

# An Efficient Successive Cancellation Polar Decoder Based on New Folding Approaches

Xiao Liang<sup>1,2</sup>, Yechao She<sup>3</sup>, Harish Vangala<sup>4</sup>, Xiaohu You<sup>2</sup>, Chuan Zhang<sup>1,2,\*</sup>, and Emanuele Viterbo<sup>4</sup>

<sup>1</sup>Lab of Efficient Architectures for Digital-communication and Signal-processing (LEADS)

<sup>2</sup>National Mobile Communications Research Laboratory, Southeast University, Nanjing, China

<sup>3</sup>Department of Electronic and Computer Engineering, HKUST, Hong Kong, China

<sup>4</sup>Department of Electrical and Computer Systems Engineering, Monash University, Melbourne, Australia

Email: <sup>2</sup>{xiao\_liang, xhyu, chzhang}@seu.edu.cn, <sup>3</sup>yshe@ust.hk, <sup>4</sup>{harish.vangala, emanuele.viterbo}@monash.edu

**Abstract**—In this paper, an efficient successive cancellation (SC) polar decoder based on new folding approaches is proposed. The main approach of this paper is called  $k$ -level decomposition with  $2^p$  sub-decoders. Adjusting  $k$  and  $p$ , the derived architecture can have a very low processing complexity with proper combinations of decomposition method and folding technique. Compared to state-of-the-art designs, hardware utilization ratio (HUR) of processing elements can be drastically improved with small latency overhead. Meanwhile, the memory complexity remains similar. Furthermore, decomposition and folding operations can also be applied to a family of hybrid polar decoders. To validate efficiency of these approaches, two folded SC decoders with  $N = 64$  and 1024 respectively are implemented with Altera Stratix V FPGA. These two demos require only 68.3% and 39.1% ALMs, compared to the non-decomposed SC decoder.

**Index Terms**—Polar code, SC decoder,  $k$ -level decomposition, folding technique, VLSI implementation.

## I. INTRODUCTION

Proposed by Arıkan, polar codes are of capacity-achieving capability for binary-input discrete memoryless channels (BDMCs) [1]. With the help of channel polarization, polar codes show advantages of low encoding-complexity and capacity-achieving performance, nevertheless, the latter is guaranteed by successive cancellation (SC) decoding and large block size. In order to achieve excellent performance which can satisfy the requirements of real applications, polar codes with thousand bit length is expected. Apparently, corresponding full size hardware design, which contains thousands of processing nodes, is impractical.

Researchers have already made a lot of efforts for the improvement of SC decoder in respect to the hardware complexity and decoding latency. In [2], an empirical method was presented to construct tree architecture SC decoder. Using pre-computation and look-ahead schedule, it reduces the hardware complexity to half and decoding latency from  $(2N - 1)$  cycles to  $(N - 1)$ . In [3], pipeline technique was combined with previous tree architecture to achieve much lower decoding latency. To further increase utilization ratio of processing units, line architecture was suggested by [4]. Afterwards, semi-parallel line architecture was proposed in [5]. [6, 7] proposed folded SC decoders to reduce hardware complexity. However, the decomposition method in [7] is limited by the factor of  $n = \log_2(N)$ . Furthermore, in order to achieve decoding

latency reduction, hybrid polar codes are proposed. For example, combination of SC and maximum likelihood (ML) decoding, or combination of SC and list SC decoding, works perfectly regarding the tradeoff between hardware complexity and decoding latency.

In this study, we apply  $k$ -level decomposition and folding operations with  $2^p$  sub-decoders to propose an efficient SC polar decoder, which obtains drastic hardware reduction. By adjusting  $k$  and  $p$ , it is flexible to balance decoding latency and hardware complexity. SC+SC polar decoders are used as an example in this paper, but similar optimization can also be applied to other hybrid polar decoders and SC list decoders. FPGA implementation with Altera Stratix V is also given to demonstrate advantages of the proposed decoder.

Remainder of the paper is organized as follows. Section II briefly reviews polar codes and gives explanation to decomposition of polar codes. Section III focuses on construction of folded SC polar decoder. Hardware architectures of 8-bit folded SC polar decoder decomposed with different levels are given as an example. Then,  $k$ -level decomposed polar code is discussed in detail. FPGA implementations are given in Section IV. Section V concludes the entire paper.

## II. POLAR CODE AND DECOMPOSITION

### A. Polar Encoding and SC Decoding

A polar code is completely specified by  $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ , where  $N$  is the code length,  $K$  is the number of information bits,  $\mathcal{A}$  is the information set, and  $u_{\mathcal{A}^c}$  are frozen bits' values.

Denote  $n = \log_2(N)$  and  $\mathbf{F} \triangleq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ . By  $n$ -fold Kronecker product of  $\mathbf{F}$ , a codeword is generated as  $\mathbf{x} = \mathbf{G}\mathbf{u} = \mathbf{F}^{\otimes n}\mathbf{u}$ . Namely,  $u_i^j$  is a vector which consists of  $[u_i, \dots, u_j]$ .

Given a received vector  $\mathbf{y}$  corresponding to a transmitted codeword  $\mathbf{x}$  through  $W_N$ , the logarithm-likelihood ratio (LLR) is defined as,

$$L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \triangleq \ln \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i = 0)}{W_n^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i = 1)}, \quad (1)$$

where  $\hat{u}_1^{i-1}$  are previously estimated bits. It is noticed that the LLRs with even and odd indices can be generated by recursively applying two equations in Eq. (2) respectively.

$$\begin{cases} L_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}) \approx \text{sgn}(L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) \text{sgn}(L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})). \\ \min(L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}), L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})) \\ L_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = (-1)^{\hat{u}_{1,o}^{2i-1}} L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) + L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}). \end{cases} \quad (2)$$

### B. $k$ -Level Decomposition of Polar Coding

By exploiting the recursive nature of polar codes, decomposition of original polar encoder into smaller ones is proposed by [8]. [7] decomposed an  $N$ -bit polar encoder into several sub-encoders with the same code length. On the contrary, this paper proposed a  $k$ -level decomposition, where  $k$  can be any number smaller than  $n$ . Denote  $\mathbf{P}$  as a permutation matrix. The encoding equation  $\mathbf{x} = \mathbf{F}^{\otimes n} \mathbf{u}$  can be rewritten by:

$$\begin{aligned} \mathbf{v} &= (\mathbf{F}^{\otimes(n-k)} \otimes \mathbf{I}_{2^k}) \mathbf{u} = \mathbf{P}^T (\mathbf{I}_{2^k} \otimes \mathbf{F}^{\otimes(n-k)}) \underbrace{\mathbf{P} \mathbf{u}}_{\mathbf{u}'} \\ &= \mathbf{P}^T \begin{bmatrix} \mathbf{F}^{\otimes(n-k)} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{F}^{\otimes(n-k)} \end{bmatrix} \begin{bmatrix} \mathbf{u}'_1 \\ \vdots \\ \mathbf{u}'_{2^k} \end{bmatrix}. \quad (3) \\ \mathbf{x} &= \mathbf{F}^{\otimes n} \mathbf{u} = (\mathbf{I}_{2^{n-k}} \cdot \mathbf{F}^{\otimes(n-k)} \otimes \mathbf{F}^{\otimes k} \cdot \mathbf{I}_{2^k}) \mathbf{u} \\ &= (\mathbf{I}_{2^{n-k}} \cdot \mathbf{F}^{\otimes k}) \cdot \underbrace{(\mathbf{F}^{\otimes(n-k)} \otimes \mathbf{I}_{2^k})}_{\mathbf{v}} \mathbf{u} \\ &= \begin{bmatrix} \mathbf{F}^{\otimes k} \cdot \mathbf{v}_1 \\ \vdots \\ \mathbf{F}^{\otimes k} \cdot \mathbf{v}_{2^{n-k}} \end{bmatrix}. \quad (4) \end{aligned}$$

$[\mathbf{u}'_1, \dots, \mathbf{u}'_{2^k}]$  are the sub-vectors of  $\mathbf{u}'$  of length  $2^{n-k}$ , for  $i = 1 : 2^k$ ,  $\mathbf{u}'_i = u'_{1+(i-1) \cdot 2^{n-k}}$ .  $[\mathbf{v}_1, \dots, \mathbf{v}_{2^{n-k}}]$  are the sub-vectors of  $\mathbf{v}$  of length  $2^k$ , for  $j = 1 : 2^{n-k}$ ,  $\mathbf{v}_j = v_{1+(j-1) \cdot 2^k}$ .

What is observed from Eq.s (3) and (4) is that the encoding process can be decomposed into two stages. Eq. (3) represents the first stage:  $\mathbf{u}$  is permuted into  $2^k$  independent polar codes of code length  $2^{n-k}$ . The  $i^{\text{th}}$  polar encoder operates on  $\mathbf{u}'_i = u'_{1+(i-1) \cdot 2^{n-k}}$  to yield  $\mathbf{v}$ .

Eq. (4) represents the second stage:  $2^{n-k}$  independent polar codes of code length  $2^k$ . The  $j^{\text{th}}$  polar encoder operates on  $\mathbf{v}_j = v_{1+(j-1) \cdot 2^k}$  to yield  $x_{1+(j-1) \cdot 2^k}^{j \cdot 2^k}$ , for  $j = 1 : 2^{n-k}$ .

Therefore, it is possible to concatenate polar decoders of length  $2^k$  and  $2^{n-k}$  to realize the function of polar decoder of length  $2^n$ . Denote  $L_k$  and  $L_d$  to distinguish the LLRs of these two decoders of different lengths. With the help of pre-computation scheme proposed in [2], the decoding scheme for  $k$ -level decomposition can be described as follows:

- Initialization:  $i = 1$ , for  $j = 1 : 2^{n-k}$ ,  $j^{\text{th}}$  decoder's  $L_{k,1}(y_1^{2^k}) = L_1(y_{1+(j-1) \cdot 2^k}^{j \cdot 2^k})$ ;
- Step1: for  $j = 1 : 2^{n-k}$ ,  $j^{\text{th}}$  decoder of length  $2^k$  computes out  $L_{k,2^k}^{(i)}$  separately;
- Step2: Input  $L_{d,1}^{(1)}(y_j) = L_{k,2^k}^{(i)}$  to decoder of length  $2^{n-k}$  to compute  $L_{d,2^{n-k}}$ , thus  $\hat{u}_{(i-1) \cdot 2^{n-k} + 1}^{i \cdot 2^{n-k}}$  is obtained;
- Step3: with pre-computation scheme, feed  $\hat{u}_{(i-1) \cdot 2^{n-k} + 1}^{i \cdot 2^{n-k}}$

back to decoders of length  $2^k$  to obtain  $L_{k,2^k}^{(i+1)}$  directly;

- Step4:  $L_{d,1}^{(1)}(y_j) = L_{k,2^k}^{(i+1)}$ , decoder of length  $2^{n-k}$  computes out  $L_{d,2^{n-k}}$  to obtain  $\hat{u}_{i \cdot 2^{n-k} + 1}^{(i+1) \cdot 2^{n-k}}$ ;
- Step5:  $i = i + 2$ , go back to step1. Iterate the process till all bits are decoded out (when  $i > 2^k$ ).

### III. FOLDING WITH $k$ -LEVEL DECOMPOSITION

According to Section II.B,  $2^{n-k}$  decoders of length  $2^k$  work in parallel to realize the  $k$ -level decomposed decoder. Multiplexing  $2^p$  sub-decoders of length  $2^k$ , hardware complexity is reduced at the cost of slightly increased decoding latency.

#### A. 2-Level Decomposed 8-Bit Polar Decoder

2-level decomposed 8-bit polar decoder can be viewed as concatenating of two 4-bit polar decoders and one 2-bit polar decoder. If only one 4-bit polar decoder works in serial ( $p = 0$ ), then decoding schedule can be modified into Table I.  $S_i$  denotes stage  $i$  in Fig. 1. In order to increase utilization ratio of processing elements, pipeline technique is also used. Underlined value indicates that pre-computation is used, therefore no extra clock cycle is required.

TABLE I  
SCHEDULING FOR FOLDED 2-LEVEL DECOMPOSED 8-BIT SC DECODER

CC	1	2	3	4	5	6	7	8	9
$S_1$	$L_2^{(1)}(y_1^2)$	$L_2^{(1)}(y_5^6)$	-	-	-	$L_2^{(2)}(y_1^2)$	$L_2^{(2)}(y_5^6)$	-	-
	$L_2^{(1)}(y_3^4)$	$L_2^{(1)}(y_7^8)$	-	-	-	$L_2^{(2)}(y_3^4)$	$L_2^{(2)}(y_7^8)$	-	-
$S_2$	-	$L_4^{(1)}(y_1^4)$	$L_4^{(1)}(y_5^8)$	-	$L_4^{(2)}(y_1^4)$	$L_4^{(3)}(y_1^4)$	$L_4^{(3)}(y_5^8)$	-	$L_4^{(3)}(y_1^4)$
	-	-	-	-	$L_4^{(2)}(y_5^8)$	-	-	-	$L_4^{(4)}(y_5^8)$
$S_3$	-	-	-	$L_8^{(1)}(y_1^8)$	$L_8^{(3)}(y_1^8)$	-	-	$L_8^{(5)}(y_1^8)$	$L_8^{(7)}(y_1^8)$
	-	-	-	$L_8^{(2)}(y_1^8)$	$L_8^{(4)}(y_1^8)$	-	-	$L_8^{(6)}(y_1^8)$	$L_8^{(8)}(y_1^8)$
out	-	-	-	$\hat{u}_1$	$\hat{u}_3$	-	-	$\hat{u}_5$	$\hat{u}_7$
	-	-	-	$\hat{u}_2$	$\hat{u}_4$	-	-	$\hat{u}_6$	$\hat{u}_8$

Corresponding architecture is shown in Fig. 1. Control signal  $s$  is used to multiplex the 4-bit polar decoder. Generation of control signals is not discussed in the paper.

#### B. 1-level Decomposed Folded 8-bit Polar Decoder

In respective to 1-level decomposed 8-bit polar decoder, it can be viewed as four 2-bit polar decoders and one 4-bit polar decoder. If we use only one 2-bit decoder to replace four 2-bit decoders ( $p = 0$ ), the architecture can be shown in Fig. 2. If we use two 2-bit decoders to replace four 2-bit decoders ( $p = 1$ ), the architecture can be shown in Fig. 3.

#### C. $k$ -level Decomposed Folded $N$ -bit Polar Decoder

Denote  $\mathbf{D}_{2^k} = [D_{2^k}^{(1)}, \dots, D_{2^k}^{(2^{n-k})}]$  as  $2^{n-k}$  decoders of length  $2^k$ . Correspondingly,  $D_{2^{n-k}}$  represents decoder of length  $2^{n-k}$ . Folding set is an ordered set of operations

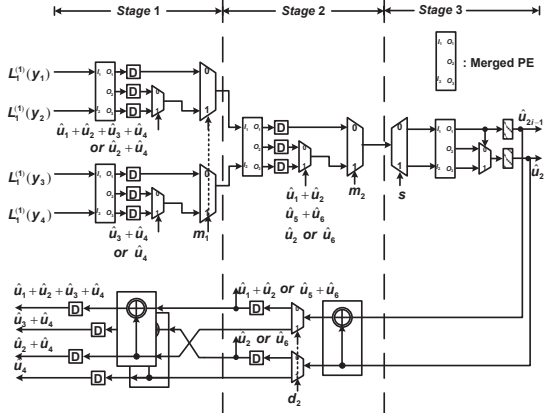


Fig. 1. 2-level Decomposed 8-bit decoder with one 4-bit polar decoder.

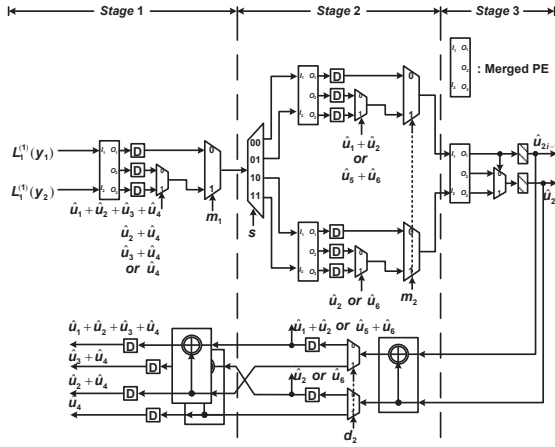


Fig. 2. 1-level decomposed 8-bit decoder with one 2-bit polar decoder.

executed by the same functional unit. Each folding set contains  $w$  entries some of which may be null operations.  $w$  is called folding factor. Consider the following folding sets:

$$\left\{ \begin{array}{l} \mathbf{D}'_{2^k,1} = \{D_{2^k}^{(1)}, \dots, D_{2^k}^{(2^{n-k}-2^p+1)}, \emptyset\}, \\ \dots \\ \mathbf{D}'_{2^k,i} = \{D_{2^k}^{(i)}, \dots, D_{2^k}^{(2^{n-k}-2^p+i)}, \emptyset\}, \\ \dots \\ \mathbf{D}'_{2^k,2^p} = \{D_{2^k}^{(2^p)}, \dots, D_{2^k}^{(2^{n-k})}, \emptyset\}; \end{array} \right. \quad (5)$$

where  $2^{n-k}$  decoders are folded into  $2^p$  decoders.  $\mathbf{D}'_{2^{n-k}}$  should be activated at the respective idle time instances of  $\mathbf{D}'_{2^k}$ , therefore folding set of  $D_{2^{n-k}}$  can be described as follows:  $\mathbf{D}'_{2^{n-k}} = \{\emptyset, \dots, \emptyset, D_{2^{n-k}}\}$ . As to the folding set above,  $w = 2^{n-k-p} + 1$ . Assume each activation cost a time partition, the decoding schedule can be described as follows:

- Initialization:  $j = 1, l = 0$ ;
- Step1: at the  $(j + wl)^{th}$  time partition, where  $j = 1 : w - 1$ , function unit  $\mathbf{D}'_{2^k,i}$  operates as the  $j^{th}$  decoder in the folding set,  $i = 1 : 2^p$ ;
- Step2: at the  $(wl+w)^{th}$  time partition,  $\mathbf{D}'_{2^{n-k}}$  is activated

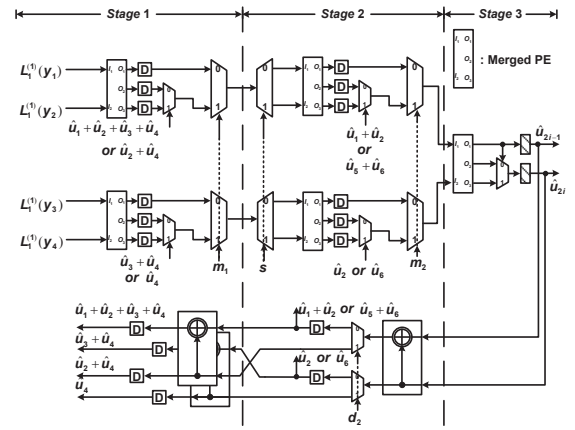


Fig. 3. 1-level decomposed 8-bit decoder with two 2-bit polar decoders.

and decoded out  $\hat{u}_{l,2^{n-k}+1}^{(l+1) \cdot 2^{n-k}}$ ; if  $l = 2^k - 1$ , all information bits have been decoded. Otherwise, continue;

- Step3: feed  $\hat{u}_1^{(l+1) \cdot 2^{n-k}}$  to  $\mathbf{D}'_{2^k,i}$ ,  $l++$  and go to Step1.

In a folded SC decoder,  $D_{2^{n-k}}$  is going to be activated  $2^k$  times. Due to the use of pre-computation and look-ahead technique, each activation costs  $(2^{n-k} - 1)$  clock cycles. As to  $\mathbf{D}_{2^k}$ , each function unit operates full decoding process for each element in the set although is interrupted for  $2^k$  times, half of these interruptions lead to selections of pre-computed values at the last stage. Consequently, only  $(w - 2) \cdot 2^{k-1}$  extra clock cycles are cost due to pipeline. Then, the latency of proposed decoder is:

$$\begin{aligned} t_{FP} &= 2^k \cdot (2^{n-k} - 1) + 2^k - 1 \\ &\quad + (2^{n-k-p} + 1 - 2) \cdot 2^{k-1} \\ &= 2^n - 1 + 2^{n-p-1} - 2^{k-1}. \end{aligned} \quad (6)$$

Define the relative speed factor of proposed decoder as:

$$\alpha_{FP} = \frac{t_{FP}}{t_{ref}} = \frac{2^n - 1 + 2^{n-p-1} - 2^{k-1}}{2^n - 1}, \quad (7)$$

where  $\alpha_{FP}$  defines the latency ratio of proposed folded polar decoder to the reference architecture [2]. During the whole process, a total of  $n \cdot 2^{n-1}$  processing nodes are updated with pre-computation scheme. The main frame of proposed polar decoder is composed of  $2^p$  polar decoders of length  $2^k$ , and one polar decoder of length  $2^{n-k}$ . Thus a total of  $(2^p \cdot (2^k - 1) + 2^{n-k} - 1)$  processing units are required. The hardware utilization rate (HUR) of proposed folded polar decoder is thus:

$$\begin{aligned} r_{FP} &= \frac{\# \text{ of required module operations}}{\# \text{ of all available module operations}} \\ &= \frac{n2^{n-1}}{(2^n - 1 + 2^{n-p-1} - 2^{k-1})(2^{p+k} - 2^p + 2^{n-k} - 1)}. \end{aligned} \quad (8)$$

The HUR of reference architecture [2] is

$$r_{ref} = \frac{n2^{n-1}}{(2^n - 1) \cdot (2^n - 1)}. \quad (9)$$

Therefore, the relative HUR ratio of proposed folded polar decoder to the reference one can be defined as:

$$\beta_{FP} = \frac{r_{FP}}{r_{ref}} = \frac{(2^n - 1)^2}{(2^n - 1 + 2^{n-p-1} - 2^{k-1})(2^{p+k} - 2^p + 2^{n-k} - 1)}. \quad (10)$$

Figs. 4 and 5 show the relationships between  $\alpha_{FP}$  and  $k$ ,

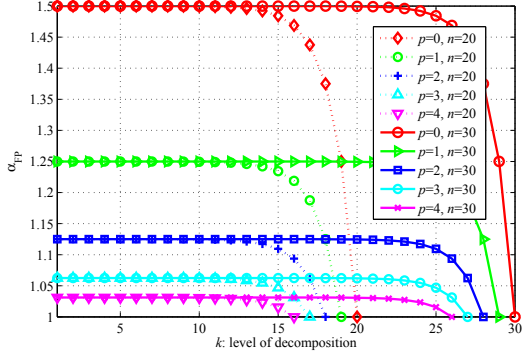


Fig. 4. Relative-speed factor  $\alpha_{FC}$  for the folded SC decoder.

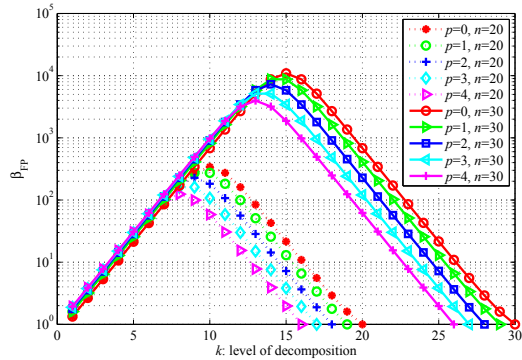


Fig. 5. Relative-utilization ratio  $\beta_{FC}$  for the folded SC decoder.

$\beta_{FP}$  and  $k$ , respectively. When  $k$  approaches to  $n$ , the folding effect will be impaired. As illustrated in Fig. 4,  $\alpha_{FP}$  of  $n = 30$  overlapped that of  $n = 20$  when  $k < n - 6$ , which shows that  $\alpha_{FP}$  is not sensitive to code length. And  $\alpha_{FP}$  is a stable value when  $k < n - 6$ . Fig. 5 shows that the peak value of  $\beta_{FP}$  appears at  $\lceil (n-p)/2 \rceil$ , where  $\lceil \cdot \rceil$  means rounding up the value. In addition, when  $p$  increases,  $\alpha_{FP}$  will be improved while  $\beta_{FP}$  will be reduced. Thus, by adjusting  $k$ ,  $p$ , it is flexible to trade off between  $\alpha_{FP}$  and  $\beta_{FP}$ , decoding latency and hardware complexity correspondingly.

#### IV. VLSI IMPLEMENTATION RESULTS

The hardware platform is set up on Altera Stratix V FPGA. The quantization of LLR is: 1 sign bit, 6 integer bits, and 1 decimal bit. As is shown in Tab. II, two groups of polar codes are considered. In the first group, folded 64-bit SC decoder ( $k = 3$ ,  $p = 1$ ) with three 8-bit sub-decoders

requires 68.3% ALMs (adaptive logical modules) compared to non-decomposed SC decoder ( $k = n = 6$ ). In the second group, 1024-bit SC decoder ( $k = 5$ ,  $p = 0$ ) with two 32-bit sub-decoders requires only 39.1% ALMs compared to non-decomposed SC decoder ( $k = n = 10$ ).

TABLE II  
IMPLEMENTATION RESULTS COMPARISON

SC Module	ALMs	Register	Latency
<b>8-bit sub-decoder</b>	325	183	7
<b>64-bit decoder (<math>k = 6</math>)</b>	2,490	1,642	63
<b>64-bit decoder (<math>k = 3</math>, <math>p = 1</math>)</b>	1,782	1,584	75
<b>32-bit sub-decoder</b>	1,424	809	31
<b>1024-bit Decoder (<math>k = 10</math>)</b>	46,572	25,888	1023
<b>1024-bit decoder (<math>k = 5</math>, <math>p = 0</math>)</b>	18,685	18,302	1519

#### V. CONCLUSION

A folded SC polar decoder based on decomposition is proposed. Adjusting the decomposition level  $k$ , and number of folding sets  $2^p$ , it is convenient to balance complexity and latency. Meanwhile, folded SC polar decoder has exactly the same error-correction performance as other SC polar decoders. As to hybrid polar decoders, we can simply replace one SC polar decoder with other kinds of decoders.

#### ACKNOWLEDGEMENT

This work is supported in part by NSFC under grant 61501116, Jiangsu Provincial NSF under grant BK20140636, Huawei HIRP Flagship under grant YB201504, the Fundamental Research Funds for the Central Universities, State Key Laboratory of ASIC & System under grant 2016KF007, ICRI for MNC, and the Project Sponsored by the SRF for the Returned Overseas Chinese Scholars of MoE.

#### REFERENCES

- [1] E. Arkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.
- [2] C. Zhang, B. Yuan, and K. K. Parhi, "Reduced-latency SC polar decoder architectures," in *Proc. IEEE International Conference on Communications (ICC)*, 2012, pp. 3471–3475.
- [3] C. Zhang and K. K. Parhi, "Low-latency sequential and overlapped architectures for successive cancellation polar decoder," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2429–2441, May 2013.
- [4] C. Leroux, I. Tal, A. Vardy, and W. J. Gross, "Hardware architectures for successive cancellation decoding of polar codes," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 1665–1668.
- [5] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for polar codes," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 289–299, Jan 2013.
- [6] C. Zhang and K. K. Parhi, "Interleaved successive cancellation polar decoders," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014, pp. 401–404.
- [7] X. Liang, C. Zhang, S. Zhang, and X. You, "Hardware-efficient folded SC polar decoder based on  $k$ -segment decomposition," in *Proc. IEEE Asia Pacific Conf. on Circ. and Syst. (APCCAS)*, Oct 2016, pp. 1–4.
- [8] S. Kahanman, E. Viterbo, and M. E. Celebi, "Folded tree maximum-likelihood decoder for kronecker product-based codes," in *Proc. Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2013, pp. 629–636.