JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE, VOL.16, NO.5, OCTOBER, 2016 ISSN(Print) 1598-1657
http://dx.doi.org/10.5573/JSTS.2016.16.5.550 ISSN(Online) 2233-4866

An Efficient List Successive Cancellation Decoder for Polar Codes

Zheyan Piao, Chan-Mi Kim, and Jin-Gyun Chung

***Abstract*—Polar codes are one of the most favorable capacity-achieving codes due to their simple structure and low decoding complexity. However, because of the disappointing decoding performance realized using conventional successive cancellation (SC) decoders, polar codes cannot be used directly in practical applications. In contrast to conventional SC decoders, list SC (SCL) decoders with large list sizes (e.g. 32) achieve performances very close to those of maximum-likelihood (ML) decoders. In SCL decoders with large list sizes, however, hardware increase is a severe problem because an SCL decoder with list size *L* consists of *L* copies of an SC decoder. In this paper, we present a low-area SCL decoder architecture that applies the proposed merged processing element-sharing (MPES) algorithm. A merged processing element (MPE) is the basic processing unit in SC decoders, and the required number of MPEs is *L*(*N*-1) in conventional SCL decoders. Using the proposed algorithm reduces the number of MPEs by about 70% compared with conventional SCL decoders when the list size is larger than 32.**

***Index Terms*—Polar codes, list SC decoder, pre-computation, low area**

I. INTRODUCTION

Polar codes, proposed in [1], are the first constructive and provable capacity-achieving error-correcting codes. Recently, due to their simple structure and low decoding complexity, polar codes have received much attention. Although polar codes have many advantages in encoding/decoding structures, the decoding performance is disappointing compared with other well-known codes, such as low-density parity check (LDPC) or Turbo codes. Thus, the list SC (SCL) decoder, an improved version of the SC decoder, was proposed [2]. An SCL decoder can approach the performances of an ML decoder with a large list size *L*.

Theoretical analysis of SCL decoders can be found in [2, 3]. Some hardware implementation issues of SCL decoders, such as latency and complexity, are studied in [4-6]. However, because an SCL decoder consists of *L* copies of an SC decoder, the hardware increase is a non-negligible problem.

In this paper, we present a low-area SCL decoder that reduces the number of merged processing elements (MPEs). An MPE is the basic processing unit in SC decoder construction, and the number of MPEs significantly affects the area of the SCL decoder. By analyzing the decoding procedure of conventional SCL decoders, it is noted that some of the MPEs can be shared among *L* copies of the SC decoder by pre-computing and sharing some computation results in the first two stages in the SCL decoders. When the list size is larger than 32, the proposed architecture can reduce the number of MPEs by 70% compared with conventional SCL decoders.

The rest of this paper is organized as follows. Section II reviews the architecture of the SC decoder. The proposed low-area SCL decoder with hardware analysis is presented in Section III. Section IV presents the hardware analysis of the proposed algorithm. Section V

Manuscript received Mar. 4, 2016; accepted Sep. 27, 2016
Department of Electronic Engineering, Chonbuk National University, Korea
E-mail : capark@jbnu.ac.kr

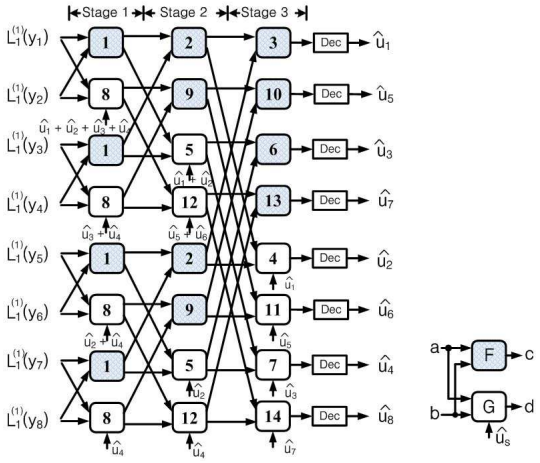


Fig. 1. The decoding procedure of a conventional SC decoder with N=8.

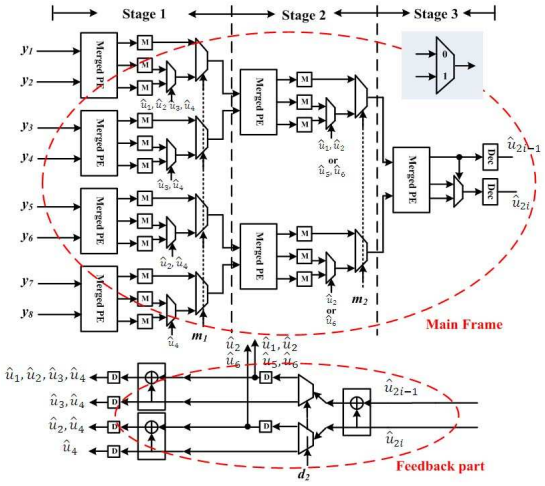


Fig. 2. Latency reduced architecture of an 8-bit SC decoder.

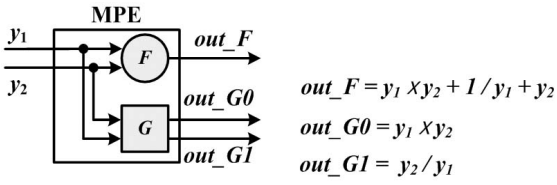


Fig. 3. Simplified block diagram of MPE.

provides the conclusions drawn.

II. Review of SC Decoding Algorithm

By recursively combining and splitting N copies of any given binary-input discrete memoryless channel (B-DMC) W, we obtain a set of N binary-input coordinate channels $\left\{W_N^{(i)},\ \ \le i \le N\right\}$. Among the N channels, only those with the highest capacity are used for data transmission and the inputs to these channels are referred to as information bits ($u_{\mathcal{A}}$). We denote the input vector as u_1^N, which consists of a random part $u_{\mathcal{A}}$ and a frozen part $u_{\mathcal{A}^c}$. The corresponding output vector through the synthesized channel W_N is denoted as y_1^N with conditional probability $W_N(y_1^N \mid u_1^N)$.

The likelihood ratios (LRs) are defined as

$$L_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1}\right) \triangleq \frac{W_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1} \mid 0\right)}{W_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1} \mid 1\right)} \tag{1}$$

and the decision is generated as [1]

$$\hat{u}_i = \begin{cases} 0, & \text{if } L_N^{(i)} \ge 1 \text{ or } \hat{u}_i \text{ is frozen,} \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

The LRs with even and odd indices can be calculated using the following recursive formulas:

$$F(a,b) = \frac{a \cdot b}{a+b}. \tag{3}$$

$$G\left(a,b,\ \hat{u}_s\right) = a^{1-2\hat{u}_s} b. \tag{4}$$

The decoding procedure of the conventional SC decoder with N=8 is shown in Fig. 1, where the number inside each processing element (PE) indicates the index of the clock cycle when the corresponding PE is activated. As can be seen in this figure, the SC decoder consists of $\log_2 N$ stages, where each stage consists of two computation types: F type and G type. F and G computation types compute (3) and (4), respectively.

Conventional SC decoders need 2N-2 clock cycles to decode an input code with a code length of N bits. Due to the long computation time, a latency reduced architecture was proposed, as shown in Fig. 2 for N=8 [7]. By merging the F and G computation types, merged PEs (MPEs) can compute all the proper data in one clock cycle, and the required computation clock cycles are reduced by N-1. Fig. 3 shows the simplified block diagram of the MPE.

As can be seen in Fig. 2, each stage needs de-multiplexers and multiplexers to select its proper data. The selected signals of the multiplexers and de-

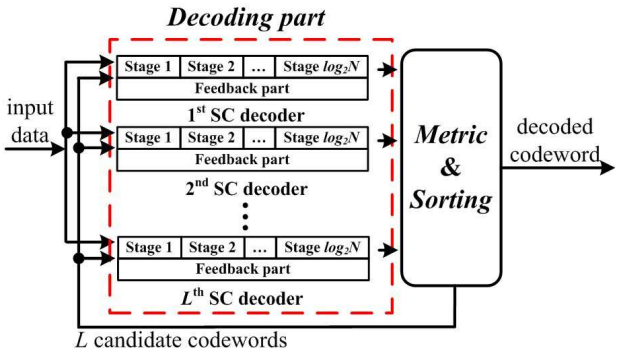


Fig. 4. The decoding procedure of a conventional SC decoder with N=8.

multiplexers in stage i are denoted as m_i and d_i ($1 \le i \le \log_2 N$), respectively.

The relationship between m_i and d_i can be summarized as follows:

$$\begin{aligned} d_{\log_2 N-1}(n) &= m_{\log_2 N-1}(n), \\ d_{\log_2 N-2}(n) &= m_{\log_2 N-2}(n-2^1), \\ d_{\log_2 N-3}(n) &= m_{\log_2 N-3}(n-2^1-2^2), \\ &\vdots \\ d_2(n) &= m_2(n-2^1-2^2-\cdots-2^{\log_2 N-3}). \end{aligned} \quad (5)$$

III. Proposed SCL Decoding Architecture

1. Conventional SCL Decoder

The concept of the SCL decoding algorithm is similar to the K-best MIMO detection algorithm. For the conventional SC decoding algorithm, only the most likely code bit is selected at each stage. However, the SCL decoding algorithm always keeps a list of L code bits at each step and it gives the most likely codeword as the final output [2, 3].

Fig. 4 shows the block diagram of a conventional SCL decoder with list size L. In general, an size L (n, k) SCL decoder consists of a metric & sorting part and a decoding part. The decoding part consists of a combination of L copies of the (n, k) SC decoder. As mentioned previously, the larger the list size L we employ, the better the decoding performance. However, we must pay the penalty of large hardware overhead with an increased list size.

Table 1. Number of MPEs in each stage

Code length (N)	Stages 1&2	Other stages
8	6 (85.7%)	1 (14.3%)
16	12 (80%)	3 (20%)
32	24 (77.4%)	7 (22.6)
64	48 (76.2%)	15 (23.8%)
1024	768 (75.1%)	255 (24.9%)
2048	1,536 (75%)	511 (25%)

Table 2. Efficiency of each stage in the SC decoder with N=64

Stage index (s)	Number of active times	Number of MPEs	Efficiency
	(7)	(6)	(8)
1	1	32	0.03125
2	2	16	0.125
3	4	8	0.5
4	8	4	2
5	16	2	8
6	32	1	32

The architecture proposed in this paper is based on the architecture shown in Fig. 2, as the architecture has obvious advantages in terms of latency compared to the conventional SC decoding algorithm.

As can be seen in Fig. 2, the number of MPEs in each stage decreases as the stage index s increases. More specifically, for code length N, the required number of MPEs in stage s is as follows:

$$\text{Num}_{\text{MPE}} = \frac{N}{2^s}, \quad s = 1, 2, \cdots, \log_2 N. \quad (6)$$

With an increasing code length N, stages 1 and 2 require about 75% of the total number of MPEs used in the SC decoder, as shown in Table 1. In the conventional SCL decoder, the total number of required MPEs is L(N-1). Thus, with an increasing list size L and large code length N, the MPEs will require a large amount of hardware area in the SCL decoders.

In addition, by analyzing the decoding procedure shown in Fig. 2, the number of active times of each stage can be calculated as

$$\text{Num}_{\text{active}} = 2^{s-1}. \quad (7)$$

Table 2 lists the efficiency of each stage in the SC decoder, where efficiency is defined as

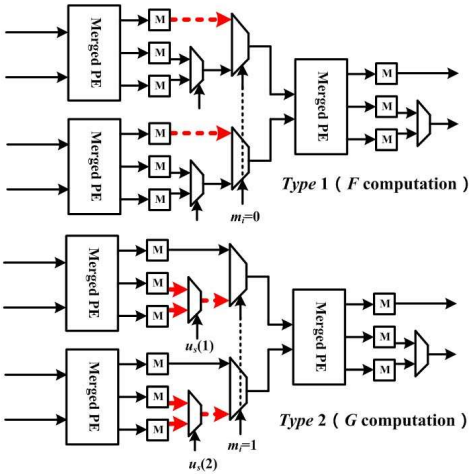


Fig. 5. Two MPE connection types between neighboring stages in the SC decoder.

$$\begin{aligned}\text{Efficiency} &\triangleq \text{Num}_{active} / \text{Num}_{\text{MPE}} \\ &= 4^s / 2N.\end{aligned} \tag{8}$$

As can be seen in Table 2, stages 1 and 2 occupy the highest number of MPEs, but they show the lowest efficiencies. Thus, in this paper, we try to reduce the number of MPEs in stages 1 and 2 in the SCL decoders.

2. Proposed SCL Decoder Architecture

In the conventional architecture of the SCL decoder shown in Fig. 4, all the stage 1s of the SC decoders in the decoding part receive the same input data from the channel. So the outputs of all stage 1s are the same. The only difference is the data passed to each stage 2, which is selected by signal m_i. Thus, instead of using L stage 1s, the output data from one stage 1 can be shared with the other SC decoders if the connection between stage 1 and 2 is well controlled.

Analysis of the decoding procedure shows that the connection between the MPEs in stage 1 and the MPEs in stage 2 can be either type 1 or type 2, as shown in Fig. 5. In addition, it should be noted that type 1 and type 2 cannot occur simultaneously.

When $m_i = 0$, type 1 is selected and the MPE in the next stage receives input data from two F computation types. From the block diagram of the MPE in Fig. 3, the F computation has only one kind of output data. Thus, the MPE in stage 2 will have only one case.

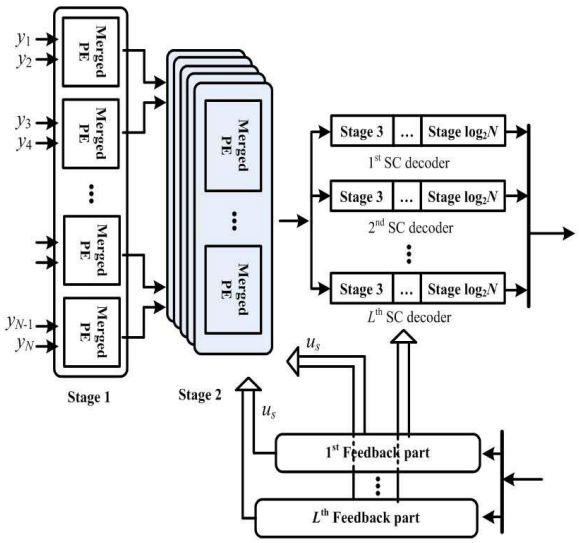


Fig. 6. Block diagram of the decoding part in the proposed SCL decoder.

In contrast to type 1, when $m_i = 1$, type 2 has different cases depending on signals $u_s(1)$ and $u_s(2)$.

The signal u_s can be computed with decoded bits, as shown in Fig. 2, and it has only two possible values, 0 or 1. Therefore, type 2 has 4 possible cases. Thus, if we use 4 MPEs to pre-compute these 4 cases and store the outputs, we do not have to use L stage 2 blocks.

When type 1 is selected, only one stage 2 block is used. When type 2 is selected, four stage 2 blocks are used to pre-compute the four different cases corresponding to the signals $u_s(1)$ and $u_s(2)$. Thus, in total, five stage 2 blocks are required, as shown in Fig. 6.

In stage 3, the proper input data is selected from the memories in stage 2, depending on the selection signals from the feedback part. Therefore, in the proposed design, as shown in Fig. 6, only one stage 1 block and five stage 2 blocks are needed. The rest of the stages require L blocks, like in the conventional SCL decoders. The number of MPEs in the proposed architecture can be calculated as follows:

$$\begin{aligned}\text{Num}_{\text{MPE}} &= \frac{N}{2} + \frac{N}{4} \times 5 + L\left(\frac{N}{4} - 1\right) \\ &= \left[N\left(7 + L\right) - 4L\right] / 4.\end{aligned} \tag{9}$$

The proposed merged processing element-sharing (MPES) algorithm is summarized as Algorithm 1.

Algorithm 1: MPES algorithm

Step 1 : Regardless of the list size, use only one stage 1 block;

Step 2 : For stage 2, use five stage 2 blocks. The output of the F computation data from stage 1 is connected to the first stage 2 block (stage 2_F). The outputs of the G computation data from stage 1 are connected to the rest of the four stage 2 blocks (stage 2_G_{00} – stage 2_G_{11});

Step 3 : The other stages remain the same, as in the conventional SCL decoders;

Step 4 : The four stage 2 blocks (stage 2_G_{00} – stage 2_G_{11}) calculate the 4 different cases corresponding to all the possible values of $u_s(1)$ and $u_s(2)$;

Step 5 : When stage 2 is active for the first time, pass the calculated data from stage 2_F to stage 3;

Step 6 : When stage 2 is active a second time, pass the calculated data from stage 2_G_{00}, stage 2_G_{01}, stage 2_G_{10} and stage 2_G_{11} to stage 3.

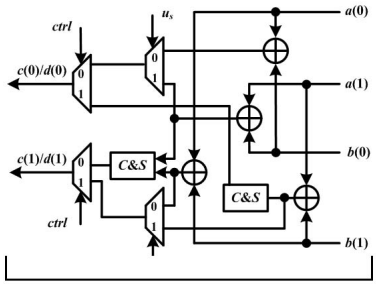


Fig. 7. Architecture of LL-based MPE.

IV. Analysis of Proposed SCL Decoding Architecture

As shown in Fig. 3, the likelihood-based MPE contains numerical stability and implementation, the SC decoding algorithm over the logarithm domain is preferable. Fig. 7 shows the log-likelihood version of the MPE proposed in [4]. Here, the *C&S* block represents the combined comparator and 2-to-1 selector. The signal *ctrl* is the same as the signal m_i in Fig. 2. Fig. 8 shows the detailed architecture of the proposed SCL decoding part with N=8 and L=4.

Table 3 compares the number of MPEs in [5] with that of the proposed architecture. Fig. 9 shows the reduction rate of the number of MPEs with the proposed algorithm compared with the conventional algorithm in [5]. As can be seen in Fig. 9, the reduction rate of MPEs is directly dependent on the list size. For example, when the list size is L, the number of MPEs in conventional and proposed architectures is $L(N-1)$ and $(N(7+L)-4L)/4$, respectively. So the reduction rate can be derived as follows:

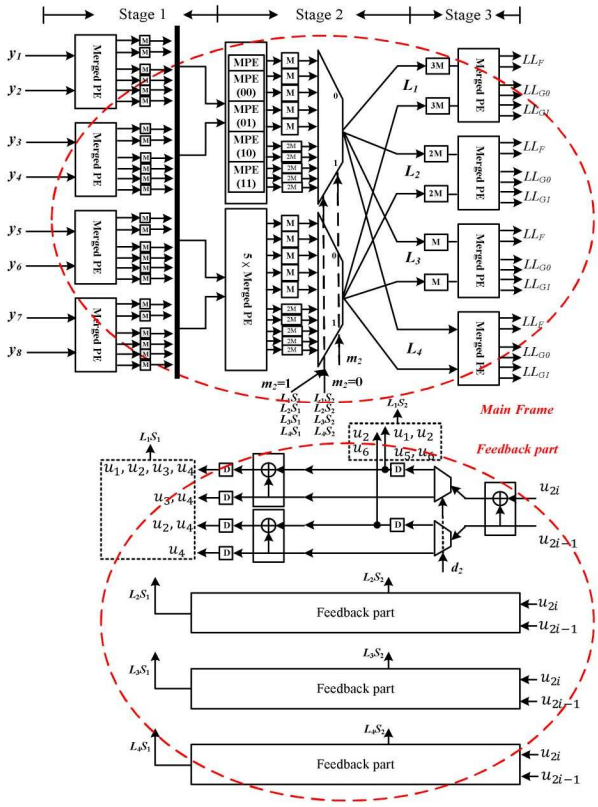


Fig. 8. Architecture of the proposed 8-bit SCL decoder with L=4.

Table 3. Comparison of number of MPEs in various code lengths N

Code length (N)	List size (L)	Number of MPEs in [5]	Number of MPEs in proposed design
512	2	1,022	1,150
	4	2,044	1,404
	8	4,088	1,912
	16	8,176	2,928
	32	16,352	4,960
1024	2	2,046	2,302
	4	4,092	2,812
	8	8,184	3,832
	16	16,368	5,872
	32	32,736	9,952
2048	2	4,094	4,606
	4	8,188	5,628
	8	16,376	7,672
	16	32,752	11,760
	32	65,504	19,936

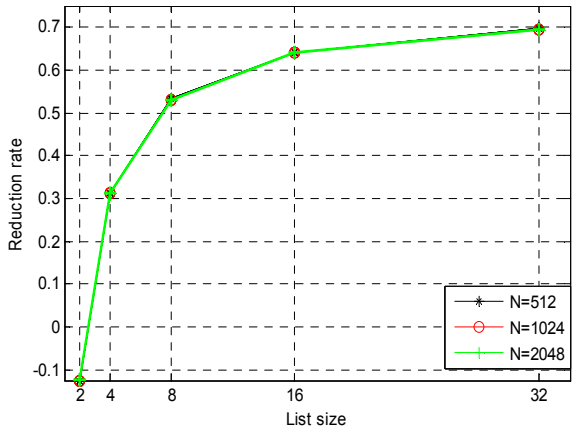


Fig. 9. Reduction rate of the number of MPEs in proposed algorithm.

Table 4. Time schedule of 8-bit SCL decoder with L=2 in conventional architecture

Stage \ Clock		1	2	3	4	5	6	7
1	L1	4 MPEs						
	L2	4 MPEs						
2	L1		2 MPEs			2 MPEs		
	L2		2 MPEs			2 MPEs		
3	L1			1 MPE	1 MPE		1 MPE	1 MPE
	L2			1 MPE	1 MPE		1 MPE	1 MPE
Sort				S	S		S	S
Out	L1			$\hat{u}_1^2$	$\hat{u}_3^4$		$\hat{u}_5^6$	$\hat{u}_7^8$
	L2			$\hat{u}_1^2$	$\hat{u}_3^4$		$\hat{u}_5^6$	$\hat{u}_7^8$

$$R_{reduction} = \frac{L(N-1)-\left[N(7+L)-4L\right]/4}{L(N-1)} = \frac{N\left(\frac{3L}{4}-\frac{7}{4}\right)}{N\left(L-\frac{L}{N}\right)}. \quad (10)$$

When $N \gg L$, (10) becomes

$$R_{reduction} \approx 3L-7/4L. \quad (11)$$

When the list size is larger than 32, the proposed architecture reduces the number of MPEs by 70% compared with conventional SCL decoders.

Table 4 shows the time schedule of the conventional 8-bit SCL decoder with L=2. In Table 4, (x MPEs) means the number of active MPEs in given clock cycle is x.

Table 5. Time schedule of 8-bit SCL decoder with L=2 in the proposed architecture

Stage \ Clock		1	2	3	4	5	6	7	8	9	10
1		4 MPEs									
2			5 × 2 MPEs								
3	L1				1 MPE		1 MPE		1 MPE		1 MPE
	L2				1 MPE		1 MPE		1 MPE		1 MPE
Sort					S		S		S		S
Out	L1				$\hat{u}_1^2$		$\hat{u}_3^4$		$\hat{u}_5^6$		$\hat{u}_7^8$
	L2				$\hat{u}_1^2$		$\hat{u}_3^4$		$\hat{u}_5^6$		$\hat{u}_7^8$

Table 6. Hardware comparison of SCL decoders with N=32

	L=4		L=8	
	[5]	Proposed	[5]	Proposed
# of MPEs	124(100%)	84(69%)	248(100%)	112(47%)
# of logic elements	13,889(100%)	10,717(77%)	27,310(100%)	13,430(49%)

Table 5 shows the time schedule of the proposed 8-bit SCL decoder with L=2. In the proposed architecture, to synchronize the data transfer between stage 2 and stage 3, additional memory blocks are used. The decoding latency in the proposed architecture can be derived as follows:

$$T_{latency} = N-1+(2^{(s=3)-1})(L-1)-1 = N+4L-6. \quad (12)$$

Table 6 compares the hardware complexity in [5] with that of the proposed architecture. When the list size is close to 32, the decoding performance is very close to that of maximum likelihood (ML) decoders. However, with the list size increasing, the implementation complexity, computation complexity and memory size are all increasing. Thus, unlike the implementation in software, large list size has significant limitation in hardware implementation.

In order to demonstrate the advantage of the proposed architecture, hardware implementation results of different list size are listed in Table 6. All designs are implemented using Altera Cyclone III FPGA with device number EP3C40F780C6.

From Table 6, the proposed architecture with list size 4 and 8 both show significant hardware reduction compared to the conventional architecture [5]. According

to Fig. 9, when the list size is 4 and 8, the reduction rate of the number of MPEs is 31% and 53%, respectively. However, due to the additional memory blocks used by the proposed architecture, the total reduction rate of hardware cost in proposed architecture with list size 4 and 8 is 23% and 51% as shown in Table 6. As commented in [7], it is suggested to use memory modules instead of heavy use of D flip-flops in the feedback part.

V. CONCLUSIONS

In this paper, we have presented a low-area SCL decoder architecture. In the proposed design, by using the MPES algorithm, we can reduce the number of MPEs by approximately 70%, compared with conventional SCL decoders, when the list size is larger than 32. Hardware analysis shows that the proposed architecture reduces the total logic elements by 51% compared with conventional designs for N=32 and L=8. The proposed MPES algorithm gives more efficient results when the list size is large.

ACKNOWLEDGEMENT

This research was supported by "Research Base Construction Fund Support Program" funded by Chonbuk National University in 2016.

REFERENCES

[1] E. Arikan, "Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. on Inf. Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009.

[2] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. IEEE Inter. Symp. on Info. Theory* (*ISIT*), pp. 1-5, 2011.

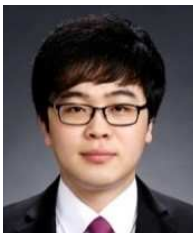
[3] K. Chen, K. Niu, and J. Lin, "List successive cancellation decoding of polar codes," *Electron. Lett.*, vol. 48, no. 9, pp. 500-501, 2012.

[4] Bo Yuan and Keshab K. Parhi, "Low-latency successive-cacellation list decoders for polar codes with multibit decision," *IEEE Trans. on VLSI sys.*, vol. 23, no. 10, pp. 2268-2280, 2015.

[5] Chuan Zhang, Xiaohu You and Jin Sha, "Hardware architecture for list successive cancellation polar decoder," in *Proc. IEEE Int. Symp. on Circuits Syst.* (*ISCAS*), pp. 209-212, 2014.

[6] Chuan Zhang, Zhongfeng Wang, Xiaohu You and Bo Yuan, "Efficient adaptive list successive cancellation decoder for polar codes," in *Proc. IEEE Asilomar Conf. on Signals, Syst. and Computers*, pp. 126-130, 2014.

[7] C. Zhang and K. K. Parhi, "Low-latency sequential and overlapped architectures for successive cancellation polar decoders," *IEEE Trans. on Signal Proc*., vol. 61, no. 10, pp. 2429-2441, 2013.



Zheyan Piao received the B.S. degree in Communication Engineering from Yanbian University, Yanji, China, in 2010 and the M.S. degree in Electronic Engineering at Chonbuk National University, Chonju, South Korea, in 2012. He is currently pursuing his Ph.D. in Electronic Engineering at Chonbuk National University. His research interests include VLSI design of digital signal processing and communication algorithms.

Chan-Mi Kim received the B.S. degree in Electronic Engineering at Chonbuk National University, Chonju, South Korea, in 2015. He is currently pursuing his M.S. in Electronic Engineering at Chonbuk National University. His research interests include VLSI design of digital signal processing and communication algorithms.

Jin-Gyun Chung received M.S. and Ph.D. degrees in Electrical Engineering from the University of Minnesota, Minneapolis, Minnesota, in 1991 and 1994, respectively. Since 1995, he has been with the Division of Electronic Engineering at Chonbuk National University. His research interests are VLSI architectures and algorithms for signal processing and communication systems, including the design of high-speed and low-power algorithms for digital filters, XRF systems, OFDM systems, and ultrasonic NDE systems.