TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 2718 - 2723

Hardware Implementation of Polar Codes for very Short Length Messages

G.Aparna
Research Scholar, Department of ECE, Osmania Univerisity, Telangana.

Article Info
Volume 82
Page Number: 2718 - 2723
Publication Issue:
January-February 2020

Article History
Article Received: *14 March 2019*
Revised: *27 May 2019*
Accepted: *16 October 2019*
Publication: *18 January 2020*

Abstract:
An emerging error-detection and correcting technique developed in the recent years is Polar codes applicable in next generation wireless communications. This coding technique does not focus on randomization of the bits like other techniques does as it is based channel polarization and achieves almost Shannon Capacity also. This paper presents a successive cancellation (SC) algorithm based FPGA implementation of Polar codes for very short length messages of 8 bits. The implementation focuses on low complexity decoder for high speed applications and is to be extended for variable length like 32,128,256,512 bits to access the hardware complexity and design aspects for further implementation. Simulation results show the performance of polar codes the scope being that these codes outperform compared to LDPC and Turbo codes in wireless applications.

Keywords:*Polar codes, Successive Cancellation, FPGA, error detection and correction, Shannon theory.*

I. INTRODUCTION

Polar codes is used in the context of a coding problem which was traditionally formulated by Shannon back in 1948, which basically talks about transmitting the binary information through a noisy channel. The noisy channels are characterised by statistical models like BI-DMS, BPSK+AWGN etc as shown in the figure 1.Here binary bits are transmitted over the channel and noisy version of those bits are being received. Now, all that has to be done is to recover the original bits that's was transmitted from the noisy version of those bits. The Shannon's idea is to improve the performance of the system by doing something called as adding redundancy. In simple words K bits are taken and are converted to N bits, and receive the noisy versions of the N coded bits, and estimate the original message from these noisy coded bits [1]. Basically Encoding and Decoding is done here. Now, this system can perform much better than the earlier system. This is called as the coding system which can achieve the Shannon capacity only if it is done is done in a very intelligent process. This has been an open problem since many years. The Polar Coding system has an encoding block, a Decoding block and a Code-construction block. The Code-construction block decides the parameters that go in encoding and decoding blocks.The polar codes are probably the first ever capacity achieving codes. The capacity achieving codes are nothing but the codes that approach capacity which is nothing but when these codes are used to simulate their performance seems to be very close to the capacity [2]. Unfortunately there is no proof that they achieve capacity asymptotically in mathematical terms.

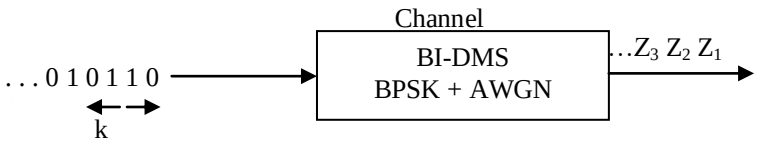

Fig. 1. Noisy channel representation

Arikan's [3] described the channel polarization process for memory less binary channels. S. Liu. Et al [4], proposed a block fading channel based polar codes. The fading process is described as a natural polarization as the fading block changes according to the fading coefficient. G. B .Ocherer [5] presented a higher-order modulation polar codes concept where

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 2718 - 2723

the bit reliabilities after successive demapping are estimated using the LM-rate, an achievable rate for mismatched decoding. Then Gaussian approximation is used to reconstruct the polar codes.

M. Mondelli, S. H. Hassani, and R. Urbanke [6] proposed the polar code with sublinear complexity and C. Condo, S. A. Hashemi, and W. J. Gross [7] presented multi-mode polar encoder and decoder with high reliability. The successive cancellation decoding algorithm was discussed by C. Condo et al [8] and S. A. Hashemi et al [9]. Minimum distance based and portioned list decoder was discussed in the literature [10]. The memory issues involved in the polar codes were also addressed [11].

II. Polar Codes

The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm, if possible.

Use 10-point type for the name(s) of the author(s) and 9-point type for the address(es) and the abstract. For the main text, please use 10-point type and single-line spacing. We recommend the use of Computer Modern Roman or Times. Italic type may be used to emphasize words in running text. Bold type and underlining should be avoided.

Papers not complying with the LNCS style will be reformatted. This can lead to an increase in the overall number of pages. We would therefore urge you not to squash your paper.

The polar code is desired with respect to three parameters namely N, K, I, where

1. N is mentioned to be the code length of the bits
2. K is utilized for representing the information present in the length in bits
3. I is utilized for representing the bit reversed as a pair of K indices

$$\mathcal{J} \subset \{0, 1, \dots, N-1\} \quad (1)$$

The above is mentioned to be the information bit indices

Bit reverse operation is nothing but simply taking n bit of integers and representing in the binary format and reverses the order into the required format takes them in the form of $\mathcal{J}$

$$b_1, b_2, \dots, b_n \triangleq b_n \dots b_2 b_1 \quad (2)$$

4. The complementary set $\mathcal{J}^c$ is mentioned to be the frozen bit indices.

2.1 Kernel

The kernel is also highly utilized technique in the coding process. This is a triangular kernel. It utilizes koneke product operations as an outcome the matrix dimensions gets doubled in each iteration. Initially it begins with the 2*2 size and then goes to 4*4 size and soon.

$$\boldsymbol{F^{\otimes n}} \triangleq F \otimes F \otimes \dots \ (n\ times) \quad (3)$$

2 by 2 matrixes

$$\boldsymbol{F^{\otimes 2}} = \begin{pmatrix} F & F \\ 0 & F \end{pmatrix} \quad (4)$$

4 by 4 matrixes

$$\boldsymbol{F^{\otimes 3}} = \begin{pmatrix} \boldsymbol{F^{\otimes 2}} & \boldsymbol{F^{\otimes 2}} \\ 0 & \boldsymbol{F^{\otimes 2}} \end{pmatrix} \quad (5)$$

The dimensions grow as the powers are smaller.

III. Encoding

Encoding vector is nothing but multiplying the vector with the kernel.

Encoding eq: (kbits → $Nbits$):

$$X = F^{\otimes n} d \ (\sim x = Gu) \quad (6)$$

Where $\begin{cases} d_{I^c} = 0, and \\ d_i = u \ \rightarrow the\ message \end{cases}$

The above process is quite similar to the traditional encoding process, where k message bits are

January - February 2020
ISSN: 0193 - 4120 Page No. 2718 - 2723

multiplied directly with the generator matrix (Gu)and obtains the information which is encoded. D is not the matrix but the value of the d is embedded inside the matrix for obtaining the efficient outcomes. In the traditional process the message gets varied with respect to the information given by the user. But in the novel encoding process the information goes efficiently without any interference of the user.

For example, if rectangular matrix is taken where the rectangular matrix is embedded inside the matrix then

Example: N = 8, K = 5, $\mathcal{J} = \{1, 3, 5, 6, 7\}$

Then x = $\boldsymbol{F^{\otimes 3}d}$

The d_0, d_2, d_4 represents the complementary information of the sets.

So due to this the developers basically use only few columns for translating the information, where the blue colored columns gets eliminated, which is quite similar to the classic form of linear encoding. It is not that much variable as compared to the traditional one. The matrix multiplication generally consists of quadratic complexity, which is not that much implementation friendly as lesser complex implementations. As the discussion is about polar codes which are asynchrotically high they need higher efficient techniques. The developers have one specific type which is necessary for them.

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} 1&1&1&1&1&1&1&1 \\ 0&1&0&1&0&1&0&1 \\ 0&0&1&1&0&0&1&1 \\ 0&0&0&1&0&0&0&1 \\ 0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&1&0&1 \\ 0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} d_0 = 0 \\ d_1 \\ d_2 = 0 \\ d_3 \\ d_4 = 0 \\ d_5 \\ d_6 \\ d_7 \end{pmatrix}$$

Fig. 2. Encoding example

A very efficient O (N log N) implementation is available. It is capable of reducing the N square order through the linear order of N log N information.

3.2Efficient O(N log_2 N) implementation

$$x = F^{\otimes n} d \ , in\ just \left(\frac{N}{2} log_2 N\right) XORs$$

(7)

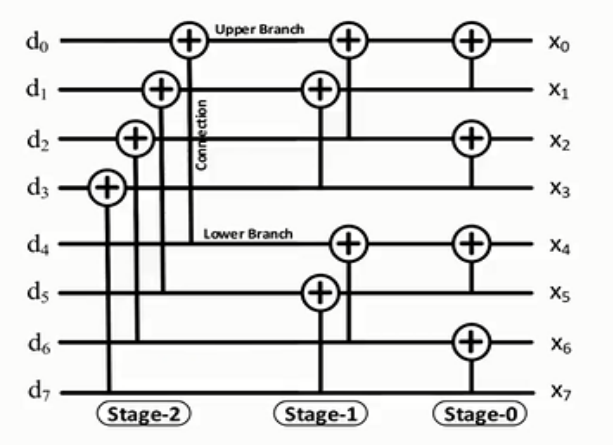

Fig. 3. Encoding computation tree

The above figure represents the edge or operation utilized in this technique. The $\boldsymbol{x_0}$value block implements one kernel as an outcome one can come to observe that whatever the input that will be executed here. In the stage 0 it consists of four f values. So by cascading in an intelligent fashion several small f values can be implemented here, whose value is efficiently computable. The above figure represents the three stage outcome. So every stretched circuit can be represented with the connection, and the upper & the lower branches. The above process is named as highly efficient process. It can be utilized by applying a mat lab code with reasonable understanding of the circuit.

Systematic coding is a method of coding which is capable of representing the entire coding process as an upending process of redundancy. So in general after making the process of encoding one cannot observe the messaged bits as same as the encoded bits. Though the redundancy being added or upended, redundancy is being added in a very linear passion between the n bits that are being output, but incase the developers can sustain the messages as a copy paste as it is in the code word that is called a symmetrical code.

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 2718 - 2723

IV. DECODER

Successive cancelation decoder is the most common and efficient decoder used in the industry for decoding polar codes. It's a greedy tree search based method, where instead of searching the binary tree in an exhaustive process one foot link is selected every time and the search is carried out.

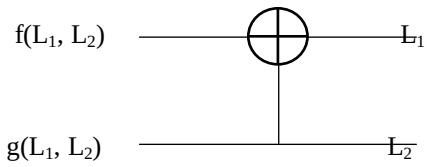

Fig. 4. Decoder basic structure

It is a two way recursion based process where if two inputs L1 and L2 are given to the fundamental construction block, two variable functions are produces as outputs. The second operation depends on the intermediate bit decisions from the upper branch.

$$\begin{pmatrix} L_1 \\ L_2 \end{pmatrix} \rightarrow \begin{pmatrix} f(L_1, L_2) \\ g(L_1, L_2) \end{pmatrix} = \begin{pmatrix} \frac{L_1 L_2 + 1}{L_1 + L_2} \\ L_1 . L_2 \ or \ L_2 / L_1 \end{pmatrix}$$

(7)

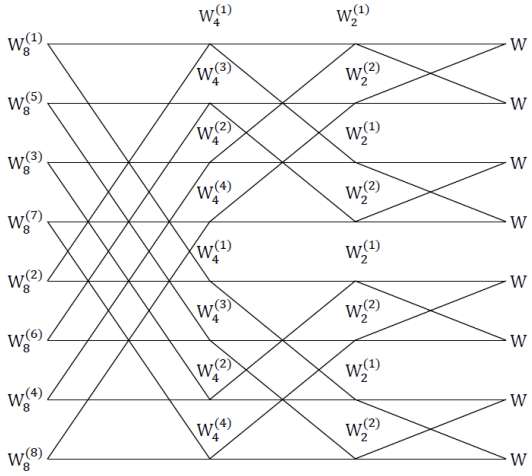

Fig. 5. Recursive transformation of decoder

Step 1: The rightmost side variable in Figure 5 are initialized with channel observations. For i=1,2,…..,N, $L_1^{(1)}(y_i) = \frac{w(y_i|u_i=0)}{w(y_i|u_i=1)}$.

(8)

Step 2: The first bit u is needed to be decoded in step 2. If u is a frozen bit, u is set to 0. Otherwise, update the likelihood ratios from the right most side to the left most side according to the rules of calculating f and g and obtain the required L . If L >1, u is determined to 0. If L<1, u is determined to 1. If L =1, u is determined to 0 or 1 with equal probability.

Step 3: In this step, u is decoded. The process is like step2 except that when calculating L , the value u is used when computing g functions. u is determined when L is obtained finally.

Step 4: Next, the remaining bits u , … . . , u are decoded in the order. Decoding each bit would make use of the knowledge of previous bits.

V. RESULTS

The polar encoder described in the above section is implemented using Verilog HDL in Xilinx 14.5 edition. The target board used is FPGA Vertex 7.

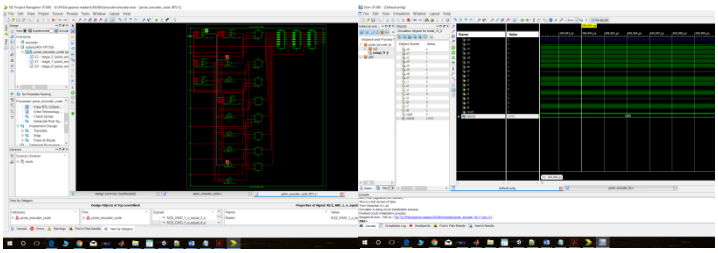

Fig. 6.Encoder schematic Gate level design

Fig. 7.Encoder Simulation Results

The simulation results show the encoded results of the 8 bit data. Table 1 presents the device utilization summary of the proposed algorithm.

Table 1.Device Utilization summary.

Logic Utilization	Used	Available
Number of Slice LUTs	17	150720
Number of fully used LUT-FF pairs	0	17
Number of bonded IOBs	20	600

January - February 2020
ISSN: 0193 - 4120 Page No. 2718 - 2723

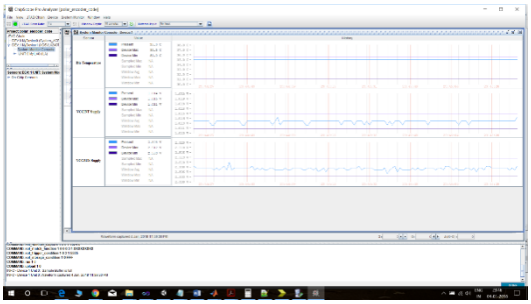

Fig. 8. Chip scope analyser results

The output obtained from the FPGA board are analysed in the chipscope analyser as shown in figure 8. The figures 9 and 10 present the schematic and gate level description of the polar decoder.

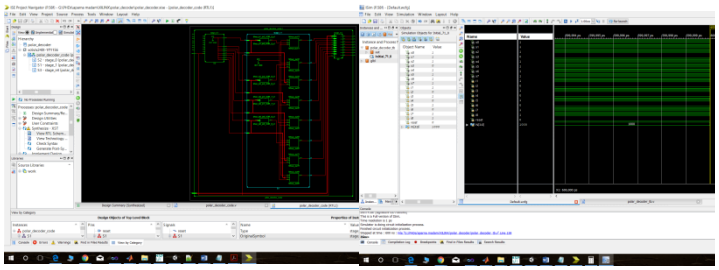

Fig. 9.Decoder Schematic Gate level design

Fig. 10. Simulation Results

Figure 10 shows the simulation result of the decoder. The decoded data matches with the input data bits given to the system.

Table 2.Device Utilization summary.

Logic Utilization	Used	Available
Number of Slice LUTs	17	150720
Number of fully used LUT-FF pairs	0	17
Number of bonded IOBs	21	600

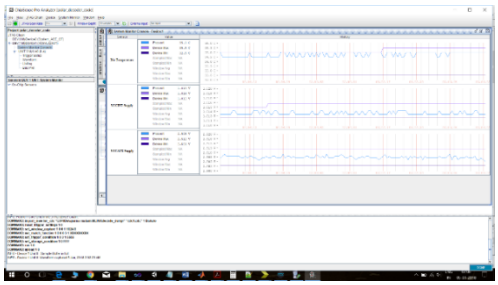

Fig. 11: Chip scope results

VI. CONCLUSION

This paper presents a successful implementation of polar codes with successive cancellation decoding in Verilog HDL. The proposed algorithm is synthesised and tested on the Vertex 7 FPGA board. The results are all tabulated and discussed in the paper. The procedure devised consumes much less power and area compared to the traditional counterparts.

VII. REFERENCES

1. S. B. Korada, E. S¸ as¸oglu, and R. Urbanke, "Polar codes: Characterization of exponent, bounds, and constructions," IEEE Trans. Inform. Theory, vol. 56, pp. 6253–6264, 2010.
2. Tal and A. Vardy, "How to construct polar codes," submitted to IEEE Trans. Inform. Theory, available online as arXiv:1105.6164v2, 2011.
3. E. Arıkan, "Channel polarization: A method for constructing capacityachieving codes for symmetric binary-input memoryless channels," IEEE Trans. Inform. Theory, vol. 55, pp. 3051–3073, 2009.
4. S. Liu, Y. Hong, and E. Viterbo, "Polar codes for block fading channels," in IEEE Wireless Commun. and Netw. Conf. Workshops, March 2017, pp. 1–6.
5. G. B ¨ocherer, T. Prinz, P. Yuan, and F. Steiner, "Efficient polar code construction for higher-order modulation," in 2017 IEEE Wireless Commun. and Netw. Conf. Workshops (WCNCW), March 2017, pp. 1–6.
6. M. Mondelli, S. H. Hassani, and R. Urbanke, "Construction of polar codes with sublinear complexity," in IEEE International Symposium on Information Theory (ISIT), Aachen, Germany, June 2017.
7. C. Condo, S. A. Hashemi, and W. J. Gross, "Efficient bit-channel reliability computation for multi-mode polar code encoders and decoders," in IEEE International Workshop on Signal Processing Systems (SiPS), Lorient, France, October 2017.
8. C. Condo, F. Ercan, and W. J. Gross, "Improved successive cancellation flip decoding of polar codes based on error distribution," in IEEE

January - February 2020
ISSN: 0193 - 4120 Page No. 2718 - 2723

Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, April 2018.

9. S. A. Hashemi, A. Balatsoukas-Stimming, P. Giard, C. Thibeault, and W. J. Gross, "Partitioned successive-cancellation list decoding of polar codes," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, March 2016.
10. V. Bioglio, F. Gabry, I. Land, and J.-C. Belfiore, "Minimum-distance based construction of multi-kernel polar codes," in IEEE Global Communications Conference (GLOBECOM), Singapore, December 2017.
11. S. A. Hashemi, C. Condo, F. Ercan, and W. J. Gross, "Memory-efficient polar decoders," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 70, no. 4, pp. 604–615, April 2017.