Multiple Folding for Successive Cancelation Decoding of Polar Codes

Sinan Kahraman, Emanuele Viterbo, and Mehmet E. Çelebi

***Abstract*—Polar coding is known as the first provably capacity-achieving coding scheme under low-complexity suboptimal successive cancelation decoding (SCD). The large error-correction capability of finite-length polar codes is mostly achieved with relatively long codes. SCD is the conventional decoder for polar codes and exhibits a quasi-linear complexity in terms of the code length. Practical decoder schemes with low latency are important for high-speed polar coding applications. In this letter, we propose a nonbinary multiple folded SCD scheme to reduce the decoding latency of standard binary polar codes. Multiple foldings were first proposed to improve the efficiency of folded tree maximum-likelihood decoder for Kronecker product-based codes. By successively applying the folding operation κ times on the SCD, for a code length N, the latency is reduced from $2N-1$ to $(N/2^{\kappa-1})-1$ time slots, assuming full parallelization. We show that multiple folded SCD can be effectively implemented for up to $\kappa=3$ foldings due to memory limitations. This decoder achieves exactly the same performance of the original SCD with significantly reduced latency.**

***Index Terms*—Polar codes, SC decoder, folding operation.**

I. Introduction

SHANNON's channel coding theorem proves the existence of capacity-achieving codes without providing an explicit construction [2]. The channel polarization phenomenon introduced in [1] makes the polar codes the first provable capacity-achieving coding scheme under a low complexity successive cancelation decoding (SCD) method, which exhibits a quasi-linear complexity in terms of the code length. It is interesting to note that the polar codes achieve capacity, as the length grows to infinity, even though SCD is a sub-optimal decoder, (i.e., not maximum likelihood). At finite lengths, the good error correction capability becomes significant only for relatively long polar codes where the implementation of SCD can become challenging due to complexity and latency. On the other hand, industrial predictions that are based on a well-known observation naming Moore's law show that transistor densities and counts in microprocessors double approximately every two or three years. Then it can be accepted that the latency is more critical issue than the space complexity issues. In this case, we consider an advanced scheme to reduce decoding latency using a new design achieving high parallelism at the cost of higher complexity and power consumption.

To overcome these limitations of polar coding, two different research directions have been undertaken. The first direction focuses on SCD implementations with a reduced complexity to extend the code length without major impact on performance. In [3]–[5] specific methods to speed up SCD in hardware have been proposed. In [3], a specific scheduling in the butterfly structure of SCD was presented to reduce complexity by the use of resource sharing. A semi-parallel decoder was proposed in [4] as a simple architecture for resource sharing with a small increase in latency. In [5], a decoding schedule of pre-computation look-ahead technique was introduced to reduce the latency of SCD by half.

In the second direction of research, higher complexity decoders have been proposed to improve the error performance of relatively shorter polar codes. For example, the sub-optimal performance of the SCD was improved by the list decoder in [6], the belief propagation in [7] and [8] and the stack algorithms in [9]. Moreover, optimal maximum-likelihood (ML) decoders have been studied in [10]–[12] for polar codes. In [11], the binary sphere decoding based ML decoder was proposed for short polar codes with code lengths up to 64. Recently, the folding operation applied to the ML tree search was used in [12] to design an efficient ML decoder based on a non-binary tree search strategy for longer Kronecker product based codes, such as polar and Reed–Muller codes of lengths up to 256.

In this letter, we apply the multiple folding operation to SCD to design a new low latency non-binary SCD for binary polar codes. We will refer to this as *multiple* or κ *folded SCD*. The butterfly structure is still preserved in the multiple folded SCD and hence the proposed method can be combined with the scheduling methods in [3]. We focus on decoding a standard polar code with frozen bits chosen according to the channel polarization. The proposed multiple folded SCD can also be used for Reed–Muller codes. Since it is known that Reed–Muller codes with SCD is very far from that of polar codes in terms of the error correction capability, we will not consider them in this letter. We show that using κ folding operations, the conventional SCD can be re-designed as a q-ary code SCD with $q=2^{2^\kappa}$ and length $N/2^\kappa$. The likelihood ratios used in the $(1+\log_2 N)$ steps of the conventional SCD architecture, are replaced by the conditional probabilities of the q-ary symbols grouping 2^κ bit using only $(1+\log_2(N/2^\kappa))$ steps in the multiple folded SCD. This provides a significant reduction of the decoder latency to $(N/2^{\kappa-1})-1$ time slots from $2N-1$ time slots under fully parallel decoder implementation for a code length N. A single folded SCD (i.e. for $\kappa=1$) was presented as a preliminary result in [13] and the dependence of the error performance on the alternative foldings was investigated. Here, we investigate the complexity of the proposed method in terms of the computational and memory requirements. Simulation results show that the proposed decoders can provide the same error performance as in the conventional SCD.

Manuscript received May 6, 2014; accepted July 21, 2014. Date of publication July 30, 2014; date of current version October 9, 2014. The work of S. Kahraman was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant 1059B141200235. The work of E. Viterbo was supported by the National Priorities Research Program (NPRP) under Grant NPRP5-597-2-241 from the Qatar National Research Fund (a member of Qatar Foundation). The associate editor coordinating the review of this paper and approving it for publication was M. Xiao.

S. Kahraman was with the Software Defined Telecommunications Laboratory, Monash University, Melbourne, VIC. 3800, Australia. He is now with Istanbul Technical University, Istanbul 34469, Turkey (e-mail: kahraman@ieee.org).

E. Viterbo is with Monash University, Melbourne, VIC. 3800, Australia (e-mail: emanuele.viterbo@monash.edu).

M. E. Çelebi is with Istanbul Technical University, Istanbul 34469, Turkey (e-mail: mecelebi@itu.edu.tr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LWC.2014.2343970

II. System Model

In this section, we consider the system model of polar codes in additive white Gaussian noise (AWGN) channel. Any given binary polar code with length N is uniquely defined by the number K of information bits and by the set of $N-K$ frozen bit indices $\mathcal{F} \subseteq \{0,1,\ldots,N-1\}$. A codeword is denoted by $\mathbf{x}=(x_0,\ldots,x_{N-1})^T$ and can be generated as from the information bits

$$\mathbf{x} = \mathbf{F}^{\otimes n}\mathbf{d}, \tag{1}$$

where the N dimensional vector $\mathbf{d}=(d_0,\ldots,d_{N-1})^T$ has $N-K$ frozen bits in positions $\mathcal{F}$ fixed to "0". The remaining K bits in vector $\mathbf{d}$ in the positions $\mathcal{F}^c=\{0,1,\ldots,N-1\}\setminus\mathcal{F}$, are used to transmit the K information bits. The frozen bit indices are selected as the least reliable bits after channel polarization and are determined by the polar code construction method, [1], [7]. The encoding matrix $\mathbf{F}^{\otimes n}$ is the n-fold iterated Kronecker product of the kernel matrix $\mathbf{F}=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}$. The transmission rate of the code will be $R=K/N$ which approaches the channel capacity as the code length tends to infinity.

We assume that the encoded bits x_k are mapped to binary antipodal modulation signals such that '1' $\rightarrow +1$, '0' $\rightarrow -1$ and the signal vector $\widetilde{\mathbf{x}}=(\tilde{x}_0,\ldots,\tilde{x}_{N-1})^T$ is transmitted over AWGN channel. The received noisy observations are given by the vector $\widetilde{\mathbf{y}}$ as

$$\widetilde{\mathbf{y}} = \widetilde{\mathbf{x}} + \mathbf{z}, \tag{2}$$

where $\mathbf{z}$ is the AWGN with zero mean and variance σ^2 and, for a given E_b/N_0 in dB, then $\sigma^2 = 1/(2R10^{[E_b/N_0]_{dB}/10})$.

Let $n=\log_2 N$ be the number of polarization steps and let $\hat{\mathbf{d}}$ be the estimated information bit vector. The conventional SCD estimates bits in the order $\alpha(0),\alpha(1),\ldots,\alpha(N-1)$, which depends on the bit-reversal operation of SCD architecture in [1]. For example, for a code length $N=8$ the order will be $\alpha=\{0,4,2,6,1,5,3,7\}$. Let $\hat{\mathbf{d}}_*^{(i)}$ be a partial estimate of $\mathbf{d}$ containing the partial decisions after the first i bit estimations. The remaining $(N-i)$ entries have not been determined yet and, at the end of the decoding procedure, the SCD decision will be $\hat{\mathbf{d}}=\hat{\mathbf{d}}_*^N$.

The conventional SCD algorithm in [1] is based on the successive estimations of bits in the desired vector, (i.e. $d_{\alpha(i)}$ for $i=0,\ldots,N-1$) using the received vector $\widetilde{\mathbf{y}}$, the frozen bits locations $\mathcal{F}$, and the previously estimated bit vector $\hat{\mathbf{d}}_*^{(i-1)}$.

The conditional probabilities for the $\alpha(i)$-th bit are denoted by $W_{\alpha(i)}^k(\widetilde{\mathbf{y}},\hat{\mathbf{d}}_*^{(i-1)}|d_{\alpha(i)}=0)$ and $W_{\alpha(i)}^k(\widetilde{\mathbf{y}},\hat{\mathbf{d}}_*^{(i-1)}|d_{\alpha(i)}=1)$ at step $k=0,\ldots,n$, which are successively computed in $(1+\log_2 N)$ steps k from 0 to n. If $d_{\alpha(i)}$ is a non-frozen information bit (i.e. $\alpha(i)\notin\mathcal{F}$), then the estimate is given by

$$\hat{d}_{\alpha(i)} = \begin{cases} 0, & \text{if } \dfrac{W_{\alpha(i)}^n\left(\widetilde{\mathbf{y}},\hat{\mathbf{d}}_*^{(i-1)}|0\right)}{W_{\alpha(i)}^n\left(\widetilde{\mathbf{y}},\hat{\mathbf{d}}_*^{(i-1)}|1\right)} \geq 1 \\ 1, & \text{otherwise.}\end{cases} \tag{3}$$

The main cause of the sub-optimality of SCD is the error propagation due to incorrect decisions.

III. Multiple Folded Successive Cancelation Decoding

Let us first consider the encoding (1) and note that it can be split into two $N/2$ dimensional equations in terms of $\mathbf{F}^{\otimes(n-1)}$, i.e.,

$$\mathbf{x} = \begin{bmatrix}\mathbf{F}^{\otimes(n-1)} & \mathbf{F}^{\otimes(n-1)}\\ \mathbf{0} & \mathbf{F}^{\otimes(n-1)}\end{bmatrix}\begin{bmatrix}\mathbf{d}'\\ \mathbf{d}''\end{bmatrix}, \tag{4}$$

where the vector $\mathbf{d}$ is split into $\mathbf{d}'=(d_0,\ldots,d_{N/2-1})^T$ and $\mathbf{d}''=(d_{N/2},\ldots,d_{N-1})^T$. This property was first observed by Dumer in [14] for Reed–Muller codes. Equivalently, considering the modulo-2 arithmetic, we have

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}'\\ \mathbf{x}''\end{bmatrix} = \begin{bmatrix}\mathbf{F}^{\otimes(n-1)}(\mathbf{d}'\oplus\mathbf{d}'')\\ \mathbf{F}^{\otimes(n-1)}\mathbf{d}''\end{bmatrix}. \tag{5}$$

Hence, we can consider the two binary polar codes with the code length $N/2$

$$\begin{cases}\begin{bmatrix}x_0\\ x_1\\ \vdots\\ x_{N/2-1}\end{bmatrix} = \mathbf{F}^{\otimes(n-1)}\begin{bmatrix}d_0\oplus d_{N/2}\\ d_1\oplus d_{N/2+1}\\ \vdots\\ d_{N/2-1}\oplus d_{N-1}\end{bmatrix}\\ \begin{bmatrix}x_{N/2}\\ x_{N/2+1}\\ \vdots\\ x_{N-1}\end{bmatrix} = \mathbf{F}^{\otimes(n-1)}\begin{bmatrix}d_{N/2}\\ d_{N/2+1}\\ \vdots\\ d_{N-1}\end{bmatrix}\end{cases} \tag{6}$$

where the first code encodes the information $\mathbf{d}'\oplus\mathbf{d}''$ and the second $\mathbf{d}''$. Here, the folding operation is based on considering non-binary bit pairs from $\mathbf{d}'\oplus\mathbf{d}''$ and $\mathbf{d}''$. It should be noted that the folding operation does not need any modification on the standard binary polar code nor its encoder and it only affects the decoder. In general, it can be shown that the pairs of bit indices which appear in $(\mathbf{d}'\oplus\mathbf{d}'')$ and $\mathbf{d}''$ have indices $\mathcal{I}_\ell = ((N/2)-\ell, N-\ell)$ for $\ell=1,\ldots,N/2$.

Moreover, any given polar code can be folded in alternative way by using suitable permutation matrices π_i, such that $\mathbf{F}^{\otimes n}=\pi_i^T\mathbf{F}^{\otimes n}\pi_i$ for $i=1,\ldots,n$. In fact, the encoding equation can be rewritten as $\mathbf{x}=\pi_i^T\mathbf{F}^{\otimes n}\pi_i\mathbf{d}$ and alternative encoding equations are given by suitably permuted vectors $\pi_i\mathbf{x}$ and $\pi_i\mathbf{d}$ such as $\pi_i\mathbf{x}=\mathbf{F}^{\otimes n}\pi_i\mathbf{d}$, using the property $\pi_i^{-1}=\pi_i^T$.

In order to describe suitable permutations, we use the *commutation matrix* $K(m,r)=\sum_{i=1}^m\sum_{j=1}^r(\mathbf{H}_{i,j}\otimes\mathbf{H}_{i,j}^T)$, where $\mathbf{H}_{i,j}$ is a $m\times r$ matrix with a "1" in its (i,j)th position and zeros elsewhere, [15]. Thanks to the permutation equivalent property in [16, Th. 9, p.47] we have $K(m,r)^T(\mathbf{A}\otimes\mathbf{B})K(m,r)=\mathbf{B}\otimes\mathbf{A}$, where $\mathbf{A}$ is $m\times m$, $\mathbf{B}$ is $r\times r$ matrices and $K(m,r)$ is the $mr\times mr$ commutation matrix. Then we can write $\mathbf{F}^{\otimes n}=K(2^{n-i},2^i)^T\mathbf{F}^{\otimes n}K(2^{n-i},2^i)$, for $i=1,\ldots,n$. Hence, the permutations $\pi_i=K(2^{n-i},2^i)$ for $i=1,\ldots n$ provide n alternative foldings.

Due to its fractal nature, $\mathbf{F}^{\otimes(n-1)}$ preserves the same structure of $\mathbf{F}^{\otimes n}$ and the folding operation can be repeated multiple times. In general, the folding operation can be successively applied for $1\leq\kappa\leq n-1$ times. The multiple folding operation $(\kappa\geq 2)$ was first introduced in [12] to implement the folded tree maximum-likelihood decoder for polar codes. In this study, we construct a non-binary multiple

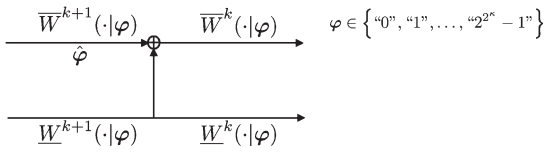

Fig. 1. Unit circuit for κ folded SCD for the 2^κ-bit symbols $\boldsymbol{\varphi}$.

folded SCD scheme with $1+\log_2(N/2^\kappa)$ steps, based on $\mathbf{F}^{\otimes(n-\kappa)}$ sub-blocks of size $N/2^\kappa$. In general, the set of indices of the group of 2^κ bits appearing in the ℓ-th non-binary level is given by $\mathcal{I}_\ell = ((N/2^\kappa)-\ell, (2N/2^\kappa)-\ell, \ldots, (2^\kappa N/2^\kappa)-\ell)$ for $\ell = 1, \ldots, N/2^\kappa$. A group of 2^κ bits corresponds to a q-ary symbol from the alphabet $\{0, 1, \ldots, 2^{2^\kappa}-1\}$ and will be denoted by $\boldsymbol{\varphi}$. In general, we write $\boldsymbol{\varphi} = \mathbf{F}^{\otimes\kappa}\mathbf{d}_{(\mathcal{I}_\ell)}$, where $\mathbf{d}_{(\mathcal{I}_\ell)}$ is the sub-vector of $\mathbf{d}$ corresponding to the indices $\mathcal{I}_\ell$. For example, for $\kappa = 2$, four bits are grouped in $\boldsymbol{\varphi} = \{(d_{(N/4)-\ell} \oplus d_{(N/2)-\ell} \oplus d_{(3N/4)-\ell} \oplus d_{N-\ell}), (d_{(N/2)-\ell} \oplus d_{N-\ell}), (d_{(3N/4)-\ell} \oplus d_{N-\ell}), (d_{N-\ell})\}$, where $\ell = 1, \ldots, N/4$.

A. Multiple Folded SCD Architecture

The bit decision rule in (3) is transformed to a decision on a group of 2^κ bits $\boldsymbol{\varphi}$ in (7),

$$\hat{\boldsymbol{\varphi}} = \arg\max_{\boldsymbol{\varphi}} \left\{ W_{\alpha(i)}^{n-\kappa} \left(\widetilde{\mathbf{y}}, \hat{\mathbf{d}}_*^{(2^\kappa \cdot i)} | \boldsymbol{\varphi} \right) \right\} \tag{7}$$

where $\hat{\mathbf{d}}_*^{(2^\kappa \cdot i)}$ is a binary vector containing previously $(2^\kappa \cdot i)$ estimated bits. It should be noted that $W_{\alpha(i)}^{k}(\widetilde{\mathbf{y}}, \hat{\mathbf{d}}_*^{(2^\kappa \cdot i)}|\boldsymbol{\varphi})$ is computed for all possible $q = 2^{2^\kappa}$ candidates of $\boldsymbol{\varphi}$ in the $k = 0, \ldots, n-\kappa$ steps. Fig. 1 shows the folded unit circuit where the conditional probabilities are successively computed from step k to step $k+1$ as $\boldsymbol{\varphi} : \{0, 1, \ldots, 2^{2^\kappa}-1\}$

$$\overline{W}^{k+1}(\cdot|\boldsymbol{\varphi}) = \sum_{\forall \boldsymbol{\psi}} \left(\overline{W}^{k}(\cdot|\boldsymbol{\psi}) \cdot \underline{W}^{k}(\cdot|\boldsymbol{\psi} \oplus \boldsymbol{\varphi}) \right), \tag{8}$$

where the sum is over all vectors $\boldsymbol{\psi}$ of 2^κ bits. Here, $\overline{W}$ and $\underline{W}$ denote upper and lower branches of the unit circuit. The second conditional probability is given by

$$\underline{W}^{k+1}(\cdot|\boldsymbol{\varphi}) = \overline{W}^{k}(\cdot|\boldsymbol{\varphi} \oplus \hat{\boldsymbol{\varphi}}) \cdot \underline{W}^{k}(\cdot|\boldsymbol{\varphi}), \tag{9}$$

where $\hat{\boldsymbol{\varphi}}$ denotes a previously decided symbol of the upper branch. In this case, the multiple folded-SC decoder only requires $(1+\log_2(N/2^\kappa))$ steps to decide for all the $N/2^\kappa$ non-binary symbols. The likelihood ratios cannot be used as in the binary case and 2^{2^κ} dimensional probability vectors need to be stored.

We can now describe the proposed decoding algorithm. In the initialization, κ is fixed to a number in the range 1 to $n-1$. Instead of N successive binary decisions, q-ary symbol decisions are made successively for $N/2^\kappa$ folded partial vectors $\boldsymbol{\varphi}$. Then, for each decided candidate $\hat{\boldsymbol{\varphi}}$ at the last step $(k = n-\kappa)$, the actual information bits can be computed as $\mathbf{d}_{(\mathcal{I}_\ell)} = \mathbf{F}^{\otimes\kappa}\hat{\boldsymbol{\varphi}}$. The proposed pseudocode is given in Table I.

For each successive q-ary $\boldsymbol{\varphi}$ decision, we need to compute the probabilities $W_{\alpha(i)}^{n-\kappa}(\cdot|\boldsymbol{\varphi})$ for all $\boldsymbol{\varphi}$ non-binary candidate symbols, as described in (8) and (9), by the use of all noisy observations $\widetilde{\mathbf{y}}$, frozen bits in $\mathcal{F}$ and previously made decisions $\hat{\boldsymbol{\varphi}}$.

TABLE I
MULTIPLE FOLDED-SC DECODER ALGORITHM

1: **for all** $i := 0, 1, \ldots, \frac{N}{2^\kappa} - 1$ **do**
2: **for all** $k := 0, 1, \ldots, n-\kappa$ **do**
3: calculate $W^k(\cdot|\boldsymbol{\varphi})$, for all $\boldsymbol{\varphi} \in \{0, \ldots, q-1\}$
4: **end for**
5: $\hat{\boldsymbol{\varphi}} := \arg\max_{\boldsymbol{\varphi}} \left\{ W_{\alpha(i)}^{n-\kappa}(\cdot|\boldsymbol{\varphi}) \right\}$
6: $\hat{\mathbf{d}}_{\alpha(\mathcal{I}_{i+1})} := \mathbf{F}^{\otimes\kappa}\hat{\boldsymbol{\varphi}}$
7: **if** there is any frozen bit in $\hat{\mathbf{d}}_{\alpha(\mathcal{I}_{i+1})}$ **then**
8: set the frozen bit $'0'$ in $\hat{\mathbf{d}}_{\alpha(\mathcal{I}_{i+1})}$
9: **end if**
10: **end for**

TABLE II
LATENCY, MEMORY AND COMPLEXITY REQUIREMENTS

	Latency		Memory		Cmp. Complexity	
κ	N=256	N=512	N=256	N=512	N=256	N=512
0	511	1023	511	1023	512	1024
1	255	511	765	1533	1024	2048
2	127	255	1905	3825	8192	16384
3	63	127	16065	32385	1048576	2097152
4	31	63	2×10^6	4×10^6	3.4×10^{10}	6.9×10^{10}

We should recall that some of the probabilities could be set to zero due to the broadcasted information of frozen bits and previous decisions. This process is accomplished by *line-(3)* in Table I. Then, the decision on the most likely $\boldsymbol{\varphi}$ can be taken by maximizing the computed probabilities as given by *line-(5)* in Table I. Thereafter, using the decision $\hat{\boldsymbol{\varphi}}$ made at stage $k = n-\kappa$, we can store the decision on the information bit values in the vector $\hat{\mathbf{d}}_{\alpha(\mathcal{I}_\ell(i))} = \mathbf{F}^{\otimes\kappa}\hat{\boldsymbol{\varphi}}$ as given by *line-(6)* in Table I. Hence, the current decisions can be passed to other levels in the non-binary folded-SCD. One can notice that the well-known butterfly structure of the SCD architecture is still preserved by the non-binary folded-SC decoder for $(1+\log_2(N/2^\kappa))$ steps. It should be noted that in the special case of $\kappa = 0$ the proposed decoder is identical to the conventional SCD.

We now compare the decoding latency of the conventional SCD in [1] to the latency of the κ folded SCD. In the construction of the conventional SCD, there are $(N/2)(1+\log_2 N)$ binary unit circuits and each has 2 processing elements (PEs), one for the upper and one for lower branches for the computation of $\overline{W}$ and $\underline{W}$ based on (8) and (9), respectively. On the other hand, the conventional SC decoding scheme can be implemented in $(1+\log_2 N)$ steps by the use of the butterfly structure and each i-th step has 2^{i-1} parallel PEs under the best possible parallelization [1]. Hence, the decoding latency of the conventional SCD is clearly given as $\sum_{i=1}^{(1+\log_2 N)} 2^{i-1} = 2N-1$.

With the κ folded SCD, only $(N/2^{\kappa+1})(1+\log_2(N/2^\kappa))$ unit circuits are used and each one has two PEs. Then the κ folded SCD has only $(1+\log_2(N/2^\kappa))$ steps, and hence its latency is given by $\sum_{i=1}^{(1+\log_2(N/2^\kappa))} 2^{i-1} = (N/2^{\kappa-1})-1$.

The multiple folded SCD requires to store the total number of conditional probabilities in the active branches in one time slot. The maximum number of active branches is $((N/2^{\kappa-1})-1)$ in the worst case (i.e. with code rate 1). Each active branch needs to store 2^{2^κ} conditional probabilities that can be normalized to store only $2^{2^\kappa}-1$ floats. Then the memory requirement is $((N/2^{\kappa-1})-1)(2^{2^\kappa}-1)$ floats. Table II shows the latency and memory and computational complexity requirements.

IV. Results and Discussion

In this section, we discuss complexity and error performance of the proposed multiple folded-SCD scheme.

The complexity of the κ folded SCD algorithm is determined by the computational complexity of the unit circuits and its memory requirements. It should be noticed that the main contribution of the proposed decoder is the requirement of a lower number of unit circuits with higher complexity, to reduce the latency up to 87%. The conventional SC decoder uses $(N/2)(1+\log_2 N)$ unit circuits with 4 multiplications to update the log-likelihood ratios. The proposed method with κ folding operations requires only $(N/2^{\kappa+1})(1+\log_2(N/2^{\kappa}))$ unit circuits and maximum $N/2^{\kappa+1}$ unit circuits are active in the same time slot with $2^{2^{\kappa+1}}$ multiplications to update the conditional probabilities of the q-ary symbols.

For $\kappa=2$ and $\kappa=3$ folding operations for SCD can be seen as an efficient tool to decrease the latency of the polar decoding at the cost of additional complexity and memory requirements. It can be seen that the unit circuits for $\kappa=4$ would need to compute and store $2^{2^4}-1=65.535$ conditional probabilities for each 2^{κ}-bit symbol $\boldsymbol{\varphi}$ in all active branches. The representation of conditional probabilities in a practical implementation needs to be more accurate for the case of a large number of κ. Hence, multiple folded SCD for $\kappa \geq 4$ would not be efficient.

Let us now consider the error performance of multiple folded SCD. In [13], it was shown that the choice of alternative foldings for any given polar code may be crucial on the decoding performance. In fact, some of the frozen information bits in $\mathbf{d}$ can be hidden in the folded group $\boldsymbol{\varphi}$ of 2^{κ} bits. For example, for $\kappa=2$ the four bit groups are $\boldsymbol{\varphi}=\{(d_{(N/4)-\ell}\oplus d_{(N/2)-\ell}\oplus d_{(3N/4)-\ell}\oplus d_{N-\ell}),(d_{(N/2)-\ell}\oplus d_{N-\ell}),(d_{(3N/4)-\ell}\oplus d_{N-\ell}),(d_{N-\ell})\}$, where $\ell=1,\ldots,N/4$. In some cases, frozen bits in $d_{(\mathcal{I}_\ell)}$ can not affect the conditional probabilities of the partial vector $\boldsymbol{\varphi}$. For example, when $d_{(3N/4)-1}$ is a frozen bit and the others are information bits in the group, there is no visible frozen bit in the group $\boldsymbol{\varphi}$. It can be seen that the hidden frozen bit, $d_{(3N/4)-1}$, can only be taken into account at the last step of the decoding scheme and it is not broadcasted to the other steps. To avoid this problem, the best alternative folding should be selected for a given polar code, so that all frozen bits are visible in $\boldsymbol{\varphi}$.

In the setup phase, we were able to test suitable alternative foldings that are used to provide folded groups of bits, where all frozen bits are visible (i.e. there are $N-K$ "folded frozen" bits in $N/2^{\kappa}$ groups with 2^{κ} folded bits) for rate 1/2 polar codes of lengths 256 and 512 that are optimized for $E_b/N_0=0$ dB. Simulation results in Fig. 2 show that bit error rate (BER) performances are the same under the conventional binary SCD and $\kappa=3$ folded SCD. Same results are obtained for $\kappa=1,2$.

V. Conclusion

To reduce the decoding latency of polar codes, we propose multiple folded SCD that is based on the folding operation of the Kronecker product based codes. In this way, the conventional SCD for a given polar code is re-designed with a non-binary architecture. The proposed decoding scheme has only $(1+\log_2(N/2^{\kappa}))$ steps when κ folding operations are applied. Note that no modification is needed at the encoder side. The decoding latency can be significantly reduced from $2N-1$ to $(N/2^{\kappa-1})-1$. By choosing the proper alternative folding, the same performance of the conventional SCD can be obtained.

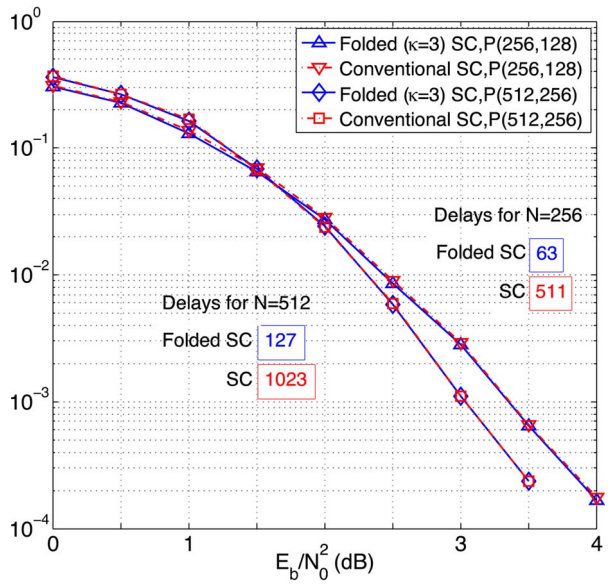

Fig. 2. BER vs. E_b/N_0 for conventional SCD and multiple folded SCD.

References

[1] E. Arıkan, "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, Jul. 2009.

[2] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948, 623–656.

[3] C. Leroux, I. Tal, A. Vardy, and W. J. Gross, "Hardware architectures for successive cancellation decoding of polar codes," in *Proc. IEEE ICASSP*, 2011, pp. 1665–1668.

[4] 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.

[5] C. Zhang, B. Yuan, and K. Parhi, "Reduced-latency SC polar decoder architectures," in *Proc. IEEE ICC*, Ottowa, ON, Canada, 2012, pp. 3471–3475.

[6] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, 2011, pp. 1–5.

[7] E. Arıkan, "A performance comparison of polar codes and Reed–Muller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447–449, Jun. 2008.

[8] N. Hussami, S. B. Korada, and R. Urbanke, "Performance of polar codes for channel and source coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Seoul, Korea, 2009, pp. 1488–1492.

[9] K. Chen, K. Niu, and J. Lin, "Improved successive cancellation decoding of polar codes," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3100–3107, Aug. 2013.

[10] E. Arıkan, K. Haesik, M. Garik, Ö. Üstün, and E. Efecan, "Performance of short polar codes under ML decoding," in *Proc. ICT Mobile Summit*, Santander, Spain, Jun. 10–12, 2009.

[11] S. Kahraman and M. E. Çelebi, "Code based efficient maximum-likelihood decoding of short polar codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, 2012, pp. 1967–1971.

[12] S. Kahraman, E. Viterbo, and M. E. Çelebi, "Folded tree maximum-likelihood decoder for Kronecker product-based codes," in *Proc. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, 2013, pp. 629–636.

[13] S. Kahraman, E. Viterbo, and M. E. Çelebi, "Folded successive cancelation decoding for polar codes," in *Proc. AusCTW*, Sydney, NSW, Australia, 2014, pp. 57–61.

[14] I. Dumer and K. Shabunov, "Soft decision decoding of Reed–Muller codes: Recursive lists," *IEEE Trans. Inf. Theory*, vol. 53, no. 3, pp. 1260–1266, Mar. 2006.

[15] J. R. Magnus and H. Neudecker, "The commutation matrix: Some properties and applications," *Ann. Statist.*, vol. 7, no. 2, pp. 381–394, Mar. 1979.

[16] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus With Applications in Statistics and Econometrics*. Hoboken, NJ, USA: Wiley, 1988.