Hardware Realization of Polar Decoder Using Reinforcement Learning Algorithm

Albert Raj A (albert@skcet.ac.in)
Sri Krishna College of Engineering and Technology https://orcid.org/0000-0002-4507-3524
Calin A
Sri Krishna College of Engineering and Technology
Latha T
St Xavier's Catholic College of Engineering
Subbulakshmi D
VIT University School of Electrical Engineering
Vijila J
Anna University Chennai University College of Engineering Nagercoil

Research Article

Keywords: Channel Code, Polar Code, Successive Cancelation, FPGA, Reinforcement learning

Posted Date: April 19th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1222830/v1

HARDWARE REALIZATION OF POLAR DECODER USING REINFORCEMENT LEARNING ALGORITHM

ABSTRACT

Error control coding is nearly ubiquitous in our information-based society. In recent days, one amongst the error correcting techniques referred to as the polar codes which attracts the researchers as it represents one of the foremost breakthroughs in 5G standard. It is built by polarization effect of polarization matrix and it is one of the capacity achieving algorithms. It is proved that Successive List decoding (SCL) algorithm improves the efficiency of Polar codes. However, when the codelength increases the latency also increases. It is expected that Reinforcement learning algorithm (RLA) will be able to reduce the latency of the decoder. Therefore, in this article, Markov decision-process algorithm is proposed. RLA uses this Markov decision process when the decoding probabilities are unknown. The same is implemented in the hardware architecture. The implementation result shows that, this method reduces decoding latency to 33% without sacrificing a frame error rate. Experiment result shows that the hardware complexity is also reduced when compared to SCL decoding algorithm. This project is developed using System Generator (Xilinx), with a target device of FPGA-Virtex6.

Keywords

Channel Code, Polar Code, Successive Cancelation, FPGA, Reinforcement learning.

1 INTRODUCTION

Polar code is one among the recent topics among researchers due to its capacity achieving property. After polarization, the channels, become either noisy or noiseless channels. For transmitting the data only noiseless channels are selected by the Polar codes. Even though number of error detection codes are available for encoding and decoding the data, polar codes [2, 3] are considered as one of the promising techniques for 5G applications. The article [4] analyzed the polar codes with 5G applications, and introduced rate matching methods for polar code and claimed that the polar codes solve the coding rate. But even though the rate matching codes solves this issue it changes the reliability of the sub channel used. In general, the channel capacity is achieved by polar codes for large code lengths, e.g., $n \geq 2^{20}$. Hence so as to handle the shorter code length, belief propagation (BP) decoding [5] has been introduced. This article compared their performance with SC decoding for same block length n. However, with the expense of an increase in decoding complexity only this gain can be achieved. Most of the block codes outperform the successive cancellation codes [6] in terms of frame error rate. Even though SCL decoding architecture overcome this issue they suffered increase in complexity. To solve this issue SC-Flip decoding [7] was introduced to improve the error correcting performance. This algorithm reduces the memory requirements but suffers the decoding performance for moderate and higher code length polar codes. A Successive-Cancelation list (SCL) decoder has been introduced in [8], in this method, when the size of list increases, efficiency of SCL decoder became almost same to ML decoder. The active computation of channel architecture was proposed in [9]. Here the author

claimed that polar codes have higher error probability than Reed Muller codes. To address this issue, a Reed Muller code was implemented [10] that was intended to increase the performance of polar decoder. To describe the importance of short length codes, multi core polar code construction is proposed in [11] it is a kind of code construction which is based on the maximization of the minimum distance. Successive Cancellation decoding experiences significance performance gains and its scaling exponent is directly proportional to the list sizes. The scaling exponent is detailed in [12]. However, these results couldn't be applied to all kind of linear codes. To solve this issue, the Monte Carlo for Belief [13] propagation algorithm for polar codes was implemented. Various theoretic characteristics related to polar decoding have been explored in literature works [14-17, 19-23]. These articles focused only on Bit error rate and none of these articles were designed in hardware. In order to minimize the Frame Error Rate (FER) while improving the return the SARSA Learning Enhancement (RL) Algorithm is proposed in [18]. The simulation results show that with a moderate amount of training, the game-based polar code constructs can match the SC decoding, however this algorithm outperform the SCL decoding in FER the latency is more for larger K and N values. The development of the reinforcement learning technique is proposed in [24, 25] where the Q learning in implemented in hardware which allows systems to be designed faster than their software equivalents but their latency and throughput performance are not upto the marks. Having considered everything, this article is structured as follows. Section II explains the Encoding and decoding of polar codes. The successive cancellation Technique is briefed Section III. The Reinforcement learning algorithm and the Markov Decision Process for SC Decoder is discussed in Section IV. The complete algorithm is implemented in FPGA design and its hardware block diagram is analyzed in V. Section VI details the performance analysis of decoder. Finally, Section VII draws the conclusions.

II. ENCODING AND DECODING OF POLAR CODES

Encoding and Decoding of Polar codes was introduced by Arikan [1]. By multiplying information bit vector (Ui) with generator matrix G_N code word C_N is created and this codeword is transmitted. The process of generating with respect to information bit is called encoding.

According to the coding methods, tree of binary codes can be used for decoding. SC Technique is one amongst the coding to unravel the decoding for binary tree. It is the primary decoding architecture for Polar code with low complexity construction. In this algorithm several unwanted low-probability paths in the code tree can be removed. This algorithm uses soft inputs and outputs and that we keep the hard output ignoring the soft inputs. And decision is taken based on hard outputs. Normally every decoder gets the received vectors and release codeword. Whereas in Successive decoding, multiple codewords will be released as output. For this purpose, the Cyclic Redundancy Check (CRC) is employed in number of article as it has excellent error correction properties. However, the latency is more for CRC.

The polar code transmits the message in the AWGN channel and the decoder receives the messages. The receive signal will contain some offsets signal if the channel have some noise. Polar codes are designed to reconstruct the original data after correcting this offset. Several algorithms are developed to improve the performance of the decoding signal. All these algorithms are found to be SC based algorithms. That is why this algorithm is again considered here in this article.

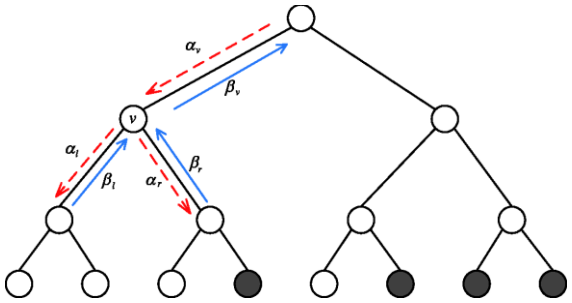

Fig (1): Polar Decoder

III. SUCCESSIVE CANCELLATION TECHNIQUE

Decoding of Successive Cancellation Technique (SC) using Log Likelihood ratio (LLR) of information bit was initiated in [3]. In the polar code arrangement, the channels are formed as binary tree. In which the parent and child nodes are available. In which, every node must perform certain operations. To begin with the received information in the parent node travel through the left child node to reach the lower node (fig.2a). Once the information is arrived at the left child the decision for the left child ($\hat{u}_1$) will be calculated. Then it goes to right child to calculate the right child decision ($\hat{u}_2$) (fig.2b). Finally, the combined decision($\hat{u}_1 + \hat{u}_2, \hat{u}_2$)will be carried out in the parent node(fig.2c).

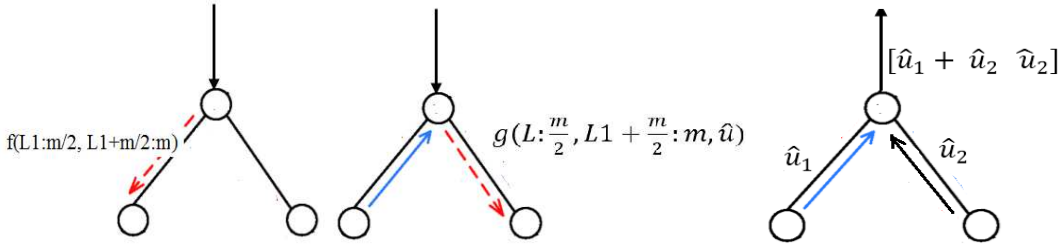

Fig 2a. Left child decision Fig 2b: Right child decision fig 2c: Combined Decision by parent

Fig 2 : Interior node operations

III a. Operations of internal nodes

Internal nodes perform the f and g operation for the incoming beliefs (fig 2). Once the belief 'L' arrives the left child the left child will execute the min-sum execution.

(f (r1, r2) = sign (r1) sign (r2), min r1r2) to find $\hat{u}_1$) ----------(1)

Once the min-sum operation is over, right child will take the control to perform the g value as below

$g(r1, r2, \hat{u}_1) = (12\hat{u}_1) + r1\ r2$ ----------(2)

Now parent gets the decision $\hat{u}_2$ from the right child node. Once it is received the $\hat{u}_1$ and $\hat{u}_2$ from both the child it will then perform the addition operation for the received values $(\hat{u}_{1+}\ \hat{u}_2, \hat{u}_2)$. In this operation when the nonfrozen bit is encountered, both 0 and 1, are considered. In this process, to evaluate the decoding path a path metric (PM) is calculated. In particular the path metric for the L^{th} path with $\tilde{u}^{k,(l)} = \left(\hat{u}_0^{(l)}, \ldots\ldots\ldots\ldots \hat{u}_k^{(l)}\right)$ at the k-th leaf node is

$$PM_k^{(l)} = \begin{cases} PM_{k-1}^{(l)} & \text{if } \hat{u}_k^{(l)} = \frac{1-\text{sgn}(\propto^{(l)})}{2} \\ PM_{k-1}^{(l)} + |(\propto^{(l)}|, & \text{otherwise} \end{cases} \text{ ---------- (3)}$$

where $(\propto^{(l)})$ is the soft output of leaf node. After computing all the possible path of PMs , the L path with the largest path remains the tree, and the others are dropped. The PM with k & n=4 for SC decoding for the tree structure is shown in fig 3. In this structure n=4 have 4 levels and the decoding bits are represented by each level.

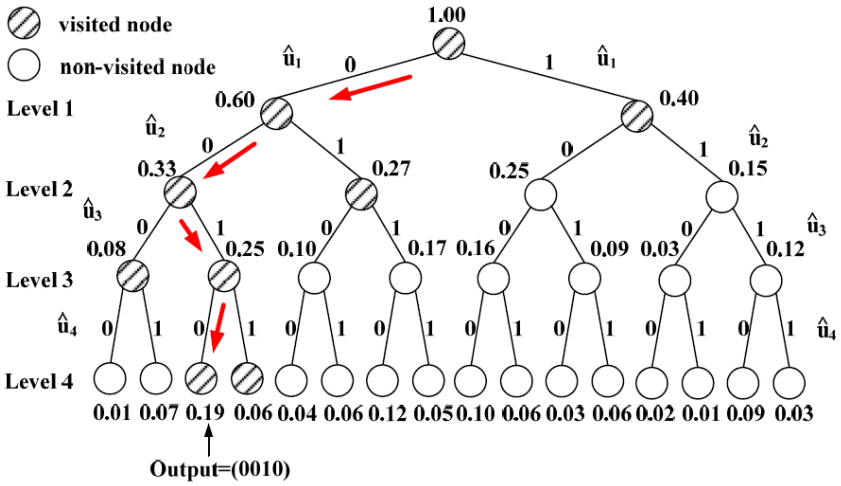

Fig 3: SC decoding path from parent to leaf node (n=4 and k=4)

In fig 3 the PM for various level is calculated. For example, the metric for level 2 Pr is $((\hat{u}_1) \hat{=} 0, (\hat{u}_2) = 0) = 0.33$ for the path $(\hat{u}_1) = 0$ and $(\hat{u}_2) \hat{=} 0$, similarly, it is 0.12 for path $(\hat{u}_1) = 1$, $(\hat{u}_2) = 1$ and $(\hat{u}_3) = 1$. Always the highest metric comes at the lowest level and vice versa. In this fig 3, highest metric (0010) i.e., 0.19, which has the PM= Pr $((\hat{u}_1) = 0, (\hat{u}_2) = 0, (\hat{u}_3) = 1, (\hat{u}_4) = 0) = 0.19$. This is the optimal path also. In order to reduce the latency this bath should be determined for all the decoding path. In order to find the shortest and efficient path in this article we propose a Reinforcement learning technique.

IV.REINFORCEMENT LEARNING ALGORITHM

Reinforcement learning algorithm is neither supervised nor unsupervised learning method rather it is to maximize a numerical reward signal in which the learner will be informed about the action

to be taken rather they need to discover the action to yield the maximum reward. The best part of the learning algorithm is the action may affect both immediate and subsequent rewards. Thus, trial and error search and the final rewards are the main features of this algorithm. This learning is defined by learning problems. This article deals with the reinforcement learning problem. Here, we use the Markov Decision Process to construct the efficient path for solving polar decoder.

IV-a Markov Decision Process for SC Decoder

Markov decision process, satisfies the Markov property. Finite MDP with the finite state and action spaces useful for reinforcement learning process. A sets of actions and states with the dynamics of environment develops the finite MDP. Given each state 's' and action 'a', the probability of every pair of successive states and rewards, 's', 'r' is represented as

$$p(s', r|s, a) = P_r\{ S_{t+1} = s', R_{t+1} = r \,|S_t = s, At = a\} \text{---------- (4)}$$

The dynamics of a finite MDP are specified by these quantities. Once the dynamics are known the details about environment (state action pairs rewards) can also be determined as given below

$$(s, a) = \mathbb{E}[R_{t+1} \mid S_t = s, A_t, = a\ ] = \sum_{r\in R} r \sum_{s\prime\in S} p(s', r|s, a) \text{ ---------- (5)}$$

And the state transition probability is

$$p(s'|s, a) = P_r\{ S_{t+1} = s'|S_t = s, A_t = a \} = \sum_{r\in R} p(s', r|s, a) \text{ ---------- (6)}$$

Finally the reward is

$$r(s, a, s') = \mathbb{E}[R_{t+1}| S_t = s, A_t = a, S_{t+1} = s'] = \frac{\sum_{r\in R} rp(s', r|s,a)}{p(s'|s,a)} \text{ ---------- (7)}$$

In this work, finite MARKOV decision-making algorithm with eligibility trace to update the value function is used by the agent uses the. Based on the reward the agent updates the value function it receives from the environment. This way the agent learns the best policy. This algorithm improves search path latency and completes node completion faster. In this decoding structure, the agent makes its decisions exclusively on the basis of the frozen bit. It can differentiate two levels as high (unfrozen) and low (frozen). Based on the state any of the following decision will be performed.

Update the PM, survival path search, and comparison. When the node is frozen, comparison is not required, because the state will not perform any action. However the agent perform the action as given below

A(high) = {Update the PM, searching the survival path}

A(low) = {Update the PM, searching the survival path, and comparison}.

If the node is non-frozen, then active survival path search can always be completed. A journey of searching that starts with a high-level leaves high with probability α and minimize it to low with probability 1−α. Conversely, when the S is low the survival path searching undertaken leaves it

low with probability β and drop the state with probability 1−β. However, in this case, the searching of node must be rescued, and the comparison will happen to get higher value. Each path collected by the engine counts as a unit reward, whereas a reward of −3 results whenever the search engine has to be renewed. The complete rewards for the transition probabilities are given in Table I.

Table I. The expected rewards for the Transition probabilities

s	**s'**	**a**	**R(s,a,s')**	**P(s'|s,a)**
low	low	Searching survival path	survival path	β
low	high	Searching survival path	-3	1−β
high	low	Searching survival path	r-survival path	1−α
high	high	Searching survival path	r-survival path	α
low	low	Update the PM	Update the PM	1
low	high	Update the PM	Update the PM	0
high	low	Update the PM	Update the PM	0
high	high	Update the PM	Update the PM	1
low	low	comparison	0	0
low	high	comparison	0	1

V. HARDWARE ARCHITECTURE

The FPGA based hardware architecture is presented in figure 4. The complete block diagram is given in this architecture, where the main function for choice of action, random number generation, the updating of the pairs of states and the architecture for the future state operation are given. This architecture is designed to function with Z actions and N states with the combination of Z × N action-state pairs. This architecture is designed for reducing the processing time by executing the algorithm in parallel. Five sub modules are given in this architecture in which random numbers are generated in GA module, EN modules determines state pair updation , RS modules, is for storing the rewards, The Path metric value is calculated in S modules and the SEL module, is used for selection of the state and the repository of the PM value function.

.

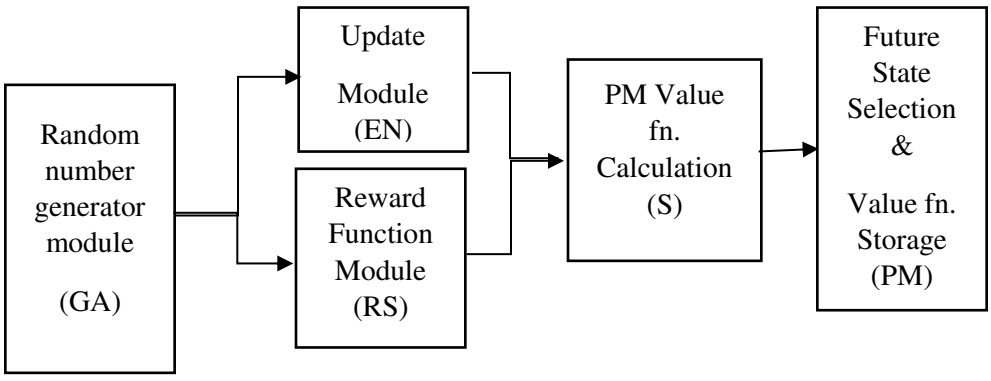

Fig 4: Hardware Architecture for Markov Decision process

VI. PERFORMANCE ANALYSIS

Few research articles proposed the path metric algorithm to reduce the latency of SC decoders [17], they have also implemented the same algorithm for SCL decoder [8]. However, when the code is more the latency of the SCL decoder is increased. The latency of any decoding path depends on the PM value. Which requires f and g of received values (equation 1 & 2). We performed decoding algorithm with and without RL. For n=8, the path metric has taken 14 clock cycle to complete one survival path to calculate U8 (parent node) (latency of 2n-2 cycles) whereas our proposed methods took just 10 clock cycle (latency of 1.5n-2). to complete the same path. The different stages of conventional SCL decoders and the proposed systems are given in the following Table II. and III respectively.

Table II. Decoding path for n=8 polar code conventional SCL (8)

Clk	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th
Stage 1	f(r)							g(r)						
Stage 2		f(r)			g(r)				f(r)			g(r)		
Stage 3			f(r)	g(r)		f(r)	g(r)			f(r)	g(r)		f(r)	g(r)
output			$\widehat{ui}$	$\hat{u}_2$		$\hat{u}_3$	$\hat{u}_4$			$\hat{u}_5$	$\hat{u}_6$		$\hat{u}_7$	$\hat{u}_8$

Table III. Decoding path for n=8 polar code of proposed algorithm

Clk	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Stage 1	f(r)					g(r)				
Stage 2		f(r)		g(r)			f(r)		g(r)	
Stage 3			f(r)		f(r)			f(r)		g(r)
output			$\hat{u}_1$& $\hat{u}_2$		$\hat{u}_3$& $\hat{u}_4$			$\hat{u}_5$& $\hat{u}_6$		$\hat{u}_7$& $\hat{u}_8$

So, when compared to the conventional SCL algorithm [8], where the overall latency is of 2n-2 our proposed algorithm reduced 33% less than SCL. As we achieve the 33% improvement in latency it is evident that the performance of tree is excellent for larger value of n. Since the searching of path completes in shortest time the power consumption can also be reduced by reducing clock cycles. This is also given here in Table IV. The simulation parameter used for transmitting 512-bit information is shown in fig 5. The simulated output of PM value (equation 3) for the received values are given in this fig 5. This program was implemented using Verilog language. The same in synthesized to get the throughput. Throughput is used for measuring the performance of the system. This parameter cannot be obtained from the tool directly rather it has to be calculated manually. It is the ratio of total number of transmitted bits per clock cycle to the latency of total number of inputs. The performance of our method (throughput) is 9.01(table IV) which is higher than SC and SCL algorithm.

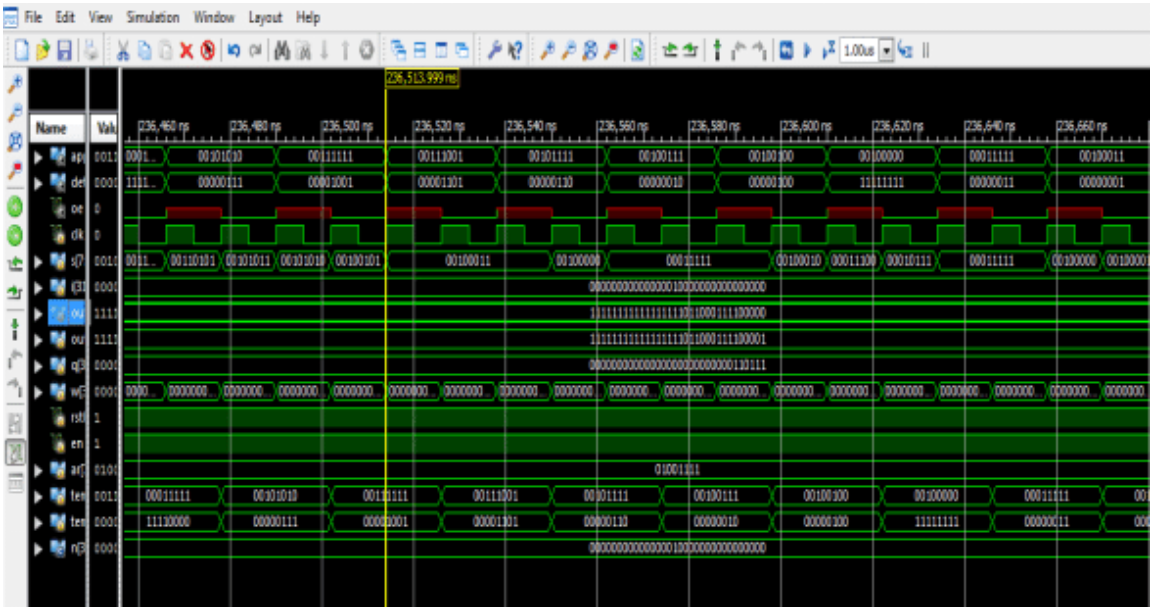

Fig 5. Simulated output for path metric calculation

Table IV The FPGA based hardware performance for n=1024 k=512.

Hardware	SC(17)	SCL(8)	Proposed Method
Area	1.476	1.06	0.7
Clock frequency(MHz)	500	600	450
Latency (Clock cycle)	1022	2046	988
Power consumption(mW)	395	321	298
Coded Throughput (GB/s)	8.9	8.2	9.01

Finally, the architecture was simulated using MATLAB R2021 to see the performance of decoding architecture for n = 1024, K = 512. At the transmitter, 512 information is encoded into code-words of length 1024. At the receiver, while recovering the corresponding information bit we observed that the FER performance is almost same for both SCL(dotted line) [9] and the proposed algorithm (straight line)as given in fig (6). So obviously without compromising the FER performance our architecture improves its latency.

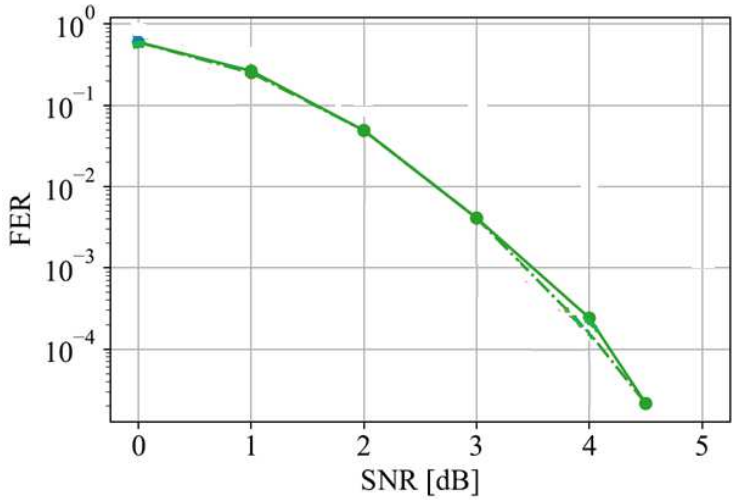

Fig (6): FER performance of the SCL decoder (dotted line) and the proposed algorithm (straight line) for N = 1024, K = 512.

VII.CONCLUSION

This article analyzed SC decoding with Reinforcement learning based path searching algorithm is analyzed. This work illustrated the hardware architecture for traversing the path metric on FPGA. We have built the Markov Decision Making Algorithm to support the latest wireless technology (5G). This algorithm was implemented in Xilinx and the hardware complexity, latency and power consumption were analyzed using synthesis result. Using Xilinx software, the hardware architecture was validated. The performance results from MATLAB shows that the received values have better FER. By analyzing the synthesis result from FPGA, we observed that the architecture has taken very less processing time low power consumption. These functionalities are used in application where the low power consumption is required for any devices.

REFERENCES

[1] E. Arıkan, "Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels," IEEE Transactions on Information Theory, vol. 55, no. 7, pp. 3051– 3073, July 2009.

[2] A.Albert raj, D.Judson,A. Beno 2021 "Efficient Construction of Successive Cancellation Decoding of Polar Codes using Logistic regression Algorithm" Journal of Physics doi:10.1088/1742-6596/1998/1/012010.

[3] A.Albert Raj, M.poorna Sankari, S.Sri Abinaya '2021 " Design of Successive Cancellation list Decoding of polar codes' ' International Conference on Communication and Electronics System(ICCES 2021) .https:// doi.org/ 10.1109/ ICCES51350.2021.9489221.

[4] Marwan DHUHEIR, Sıtkı ÖZTÜRK "Polar Codes Analysis of 5G Systems" 2018 6th International Conference on Control Engineering & Information Technology (CEIT), 25-27 October 2018, Istanbul, Turkey

[5] Weihong Xu, Xiaosi Tan, Yair Be'ery, Yeong-Luh Ueng,2020 "Deep Learning-Aided Belief Propagation Decoder for Polar Codes," IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS, DOI: 10.1109/JETCAS.2020.2995962.

[6] O. Afisiadis, A. Balatsoukas-Stimming and A. Burg, "A low-complexity improved successive cancellation decoder for polar codes," 2014 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 2014, pp. 2116- 2120.

[7] L. Chandesris, V. Savin and D. Declercq, "An Improved SC Flip Decoder for Polar Codes," 2016 IEEE Global Communications Conference (GLOBECOM), Washington,DC, 2016, pp. 1-6.

[8] B. Yuan and K.K. Parhi, "Successive Cancellation List Polar Decoder using Log-likelihood Ratios," in Proc. of 2014 Asilomar Conf. on Signal, Systems and Computers, 2014.

[9] 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, Oct. 2017.

[10] M. Mondelli, S.H. Hassani, and R.L. Urbanke, "From polar to Reed- Muller codes: A technique to improve the finite-length performance," IEEE Transactions on Communications, vol. 62, no. 9, pp. 3084–3091, Sept. 2014.

[11] 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, Dec. 2017.

[12] M. Mondelli, S. H. Hassani, and R. L. Urbanke, "Scaling exponent of list decoders with applications to polar codes," IEEE Transactions on Information Theory, vol. 61, no. 9, pp. 4838–4851, Sept. 2015.

[13] J. Guo, M. Qin, A. Guill`en i F´abregas, and P. H. Siegel, "Enhanced belief propagation decoding of polar codes through concatenation," in IEEE International Symposium on Information Theory (ISIT), 2014, Honolulu, HI, USA, June 2014.

[14] U. U. Fayyaz and J. R. Barry, "Low-complexity soft-output decoding of polar codes," IEEE Journal on Selected Areas in Communications, vol. 32, no. 5, pp. 958–966, May 2014.

[15]Dr.A.Albert Raj, J.Vijila, 2020 "Design of Indoor Air Quality Monitoring System to Ensure a Healthy Universe" International Conference on Smart Electronics and Communication (ICOSEC) pp 1123-1127.

[16]P. Yuan, T. Prinz, and G. Bocherer, "Polar code construction for list decoding," in IEEE International ITG Conference on Systems, Communications and Coding (SCC), Rostock, Germany, Feb. 2019.

[17] Giard, Pascal; Sarkis, Gabi; Balatsoukas-Stimming, Alexios; Fan, YouZhe; Tsui, Chi-ying; Burg, Andreas; Thibeault, Claude; Gross, Warren J. 2016 IEEE International Symposium on Circuits and Systems (ISCAS) - Hardware decoders for polar codes: An overview

[18]B. Li, H. Shen, and D. Tse, "An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check," IEEE Comm. Letters, vol. 16, pp. 2044–2047, Dec. 2012.

[19] B. Li, H. Shen and D. Tse, "Parallel decoders of polar codes," arXiv: 1309.1026v1, Sep. 2013.

[20] E. Abbe, and E. Telatar, "MAC polar codes and matroids," Information Theory and Applications Workshop (ITA), pp. 1-8, 2010.

[21] N. Goela, S. B. Korada, and M. Gastpar, "On LP decoding of polar codes," in Proc. IEEE Information Theory Workshop (ITW), pp. 1-5, Aug. 2010.

[22] E. Arikan, "Systematic polar coding," IEEE Commun. Lett., vol. 15, no. 8, pp. 860-862, 2011.

[23] A. Balatsoukas-Stimming, M. Bastani Parizi and A. Burg, "LLR-based successive cancellation list decoding of polar codes," in Proc. of 2014 IEEE Intl. Conf. on Acoustics, Speech and Signal Procesing (ICASSP), pp. 3903–3907, May 2014.

[24] R. S. Sutton, Reinforcement Learning: A Special Issue of Machine Learning on Reinforcement Learning, 8th ed. Kluwer Academic Publishers, 1992.

[25] N. C. de Almeida, M. A. C. Fernandes, and A. D. D. Neto, “Beamforming and power control in sensor arrays using reinforcement learning,” Sensors, vol. 15, 2015.