iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** II **Month of publication:** February 2024

DOI: https://doi.org/10.22214/ijraset.2024.58334

Implementation of High Speed and Area Efficient Polar Encoder

Dr.K.Usha[1], Dr. Vibha D.Kulkarni[2]
[1]Professor, [2]Asst.Prof
Department of Electronics and Communication Engineering,
[1]Maturi Venkata Subba Rao (MVSR) Engineering college, Hyderabad, India
[2]Vasavi College of Engineering, Hyderabad, India

Abstract: Polar codes were introduced by E.Arıkan in 2008. They are the first family of error-correcting codes that attain the capacity of binary memoryless and symmetric channels with efficient encoding, decoding, and construction algorithms. In this paper, implementation of high speed and area efficient polar encoder for systematic polar codes is presented. According to an iterative property of the generator matrix and particular lower triangular structure of the matrix, the number of XOR computations are reduced. In this implementation, total area of the encoder is decreased by 33.69%, delay is decreased by 54.6% and power consumption is decreased by 39.44%.
Index Terms: Polar codes, Generator matrix, encoder

I. INTRODUCTION

Polar codes, originally proposed by Arıkan [1], have gained enormous interest due to a number of distinctive features. For instance, polar codes have explicit coding structure and can achieve the capacity of Symmetric Binary Memoryless Channels (S-BMC). Moreover, polar codes with finite length yield competitive performance when compared to LDPC [2] and Turbo codes [3] in addition to having low encoding and decoding complexity. The standard polar codes are in nonsystematic form where both frozen bits and information bits (also referred to as user bits) are placed on the polarized bit-channels of the polarization structure and the user bits do not appear in the polar codeword. However, information bits as part of the codeword are required in some scenarios, such as the famous Turbo codes [4] whose component codes are systematic codes that can exchange information between modules in turbo decoding. To construct systematic polar codes (SPC), Arıkan proposed the idea of shifting the user bits from polarized bit-channels to unpolarized bit-channels [5], which makes the frozen and user bits lie on two different extremes of polarization structure. Arıkan showed that systematic polar codes outperform nonsystematic polar codes (NSPC) in terms of bit error rate (BER) and the performance have also been investigated in [6].

II. SYSTEMATIC AND NON-SYSTEMATIC POLAR CODES

Based on the channel polarization theory, the length of polar codes is N = 2n, n ≥ 1, where n is a positive integer. Let A represent the set of the indices of the information bits. The code rate is R = K/N, where K represents the length of information bits and is the number of the elements in a set of A. Information bits selection is determined by Bhattacharyya parameters [17]. Let Ac represent the complement of a set of A. The index set Ac = {0,1,…,N − 1} − A is for the frozen bits, and the length of Ac is N − K. $u = (u_0, u_1, \cdots, u_{N-1})$ represents the message vector, where u_i denotes an arbitrary element of the vector of u. $x = (x_0, x_1, \cdots, x_{N-1})$ represents a codeword vector, where x_i indicates a random component in the vector of x. The generator matrix G_N is defined as

$$G_N = F^{\otimes n} \tag{1}$$

where ⊗ denotes Kronecker power operation, $n = \log_2(N)$, and F denotes two-dimensional matrix F = [1, 0; 1, 1]. Applying the property of Kronecker product, we can construct a generator matrix as

$$G_N = \begin{bmatrix} F^{\otimes n-1} & 0 \\ F^{\otimes n-1} & F^{\otimes n-1} \end{bmatrix} = \begin{bmatrix} G_{N/2} & 0 \\ G_{N/2} & G_{N/2} \end{bmatrix} \tag{2}$$

The codeword vector of x for NSPC can be represented by the encoding Eq. (1)

$$x = uG_N = u_A G_A + u_{Ac} G_{Ac} \tag{3}$$

Where G_A is a submatrix of G_N, and it is constructed by the rows of indices in A. G_{Ac} is a submatrix of G_N and is constructed by the rows of indices in Ac. u_A = (ui: i∈A), u_A u and u_{Ac} = (u_j: j∈Ac), u_{Ac} u, and $u_{Ac} = u - u_A$. The symbol of ⊕ represents a mod-2 addition or a logical XOR operation in the binary domain.

Arikan [9] first proposed the mathematical formula shown in Eqs. (4) and (5) for SPC:

$$x_A= u_AG_{AA}+ u_{Ac}G_{AcA} \quad (4)$$

$$x_{Ac} = u_AG_{AAc}+ u_{Ac}G_{AcAc} \quad (5)$$

where G_{AA} denotes the submatrix of G_N consisting of the array of elements ($G_{i,j}$) with i∈A, j∈A, and G_{AA} = ($G_{i,j}$: i∈A, j∈A). Similarly for the other submatrices of G_{AcA} = ($G_{i,j}$: i∈Ac, j∈A), G_{AAc} = ($G_{i,j}$: i∈A, j∈Ac), and G_{AcAc} = ($G_{i,j}$: i∈Ac, j∈Ac). There is the same denotation for x_A = (x_i: i∈A) and x_{Ac} = (x_j: j∈Ac). If the matrix G_{AA} is invertible and the inputs to SPC encoder are u_{Ac} and $x_{A,}$ then the output x_{Ac} from the SPC encoder is

$$x_{Ac} =(x_A-u_{Ac}G_{AcA}\, G^{-1}{}_{AA}G_{AAc}+ u_{Ac}G_{AcAc} \quad (6)$$

Consider that the decoding results are not affected by the value of frozen bits [1], we can simplify the encoding procedure by setting zero values to all frozen bits, namely u_{Ac} = (ui = 0: i∈Ac). Then, x_{Ac} in Eq. (6) can be simplified to

$$x_{Ac}= x_AG^{-1}{}_{AA}G_{AAc} \quad (7)$$

where $G_{AA}{}^{-1}$ is a lower triangular matrix with ones on the diagonal and the submatrix of G_{AAc} also includes a lower triangular matrix. Hence, the matrix product of $G_{AA}{}^{-1}$ G_{AAc} has the same structure as G_{AAc}. It has a submatrix including a lower triangular matrix.

III. METHOD

The computational complexity can be decreased by reducing the number of logical XOR computing units. The following example illustrates the procedure of omitting XOR computing units. After logical XOR operations, the output results are the same either from the approach to apply a generator matrix or the method to apply an encoding diagram scheme. Figure 1 shows the SPC encoding diagram with N = 8 and A = {3,5,6,7}. In Fig. 1, the encoding direction is from bottom to top. The encoding process for information bits starts from right to left. Then, the encoding process for frozen bits is from left to right. The gray circles on the rightmost represent the information bits, and the black circles on the leftmost represent the frozen bits. The black arrow on the right represents the computing direction of the information bits. The black arrow on the left represents the computing direction of the frozen bits. Since the value of the frozen bits does not affect the encoding result, we set them to zero. The final outputs of the encoding are the value of the rightmost white circles.

For a SPC encoder in Fig. 1, we can reduce the XOR operation to obtain the output of x_0,x_1, x_2, and x_4. From Fig. 1, we can obtain

$$x_0 =u_0\oplus u_4\oplus u_2\oplus u_6\oplus u_1\oplus u_5\oplus u_3\oplus u_7 \quad (8)$$

Consider that u_0, u_1,u_2, and u_4 are frozen bits which are set zero. Then, Eq. (8) can be rewritten as

$$x_0=u_6\oplus u_5\oplus u_3\oplus u_7 \quad (9)$$

From Fig. 1, we also have the following relations

$$\begin{cases} u_7 = x_7; \\ u_6 =x_6\oplus x_7; \\ u_5 =x_5\oplus x_7; \\ u_3 =x_3\oplus x_7; \end{cases} \quad (10)$$

Substitute Eq. (10) into Eq. (9), x_0 becomes

$$x_0= (x_6\oplus x_7) \oplus(x_5\oplus x_7) \oplus(x_3\oplus x_7)\oplus x_7$$

After applying logical XOR operations, the output results become zero if the input values are the same. Thus, Eq. (11) can be rewritten as

$$x_0 =x_6\oplus x_1\oplus x_7$$

Similarly, x_2 and x_4 can be derived

$$x_2= u_2\oplus u_6\oplus u_3\oplus u_7=u_6\oplus u_3\oplus u_7=x_6\oplus x_7\oplus x_3\oplus x_7\oplus x_7 =x_6\oplus x_3\oplus x_7$$

$$x_4= u_4\oplus u_6\oplus u_5\oplus u_7= u_6\oplus u_5\oplus u_7 = x_6\oplus x_7\oplus x_5\oplus x_7\oplus x_7=x_6\oplus x_5\oplus x_7$$

By combining Eqns. (12), (13), and (14), we simplify the encoding diagram illustrated in Fig. 1 to the diagram in Fig. 2. Compared with the computation complexity of original algorithm [11], the proposed OEA reduces the computational units without increasing the memory bits. Figures 1 and 2 show the difference of the number of computing units.

There are only four XOR computing units to be reduced in Fig. 1. However, 9 XOR operations are omitted in Fig. 2. u_3, u_5, u_6, and u_7 are intermediate variables that can be ignored. The gray nodes on the rightmost represent the information bits, and the black nodes on the leftmost represent the frozen bits. The right black nodes are unknown. The outputs of the SPC encoder are x_0,x_1, x_2 and x_4.

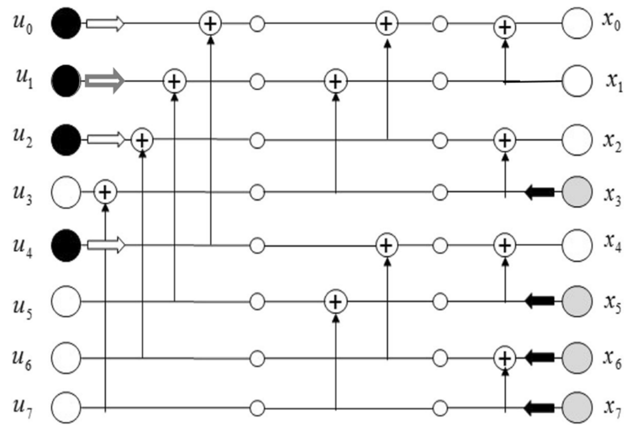
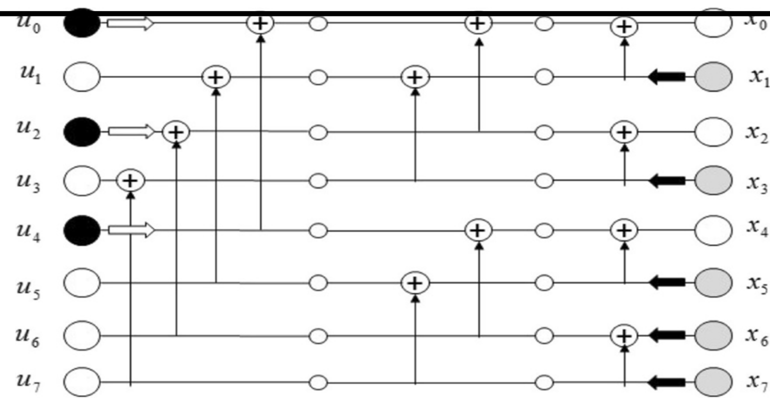


Fig. 1: SPC encoding diagram with N = 8 and A = {3, 5, 6, 7}: The encoding direction is from bottom to top. The encoding process for information bits starts from right to left. Then, the encoding process for frozen bits is from left to right. The gray circles on the rightmost represent the information bits, and the black circles on the leftmost represent the frozen bits. The black arrow on the right represents the computing direction of the information bits. The black arrow on the left represents the computing direction of the frozen bits. The final outputs of the encoding are the value of the rightmost white circles.



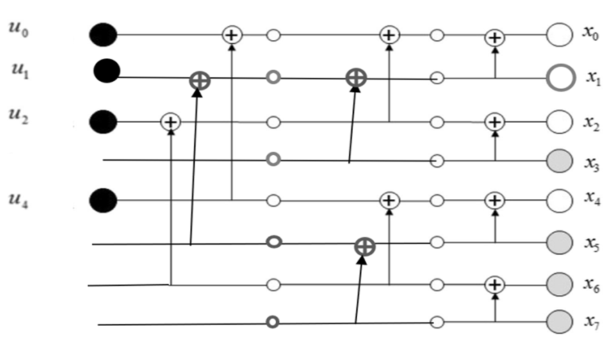


Fig. 2: SPC encoding optimization algorithm diagram: The gray nodes on the rightmost represent the information bits, and the black nodes on the leftmost represent the frozen bits. The right black nodes are unknown and the outputs of the SPC encoder are to obtain x_0, x_1,x_2, and x_4.

IV. SIMULATION RESULT

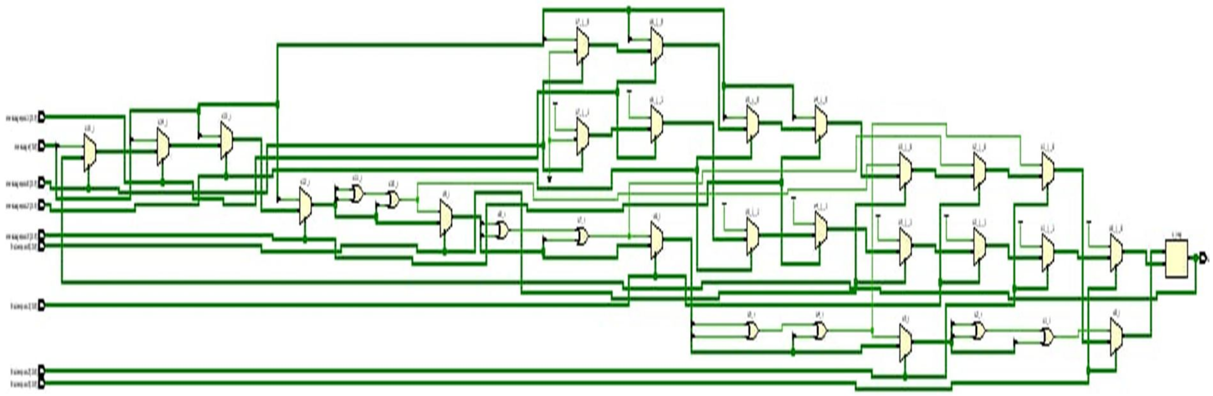


Figure 3: schematic of encoder

The figure3 shows the schematic of 8-bit high speed and area efficient polar encoder. There are 10 XOR gates which are less than the XOR gates present in low complexity polar encoder [4]

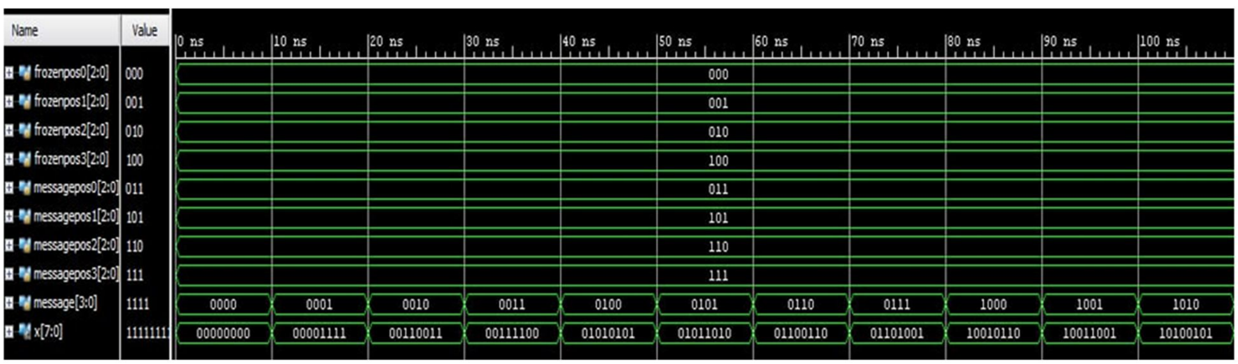


Figure 4: Verilog simulation results.

The fig 4 shows the simulation graph generated by using Verilog HDL in cadence, contains the encoded vector output for various 4 bit message vectors.

s.no	parameter	Previous work [3]	Present work
1	Area	4690.2 μm^2	3110.18 μm^2
2	Delay	10098ps	4581ps
3	Power consumption	179.296µW	108.576 µW

Table 4.1: Comparison of systematic encoders.

The above table shows the comparison between Area, Delay and Power Consumption of previous work [4] and present work .we can see that Area is decreased by 33.69% ,Delay is decreased by 54.6% and power consumption is decreased by 39.44%.

V. CONCLUSION

Polar codes are considered as a major breakthrough in channel coding area as they achieve the maximum channel capacity. There are various methods of polar encoding. In this project we implemented a high speed and area efficient polar encoder , generated its area, power and delay reports. We have compared with existing low complexity polar encoder and observed that the implemented polar encoder has reduced area, power and delay. Area is decreased by 33.69% , Delay is decreased by 54.6% and power consumption is decreased by 39.44%.

VI. FUTURE SCOPE

The optimal polar encoding algorithm not only reduces the number of XOR computing units compared with the existing non-recursive algorithms, but also is beneficial to hardware implementation compared with the existing recursive algorithms.

REFERENCES

[1] E. Arikan, Channel Polarization: A method for constructing capacity achieving codes for symmetric Serbian-input memoryless channels. IEEE Trans. Inf. Theory 55, 3051–3073 ,2009 .
[2] E. Arikan, Systematic polar coding. IEEE Commun Lett 15(8), 860– 862 ,2011 .
[3] H. Vangala, Y. Hong, E. Viterbo, Efficient algorithms for systematic polar encoding. IEEE Commun Lett 20(1),17–20 ,2016 .
[4] G. TaiChen, Z. Zhaoyang, Z. Caijun, Z. Liang, A low complexity encoding algorithm for systematic polar codes. IEEE Commun Lett 20(7), 1277–1280 ,2016 .