}\to\text{Q}}^{\text{FF}} + 2\tau_{\text{NOT}} + \tau_{\text{NAND}} + 2\tau_{\text{AND}} + \tau_{\text{OR}} + \tau_{\text{setup}}^{\text{FF}}. \tag{5}$$

For the counter, the following delays are of particular interest: $\tau_{\mathrm{CLK}\to\mathrm{Q}}^{\mathrm{counter}}$ - the delay between the start of the clock pulse and the change in the counter output; $\tau_{\mathrm{setup}}^{\mathrm{counter}}$ - the minimum time the signal at the counter's input must remain stable before the rising edge of the clock; and $\tau_{\mathrm{R}\to\mathrm{Q}}^{\mathrm{counter}}$ - the time needed for the counter to reset after receiving a signal at the R input. When γ is equal to 1, the conversion is in progress, and the counter is in normal counting mode; hence, the delays of interest are $\tau_{\mathrm{CLK}\to\mathrm{Q}}^{\mathrm{counter}}$ and $\tau_{\mathrm{setup}}^{\mathrm{counter}}$. When γ becomes 0, the converter has reached the reference state and the counter is resetting, making the delay of interest $\tau_{\mathrm{R}\to\mathrm{Q}}^{\mathrm{counter}}$.

According to the previous analysis, the maximum delay in the serial code converter circuit for the case when γ = 0 is given by:

$$\tau^{\gamma=0} = \max\{\tau_D^{\gamma=0}, \tau_{\mathrm{R}\to\mathrm{Q}}^{\mathrm{counter}}\}, \tag{6}$$

whereas when γ = 1, the maximum delay is given by:

$$\tau^{\gamma=1} = \max\{\tau_D^{\gamma=1}, \tau_{\mathrm{CLK}\to\mathrm{Q}}^{\mathrm{counter}} + \tau_{\mathrm{setup}}^{\mathrm{counter}}\}. \tag{7}$$

Finally, the maximum propagation delay of the serial code converter circuit is:

$$\tau_{\max} = \max\{\tau^{\gamma=0}, \tau^{\gamma=1}\}. \tag{8}$$

5. Performance Analysis of the Pseudorandom Absolute Position Encoder

This section delves into the examination of two crucial performance metrics related to pseudorandom absolute position encoders: the maximum operating frequency and the maximum absolute error. It is important to emphasize that this analysis is generalized, ensuring its validity across various resolution values.

5.1. Maximum operating frequency of the pseudorandom absolute position encoder

The conversion of a pseudorandom code word, which is p positions away from the reference code word, will require (p + 1) clock cycles: one clock cycle to write the code word into the shift register of the converter, followed by an additional p clock cycles to reach the reference state. Starting from 0, the counter reaches the value of p after (p + 1) clock cycles. The duration of the conversion is:

$$\delta = (p+1)\cdot T^{clock}, \tag{9}$$

where T^{clock} is the period of the clock pulses of the converter and $f^{clock} = 1/T^{clock}$ is clock frequency. The longest conversion time occurs for the code word that is furthest from the reference code word, i.e., corresponding to the maximum position $p_{\max} = 2^n - 2$, and is equal to:

$$\delta_{\max} = (p_{\max}+1)\cdot T^{clock} = (2^n - 1)\cdot T^{clock}. \tag{10}$$

Let f denote the rotation frequency (i.e., the number of revolutions per second) of the rotating system whose angular position is being measured. Then, $T = 1/f$ represents the

rotation period, i.e., the time it takes for the rotating system to complete one full revolution. For a resolution of n, the time it takes for the rotating system to traverse the width of one angular sector is:

$$\theta = \frac{T}{2^n - 1} = \frac{1}{f(2^n - 1)}, \tag{11}$$

which essentially represents the time interval between the readings of two consecutive code words from the code disk. In order for the pseudorandom absolute position encoder to function properly, the conversion of the current pseudorandom code word must be completed before a new code word is read. Therefore, the following condition must be satisfied:

$$\delta_{\max} \le \theta, \tag{12}$$

meaning that the maximum conversion duration must be shorter than the time between the readings of two consecutive code words.

Based on (10), (11) and (12), the following is obtained:

$$(2^n - 1) \cdot T^{clock} \le \frac{1}{f(2^n - 1)}. \tag{13}$$

From (13), it follows that:

$$f \le \frac{1}{(2^n - 1)^2 \cdot T^{clock}}. \tag{14}$$

The maximum operating frequency $f_{\max}$, at which the pseudorandom absolute position encoder can function correctly for a given resolution n, based on (14), is:

$$f_{\max} = \frac{1}{(2^n - 1)^2 \cdot T_{\min}^{clock}}, \tag{15}$$

where $T_{\min}^{clock}$ represents the minimum allowed value of the clock period. For the serial code converter to function correctly, the clock period must not be shorter than the maximum propagation delay in the code converter circuit $\tau_{\max}$, defined by (8), implying that:

$$T_{\min}^{clock} = \tau_{\max}. \tag{16}$$

Hence, the maximum allowed frequency of the clock pulses is:

$$f_{\max}^{clock} = \frac{1}{T_{\min}^{clock}} = \frac{1}{\tau_{\max}}. \tag{17}$$

Based on (15) and (16), the final expression for the maximum operating frequency is obtained:

$$f_{\max} = \frac{1}{(2^n - 1)^2 \cdot \tau_{\max}}. \tag{18}$$

Based on (18), it can be concluded that the maximum operating frequency of the pseudorandom absolute position encoder $f_{\max}$ is inversely proportional to the resolution n and significantly decreases as n increases. For a given resolution n, $f_{\max}$ depends on the maximum propagation delay $\tau_{\max}$ in the code converter circuit: the smaller the $\tau_{\max}$, the larger the $f_{\max}$ will be.

5.2. Absolute error of angular position measurement

Using pseudorandom absolute encoders, the angular sector in which the rotating system is located can be determined, but the exact angular position within that sector cannot. Considering that the angular position is measured in the middle of angular sectors, the maximum absolute error in measuring the angular position equals half the width of an angular sector:

$$\Delta_{max} = \frac{360^{\circ}}{2 \cdot (2^{n} - 1)}. \tag{19}$$

It can be seen that as the resolution n increases, the absolute error significantly decreases.

6. Numerical and Simulation Results

From expressions (2) through (8), it is observed that changing the resolution n only affects the parameter k, which represents the number of XOR gates in the feedback loop of the shift register, while all other parameters that influence the maximum propagation delay remain the same. For each specific resolution value n, different m-sequences can be generated using various generator polynomials, each with a distinct k value. However, to streamline practical implementations and reduce complexity, cost, and delay, the minimal value of k is commonly used for each specific resolution. Analysis of generator polynomials for all resolutions n of interest (ranging up to several tens) shows that the minimal k value can either be 1 or 3, depending on the resolution n [26-29]. Therefore, two resolutions will be considered: $n = 6$, as a representative example of a serial converter with one XOR logic gate in the feedback loop of the shift register, and $n = 8$, as a representative example of a serial converter with three XOR logic gates in the feedback loop.

To evaluate the performance of these code converters, they were implemented using the widely used 74LVC family of logic circuits. The typical propagation delays for these circuits are: $\tau_{AND} = 4.1$ ns, $\tau_{OR} = 3.8$ ns, $\tau_{NAND} = 6.3$ ns, $\tau_{NOT} = 4$ ns, $\tau_{XOR} = 5$ ns, $\tau_{setup}^{FF} = 7.4$ ns, $\tau_{CLK \to Q}^{FF} = 5.2$ ns, $\tau_{setup}^{counter} = 2.5$ ns, $\tau_{CLK \to Q}^{counter} = 7.3$ ns and $\tau_{R \to Q}^{counter} = 6.4$ ns. Theoretical values for the maximum clock frequency f_{max}^{clock} of the serial code converter derived from expression (18), and the maximum operating frequency of the pseudorandom encoder f_{max} defined by (19), are detailed in Table 1, for resolutions $n = 6$ and $n = 8$.

To validate the theoretical results, simulations of the considered 6-bit and 8-bit serial code converters were performed using NI Multisim, an industry-standard SPICE simulation tool for analog and digital electronics, which enables the analysis of functionality and propagation delays. The layout of the 6-bit serial code converter, as implemented in the Multisim simulator, is shown in Fig. 6. Multisim facilitates simulations across different clock frequencies, with the aim to determine the maximum clock frequency f_{max}^{clock} at which the serial code converter functions correctly. The values of f_{max}^{clock} obtained from these simulations are also presented in Table 1.

Table 1 also shows the maximum absolute error values Δ_{max} for angular position measurement with the pseudorandom absolute encoder, at resolutions of $n = 6$ and $n = 8$.

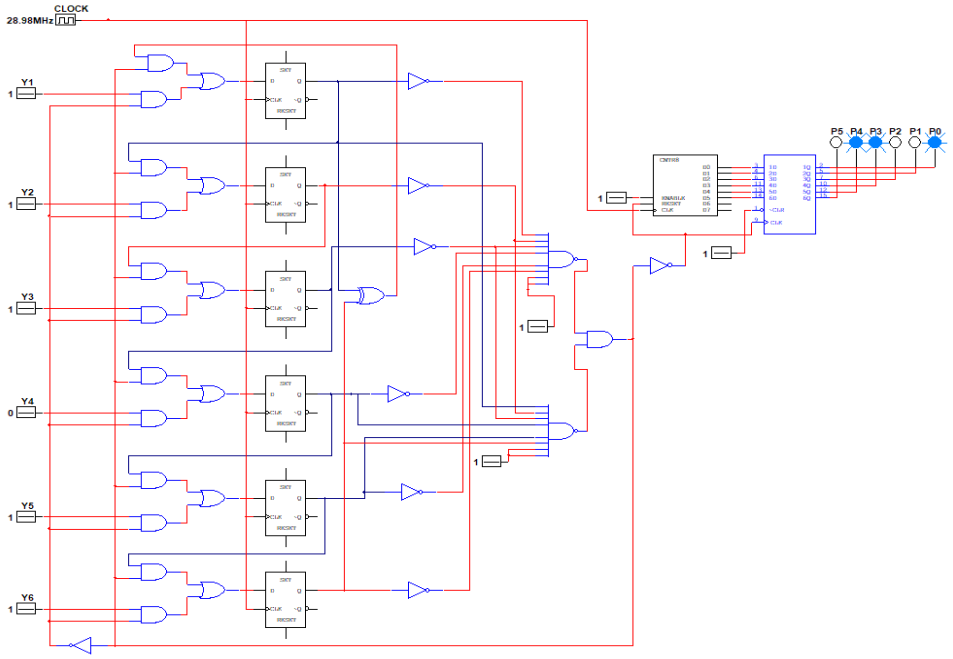


Fig. 6 Simulation of the serial pseudorandom/natural code converter in NI Multisim software for resolution $n = 6$

Table 1 Performance of the 6-bit and 8-bit serial code converters obtained by theory and by simulations

n	k	Theory/Simulation	$f_{\max}^{clock}$	$f_{\max}$	$\Delta_{\max}$
6	1	Th. Sim.	29.85 MHz 28.98 MHz	7.4 kHz	2.86 °
8	3	Th. Sim.	29.85 MHz 28.98 MHz	457.27 Hz	0.71 °

From Table 1, it can be seen that the theoretical and simulation values of $f_{\max}^{clock}$ are very close, indicating that the simulations validate the theoretical analysis. The slight difference between the theoretical and simulation values of $f_{\max}^{clock}$ is expected, as the actual delays of logic gates are never exactly the same as the given values but vary within a certain range around them.

Furthermore, Table 1 shows that the values of $f_{\max}^{clock}$ for $n = 6$ and $n = 8$ are identical. This is because increasing the number of XOR logic gates from 1 to 3 in the feedback loop of the shift register did not increase the maximum propagation delay, as other logic gates in the serial converter circuit predominantly influence it.

Based on (18) and (19) and the results in Table 1, it can be concluded that increasing the resolution n reduces the absolute measurement error in angular position $\Delta_{\max}$, which is advantageous. However, it also decreases the maximum frequency of the pseudorandom

absolute position encoder f_{max}, which is a drawback. Therefore, it is crucial to find the optimal value of resolution n for each specific application, considering both the required accuracy of angular position measurement and the desired maximum rotation frequency.

7. CONCLUSION

The paper is dedicated to determining the maximum clock frequency of the serial pseudorandom/natural code converter, as well as the performance metrics (maximum operating frequency and maximum absolute error) of the pseudorandom absolute position encoder. Initially, the paper provides a comprehensive explanation of the structure and operating principles of the serial code converter. Subsequently, an expression for the maximum clock frequency is derived through an in-depth analysis of propagation delays across all paths within the serial converter circuit. Additionally, expressions for the maximum operating frequency and maximum absolute error of the pseudorandom encoder are derived. Notably, the analysis is conducted in a generalized manner, applicable to any resolution value, thus ensuring broad applicability.

Assuming the implementation of the serial code converter using 74LVC family logic circuits, the maximum clock frequency values are determined based on theoretical analysis and simulations conducted using Multisim software. The close agreement between the theoretically derived and simulated maximum clock frequency values confirms the accuracy of the theoretical analysis. The analysis conducted in this paper has shown that the main factor affecting the propagation delay of the serial code converter is the logic circuits for writing bits into the flip-flops of the shift register and detecting the reference state, rather than the shift register's feedback branch. This was supported by numerical results (theoretical and simulated), yielding identical maximum clock frequencies for two possible values (1 or 3) of the number of XOR logic gates in the feedback branch of the shift register. This result can be used to focus future research related to increasing the operating frequency of the serial code converter on finding more efficient methods for writing bits into the flip-flops of the shift register.

Furthermore, the paper demonstrates that increasing the resolution decreases the absolute error in angular position measurement, while also reducing the maximum operating frequency of the pseudorandom position encoder. Therefore, for any specific application, it is essential to find an optimal resolution value considering both parameters. The performance expressions derived in this paper, which are valid for any resolution value, are of significant importance for selecting the optimal resolution value of the pseudorandom position encoder.

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", *Opt. Commun.*, vol. 546, 129741, 2023.

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

[3] 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.

[4] Y. Shi, Q. Zhou, X. Li, K. Ni and X. Wang, "Design and Testing of a Linear Encoder Capable of Measuring Absolute Distance", *Sensor. Actuat. A: Phys.*, vol. 308, p. 111935, 2020.

[5] I. Stojković, G. Miljković, D. Denić and D. Živanović, "Application of Pseudorandom Position Encoder for Crane Positioning", In Proceedings of the 2017 International Scientific Conference UNITECH 2017, Gabrovo, Bulgaria, 2017, vol. 3, pp. 465-468.
[6] B. Shorkry, R. M. Daoud and H. H. Amer, "Fault-Tolerant Rotary Gray Encoder for Industrial Applications", In Proceedings of the 12th Mediterranean Conference on Embedded Computing (MECO 2023), Budva, Montenegro, 2023, pp. 1-5.
[7] K. M. Lee, T. Gu and Y. B. Bang, "Analysis of Accuracy and Measuring Range of Dual Absolute Encoder System", *IEEE Sens. J.*, vol. 20, no. 6, pp. 2997-3004, 2020.
[8] E. M. Petriu, "Absolute Position Measurement Using Pseudorandom Binary Encoding", *IEEE Trans. Instrum. Meas.*, vol. 1, no. 3, pp. 19-23, 1998.
[9] R. Akkaya and F. A. Kazan, "A New Method for Angular Speed Measurement with Absolute Encoder", *Elektron. Elektrotech*, vol. 26, no. 1, pp. 18-22, 2020.
[10] B. Zhan, C. Wu, X. Liu, P. Huang and P. Yu, "An Integrated High Precision Absolute Angular Displacement Sensor", *Meas. Sci. Technol*., vol. 35, no. 9, 2024.
[11] S. Wang, L. Luo, J. Zhu, N. Shi and X. Li, "An Ultra-Precision Absolute-Type Multi-Degree-of-Freedom Grating Encoder", *Sensors, vol.* 22, no. 23, p. 9047, 2022.
[12] D. Gurauskis, D. Marinkovic, D. Mažeika and A. Kilikevičius, "Self-Calibratable Absolute Modular Rotary Encoder: Development and Experimental Research", *Micromachines*, vol. 15, no. 9, p. 1130, 2024.
[13] J. Yu, R. Dou, W. Zhang, X. Wang, J. Xu, J. Liu, N. Wu and L. Liu, "Precision Encoder Grating Mounting: A Near-Sensor Computing Approach", *Sci. Rep*., vol. 14, p. 22249, 2024.
[14] E. Celik and A. H. Obdan, "A New Absolute Encoder Design Based on Piecewise Pseudo-Linear Signals", *IEEE Sens. J.*, vol. 24, no. 16, pp. 25590-25600, 2024.
[15] J. G. Webster and H. Eren, *Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement*, 2nd edition. CRC Press, 2014.
[16] J. M. Fuertes, B. Balle and E. Ventura, "Absolute-Type Shaft Encoding Using LSFR Sequences with Prescribed Length", *IEEE Trans. Instrum. Meas*., vol. 57, no. 5, pp. 915-922, 2008.
[17] D. Denić, G. Miljković, J. Lukić and M. Arsić, "Pseudorandom Position Encoder with Improved Zero Position Adjustment", *FU: Elec. Energ*., vol. 25, no. 2, pp. 113-120, 2012.
[18] D. Denić, M. Dinčić, G. Miljković and Z. Perić, "A Contribution to the Design of Fast Code Converters for Position Encoders", *Int. J. Electron*., vol. 103, no. 10, pp. 1654-1664, 2016.
[19] D. Denić and G. Miljković, "Code Reading Synchronization Method for Pseudorandom Position Encoders", *Sensor. Actuat. A: Phys.*, vol. 150, no. 2, pp. 188-191, 2009.
[20] G. Miljković, D. Denić, M. Simić and D. Živanović, "Implementation of New Serial Pseudorandom/Natural Code Converter applied to Position Encoders", In Proceedings of the 13th International Conference Research and Development in Mechanical Industry – RaDMI 2013, Kopaonik, Serbia, 2013, vol. 2, pp. 869-874.
[21] G. Miljković and D. Denić, "Redundant and Flexible Pseudorandom Optical Rotary Encoder", *Elektron. Elektrotech*., vol. 26, no. 6, pp. 10-16, 2020.
[22] T. Etzion, *Sequences and the de Bruijn Graph: Properties, Constructions, and Applications*, 1st edition. Academic Press, 2024.
[23] T. Etzion, "Pseduo- Random and de Bruijn Array Codes", In Proceedings of the 2024 IEEE International Symposium on Information Theory (ISIT), Athens, Greece, 2024, pp. 1742-1747.
[24] X. Jing, A. Zhang and K. Feng, "Arithmetic Autocorrelation Distribution of Binary m-Sequences", *IEEE Trans. Inform. Theory*, vol. 69, no. 9, pp. 6040-6047, 2023.
[25] X. Jing and K. Feng, "Arithmetic Crosscorrelation of Binary M-Sequences with Coprime Periods", *Finite Fields Theor. Appl.*, vol. 96, p. 102424, 2024.
[26] S. Hassan and M. U. Bokhari, "Design of Pseudo Random Number Generator using Linear Feedback Shift Register", *Int. J. Eng. Adv. Technol. (IJEAT)*, vol. 9, no. 2, pp. 1956-1965, 2019.
[27] Z. Zulfikar, Y. Away and R. S. Noor, "FPGA-based Design System for a Two-Segment Fibonacci LFSR Random Number Generator", *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 7, no. 4, pp. 1882-1891, 2017.
[28] J. Ramasamy and D. Samiappan, "A Modified PRBS: Vertical Stacked LFSR Primitive Polynomial for Secure Data Communication", *Procedia Comput. Sci*., vol. 215, pp. 947-954, 2022.
[29] J. Ramasamy and D. Samiappan, "Design and Analysis of Non-vulnerable PRBS Generation with Internal Shift and XOR of LFSR", In Proceedings of the 4th International Conference on Electrical and Electronics Engineering, Himachal Pradesh, India, 2023, vol. 1, pp. 573-582.
[30] K. Onuma and S. Miyata, "An Improved Intensity Factor of Correlation-Based Steganography Using M-Sequence and DCT", *SN Comput. Sci.*, vol. 5, no. 12, 2024.
[31] 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", *Sustain. Cities Soc.*, vol. 101, p. 105205, 2024.
[32] F. Asadi, *Essential Circuit Analysis using NI Multisim™ and MATLAB.* Springer, 2022.