EMITTER International Journal of Engineering Technology
Vol. 10, No. 1, June 2022, pp. 170~182
DOI : 10.24003/emitter.v10i1.650

Low Power, Area Efficient Architecture for Successive Cancellation Decoder

Sujanth Roy J[1], G Lakshminarayanan[2]

[1,2] Department of Electronics and Communication Engineering,
National Institute of Technology Tiruchirappalli, Trichy – 620015, India
Correspondence Author: 408115052@nitt.edu

Received February 10, 2022; Revised March 14, 2022; Accepted April 18, 2022

Abstract

Polar codes have recently emerged as an error-correcting code and have become popular owing to their capacity-achieving nature. Polar code based communication system primarily consists of two parts, including Polar Encoder and Decoder. Successive Cancellation Decoder is one of the methods used in the decoding process. The Successive Cancellation Decoder is a recursive structure built with the building block called Processing Element. This article proposes a low power, area-efficient architecture for the Successive Cancellation Decoder for polar codes. Successive Cancellation Decoder with code length 1024 and code rate 0.5 was designed in Verilog HDL and implemented using 45-nm CMOS technology. The proposed work focuses on developing an area-efficient Successive Cancellation Decoder architecture by presenting a new Processing Element architecture. The proposed architecture has produced about 35% lesser area with a 12% reduced gate count. Moreover, power is also reduced by 50%. A substantial reduction in the latency and improvement in the Technology Scaled Normalized Throughput value was observed.

Keywords: Polar Codes, Channel Coding, Successive Cancellation Decoder, FPGA, ASIC.

1. INTRODUCTION

Proposed by E Arikan, the polar codes stand as one of the finest capacity-achieving codes having low encoding and decoding complexity of the order O(N log N), where N is the code length [1]. Polar codes are established on a recursive concatenation of the short core functions that convert physical channels to virtual channels. As the number of virtual channels increases, they tend to have either low reliability or high reliability i.e., they polarize. This feature enables the message bits to be allotted to the most reliable channel.

Several architectures have been proposed for the Successive Cancellation decoder (SCD). In SCD, there are three major blocks, including the Processing Element (PE), Partial Sum Generation Unit (PSG) and Memory Unit (MU). In this work, an area-efficient architecture for the PE and hence the decoder is proposed. An architectural modification has been carried out on the PE, which reduces the overall area to a great extent. SCD is the widely used algorithm that can assure exceptional error-correcting performance in polar code decoding. The hardware architectures of the conventional SCD occupies a large silicon area. Improvements in polar decoding techniques and systems are eagerly awaited by the 5G telecommunication industry, where even more complex systems are being developed that require less power, area and obtain high throughput and speed. The challenge in the 5G wireless system is to produce a suitable channel coding scheme to ensemble increasing spectral efficiency. Research in SCD has been going on recently, and therefore, it is worthwhile to come out with a new architecture for the same.

The remaining paper is arranged as follows: In section 2, the related works to this model are discussed. Section 3 highlights the originality of the work and section 4 briefs the proposed system design consisting of the processing element and the proposed decoder. Section 5 deliberates the implementation results, and the paper is concluded in section 6.

2. RELATED WORKS

B. Yuan and K.K. Parhi proposed a parallel decoder, which decomposes the polar code into a series of constituent codes and thus reducing the decoder latency by 25% [2]. [5] - [7] have proposed an upfront implementation of the SCD. SCD with merged PEs and pre-computation techniques was detailed in [8], which enabled the 2-bit decoding scheme and was then expanded widely. H. Vangala et. al. proposed a Multi-Folded SCD [10] that incorporates Folding technique which was first introduced in Maximum Likelihood (ML) decoder algorithm for polar codes with code lengths up to 256. The complexity increased by 22K in best implementation, where K represents the number of folding [10]. Multiple folding are employed to reduce the latency in the decoder [11] but increases the computation complexity. Feedback and feed-forward implementations of the encoder reduced the complexity and enhanced the speed of operation [14]. A non-recursive method to create a decoding schedule without affecting the performance has been portrayed in [15]. In [16], path splitting selection (PSS) strategy aided decoder was proposed to lessen the decoding complexity with negotiable performance loss. Based on PSS, two schemes were suggested to locate flawed information bits more precisely. Apart from the PE and PSG, the memory unit is another sub-block that can be optimized according to the requirement [17]. The logarithmic implementation has significantly reduced the hardware complexity [18]. The PE operates with log-likelihood ratio (LLR) instead of likelihood ratio (LR). Hence, polar codes are preferred over the Low Density Parity-check codes (LDPC) on wireless communication channels due to their better performance [19]. There are rate-less polar code implementations [20], [21] where the number of frozen bits are varied based on the length of the information bits. This feature provides the extendibility of the decoder for various code rates [24]. Polar codes are extended to non-identically distributed channels and they were found to be capacity achieving but the latency and complexity were sacrificed [25].

3. ORIGINALITY

In this paper, a low complexity SCD with area efficient merged processing element (MPE) has been proposed and its performance was compared to other existing models. In SCD, processing element is the fundamental building element which processes the log-likelihood ratio (LLR) values to decode the message. The processing element is initially optimized in the proposed model, and the same was used in the proposed decoder. The model highlights that the signed to 2’s complement conversion is removed, and only one adder block is incorporated for the implementation, thus reducing a considerable amount of area. Thus, an area-efficient processing element is presented in this paper, which can generate two bits of output simultaneously, thereby reducing the latency and also the overall hardware consumes less silicon area.

4. SYSTEM DESIGN

Polar codes fall under the banner of linear block codes and are one of the available forward error correction (FEC) codes. They have low computational complexity and are capacity achieving codes. Polar codes with rate R=I/N have a code length of N=2^n and I (0≤I≤N), denotes the number of information bits whereas N-I denotes the number frozen bits that are added to the information bits.

4.1 Polar Code - Encoding

The encoding approach of polar codes follow the equation (1) given below [3]

$$x_1^N = u_1^N G_N = u_1^N B_N F^{\otimes n} = u_1^N F^{\otimes n} B_N \tag{1}$$

Where $G_N = B_N F^{\otimes n}$ denotes the generator matrix, F = [1 0;1 1], $(.)^{\otimes n}$ represents the nth Kronecker power, B_N denotes the bit reversal vector, the input vector represented as $u_1^N = u_1, u_2, \ldots u_N$ having I information bits and (N-I) frozen bits. Also, $x_1^N = x_1, x_2, \ldots x_N$ represents the encoded value [12]. The position of frozen bits are identified by the method explained in [4].

4.2 Polar Code - Decoding

The decoding approach is carried out with the widely used Successive Cancellation decoding algorithm. In the decoding scheme, the information bits $u_1^N = u_1, u_2, \ldots u_N$ are retrieved sequentially from the received vector

$y_1^N = y_1, y_2, \dots y_N$. The output bits at stage t can be decoded by processing the LLR function as given below in (2) and (3)

$$u_i = O(LL(i,t)) \tag{2}$$

where,

$$u_i = \begin{cases} 1, \text{if } \mathrm{LL(i,\ t)} < 0 \text{ and when i is free} \\ 0, \text{if } \mathrm{LL(i,\ t)} \geq 0 \text{ and when i is frozen} \end{cases} \tag{3}$$

Both encoder and decoder facilitate neat processing structures and proper component sharing owing to their recursive build. Every stage is composed of the kernels f and g appropriately scheduled to decode the data. The f and g kernels function based on equations (4) and (5) given below.

$$f = sign(c)sign(d)\min(|\ c\ |, |\ d\ |) \tag{4}$$

$$g = c(-1)^{\hat{u}_{sum}} + d \tag{5}$$

where $\hat{u}_{sum}$ decides between addition and subtraction in the kernel g, c and d represent the LLR inputs. $\hat{u}_{sum}$ represents the partial sum of the subset of previously decoded bits. The conventional SCD takes 2(N-1) clock cycles to decode a code of length N [1]. Po et al. proposed a polar code decoder with variable R and N [22].

4.3 Proposed Processing Element

The LLRs are processed by the PEs to produce a signed value as output and passed on to further stages. The PE consists of the f and g kernels that perform their corresponding functions appropriately to build the decoding tree. The f and g kernels operate according to the equations (4), (5) mentioned above.

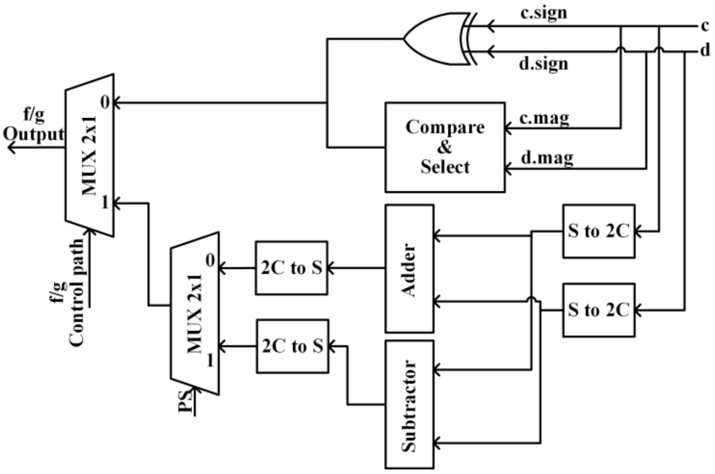

Figure 1. Standard Merged Processing Element

The conventional merged PE has a comparator, signed to 2’s complement block (s to2C), 2’s complement to signed (2C to S), separate adder and subtractor. The architecture of the standard merged PE is shown in Figure 1. The proposed PE is designed to function based on the logic discussed below. Let c, d be two input LLRs and δ=mag(d)-mag(c), ε= mag(d) +mag(c), λ=sign(c)⊕sign(d). The truth table of the f kernel and g kernel of the decoder is given below in Table 1 & 2. Here, mag represents the value, and sign represents the sign bit of the LLR.

Table 1. Truth table of f-Kernel functionality

Input				**Output**	
msb(mag(c))	**msb(mag(d))**	**msb(δ)**	**msb(ε)**	**sign(f)**	**mag(f)**
0	0	0	x	λ	mag(c)
0	1	x	1	λ’	mag(c)
1	0	x	0	λ	mag(c)
1	1	1	x	λ’	mag(c)
0	0	1	x	λ	mag(d)
0	1	x	0	λ	mag(d)
1	0	x	1	λ’	mag(d)
1	1	0	x	λ’	mag(d)

x = don’t care

Table 2. Truth table of g-Kernel functionality

Input		**Output**			
sign(c)	**sign(d)**	**sign(g^0)**	**mag(g^0)**	**sign(g^1)**	**mag(g^1)**
0	0	0	ε	0	δ
0	1	1	δ	1	ε
1	0	0	δ	0	ε
1	1	1	ε	1	δ

The operations are carried out in a fashion where the sign bit and the magnitude bit are processed separately. On careful analysis of the existing processing element, a scope to reduce the on chip area of the same was found. The architecture of the proposed PE is shown in Figure 2. The proposed PE has a lesser chip area, gate count and power compared to the existing model. Also the hardware utilization has also been optimized considerably.

According to the truth table (Table 1 and 2), the magnitude and the sign of input LLR are processed separately. The proposed PE does not include the subtractor block, and the necessary operations have been performed suitably with the adder without causing any change in the desired output. Though this might be a minor modification in the PE, when the decoder is constructed for higher block lengths, using the proposed processing element, there is a considerable change in the parameters, thanks to the recursive construction.

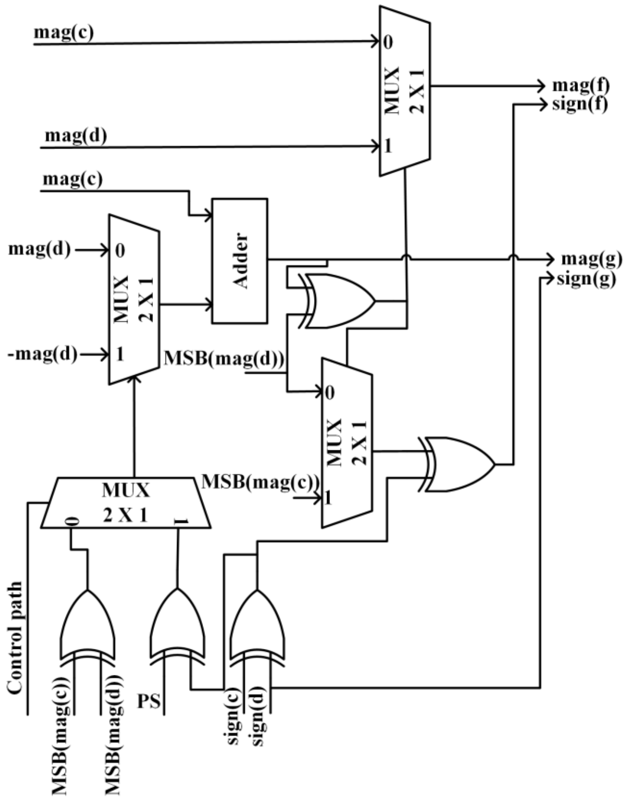

Figure 2. Proposed Processing Element

The functioning of the proposed PE based on the truth table (Table 1 and 2) is shown in the Table 3 below.

Table 3. Functioning Table of f & g Kernels

f - kernel				**g - kernel**		
Input	sign(c)		1	**Input**	sign(c)	1
	sign(d)		-1		sign(d)	-1
	λ		-1	**Output**	sign(g^0)	1
	mag(c)		1		mag(g^0)	δ
	mag(d)		4		sign(g^1)	1
	δ		3		mag(g^1)	ε
	ε		6			
Output	f	sign(f)	-1			
		mag(f)	1			

let, c = 1 and d = -4

The f and g kernels function based on the equations (4) and (5). The control path chooses the function between f and g kernels in the proposed architecture. Here, 2's complement approach was used because only addition/subtraction is the main computational sub-block involved in the architecture. From equation (4), the minimum magnitude of the input LLRs is required. The minimum magnitude is determined by borrow/carry when both the inputs are positive or negative. When the inputs have complementary sign bits, then it was found that the (msb(d) xor msb(adder out)), determined the minimum magnitude of the input LLRs c and d. This was found to be true throughout the research cases. From equation (5), the g-kernel calculates the sum/difference based on the previous partial sum (PS) bit. Hence the calculations are carried out in 2's complement form itself. Since only one adder is used, the decision between addition/subtraction must be taken before the adder block. When PS=0 the sum output is calculated and when PS=1 the difference output is calculated. Since the sign bit of the g output is required for the next stage, it has to be processed separately. On analysis, it was found that the sign(g) followed the same pattern as the sign(d) for all test cases. This is shown in Table 2 of the manuscript.

Here, c=1 and d=-4 are considered for example and the functioning is elaborated. From Table 3, the output of f kernel is -1. Based on equation (2) and (3), the decoded bit is 1. Therefore, the g1 output is the actual output calculated in the g kernel as the $\hat{u}_{sum}$ value is 1. In the proposed architecture, the δ and ε values are calculated using the same adder. All the values are fed to the PE in 2's complement format. The value of mag(d) and –mag(d) are both magnitude values with negation only in the msb. This facilitated the entire process to be carried out using one adder. The PS in Figure 2 denotes the partial sum input for the g kernel and the control path is used to switch between f and g kernels appropriately. Thus the proposed architecture was able to yield the proper functioning of the f and g kernels correspondingly.

The critical path of the proposed decoder is $t_{Adder} + 3 * t_{2:2MUX} + 3 * t_{2-InputXor}$ where, $t_{Adder}, t_{2:2MUX}, t_{2-InputXor}$ are the logic delays of the adder, mux and xor blocks, respectively.

4.4 SCD using the Proposed Processing Element

SCD is an electronic circuitry developed with the purpose to receive the channel information and decode the message embedded within it. The proposed processing element is used to construct the decoder. The decoding tree is similar to the conventional tree using the MPE. The basic building block is the processing element and the recursive structure of the same builds the decoder. The conventional decoder mentioned in [1] takes about 2N-2 clock cycles to generate the decoded bits where N signifies the code length. The decoding tree for polar code with N=8 is constructed using seven PEs. In the final stage the bits are decoded based on equation (2) and (3) from the signed values generated. The decoding tree is shown in the Figure 3 below. From the figure, it is clear that for N=8, the SCD has three stages. Stage 1 takes the LLR values received from the channel as inputs and generates the intermediate signed values for the next stage. In stage 1, four PEs are used. In stage 2, two PEs are used to generate the intermediate values for stage 3 which comprises of one PE. At the end of stage 3, the bit decision is made using h-block. The proposed SCD tree is thus constructed using total seven PEs. This decoding tree functions continuously till all 8 bits are decoded. During this decoding process, the partial sum block also will be working to provide the partial sum, $\hat{u}_{sum}$ at the appropriate timing [1]. The same procedure is extended for higher block lengths by appropriately extending the decoding tree. For a block length of N=2n, there will be ‘n’ stages in the decoding tree. Pipelining has been introduced between the stages helps to reduce the number of decoding cycles.

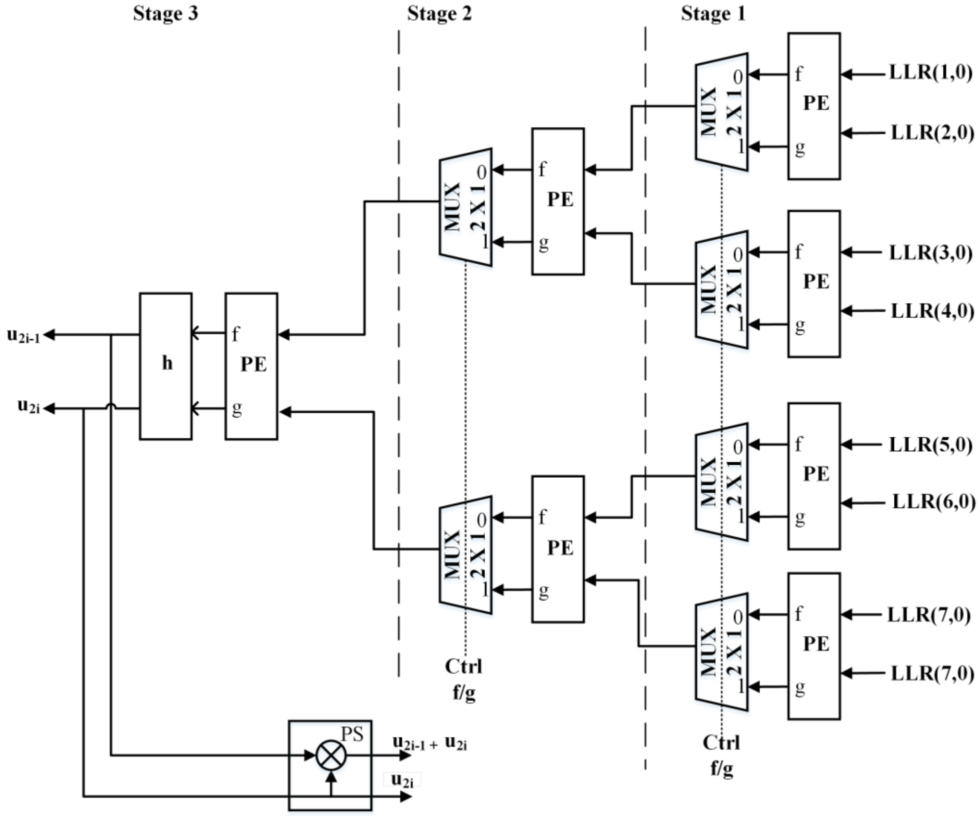

Figure 3. Proposed decoding tree

This facilitated faster decoding by avoiding idle states in the decoding tree. The proposed PE is optimized for area and hence while the decoder is constructed using the proposed PE, there is a substantial reduction in the on-chip area for higher block lengths.

5 RESULT AND DISCUSSIONS

The proposed decoder was modelled in Verilog HDL, implemented in Xilinx xc7vx980t FPGA and synthesized in Cadence 45nm CMOS technology. The area, power, and gate count were compared with other existing models. The overall chip area has been reduced by 35% approximately with respect to the [8] and [9]. The decoder tree was constructed using the proposed PE and was implemented and tested for code length up to 1024. Table 4 below shows the comparison results of the overall chip area.

Table 4. Area Comparison using Cadence 45nm CMOS Technology

Block length	Area (μm²)			% change
	[8]	[9]	Proposed SC decoder	
8	3621	2509	1040	58.55
16	7824	5466	2296	58
32	14917	10548	5356	49.22
64	30141	21278	11532	45.80
128	60899	43028	24896	42.14
256	121925	86053	50711	41.07
512	243891	172030	105627	38.60
1024	444944	343684	219512	36.13

A considerable reduction in power consumption was also observed in the proposed SCD. Unlike convolutional codes [26], the polar codes have low power consumption and less complexity. Table 5 shows that there is a reduction of more than 50% in power consumption which would be an appreciable achievement from the proposed model.

Table 5. Power Comparison using Cadence 45nm CMOS Technology

Block length	Power (μW)			% change
	[8]	[9]	Proposed SC decoder	
8	217.89	149.84	38.1	74.57
16	485.48	346.67	147.5	57.45
32	842.23	610.71	306.01	49.89
64	1725.03	1267.67	613.72	51.58
128	3567.91	2602.17	1245.72	52.12
256	7049.74	5199.48	2516.55	51.60
512	14002.7	10243.2	5051.95	50.68
1024	27220.4	20274.7	10110.90	50.13

Moreover, about a 30% reduction in the overall gate count was also observed. Table 6 compares the gate count of the proposed model with other existing models.

Table 6. Gate Count Comparison (NAND2) using Cadence 45nm CMOS Technology

Block length	Gate Count			% change
	[8]	[9]	Proposed SC decoder	
8	1584	978	734	24.95
16	3316	2074	1688	18.61
32	6310	4004	3234	19.23
64	12672	8039	6739	16.17
128	25529	16210	13210	18.51
256	51033	32379	27476	15.14
512	102012	64659	55959	13.45
1024	198625	129141	113541	12.07

Table 4, 5, and 6 clearly show that the proposed architecture has significantly reduced the overall chip area, on-chip power, and gate count, respectively. The proposed scheme produced the same functionality as the SCD with 35% lesser area, and 12% lesser gate count. Moreover, a 50% reduction in the power was also observed. The precomputation approach can be merged with the proposed model to reduce the circuit latency and thus enable high-speed operations [13]. The proposed decoder's performance was compared with other existing decoders and results are shown in Table 7 below.

Table 7. Performance Comparison of proposed SC decoder architecture with other state-of-art SC decoder

Parameters	[7]	[2]	[6]	[8]	[9]	Proposed
CMOS Technology	180 nm	45 nm	65 nm	-	45 nm	45 nm
Code Length	1024	1024	1024	1024	1024	1024
Code Rate	0.5	0.5	0.5	0.5	0.5	0.5
Clock Frequency (MHz)	150	750	500	-	750	400
Throughput (Mbps)	49	1000	246	400	750	186
Gate (NAND)	183637	338499	214370	198625	129141	113541
Efficiency (Mbps/kGate)	0.267	2.954	1.148	2.01	5.807	1.841
Latency	1568	767	2080	1023	767	1647
TNST (scaled to 45nm)	1.07	2.95	1.65	-	5.81	1.84

The proposed scheme has about 35% reduction in the overall area, but there is a small increase in the latency. But the better hardware efficiency (Throughput/kGate) [23] provides an edge over the other models. Owing to the considerable reduction in area and power, this can be negligible and can find its application where area reduction is a major concern. The Technology Scaled Normalized Throughput (TNST) is defined by [12],

$$\mathrm{TNST} = \left(\frac{\mathrm{Throughput}}{\mathrm{Gate\,Count}}\right) * \left(\frac{\mathrm{Technology}}{\mathrm{Target\,Technology}}\right) \tag{6}$$

As area and delay are inversely proportional, there is a need to compare the area-delay product to fairly validate the performance of the proposed decoder. Table 8 below compares the proposed model with [8] and [9] and provide the comparison result.

Table 8. Area-Delay Product Comparison

Parameters	[8]	[9]	Proposed
Critical path delay (ns)	1.704	5.18	1.561
Area (μm^2)	444944	343684	219512
Area Delay Product (nm^{2}s)	0.76	1.77	0.34
Power Delay Product (pJ)	46.383	105.062	15.783

Table 8 clearly shows that the proposed model competes with the architectures proposed in [8] and [9]. Thus the proposed architecture has reduced the chip area and power without compromising on the performance of the decoder.

6. CONCLUSION

In this paper, an area-efficient, low power decoder architecture was proposed. A new processing element was proposed, which was then used to construct the decoder. The proposed model produced a significant reduction in the on-chip area, power, and gate count. The proposed architecture finds its application in modern communication systems as it facilitates faster decoding while consuming less power and occupying minimal on-chip area. In future works, look-ahead techniques and precomputation can be combined with the proposed model to speed up the decoder further.

Acknowledgements

The authors thank Visvesvaraya Ph.D. Scheme (VISPHDMEITY-1713), Ministry of Electronics and Information Technology (MeiTY), Government of India for funding this research, and SMDP-C2SD and DST-FIST (Grant number: DST/ETI-324/2012), Government of India for providing the lab facilities.

REFERENCES

[1] E. Arikan, **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, Jul. 2009.

[2] B. Yuan and K. K. Parhi, **Low-Latency Successive-Cancellation Polar Decoder Architectures Using 2-Bit Decoding**, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 4, pp. 1241–1254, Apr. 2014.

[3] C. Zhang and K. K. Parhi, **Latency Analysis and Architecture Design of Simplified SC Polar Decoders**, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 2, pp. 115–119, Feb. 2014.

[4] C. Kim, H. Yun, S. Ajaz, and H. Lee, **High-Throughput Low-Complexity Successive-Cancellation Polar Decoder Architecture using One's Complement Scheme**, *Journal of Semiconductor Technology and Science*, vol. 15, no. 3, pp. 427–435, Jun. 2015.

[5] C. Leroux, A. Raymond, G. Sarkis, I. Tal, A. Vardy, and W. Gross, **Hardware Implementation of Successive Cancellation Decoders for Polar Codes**, *Journal of Signal Processing Systems*, vol. 69, Nov. 2011.

[6] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, **A Semi-Parallel Successive-Cancellation Decoder for Polar Codes**, *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 289–299, Jan. 2013.

[7] A. Mishra *et al.*, **A successive cancellation decoder ASIC for a 1024-bit polar code in 180nm CMOS**, in *2012 IEEE Asian Solid State Circuits Conference (A-SSCC)*, Nov. 2012, pp. 205–208.

[8] C. Zhang and K. K. Parhi, **Low-Latency Sequential and Overlapped Architectures for Successive Cancellation Polar Decoder**, *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2429–2441, May 2013.

[9] G. Sathees Babu, L. R. Madala, L. Gopalakrishnan, and M. Sellathurai, **Low-complex processing element architecture for successive cancellation decoder**, *Integration*, vol. 66, pp. 80–87, May 2019.

[10] H. Vangala, E. Viterbo, and Y. Hong, **A new multiple folded successive cancellation decoder for polar codes**, in *2014 IEEE Information Theory Workshop (ITW 2014)*, Nov. 2014, pp. 381–385.

[11] S. Kahraman, E. Viterbo, and M. E. Çelebi, **Multiple Folding for Successive Cancelation Decoding of Polar Codes**, *IEEE Wireless Communications Letters*, vol. 3, no. 5, pp. 545–548, Oct. 2014.

[12] M. Bohr, **A 30 Year Retrospective on Dennard's MOSFET Scaling Paper**, *IEEE Solid-State Circuits Society Newsletter*, vol. 12, no. 1, pp. 11–13, 2007.

[13] S. J. Roy, G. Lakshminarayanan, and S.-B. Ko, **High Speed Architecture for Successive Cancellation Decoder with Split-g Node Block**, *IEEE Embedded Systems Letters*, pp. 1–1, 2020, doi: 10.1109/LES.2020.3021144.

[14] C. Zhang, J. Yang, X. You, and S. Xu, **Pipelined implementations of polar encoder and feed-back part for SC polar decoder**, in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 3032–3035.

[15] D. Le, X. Wu, and X. Niu, **Decoding schedule generating method for successive-cancellation decoder of polar codes**, *IET Communications*, vol. 10, no. 5, pp. 462–467, 2016.

[16] C. Gao, R. Liu, B. Dai, and X. Han, **Path Splitting Selecting Strategy-Aided Successive Cancellation List Algorithm for Polar Codes**, *IEEE Communications Letters*, vol. 23, no. 3, pp. 422–425, Mar. 2019.

[17] 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. 7, no. 4, pp. 604–615, Dec. 2017.

[18] C. Leroux, I. Tal, A. Vardy, and W. J. Gross, **Hardware architectures for successive cancellation decoding of polar codes**, in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 1665–1668.

[19] P. Shi, W. Tang, S. Zhao, and B. Wang, **Performance of polar codes on wireless communication channels**, in *2012 IEEE 14th International Conference on Communication Technology*, Nov. 2012, pp. 1134–1138.

[20] B. Li, D. Tse, K. Chen, and H. Shen, **Capacity-achieving rateless polar codes**, in *2016 IEEE International Symposium on Information Theory (ISIT)*, Jul. 2016, pp. 46–50.

[21] S.-N. Hong, D. Hui, and I. Marić, **Capacity-Achieving Rate-Compatible Polar Codes**, *IEEE Transactions on Information Theory*, vol. 63, no. 12, pp. 7620–7632, Dec. 2017.

[22] J.-H. Po, S.-J. Chen, and C. Yu, **Variable code length soft-output decoder of polar codes**, in *2015 IEEE International Conference on Digital Signal Processing (DSP)*, Jul. 2015, pp. 655–658.

[23] H. Hsu, A.-Y. Wu, and J.-C. Yeo, **Area-Efficient VLSI Design of Reed–Solomon Decoder for 10GBase-LX4 Optical Communication Systems**, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 11, pp. 1245–1249, Nov. 2006.

[24] B. Feng, Q. Zhang, and J. Jiao, **An Efficient Rateless Scheme Based on the Extendibility of Systematic Polar Codes**, *IEEE Access*, vol. 5, pp. 23223–23232, 2017.

[25] J. Kim and J. Lee, **Polar codes for non-identically distributed channels**, *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 287, Dec. 2016.

[26] Briantoro H, Astawa IGP, Sudarsono A, **An Implementation of Error Minimization Data Transmission in OFDM using Modified Convolutional Code**, *EMITTER International Journal of Engineering Technology*, vol. 3, no. 2, 43–59, 2015.