SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

October 2020, Vol. 63 202303:1–202303:19
https://doi.org/10.1007/s11432-019-2924-6

Efficient stochastic successive cancellation list decoder for polar codes

Xiao LIANG[1,2,3,4†], Huizheng WANG[1,2,3,4†], Yifei SHEN[1,2,3,4], Zaichen ZHANG[1,2,3,4], Xiaohu YOU[1,2,3,4] & Chuan ZHANG[1,2,3,4*]

[1]*Laboratory of Efficient Architectures for Digital-communication and Signal-processing (LEADS), Southeast University, Nanjing* 211100, *China;*
[2]*National Mobile Communications Research Laboratory, Southeast University, Nanjing* 211100, *China;*
[3]*Quantum Information Center of Southeast University, Nanjing* 211100, *China;*
[4]*Purple Mountain Laboratories, Nanjing* 211100, *China*

Received 29 December 2019/Revised 13 March 2020/Accepted 22 May 2020/Published online 21 September 2020

Abstract Polar codes are one of the most favorable capacity-achieving codes owing to their simple structures and low decoding complexity. Successive cancellation list (SCL) decoders with large list sizes achieve performances very close to those of maximum-likelihood (ML) decoders. However, hardware cost is a severe problem because an SCL decoder with list size L consists of L copies of a successive cancellation (SC) decoder. To address this issue, a stochastic SCL (SSCL) polar decoder is proposed. Although stochastic computing can achieve a good hardware reduction compared with the deterministic one, its straightforward application to an SCL decoder is not well-suited owing to the precision loss and severe latency. Therefore, a doubling probability approach and adaptive distributed sorting (DS) are introduced. A corresponding hardware architecture is also developed. Field programmable gate array (FPGA) results demonstrate that the proposed stochastic SCL polar decoder can achieve a good performance and complexity tradeoff.

Keywords SCL polar decoder, stochastic computing, 2-bit decoding, distributed sorting, hardware

Citation Liang X, Wang H Z, Shen Y F, et al. Efficient stochastic successive cancellation list decoder for polar codes. Sci China Inf Sci, 2020, 63(10): 202303, https://doi.org/10.1007/s11432-019-2924-6

1 Introduction

Polar codes, proposed by Arıkan's breakthrough paper [1], are an exciting new class of channel codes that can asymptotically achieve the capacity for symmetric binary-input discrete memoryless channels. Because of its FFT-like structure and low complexity with $\mathcal{O}(N \log N)$ where N denotes code length, successive cancellation (SC) decoding algorithm has become one of the most popular polar decoding algorithms. Nevertheless, it cannot be denied that compared to maximum likelihood (ML) decoder [2], the decoding performance of SC polar decoder still suffers from an evident degradation. To narrow the performance gap caused by the sub-optimality of traditional successive cancellation decoder, the successive cancellation list (SCL) polar decoding algorithm is developed in [2,3]. Simulation results have revealed that SCL polar decoder can outperform the low density parity check (LDPC) codes even within high error-rate regions. To realize the spectrum efficiency and low-latency required by next generation wireless system, Refs. [4,5] proposed two latency-reduced SCL decoders for polar codes, respectively.

The main drawback of SCL polar decoder is its complexity increases linearly with the list size L. To realize satisfied performance, the SCL polar decoder suffers from high complexity cost, especially when

* Corresponding author (email: chzhang@seu.edu.cn)
† Xiao LIANG and Huizheng WANG have the same contribution to this work.

L is considerably large. According to the simulation over the BPSK-AWGN channel [2], it is required $L \geqslant 32$ to achieve frame error rate (FER) less than 10^{-5} when cyclic redundancy check (CRC) is 16 bits and the signal to noise (SNR) is 2 dB, where L denotes list size. Owing to the complexity $\mathcal{O}(LN \log N)$ of SCL decoding, large L will introduce huge complexity that obstructs the efficient implementation of SCL decoder. To address the issue, a stochastic computing based SCL polar decoding algorithm is proposed in this paper.

Stochastic computing [6], as an approximate computing technique, has the potential to provide significantly low hardware footprint with high energy efficiency and scalability [7]. In stochastic calculation framework, a probability number is represented by a bit-stream, hence the fundamental arithmetic operations such as additions and multiplications can be realized as simple as multiplexers (MUX) and AND gates, respectively [8]. Moreover, even some complex arithmetic operations can also be implemented with very elementary hardware logic [9,10]. In [11], a fully parallel stochastic Markov Chain Monte Carlo (MCMC) multiple-input and multiple-output (MIMO) detector is developed, which can achieve a higher throughput with lower hardware cost compared with conventional MIMO detectors. Further, the corresponding design supporting iterative detection is proposed in [12]. Considering the low-complexity and extensive computation in SCL decoding algorithm, stochastic computing offers a colossal design space for SCL optimization owing to its advantages in area reduction and soft error resiliency.

In [13], stochastic computing is incorporated with SCL decoder with $L = 1, 2$. Thanks to stochastic computing, the stochastic SCL (SSCL) decoder is expected to be an efficient paradigm which can realize complexity reduction, satisfied performance, and fault tolerance. Nevertheless, applying stochastic computing to SCL decoder directly will lead to unacceptable performance degradation and latency. This paper devotes itself in proposing solutions to reduce SCL decoder hardware complexity. The major novelties of this paper are summarized as follows. First, channel message scaling and enlarging probability (EP) are proposed to mitigate the randomness loss of stochastic computing. Second, to reduce decoding latency, we propose a stochastic based double level (DL) decoding scheme that can estimate two bits each time. Furthermore, 2^p-level decoding method is developed to realize arbitrary tradeoff between performance and latency. Third, based on distributed sorting (DS), we develop an adaptive DS (ADS) scheme that can compromise different sorting scales in stochastic multiple-level decoding. Finally, the detailed hardware architecture for the proposed stochastic SCL decoder is developed. To the best of our knowledge, this is the first paper to implement stochastic SCL decoder to field programmable gate array (FPGA) platform. Compared with the deterministic-based decoding framework, it shows advantages in the balance between performance and complexity.

The remainder of this paper is organized as follows. Section 2 re-explains the SSCL decoding algorithm with $L = 1$ that is proposed by [13] firstly. Section 3 presents the SSCL decoding with $L = 2$. Further, the SSCL decoding with $L = 2^p$ is presented in Section 4. The implementation of SSCL decoder is developed in Section 5. Comparison and conclusion are presented in Sections 6 and 7.

2 Stochastic SCL decoder for polar codes with single-level decoding method

2.1 Deterministic SCL decoding

We use the notation a_1^N as shorthand for denoting a row vector $(a_1, \ldots, a_N)$. Given a vector a_1^N, we write a_i^j, $1 \leqslant i, j \leqslant N$, to denote the subvector $(a_i, \ldots, a_j)$; if $j < i, a_i^j$ is regarded as void. Given a_1^N and $\mathcal{A} \subset \{1, \ldots, N\}$, we write $a_{\mathcal{A}}$ to denote the subvector $(a_i : i \in \mathcal{A})$.

A polar code can be defined by parameters $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ [1], where N, K, $\mathcal{A}$, and $u_{\mathcal{A}^c}$ are the code length, the number of information bits, the set of information bits, and the frozen bit whose value is always zero, respectively. The transition probabilities of binary-input discrete memoryless channels (B-DMCs) are defined as $W(y|x)$, and for simplicity, let (N, K) denote a polar code in this study.

For a polar encoding process, we denote $n = \log_2 N$ and $\boldsymbol{F} \triangleq \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$. Then the codeword $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]$ can be generated from $\boldsymbol{x} = \boldsymbol{F}^{\otimes n}\boldsymbol{u}$. For SC decoding, the estimated bit sequence is defined as $\hat{\boldsymbol{u}} = [\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_N]$. If u_i is a frozen bit, we can simply assign $\hat{u}_i = 0$. Otherwise, the decoded bit can

be determined by

$$\hat{u}_i = \begin{cases} 0, \text{ if } W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0) \geqslant W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1), \\ 1, \text{ if } W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0) < W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1), \end{cases} \tag{1}$$

where $W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|u_i = \hat{u}_i)$ is defined as the bit channel transition probability, $(y_1^N, \hat{u}_1^{i-1})$ denotes the output of $W_N^{(i)}$ and u_i is input. The deterministic successive cancellation decoder calculates this log-likelihood ratio (LLR) by recursively utilizing two basic processing nodes, namely f node and g node, whose functions are shown as

$$f(a,b) = \frac{1+ab}{a+b}, \quad g(a,b,\hat{u}_{\text{sum}}) = a^{1-2\hat{u}_{\text{sum}}}b, \tag{2}$$

where $a, b, \hat{u}_{\text{sum}} \in \{0,1\}$.

Regarded as the generalization of the SC decoding, the SCL decoding algorithm was firstly proposed by [3, 14]. In SC decoding, only the most likely estimation code bit can survive. Whenever a certain bit is incorrectly decoded, not only this bit decoding fails, but the wrong feedback information also causes influence on the following bit decoding. To this end, the SCL decoder utilizing the similar algorithm as K-best MIMO detection [15] always expands a list of L code bits at each step on the full binary-tree, and outputs the most likely codeword when the last step is completed. The expanding decoding steps of SCL algorithm with list size $L = 2$ is illustrated in Figure 1. The SCL decoder improves the performance at the expense of increasing hardware. One possible solution to lower the hardware complexity of basic processing nodes is to implement the algorithm with stochastic computing.

2.2 Stochastic SCL decoding

Stochastic decoding utilizes bit-streams that are composed of '0' and '1' to represent a number rather than employs weighted sum to represent values like conventional deterministic decoding. In stochastic computing, the portion of 1's in the whole bitstream denotes corresponding value. The first stochastic SC decoder for polar codes was proposed by [16], which mentioned that when the normalization processing is applied, the bit conditional probability denoted in (3) can be utilized as the decision metric for SC polar decoder.

$$p(u_i = \hat{u}_i|\hat{u}_1^{i-1}) = \begin{cases} \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|u_i=\hat{u}_i)}{\sum\limits_{u_i\in\{0,1\}} W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|u_i)}, & \text{if } i \in \mathcal{A}. \\ \mathbf{1}_{u_i=0}, & \text{if } i \in \mathcal{A}^c. \end{cases} \tag{3}$$

According to the definition of likelihood ratio (LR) [16], $a = \frac{\Pr(A=0)}{\Pr(A=1)}$ and $b = \frac{\Pr(B=0)}{\Pr(B=1)}$, the equation of f node can be written as

$$f(a,b) = \frac{1+ab}{a+b} = \frac{\Pr(A=1)\Pr(B=1) + \Pr(A=0)\Pr(B=0)}{\Pr(A=0)\Pr(B=1) + \Pr(A=1)\Pr(B=0)} = \frac{\Pr(F=0)}{\Pr(F=1)}, \tag{4}$$

where $\Pr(F=0) + \Pr(F=1) = 1$. Therefore, we can get the equation $\Pr_f = \Pr(F=1) = (1 - \Pr(A = 1))\Pr(B=1) + \Pr(A=1)(1-\Pr(B=1))$.

To derive stochastic version of g node equation, we first set $\hat{u}_{\text{sum}} = 0$, then,

$$g(a,b,0) = ab = \frac{\frac{\Pr(A=0)\Pr(B=0)}{\Pr(A=1)\Pr(B=1)+\Pr(A=0)\Pr(B=0)}}{\frac{\Pr(A=1)\Pr(B=1)}{\Pr(A=1)\Pr(B=1)+\Pr(A=0)\Pr(B=0)}} = \frac{\Pr(G=0)}{\Pr(G=1)}, \tag{5}$$

where both numerator and denominator are divided by $\Pr(A = 1)\Pr(B = 1) + \Pr(A = 0)\Pr(B = 0)$ in order to meet the requirement that $\Pr(G = 0) + \Pr(G = 1) = 1$. Then we can get $\Pr_g = \Pr(G = 1) = \frac{\Pr_a\Pr_b}{(1-\Pr_a)(1-\Pr_b)+\Pr_a\Pr_b}$. Secondly, we set $\hat{u}_{\text{sum}} = 1$, similarly we can get $\Pr_g = \Pr(G = 1) = \frac{(1-\Pr_a)\Pr_b}{\Pr_a(1-\Pr_b)+(1-\Pr_a)\Pr_b}$. In this instance, if we define $\boldsymbol{b} = \mathbf{1}_{\hat{u}_{\text{sum}}=0}$, the stochastic form (2) can be

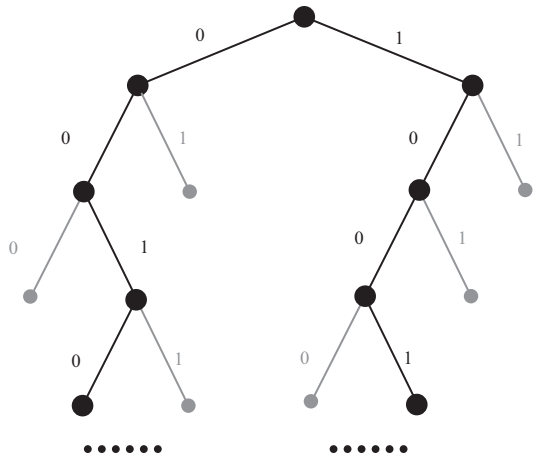

Figure 1 SCL decoding steps with $L = 2$.

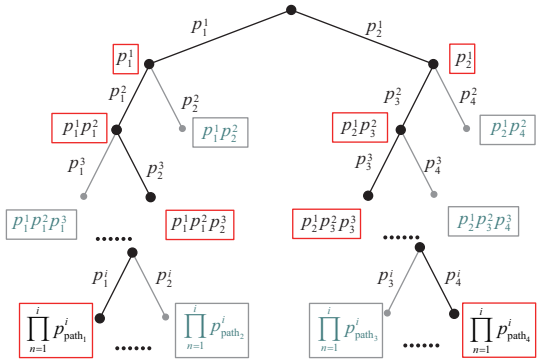

Figure 2 (Color online) Stochastic SCL decoding with $L = 2$.

rewritten as

$$\mathrm{Pr}_f = \mathrm{Pr}_a(1-\mathrm{Pr}_b) + \mathrm{Pr}_b(1-\mathrm{Pr}_a), \tag{6}$$

$$\mathrm{Pr}_g = \frac{(\boldsymbol{b}\mathrm{Pr}_a + (1-\boldsymbol{b})(1-\mathrm{Pr}_a))\mathrm{Pr}_b}{(1-\boldsymbol{b})\mathrm{Pr}_f + \boldsymbol{b}(1-\mathrm{Pr}_f)}. \tag{7}$$

From (6), bit conditional probabilities in SC decoding can be updated by recursively applying

$$\begin{cases} \mathrm{Pr}_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}) = \mathrm{Pr}_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})\big(1-\mathrm{Pr}_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})\big) \\ \qquad\qquad + \big(1-\mathrm{Pr}_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})\big)\mathrm{Pr}_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}). \\ \mathrm{Pr}_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = \frac{((1-\hat{u}^{2i-1})\mathrm{Pr}_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) + \hat{u}^{2i-1}(1-\mathrm{Pr}_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})))\mathrm{Pr}_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})}{\hat{u}^{2i-1}\mathrm{Pr}_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}) + (1-\hat{u}^{2i-1})(1-\mathrm{Pr}_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}))}. \end{cases} \tag{8}$$

Please note that the two formulas only calculate the bit conditional probability instead of the path probability $P(\hat{u}_1^i)$. After calculating bit conditional probability as denoted in

$$p(u_i = \hat{u}_i | \hat{u}_1^{i-1}) = \begin{cases} \mathrm{Pr}_N^i, & \text{if } i \in \mathcal{A} \text{ and } \hat{u}_i = 1, \\ 1-\mathrm{Pr}_N^i, & \text{if } i \in \mathcal{A} \text{ and } \hat{u}_i = 0, \\ \mathbf{1}_{u_i=0}, & \text{if } i \in \mathcal{A}^c, \end{cases} \tag{9}$$

in order to obtain path probability as the path metric, all bit conditional probability in one path $[p(u_1 = \hat{u}_1), \ldots, p(u_N = \hat{u}_N)]$ has to be multiplied as

$$P(\hat{u}_1^i) = P(\hat{u}_1^{i-1})p(u_i = \hat{u}_i | \hat{u}_1^{i-1}) = \prod_{n=1}^{i} p(u_n = \hat{u}_n | \hat{u}_1^{n-1}). \tag{10}$$

Take the SSCL decoding with $L = 2$ depicted in Figure 2 as an example. At each level, SCL decoding tree expands paths and updates path probabilities, and then selects paths with the largest L path probabilities instead of only keeping the best path. In general, at the i-th decoding level of SSCL decoder, a total of $2L$ ordered probabilities $P_{\{1,\ldots,2L\}}(\hat{u}_1^i)$ are calculated as the metric to select the L most probable candidates.

Therefore, the path probabilities at the i-th decoding level can be calculated through multiplying the $(i-1)$-th level's path probabilities with the corresponding ordered bit conditional probability. Revealed by (10), to implement path probabilities, only one two-input AND gate is needed for multiplication,

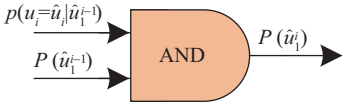

Figure 3 (Color online) The architecture for generating path probability in stochastic SCL decoding.

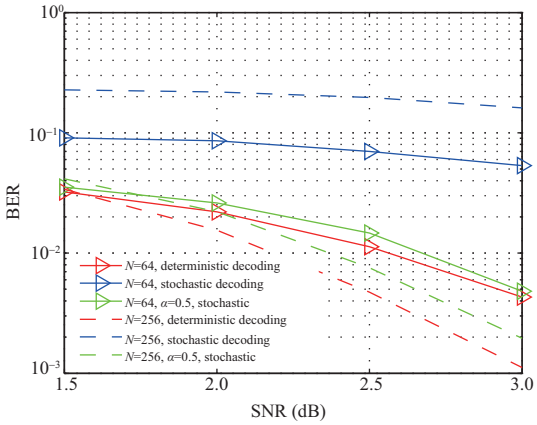

Figure 4 (Color online) Comparison of different SC polar decoders.

with the corresponding architecture depicted in Figure 3. Moreover, the calculation of the conditional probability of each corresponding ordered bit entails the recursive operation between f node and g node. In the successive way, the most probable L candidates in each stage can always be kept and the best one at the last stage can also be selected.

2.3 Channel message scaling

As illustrated in Figure 4, compared with the deterministic design, the decoding performance of stochastic SC suffers from a sharp degradation. The reason is that the channel message represented by bitstreams will lose the randomness in high SNR regions. For stochastic LDPC decoders, a new and essential scaling method for stochastic decoders was proposed in [17], which is used to provide a similar level of switching activity over different ranges of SNRs. This method can efficiently improve bit error rate (BER) performance for stochastic decoders at a high level.

In the scaling method, the received channel reliabilities are scaled by a factor that is proportional to the noise in channel. The scaled LLRs calculated by

$$\mathrm{LR}(y_i) = \log\left(\frac{\Pr\left(x_i = +1|y_i\right)}{\Pr\left(x_i = -1|y_i\right)}\right) \tag{11}$$

are independent towards channel noise. That is to say, the decoder does not need to estimate the noise in the channel. Furthermore, according to the work in [17], the scaled LLR for the i-th received symbol is $\mathrm{LR}'(y_i) = \alpha N_0 \mathrm{LR}(y_i)$, where the initial channel likelihood ratio can be denoted by $\mathrm{LR}(y_i) = 4y_i/N_0$, and N_0 is the single-sided noise power density. The channel message scaling approach introduces a scaling coefficient α and the scaled channel likelihood ratio becomes $\mathrm{LR}'(y_i) = 4\alpha y_i$, which is now used for generating the input stochastic data bitstreams for SC polar decoders. The input probability can therefore be rewritten as

$$\Pr(y_i = 1) \approx \frac{1}{\mathrm{e}^{-\mathrm{LR}(y_i)} + 1} = \frac{1}{\mathrm{e}^{-4\alpha y_i} + 1}. \tag{12}$$

Considering that the choices of α would result in different decoding performances and previous work [18] has proven the best performance of stochastic SC decoder can be achieved with $\alpha = 0.5$, the scaling factor $\alpha = 0.5$ is employed in all numerical simulations in the following part of this paper.

To testify the effectiveness of the channel message scaling method, simulation comparisons of deterministic design, stochastic design, and stochastic design with scaling factor $\alpha = 0.5$ are depicted in Figure 4. Simulation results for both $(64, 32)$ code and $(256, 128)$ code have demonstrated that the stochastic SC decoders with the scaling approach achieve improvement over the ones without.

2.4 Enlarging probability approach

The decision metric in SSCL decoder is denoted as the path probability $P(\hat{u}_1^i)$ calculated by (10) rather than $p(u_i = \hat{u}_i|\hat{u}_1^{i-1})$. Especially, $P(\hat{u}_1^i)$ denotes the multiplicative of i ordered bit conditional probabilities, and all the values of bit conditional probability are smaller than one. For stochastic decoding, path probability $P(\hat{u}_1^i)$ is denoted as bit 1's occurrence probability of a fixed-length binary bit-stream, and then the value of $P(\hat{u}_1^i)$ keeps shrinking with the increase of decoding level i. Therefore, $P(\hat{u}_1^i)$ will become considerable small in stochastic decoding process, and then the bitstream will lose its randomness characteristic and cause performance degradation.

To improve the randomness loss of path probability $P(\hat{u}_1^i)$ in SSCL decoding process, an efficient approach named enlarging probability is developed in [19]. Before all L path probabilities $P(\hat{u}_1^i)$ become too small, all path probability values are re-scaled by an identical product factor β, and therefore we still keep the proportional relation between L path probabilities and ensure their randomness characteristic. For instance, if all L candidates for SSCL decoder are smaller than 0.5, we set $\beta = 2$ otherwise $\beta = 1$. Therefore, the path probabilities of the L most possible candidates are always remained in the region of $(0.5, 1]$.

2.5 Distributed sorting

The complexity of path probabilities' comparison and memory management grows drastically with L. Thus, SSCL decoder was designed to reduce the heavy consumption. However, it is difficult to sort directly by stochastic computation representation, in other words, the stochastic data streams of path probabilities should be changed into deterministic data. Here, DS [20] is developed for path selection.

In order to change stochastic data streams into deterministic values, a counter is applied after the path probability of each candidate is calculated. The counter computes the number of '1' in stochastic streams, and this number is denoted as the path probability metric (PPM). Then paths with the largest L PPM values are selected.

For clarity, children from the same father node with larger and smaller PPM are denoted by the first child (FC) and the next child (NC), respectively. In traditional direct sorting algorithms, such as insertion and bubble sorting, when the m-th maximum NC node and the m-th minimum FC node are found in the m-th round, top m maximum NCs are all compared to the m-th minimum FC value. In DS algorithm, this step can be simplified to only compare the m-th maximum NC and the m-th minimum FC, and then the comparison result decides whether the next round of comparison needs to be conducted. The DS reduces the comparison complexity from $O(L^2)$ to $O(L)$, and an upper rounds coefficient k is set to reduce the latency from kL^2 to kL [20]. By employing DS algorithm, the selection of candidate paths is depicted in Figure 5. In our stochastic decoding system, L FC PPM values are always superior to L NC PPM values in most levels on the binary-tree, that is to say, when coefficient k is canceled, the average round's number of comparison required is still quite small and the DS method will have no performance gap compared to direct sorting.

For stochastic polar SCL decoder, the scaling factor is set as $\alpha = 0.5$, the bitstream length is chosen as $l = 1024$, and doubling probability approach is employed in the stochastic system. Figure 6 shows the numerical results for different decoder with $(256, 128)$ codeword. From Figure 6, it can be observed that the performance of DS-based SSCL decoding with list size $2L$ is similar to the performance of deterministic SCL decoding with list size L.

The latency of DS algorithm is relevant to the average rounds of comparison. For $(256, 128)$ polar decoder with $L = 8$, the average comparison rounds of DS are illustrated in Figure 7. When SNR is in $[1.5, 3.0]$, the average rounds are no larger than 1.76, hence the low latency of DS algorithm is guaranteed.

3 Stochastic SCL polar decoder with a double-level decoding method

The biggest problem of SSCL decoder is decoding latency. Single-level decoding mentioned in Section 2 for stochastic polar code makes every bit decoding consume too many clock periods, which almost equals

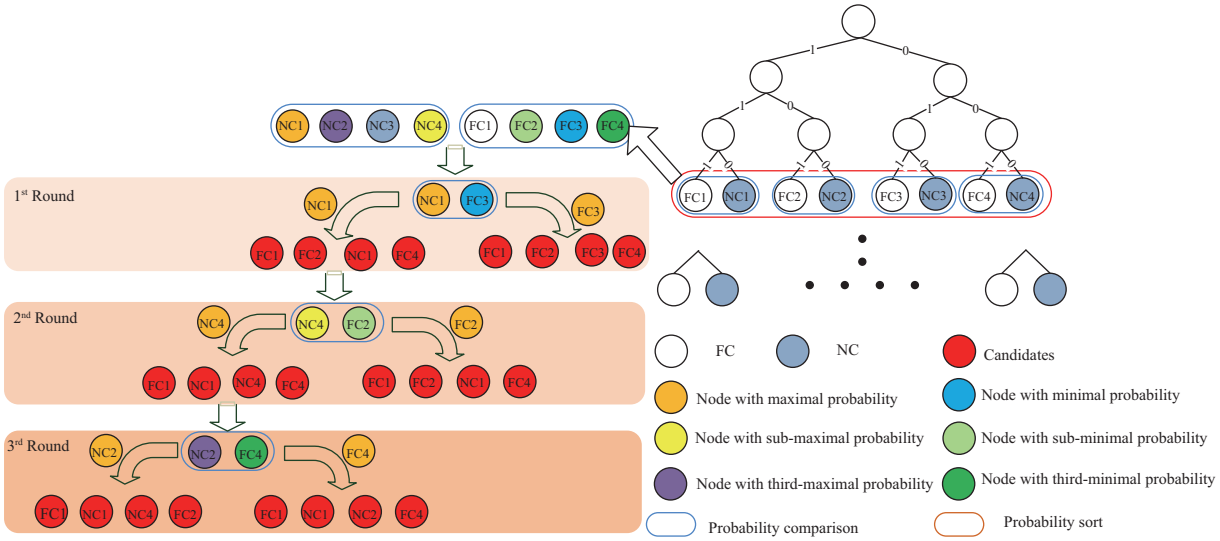

Figure 5 (Color online) Candidate paths' selection in polar SCL decoder ($L = 4$).

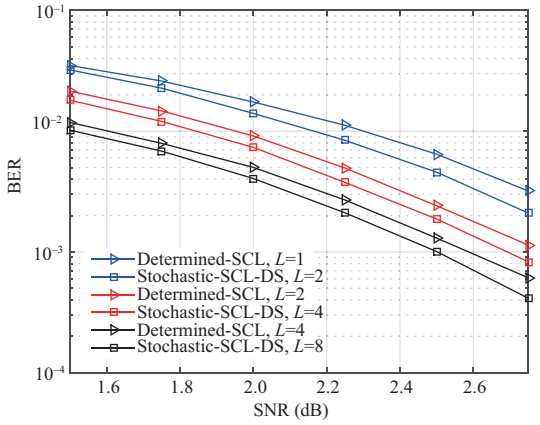

Figure 6 (Color online) Performance comparison of (256, 128) deterministic and stochastic decoders with single-level DS ($l = 1024$).

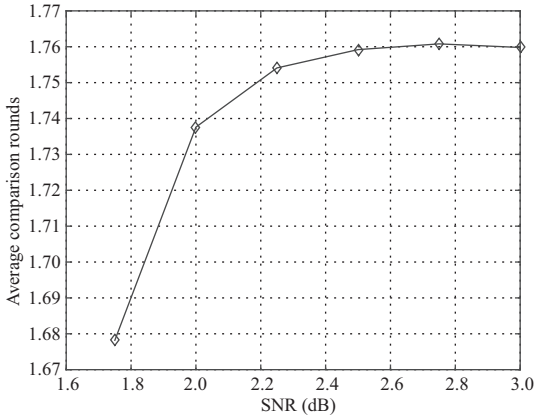

Figure 7 Comparisons in (256, 128) stochastic SCL decoder with single-level DS decoding ($L = 8$, $l = 1024$).

to the length of stochastic bitstream. To deal with the dilemma, in this section, double-level decoding method is proposed to reduce decoding latency. Each time we estimate two bits no matter they are information bits or frozen bits, and then utilize the feedback signal to calculate the next two bits codes.

3.1 Stochastic SCL decoder in double-level tree

According to (8), when $\mathrm{Pr}_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})$ and $\mathrm{Pr}_{N/2}^{(i)}(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2})$ have been calculated, $\mathrm{Pr}_{N}^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2})$ can be computed. Then after estimating $\hat{u}^{2i-1}$, $\mathrm{Pr}_{N}^{(2i)}(y_1^N, \hat{u}_1^{2i-1})$ can be calculated. It is notable that $\hat{u}^{2i-1}$ only has two possible values '1' or '0', and hence $\mathrm{Pr}_{N}^{(2i)}(y_1^N, 0)$ and $\mathrm{Pr}_{N}^{(2i)}(y_1^N, 1)$ can be pre-computed before estimating $\hat{u}^{2i-1}$. Stochastic streams $\mathrm{Pr}_{N}^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2})$, $\mathrm{Pr}_{N}^{(2i)}(y_1^N, 0)$ and $\mathrm{Pr}_{N}^{(2i)}(y_1^N, 1)$ can output simultaneously, that is to say, two neighbouring bit conditional probabilities $p(u_{2i-1} = \hat{u}_{2i-1}|\hat{u}_1^{2i-2})$ and $p(u_{2i} = \hat{u}_{2i}|\hat{u}_1^{2i-1})$ can be output at the same time. The path probability (10) is rewritten in the double-level scheme:

$$P(\hat{u}_1^{2i}) = P(\hat{u}_1^{2i-2})p(u_{2i-1} = \hat{u}_{2i-1}|\hat{u}_1^{2i-2})p(u_{2i} = \hat{u}_{2i}|\hat{u}_1^{2i-1}) = \prod\nolimits_{n=1}^{2i} p(u_n = \hat{u}_n|\hat{u}_1^{n-1}). \tag{13}$$

This double-level method is very similar to the pre-computation scheme in deterministic SC de-

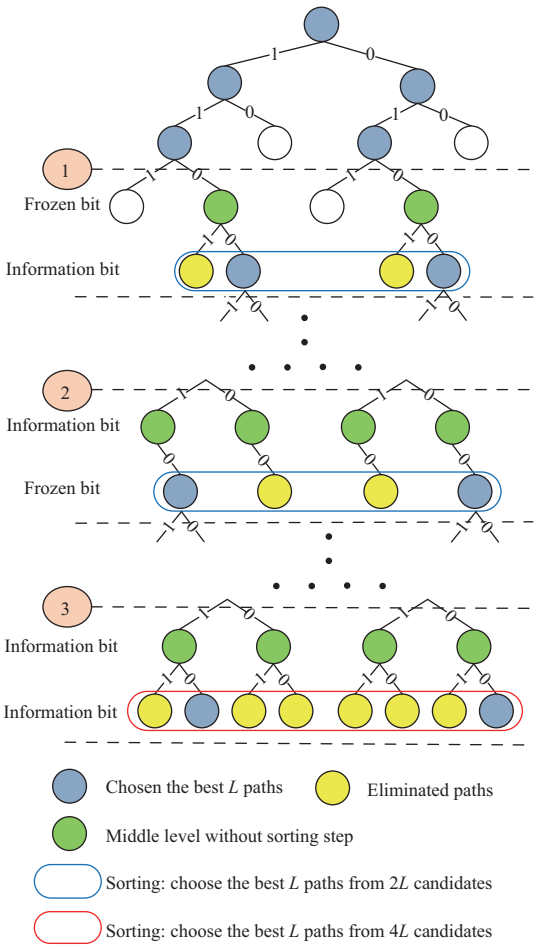

Figure 8 (Color online) Binary-tree of stochastic SCL decoder in double-level ($L = 2$).

coding [21]. Ref. [21] proposed a pre-computation scheme in all $\Pr_j^{(2i)}(y_1^j, \hat{u}_1^{2i-1})$ calculation when $j = 1, 2, \ldots, N/2, N$, in order to reduce one clock period when estimating one bit value in deterministic decoding. In stochastic calculation, our proposed double-level scheme only employs pre-computation in $\Pr_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1})$ to ensure two neighbouring bit conditional probabilities $p(u_{2i-1} = \hat{u}_{2i-1}|\hat{u}_1^{2i-2})$ and $p(u_{2i} = \hat{u}_{2i}|\hat{u}_1^{2i-1})$ output simultaneously. For decoding each bit conditional probability needs almost l clock periods, the double-level scheme can save $Nl/2$ clock periods. In stochastic decoding, pre-computation in other stages of $\Pr_j^{(2i)}(y_1^j, \hat{u}_1^{2i-1})$ calculation when $j = 1, 2, \ldots, N/2$ is meaningless.

Double-level decoding scheme is illustrated in Figure 8. For each father node, there are two children at the first level (the $(2i-1)$-th level) and four sub-children (the children of children nodes) at the second level (the $2i$-th level). According to different distributions of information bits and frozen bits, there are three cases in the double-level scheme. As shown in Figure 8, the first and second cases have a frozen bit and an information bit in two neighbouring levels, then DS is utilized to select paths with largest L PPM values from $2L$ children. The third case has two information bits in both levels, and then an adaptable DS is employed to select L candidates from $4L$ children. Details of adaptable DS are in Subsection 3.2. In double-level SCL decoding, every two levels need one sorting process.

3.2 Adaptable distributed sorting in double-level method

The ADS method is proposed to meet the sorting requirement for stochastic double-level SCL decoding. Employing the adaptable DS algorithm, the selection process of candidate paths from $4L$ to L is illustrated in Figure 9. From one father node, four sub-children are divided into one FC and three NCs. The ADS algorithm is composed of at most $L-1$ rounds of comparison. Hence, it is dynamic and approximate to

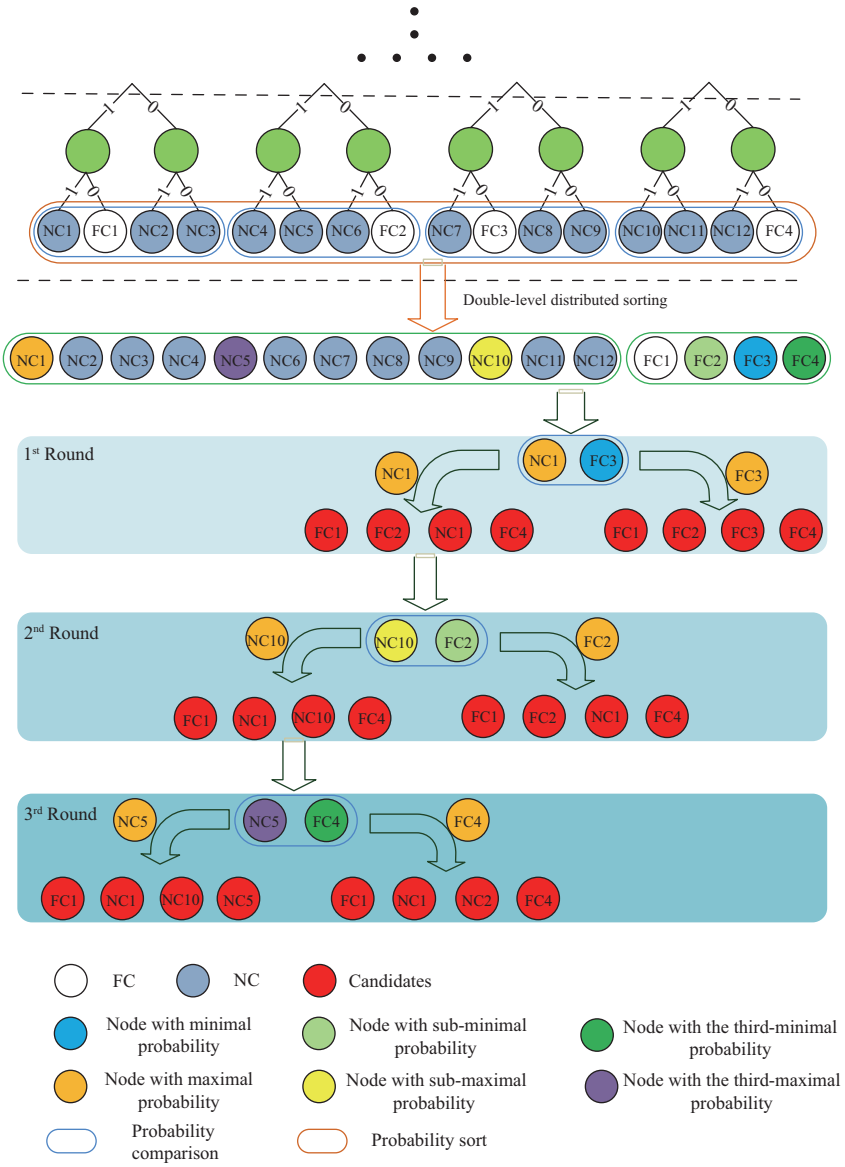

Figure 9 (Color online) Candidate paths' selection in polar SCL decoder with double-level ($L = 4$).

DS. Firstly, all L FCs are picked out, and the remainders are $3L$ NCs. In the first round comparison, NC with maximal PPM (NC1) and FC with minimal PPM (FC3) are chosen out (requiring $4L - 2$ comparison times), and then compare NC1 and FC3 (requiring 1 comparison time). If the PPM of FC3 is larger, the sorting will be finished with the selected candidates of FC1, FC2, FC3, and FC4. In this case, only one round of comparison is required with $4L-1$ PPM comparisons. Otherwise, FC3 is replaced by NC1, and these two PPMs will not be compared in the following process any more. And then the second round is operated in similar pattern. Next, NC with submaximal PPM (NC10) and FC with subminimal PPM (FC2) are chosen out (requiring $4L - 4$ comparison times), and then compare NC10 with FC2 (requiring 1 comparison time). If the PPM of FC2 is larger, the sorting process will finish with the selected candidates of FC1, FC2, NC1, and FC4. In this case, two rounds of comparison are required with $8L - 4$ PPM comparisons. Otherwise, FC2 is replaced by NC10, the step of final round comparison repeats the above process, and the final candidates can be obtained after $12L - 9$ PPM comparisons.

To reasonably evaluate the performance of the proposed stochastic SCL decoder, numerical simulation is carried out for 1/2-rate $(1024, 512)$ codeword, where the scaling factor $\alpha = 0.5$ and bit-stream length $l = 1024$. As shown in Figure 10, owing to the random fluctuations of stochastic decoding, the performance

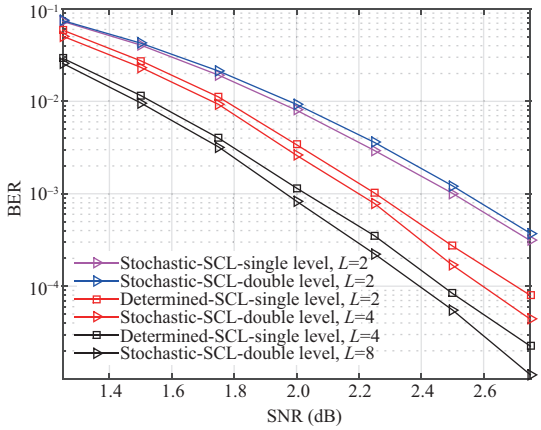

Figure 10 (Color online) Performance comparison of the 1/2-rate $(1024, 512)$ deterministic SCL decoder and stochastic SCL decoder with different decoding levels.

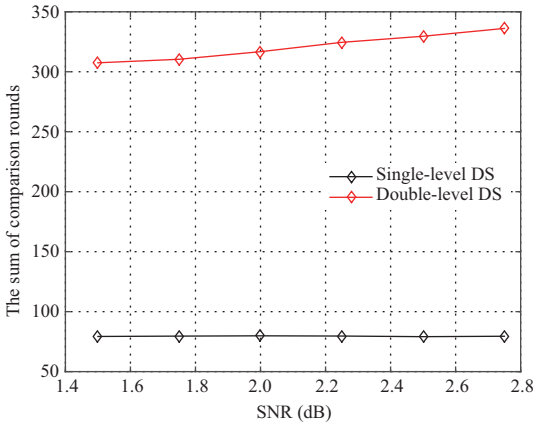

Figure 11 (Color online) The sum of rounds in SCL decoder with double-level DS 1/2-rate $(1024, 512)$ decoding $(L = 4)$.

gap, resulting from deterministic SCL decoding and SSCL decoding, does exist. And the performance of (double level stochastic SCL) DL-SSCL decoding with list size $2L$ can transcend the performance of single level deterministic SCL decoder with list size L. Furthermore, the performance gap between single-level and double-level stochastic decoding scheme results from multiplying two bit conditional probabilities into the path probability simultaneously, and enlarging probability approach in double-level decoding is applied half times than that in single-level decoding scheme. Accordingly, the randomness of PPM in double-level scheme is worse than that in single-level scheme, but the performance gap of stochastic decoders with list 2, depicted in Figure 10, can be still accepted.

In stochastic double-level SCL decoding, ADS method needs to select L best paths from $2L$ or $4L$ children. These two kinds of sorting require different comparison times in each round. Therefore, the latency of ADS is relevant to the average rounds of comparison. For $(1024, 512)$ stochastic polar decoder with $L = 4$, the sum of comparison rounds is shown in Figure 11, the red line presents the sum of comparison rounds resulting from selecting L best paths from $4L$ children, and the black line denotes the sum of comparison rounds resulting from selecting L best paths from $2L$ children.

According to Figures 5 and 9, the DS from $2L$ or $4L$ to L approximately requires $2L$ or $4L$ comparisons per round, respectively. Using $2L$ comparisons in one round as an evaluation standard, Figure 12 summarizes the average rounds for each information bit in double-level decoding. When SNR is in $[1.5, 3.0]$, the average rounds are no larger than 1.47, so the low latency of ADS algorithm is guaranteed. Hence, the ADS can compromise different sorting scales in stochastic double-level decoding, as concluded in Algorithm 1. The FER performance comparison with other FPGA-based SCL decoders is given in Figure 13.

4 Stochastic SCL decoder for polar codes with 2^p-level decoding method

As analyzed in Section 3, multiple-level scheme in binary tree is efficient for SSCL decoder, for it can estimate several bits simultaneously rather than bit by bit. However, multiple-level scheme will cause a larger sorting scale. Direct sorting cannot afford such large scale complexity, and the dramatically increasing sorting complexity will cancel out the advantages brought by stochastic computing. To this end, ADS is improved to maintain the complexity in multiple-level decoding in $O(L)$.

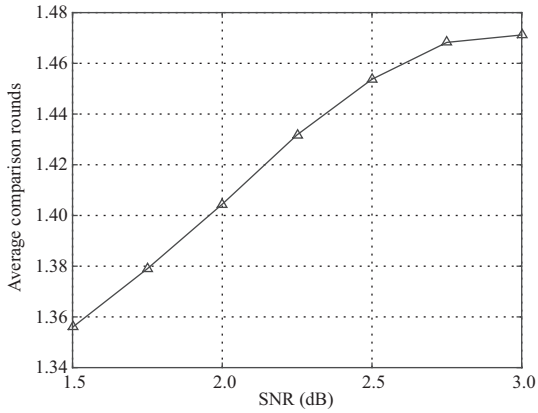

Figure 12 Average rounds in SCL decoder with single-double-level DS 1/2-rate (1024, 512) decoding ($L = 4$).

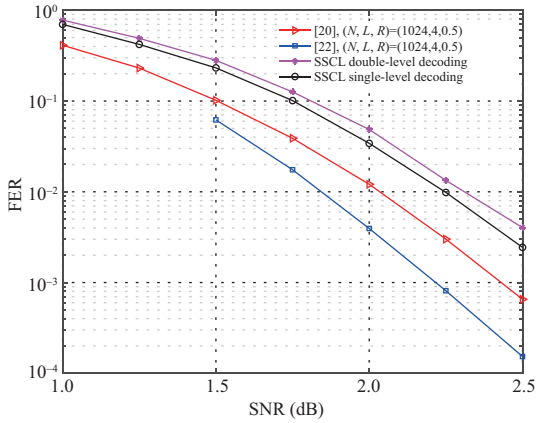

Figure 13 (Color online) The FER performance of different SCL decoders with 1/2 rate (1024, 512) codeword ($L = 4$, $l = 1024$).

Algorithm 1 Proposed ADS algorithm

Require:
$s1 : jk = \{00, 01\}$; $s2 : jk = \{00, 10\}$; $s3 : jk = \{00, 01, 10, 11\}$; $\mathbf{I}$: L previous path numbers;
$[\mathbf{PPM}_{\rm FC}, \mathbf{JK}] = \max_{s1,s2,s3}(\mathbf{PPM}(\hat{u}_1^{2i-2}|\hat{u}_1^{2i-1} = j, \hat{u}_1^{2i} = k))$;
$\mathbf{PPM}_{\rm NC}$, expanded paths list $\mathbb{P}$.
Ensure:
Updated paths list $\mathbb{P}_{\rm new} = [\mathbb{P}(\mathbf{I}), \mathbf{J}, \mathbf{K}]$ and PPM list $\mathbf{PPM}_s$.
$[\mathbf{PPM}_s, \mathbf{Ix}, \mathbf{JKx}] = \text{sort}(\mathbf{PPM}_{\rm FC}, \text{ascend})$;
$[\mathbf{PPM}_1, \mathbf{Iy}, \mathbf{JKy}] = \text{sort}(\mathbf{PPM}_{\rm NC}, \text{descend})$;
for $i = 1 : L - 1$ **do**
  $f = 0$;
  if $(\text{PPM}_{1,i} \leqslant \text{PPM}_{s,i})$ **then**
    $f = 1$;
  else
    $f = 0$;
  end if
  if $(f == 0)$ **then**
    Update $\mathbf{I}$: replace $\mathbf{I}\big(Ix(i)\big)$ with $\mathbf{I}\big(Iy(i)\big)$;
    Update $\boldsymbol{u}_{2i-1}\boldsymbol{u}_{2i}$: $\mathbf{JK}\big(JKx(i)\big) = \mathbf{JK}\big(JKy(i)\big)$;
  else
    break;
  end if
end for
Updated paths list $\mathbb{P}_{\rm new} = [\mathbb{P}(\mathbf{I}), \mathbf{J}, \mathbf{K}]$ and PPM list $\mathbf{PPM}_s$.

4.1 Stochastic SCL decoder in 2^p-level scheme

The multiple-level scheme provides different depth of pre-computation in stochastic decoding process. Denote p as the number of stages to perform pre-computation operations. In the single-level scheme, there is not any pre-computation employed in decoding process, that is to say, $p = 0$ and $2^p = 1$. Therefore, the decoding binary tree selects L candidates bit by bit. In double-level scheme, one stage of pre-computation is applied with $[\Pr_N^{(2i)}(y_1^N, 0),\ \Pr_N^{(2i)}(y_1^N, 1)]$ pre-calculated without feedback information, and $p = 1$ and $2^p = 2$. Therefore, the decoding binary tree selects L candidates from $2L$ or $4L$ nodes every double levels. And so on, when increasing the stages of pre-computation to two ($p = 2$, $2^p = 4$), $[\Pr_{N/2}^{(i)}(y_1^{N/2}, 0),\ \Pr_{N/2}^{(i)}(y_1^{N/2}, 1)]$ and corresponding $[\Pr_N^{(2i)}(y_1^N, 0),\ \Pr_N^{(2i)}(y_1^N, 1)]$ are pre-calculated without feedback information. The decoding binary tree selects L candidates from $2L$, $4L$, $8L$ or $16L$ nodes every four levels ($2^p = 4$).

In general, when the number of pre-computation stages is p, the 2^p-level scheme of SSCL decoder is proposed. The last p stages $[\Pr_{N/2^{p-1}}^{(i/2^{p-2})}(y_1^{N/2^{p-1}}, 0),\ \Pr_{N/2^{p-1}}^{(i/2^{p-2})}(y_1^{N/2^{p-1}}, 1),\ p \geqslant 2]$ to corresponding $[\Pr_N^{(2i)}(y_1^N, 0),\ \Pr_N^{(2i)}(y_1^N, 1)]$ need to be pre-calculated without feedback information. For the information bits are dispersed in N-bit code, the sorting scale is dynamic. The decoding binary tree selects L

candidates from $2L, 4L, \ldots, 2^{2^p}L$ nodes every 2^p levels. The proposed 2^p-level scheme reduces decoding latency from Nl to $Nl/2^p$. It really improves the efficiency of stochastic decoding. For data are calculated between bit and bit in the stochastic computation, the increasing computational elements for pre-computation are negligible, and more details of architecture design are presented in Section 5.

4.2 Adaptive distributed sorting in 2^p-level method

In 2^p-level decoding scheme, the sorting scale is dynamic and large. The worst situation is selecting L best candidates from $2^{2^p}L$ nodes. The DS scheme picks L FC nodes from L paths, and then sequentially chooses the worst FC and the best NC and compares them until the possibility is not changing. However, when p is large, the number of NC nodes is $(2^{2^p}-1)L$, the comparisons of choosing the best NC are difficult. To this end, in the first round, L best NCs are picked from L paths in parallel, and then compare L NCs to select the best one, and this round needs $(2^{2^p}-1)L$ NC comparisons. In the following rounds, only the path of last selected NC needs to be compared. So these rounds need only (2^p+L) NC comparisons. Algorithm 2 shows that only the first round requires $(2^{2^p}-1)L$ NC comparisons (line 7), the following rounds need (2^p+L) NC comparisons (lines 10 and 18).

Algorithm 2 ADS algorithm for 2^p level decoder

Require:
2^p-level scheme, list size is L;
Ensure:
Final L **candidates**.
1: **for** $i = 1 : L$ **do**
2: Choose the best PPM as FC_i, the rest as $\mathbf{NC}_i = [\mathrm{NC}_{i,1}, \mathrm{NC}_{i,2}, \ldots, \mathrm{NC}_{i,(2^{2^P}-1)L}]$;
3: Set $\mathbf{FC} = [\mathrm{FC}_1, \mathrm{FC}_2, \ldots, \mathrm{FC}_L]$;
4: **end for**
5: Set **candidates**$= [\mathrm{FC}_1, \mathrm{FC}_2, \ldots, \mathrm{FC}_L]$;
6: **for** $i = 1 : L$ **do**
7: Choose the best PPM of each $\mathbf{NC}_i$;
8: $\mathbf{NC^k} = [\mathrm{NC}_{1k}, \mathrm{NC}_{2k}, \ldots, \mathrm{NC}_{Lk}]$;
9: **end for**
10: Select the worst PPM FC_k of **FC**;
11: Select the best PPM NC_{pk} of $\mathbf{NC}^k$, p denotes NC_{pk} is from $\mathbf{NC}_p$;
12: **if** $\mathrm{NC}_{pk} < \mathrm{FC}_k$ **then**
13: Jump to line 21;
14: **else**
15: Keep on;
16: **end if**
17: Replace FC_k with NC_{pk} in **candidates**;
18: Cancel FC_k from **FC**, cancel NC_{pk} from $\mathbf{NC}_p$;
19: Select the best PPM of $\mathbf{NC}_p$ as NC_{pk};
20: Jump to line 10;
21: Output final L **candidates**.

4.3 Performance analysis of 2^p-level decoding method

In stochastic 2^p-level SCL polar decoding, when p increases, the decoding latency will be reduced from Nl to $Nl/2^p$. However, increasing p also causes performance loss to some extent. As shown in Figure 10, there is a slight performance gap between double-level scheme and single-level scheme. To update path probability every 2^p-level, the 2^p bit probabilities will be multiplied together and multiplied with the old path probability. Meanwhile, to keep the probabilities for stochastic computing maintained in a reasonable range, enlarging probability approach is employed. Every time updating the path probability, we re-scale all the path probabilities with the same multiple factor to make the best PPM maintained in the range of $[0.5, 1]$. However, the re-scaling times will decrease when p grows. Consequently, the 2^p bit probabilities multiplying at the same time with one re-scaling approach make stochastic randomness worse with p increasing. To conclude, increasing p lowers the latency of stochastic 2^p-level SCL polar decoding at the expense of slight performance loss. Therefore, the selection of size p depends on the realistic situations and requirements of designers.

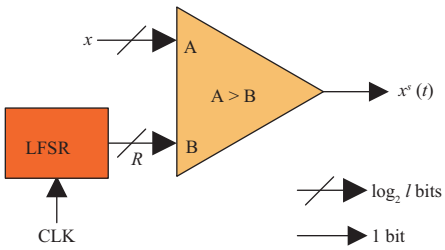

Figure 14 (Color online) The bit stream generation module.

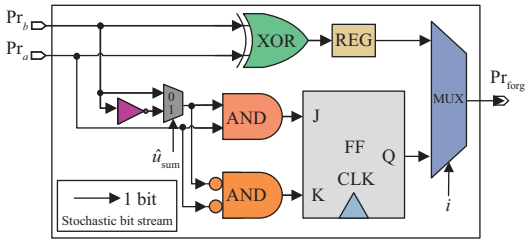

Figure 15 (Color online) The architecture of the proposed mixed node I.

Table 1 The complexity of different modules of stochastic SCL decoder[a)]

Modules	Mixed node I	Mixed node II	Mixed node III	SC decoder[a)]	SCL decoder[b)]
XOR	1	1	1	$N-1$	$NL/2-L+N/2$
NOT	3	6	5	$3N$	$3NL/2+5N/2$
AND	2	8	4	$2N+4$	$NL+4L+2N$
1-bit register	1	1	1	$N-1$	$NL/2-L+N/2$
JK-FF	1	2	2	N	$NL/2+N$
MUX	2	–	–	$2(N-2)$	$NL+N-4L$
Counter	–	–	–	4	$4L$

a) N and L denote the code length and list size, respectively;
b) Double-level decoding.

5 Stochastic SCL decoder architecture

Considering the tradeoff between latency and performance of SCL decoder, we employ $p = 1$ here as our proposed architecture design. This section focuses on designing details of every module in the corresponding proposed SSCL decoder architecture.

5.1 Stochastic bit stream generation module

As illustrated in Figure 14, the deterministic value of x is compared with a random number R with $R \in [0,1]$. In reality, R is pseudo-random, generated by a linear feedback shift register (LFSR). The stochastic bitstream output of the comparator is denoted by $x^s(t)$. If $x > R$, $x^s(t) = 1$, otherwise, $x^s(t) = 0$. Consequently, the deterministic value of a stochastic stream x^s is obtained by $x = \frac{1}{l}\sum_{t=1}^{l} x^s(t)$, where l denotes the length of a stochastic bitstream. This module converses the deterministic value of channel transition probability into stochastic streams. To decode an N-bit polar code, the N paralleling stochastic bit stream generation (SG) modules are required as the interface to connect to the main stochastic decoder. For N paralleling SG sharing the same pseudo-random sequence, the complexity of this interface is determined by that of the comparator, which is based on the length of bitstream l. To this end, the complexity of this interface module is $O(Nl)$.

5.2 Mixed node I

Through adopting similar pre-computation [21], a mixed node I is defined here to realize single level decoding. More specifically, f node and g node are merged in mixed node I to make the whole decoder succincter. On the basis of the elementary nodes presented in [16], the mixed node I is depicted in Figure 15.

This node can execute the function of f and g nodes based on stochastic computing. Table 1 summarizes the hardware complexity of each mixed node I. Although mixed node I can complete the general stochastic element processing, it is unsuitable to realize the proposed double-level decoding.

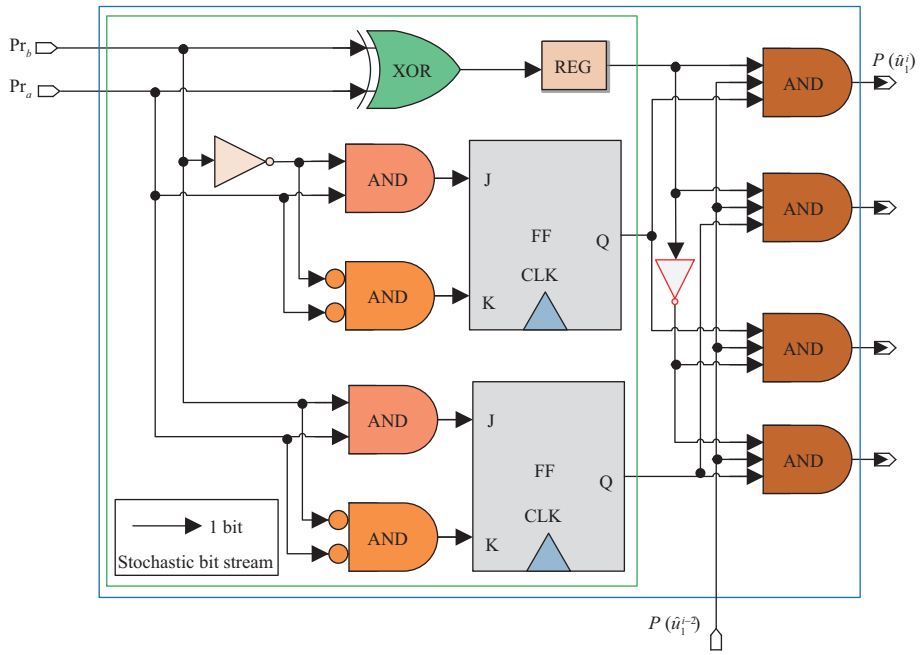

Figure 16 (Color online) Mixed node II for the last stage, also mixed node III for the first stage in SCL decoding.

5.3 Mixed node II

To realize the double-level dcoding, mixed node II that can improve the efficiency of stochastic polar decoding is proposed. To implement the proposed double-level decoding algorithm, the stochastic bit-stream of $p(u_{2i} = \hat{u}_{2i}|\hat{u}_1^{2i-1})$ needs to be pre-computed as the simultaneous output with the stream of $p(u_{2i-1} = \hat{u}_{2i-1}|\hat{u}_1^{2i-2})$, and hence the node module in the last stage should be changed as mixed node II when performing double-level decoding.

As shown in Figure 16, mixed node II (the architecture in blue box) is employed in the last stage of the SCL decoder. According to the property of probabilities, only an inverter is required to implement the complementary operation. The inverter in Figure 16 calculates stochastic stream of $p(\hat{u}_{2i-1} = 0|\hat{u}_{\text{sum}})$. Different with mixed node I outputting only one-bit stream, mixed node II simultaneously outputs four-bit streams of the production of two adjacent bit probabilities, which are $p(\hat{u}_{2i} = 1, \hat{u}_{2i-1} = 1|\hat{u}_{\text{sum}})$, $p(\hat{u}_{2i} = 1, \hat{u}_{2i-1} = 0|\hat{u}_{\text{sum}})$, $p(\hat{u}_{2i} = 0, \hat{u}_{2i-1} = 1|\hat{u}_{\text{sum}})$ and $p(\hat{u}_{2i} = 0, \hat{u}_{2i-1} = 0|\hat{u}_{\text{sum}})$.

5.4 Mixed node III

The SCL decoding architecture consists of L SC decoders in parallel. However, the parallelling design extends the hardware complexity dramatically. For the first stage of SCL decoding, only three outputs are possible for each node module (one f node calculation result and two possible g node calculation results). That is to say, the first stage does not need to use the L parallelling architecture. Therefore, mixed node III is proposed to output all possible results for the second stage calculation. As shown in Figure 16, mixed node III (the architecture in green box) is applied in the first stage in SCL decoder.

5.5 Architecture of stochastic successive cancellation decoder with double-level decoding

As illustrated in Figure 17, the architecture of stochastic SC decoder with double-level decoding is composed of two blocks, the interface and the main decoder. The figure presents an example of 8-bit stochastic decoder, and N-bit decoder obeys the same design. The interface includes N SG modules, which generate stochastic bit streams of channel bit probabilities as the inputs of main decoder.

The main decoder consists of three parts: the node process, estimation and feedback module. The node process of an N-bit SSCL decoder consists of $(N-2)$ mixed nodes I and one mixed node II on a total of $\log_2 N$ stages. For double-level decoding, mixed node II is applied on the last stage of the decoder.

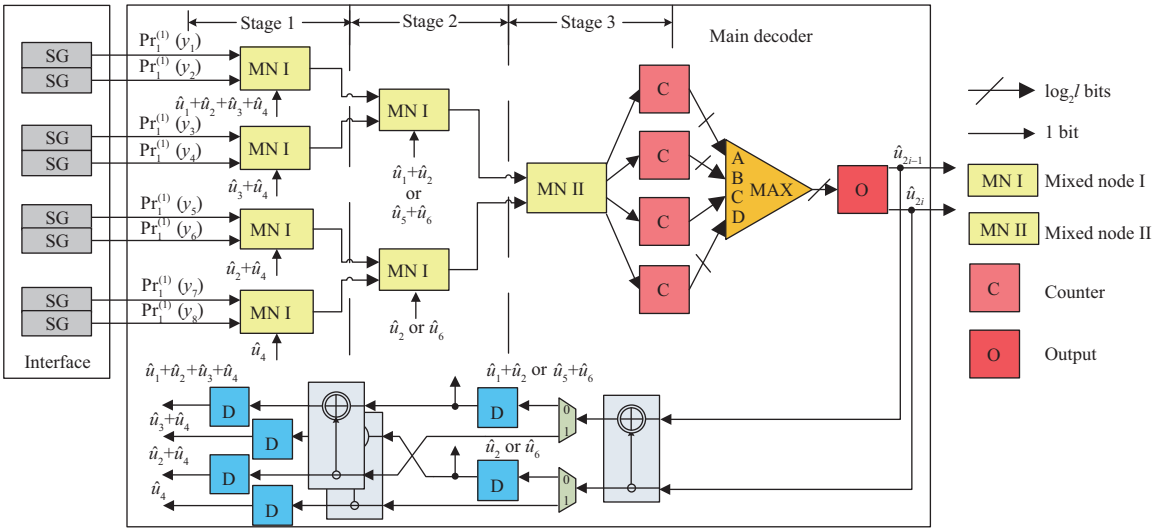

Figure 17 (Color online) The 8-bit stochastic SC decoder with double-level decoding.

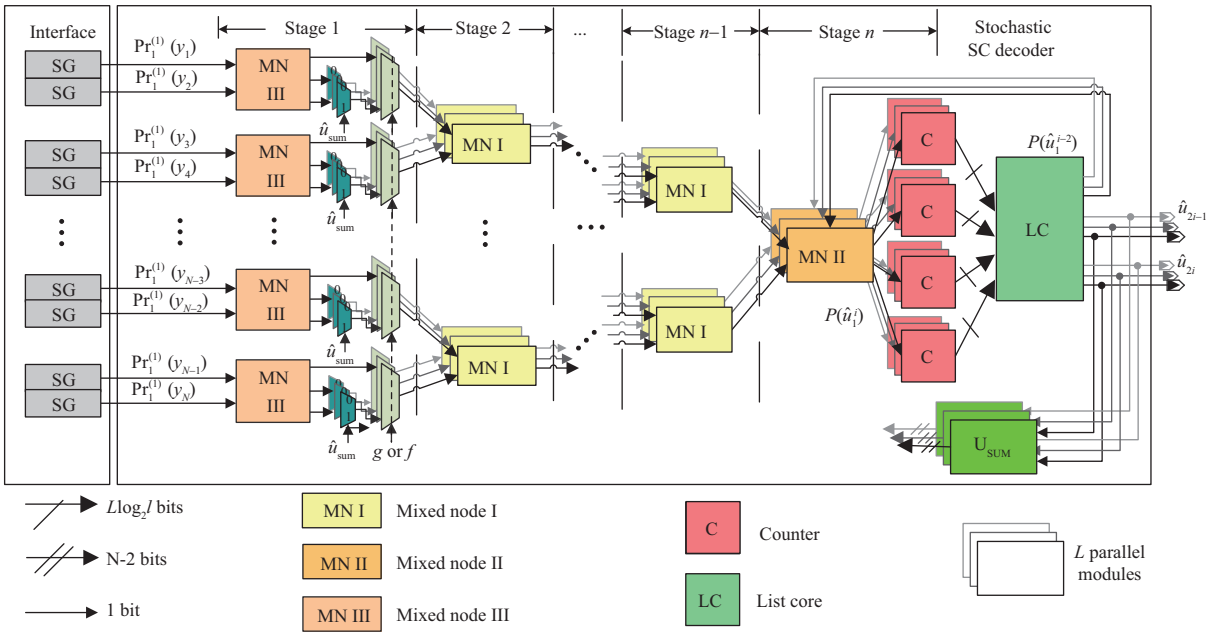

Figure 18 (Color online) The N-bit stochastic SCL decoder with double-level decoding.

Because stochastic bit streams are not suitable for comparing, four counters are employed to convert the stochastic streams into corresponding deterministic data, and then a maximum module selects the best probability. Output in Figure 17 is an estimation module to output the paths $\hat{u}_{2i-1}$ and $\hat{u}_{2i}$, which corresponds to selected probability.

The feedback part is utilized to produce $\hat{u}_{\text{sum}}$ signal for the g node calculation. Considering that the architecture has been researched well [21], we omit its details here. It is worth mentioning that for stochastic decoding, the clock set in feedback part is different from that in stochastic node processing framework. The clock in feedback part is l times than that in stochastic calculation.

5.6 Architecture of stochastic SCL decoder with double-level decoding

As shown in Figure 18, the architecture of SSCL decoder with double-level scheme is similar to that of SC design. The interface block includes N SG modules, while the main decoder expands to L paralleling basic frames. However, the node modules in the first stage can merely output three possible possibilities,

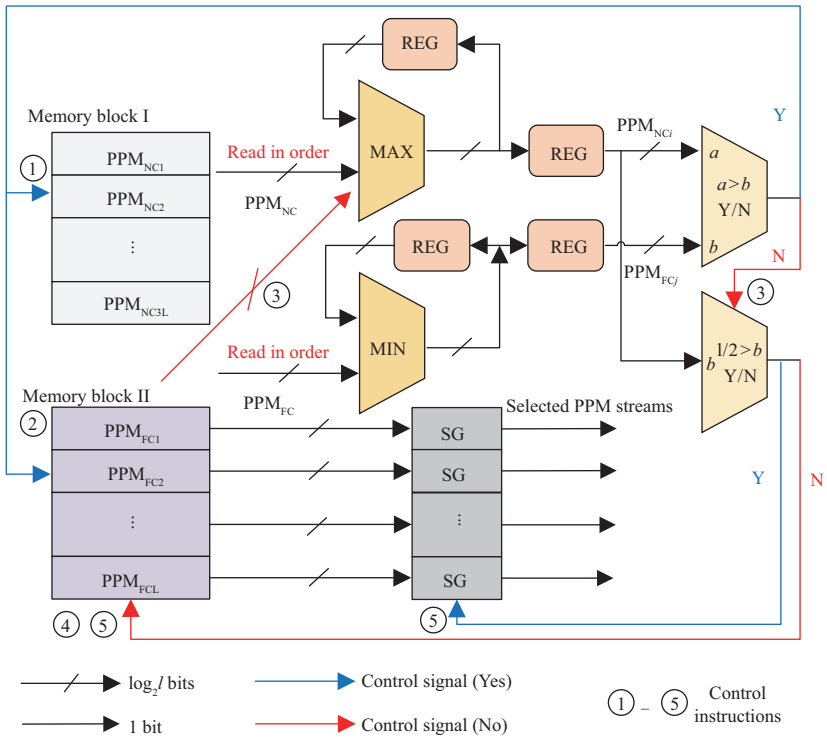

Figure 19 (Color online) The architecture of list core module with ADS method and enlarging probability approach.

as mentioned in Subsection 5.4. To this end, $N/2$ mixed nodes III are employed in the first stage, instead of expanded $NL/2$ mixed nodes I. All nodes in stage 2 to stage n expand to L paralleling modules. The LC denotes the list core module, which uses ADS method to select L best paths from $4L$ candidates.

5.7 List core architecture

The list core module implements the improved DS and enlarging probability approach in the proposed SSCL decoder. To realise the ADS method, PPMs have been categorized into two groups, FCs and NCs. Subsequently, the L FCs are saved into memory block II, and $3L$ NCs are stored in memory block I. As presented in the ADS method, PPMs in memory block II have higher priorities to be selected. As illustrated in Figure 19, two multiplexers respectively read data from memory block I or II in order in each sorting round, and select the best PPM NC_i from block I and the worst PPM FC_j from block II. If NC_i is better, the control signal is activated to execute the instructions ① and ②. The instruction ① removes NC_i from block I, and the instruction ② replaces the value of FC_j with that of NC_i. Then, the next round of sorting keeps active. If FC_j is better, the sorting process is over and data in memory block II are the selected L best paths, the instruction ③ is executed. The instruction ③ is an indicator of the end of sorting and the beginning of enlarging probability. The corresponding L best paths are output.

To realize the enlarging probability approach, the instruction ③ is executed to start-up the enlarging probability approach. If the best-selected path is smaller than $l/2$ (the corresponding probability is smaller than 0.5), the instruction ④ executes left shifting operations to all PPMs in block II. Then the instruction ⑤ employs L SG modules to regenerate enlarged PPM stochastic streams. If the best-selected path is larger than $l/2$, the instruction ⑤ is executed directly.

6 Comparison of stochastic SCL decoding and deterministic decoding

As shown in Figure 20, the decoding schedule can be denoted by three parts: node processing, DS and enlarging probability approach. When decoding every two code bits $\hat{u}_{2i}$ and $\hat{u}_{2i+1}$, all stochastic streams flow through n stages ($n = \log_2 N$), and the latency is $(n + l - 1)$ clock cycles. The latency of DS is

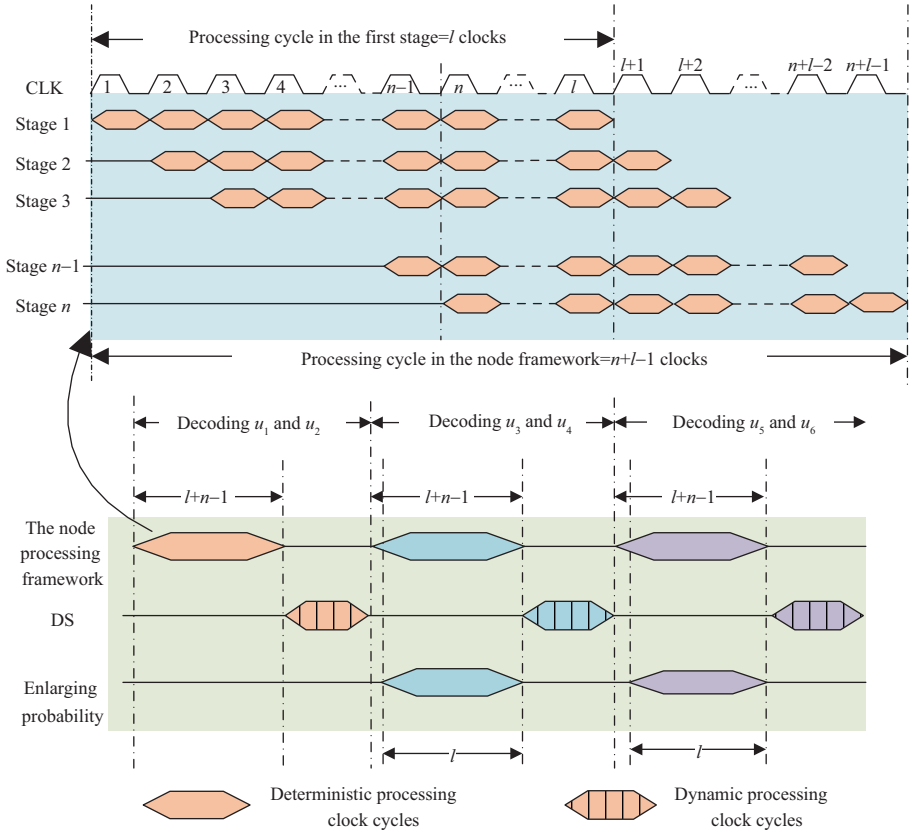

Figure 20 (Color online) Decoding schedule for N-bit stochastic SCL decoder with double-level decoding.

Table 2 Comparison of the implementation results of several FPGA-based SCL architectures

Decoder	[20]	[22]	SSCL[a)] with 1-level decoding	SSCL[a)] with 2-level decoding
FPGA device[b)]	Altera stratix V	Xilinx Kintex 7	Altera stratix V	Altera stratix V
(N, L, R)	$(1024, 4, 0.5)$	$(1024, 4, 0.5)$	$(1024, 4, 0.5)$	$(1024, 4, 0.5)$
ALMs(A)/LUTs(X)[c)]	101160	142961	8080	8146
Registers	13544	19795	2824	2862
f_{op} (MHz)	–	42.66	462.9	445.2
Latency (cycles)	4064	371	2^{20}	2^{19}
TP (Mbps)	–	115	0.452	0.871

a) Stream length $l = 1024$.
b) All the FPGAs are manufactured on 28 nm process technology.
c) An ALM on Altera FPGA can be used as a 6-input LUT.

dynamic, and more details are presented in Subsection 2.5. Enlarging probability approach includes an SG module, and the latency of output is l clock cycles. When decoding an N-bit codeword, the latency $\mathcal{L}$ is

$$\mathcal{L} = (n + l - 1) \times \frac{N}{2} + \mathcal{L}(\mathrm{DS}) \approx \frac{Nl}{2} + \mathcal{L}(\mathrm{DS}). \tag{14}$$

Table 2 compares the proposed SSCL decoder with other FPGA-based SCL decoders. Compared with deterministic SCL designs [20,22], the single-level SSCL only requires around 8%, 5.6% ALMs and 20.8%, 14.2% registers at the expense of increasing latency and reduced throughput. With double-level decoding, the SSCL can efficiently improve its throughput performance at slight cost of hardware consumption. Moreover, the FER performances of these SCL decoders are shown in Figure 13. Compared with [20,22], the single-level SSCL decoder has a loss of about 0.5 and 1 dB, respectively, and the loss of double-level is little larger. Although double-level decoding expends little more hardware resource compared to single-level decoding, it can achieve more excellent performance in throughput and latency. Therefore,

our proposed SSCL decoder can successfully realize a good hardware cost and decoding performance tradeoff.

7 Conclusion

Stochastic SCL polar code decoders are proposed. Approaches that can improve decoding performance have been discussed. It is shown that the stochastic SCL decoder can achieve similar error-correcting performance to its deterministic counterpart and considerable hardware reduction at the expense of increasing latency. Hence, it can achieve a suitable tradeoff between hardware cost and error performance. To our knowledge, this is the first stochastic SCL polar decoder, which has potential applications for wearable and IoT devices that do not require high speed but have strict hardware constraints.

Acknowledgements This work was supported in part by National Key R&D Program of China (Grant No. 2020YFB2205503), National Natural Science Foundation of China (Grant Nos. 61871115, 61501116), Jiangsu Provincial NSF for Excellent Young Scholars (Grant No. BK20180059), Six Talent Peak Program of Jiangsu Province (Grant No. 2018DZXX-001), Distinguished Perfection Professorship of Southeast University, Fundamental Research Funds for the Central Universities, and Student Research Training Program of Southeast University.

References

1 Arıkan E. Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Trans Inform Theor, 2009, 55: 3051–3073
2 Tal I, Vardy A. List decoding of polar codes. In: Proceedings of the IEEE International Symposium on Information Theory Proceedings, Petersburg, 2011. 1–5
3 Chen K, Niu K, Lin J R. List successive cancellation decoding of polar codes. Electron Lett, 2012, 48: 500–501
4 Zhou H Y, Zhang C, Song W Q, et al. Segmented CRC-aided SC list polar decoding. In: Proceedings of 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring), 2016. 1–5
5 Shen Y F, Zhang C, Yang J M, et al. Low-latency software successive cancellation list polar decoder using stage-located copy. In: Proceedings of IEEE International Conference on Digital Signal Processing (DSP), 2016. 84–88
6 Gaines B R. Stochastic computing systems. In: Advances in Information Systems Science. Boston: Springer, 1969. 37–172
7 Moons B, Verhelst M. Energy-efficiency and accuracy of stochastic computing circuits in emerging technologies. IEEE J Emerg Sel Top Circuits Syst, 2014, 4: 475–486
8 Brown B D, Card H C. Stochastic neural computation. I. Computational elements. IEEE Trans Comput, 2001, 50: 891–905
9 Wang H Z, Zhang Z C, You X H, et al. Low-complexity Winograd convolution architecture based on stochastic computing. In: Proceedings of the IEEE International Conference on Digital Signal Processing, Shanghai, 2018. 1–5
10 Ceroici C, Gaudet V C. FPGA implementation of a clockless stochastic LDPC decoder. In: Proceedings of the IEEE Workshop on Signal Processing Systems, Belfast, 2014. 1–5
11 Chen J N, Hu J H, Sobelman G E. Stochastic MIMO detector based on the Markov chain Monte Carlo algorithm. IEEE Trans Signal Process, 2014, 62: 1454–1463
12 Chen J N, Hu J H, Sobelman G E. Stochastic iterative MIMO detection system: algorithm and hardware design. IEEE Trans Circ Syst I, 2015, 62: 1205–1214
13 Liang X, Zhang C, Xu M H, et al. Efficient stochastic list successive cancellation decoder for polar codes. In: Proceedings of the IEEE International System-on-Chip Conference, Beijing, 2015. 421–426
14 Tal I, Vardy A. List decoding of polar codes. In: Proceedings of the IEEE International Symposium on Information Theory Proceedings, Petersburg, 2011. 1–5
15 Bakulin M, Kreyndelin V, Rog A, et al. MMSE based K-best algorithm for efficient MIMO detection. In: Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, Munich, 2017. 258–263
16 Yuan B, Parhi K K. Successive cancellation decoding of polar codes using stochastic computing. In: Proceedings of the IEEE International Symposium on Circuits and Systems, Lisbon, 2015. 3040–3043

17 Tehrani S S, Mannor S, Gross W J. Fully parallel stochastic LDPC decoders. IEEE Trans Signal Process, 2008, 56: 5692–5703

18 Xu Z L, Niu K. Successive cancellation decoders of polar codes based on stochastic computation. In: Proceedings of the IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication, Washington, 2014. 908–912

19 Zhang Z Y, Zhang L, Wang X B, et al. A split-reduced successive cancellation list decoder for polar codes. IEEE J Sel Areas Commun, 2016, 34: 292–302

20 Liang X, Yang J M, Zhang C, et al. Hardware efficient and low-latency CA-SCL decoder based on distributed sorting. In: Proceedings of the IEEE Global Communications Conference, Washington, 2016. 1–6

21 Zhang C, Parhi K K. Low-latency sequential and overlapped architectures for successive cancellation polar decoder. IEEE Trans Signal Process, 2013, 61: 2429–2441

22 Xiong C, Zhong Y, Zhang C, et al. An FPGA emulation platform for polar codes. In: Proceedings of the IEEE Workshop on Signal Processing Systems, 2016. 148–153