Early Stopping Criteria for Energy-Efficient Low-Latency Belief-Propagation Polar Code Decoders

Bo Yuan and Keshab K. Parhi, *Fellow, IEEE*

***Abstract*—Capacity-achieving polar codes have gained significant attention in recent years. In general, polar codes can be decoded by either successive cancellation (SC) or the belief propagation (BP) algorithm. However, unlike SC decoders, performance optimizations for BP decoders have not been fully explored yet. In this paper, we explore novel early stopping criteria for polar BP decoding to significantly reduce energy dissipation and decoding latency. First, we propose two *detection-type* novel early stopping criteria for detecting valid outputs. For polar (1024, 512) codes, these two stopping criteria can reduce the number of iterations by up to 42.5% at 3.5 dB. Then, we propose a novel channel condition estimation approach, which can help select different stopping criteria in different SNR regions. Furthermore, the hardware architectures of polar BP decoders with the proposed stopping criteria are presented and developed. Synthesis results show that with the use of the proposed stopping criteria, the energy dissipation, and average latency of polar (1024, 512) BP decoder can be reduced by $10\% \sim 30\%$ with $2\% \sim 5\%$ hardware overhead, and average throughput can be increased by $20\% \sim 55\%$.**

***Index Terms*—Belief propagation (BP), early stopping criteria, energy-efficient, low-latency, polar codes, VLSI.**

I. Introduction

POLAR codes have been recognized as important channel codes due to their capacity-achieving property [1]. For practical polar code decoding, successive-cancellation (SC) algorithm [1] and belief propagation (BP) algorithm [2], [3] are the two commonly used approaches. In general, SC algorithm requires less computation complexity. Based on this property, several area-efficient SC decoders have been reported in [4], [5]. In addition, with reformulation at algorithm or architecture level, the latency of the SC decoders has been significantly reduced in [6]–[13], [30], [31]. Moreover, another advantage of SC algorithm is its ability to achieve excellent error-correcting performance. With the aid of list-decoding or stack-decoding strategy, the improved SC algorithms in [14]–[16] can achieve much better decoding performance than the original SC and BP algorithms. As a result, hardware and decoding performance of latest SC decoders have been improved significantly.

On the other hand, the iterative BP algorithm is inherently parallel. However, the need of a large number of iterations makes BP decoders suffer from high computation complexity. Although some efforts were addressed in [17]–[20] to improve the decoding and hardware performance of BP decoders, to date the polar BP decoders are still not as attractive as their SC counterparts. As a result, BP decoders need further improvement for their practical applications. In particular, it is necessary to exploit early stopping criteria for BP decoders to reduce the decoding latency and energy consumption.

Use of early stopping criteria for polar BP decoders is important as this can greatly reduce energy dissipation and decoding latency. Note that for iterative decoders (such as polar BP decoders), their required latency and energy dissipation increase linearly with the number of iterations. In practice, the iterative decoders can often converge at an early stage of the iteration process or never converge. In these scenarios, the decoder should stop earlier than reaching the pre-set maximum iteration number. Therefore, early stopping criteria can be used in iterative decoding for early termination, leading to linear reduction in latency and energy dissipation.

Early stopping criteria have been extensively explored in many prior iterative decoders, such as LDPC and Turbo decoding [21]–[26]. Based on the different targeted SNR regions, early stopping criteria can be grouped into different types. Among them the most important type is the *detection-type*, which is used for detecting whether the valid output *has been* already decoded or not. If so, the decoder will stop at an early iteration due to decoding success; this is illustrated in Fig. 1. This detection-type stopping criterion is very useful in high SNR regions, since valid output can be usually obtained after few iterations. Notice that the well-known $\boldsymbol{H}$-matrix-based method [27], which uses parity $\boldsymbol{H}$ matrix to check the output of decoder is a valid codeword or not, can be viewed as a detection-type stopping criterion for block codes (such as LDPC codes).

Considering the similar iterative property of polar BP decoding, it is believed that efficient early stopping criteria should exist for polar codes as well. However, no prior research on this topic has been reported yet. In addition, due to the special procedure of polar decoding (see Section III-A), prior LDPC/Turbo-oriented early stopping criteria (such as $\boldsymbol{H}$-matrix-based

Manuscript received January 03, 2014; revised June 24, 2014 and October 07, 2014; accepted October 16, 2014. Date of publication October 31, 2014; date of current version November 14, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Warren J. Gross.

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities, Minneapolis, MN 55455-0170 USA (e-mail: yuan0103@umn.edu; parhi@umn.edu).

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/TSP.2014.2366712

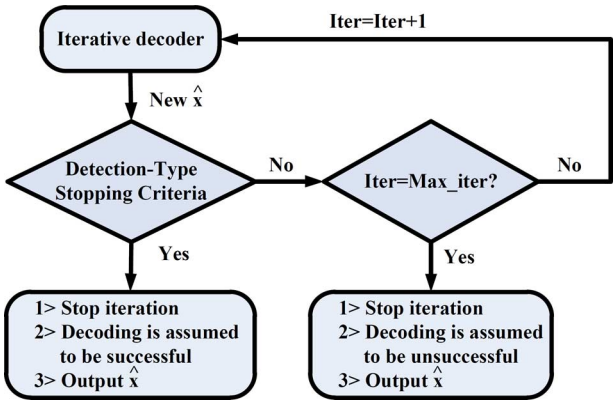


Fig. 1. Iterative decoder with detection-type stopping criteria.

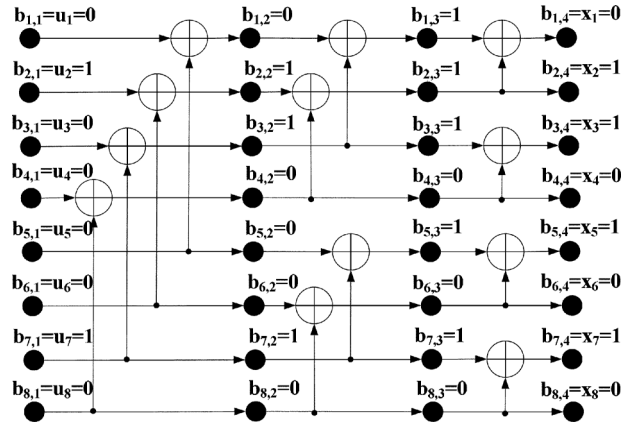


Fig. 2. Hardware implementation of $\boldsymbol{G} = \boldsymbol{F}^{\otimes 3}$.

method) cannot be directly used for polar codes. As a result, the exploration on specific early stopping criteria for polar codes is non-trivial and important.

This paper is the *first* work that presents a systematic study on early stopping criteria targeted to polar codes. In this paper we present various early stopping criteria, which can lead to energy-efficient and low-latency BP decoders.

First, we propose two novel detection-type early stopping criteria. Notice that although polar codes are a type of block codes, the aforementioned traditional $\boldsymbol{H}$-matrix-based stopping criterion cannot be used in polar decoding. In order to solve this problem, we present two new stopping criteria, referred as $\boldsymbol{G}$-*matrix* and *minLLR*, to detect a valid output of the polar decoder. Simulation results show that these two stopping criteria can lead to significant reductions in the average number of iterations. For (1024, 512) polar codes, the number of iterations is reduced by 42.5% at 3.5dB without performance loss. Therefore, energy dissipation and average decoding latency of polar BP decoders are greatly reduced.

In addition, we present a novel channel condition estimation approach for polar decoding, which can roughly estimate the noise level of the transmission channel. With the help of this estimation method, different early stopping criteria can now be specifically selected. As a result, adaptive early stopping criterion can be developed, which are able to reduce the number of iterations over the entire SNR range.

With the help of stopping criteria, hardware performance of polar BP decoder can be significantly improved. For a polar (1024, 512) BP decoder, synthesis results show that with small overhead, the use of the stopping criteria can lead to $10\% \sim 30\%$ reduction in energy dissipation and average latency. Meanwhile the average throughput can be improved by $20\% \sim 55\%$.

The rest of this paper is organized as follows: Section II gives a brief review of polar codes and the BP algorithm. Section III presents two detection-type early stopping criteria for high SNR regions. Section IV presents a channel condition estimation approach. In addition, the adaptive early stopping criterion is also proposed in this section. Section V reviews the hardware design of polar BP decoders and then presents the architecture of the proposed stopping criteria. The improvement on hardware performance with the use of the proposed stopping criteria is presented in Section VI. Section VII draws conclusions.

II. Review of Polar Codes and Polar BP Algorithm

A. Polar Codes and Encoding

Polar codes are derived from the phenomenon of channel polarization. With a special recursive encoding process, the post-decoding reliability of information bits will be polarized based on their positions in the decoded codeword. Therefore, an (n, k) polar code can be generated in two steps. First, an n-bit message $\boldsymbol{u}$ is constructed by assigning the reliable and unreliable positions as k information bits and $(n-k)$"0" bits, respectively. Those information bits are referred as "free" bits, while those forced "0" bits are referred as "frozen" bits. Then, the n-bit $\boldsymbol{u}$ is multiplied with generator matrix $\boldsymbol{G}$ to generate an n-bit transmitted codeword $\boldsymbol{x}$. Here $\boldsymbol{G} = \boldsymbol{F}^{\otimes \boldsymbol{m}}$, where $\boldsymbol{F}^{\otimes \boldsymbol{m}}$ denotes the m-th Kronecker power of $\boldsymbol{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ and $m = \log_2 n$. Fig. 2 shows the hardware implementation of $\boldsymbol{G} = \boldsymbol{F}^{\otimes 3}$. For further details of the polar encoding process, the reader is referred to [1].

B. Polar BP Algorithm With Min-Sum Approximation

Belief propagation (BP) algorithm is a widely used approach for iterative decoding over factor graph [28]. Since polar codes can be represented in the form of a factor graph [3], the BP algorithm can be applied for polar decoding as well. In general, because an $(n,\ k)$ polar code $(n = 2^m)$ can be represented by an m-stage factor graph, the polar BP algorithm is iteratively applied over this factor graph containing $(m+1)n$ nodes. Here each (i, j)-index node is associated with left-to-right and right-to-left likelihood messages. During the whole iteration process, these soft messages are updated and propagated among adjacent nodes. Fig. 3(a) illustrates the factor graph for an (8, 4) polar code. This graph contains $m = \log_2 8 = 3$ stages, where each stage contains $n/2 = 4$ processing elements (PEs) (see Fig. 3(b)). With the help of updating principles in [3], these PEs are used to update the propagating messages in each iteration.

In [4], a hardware-friendly min-sum approximation was first proposed for polar decoding. In order to reduce complexity of polar BP decoding, this approximation was adopted in [18] as well. However, unlike the case for SC decoders, the min-sum approximation for BP decoders causes performance loss. As

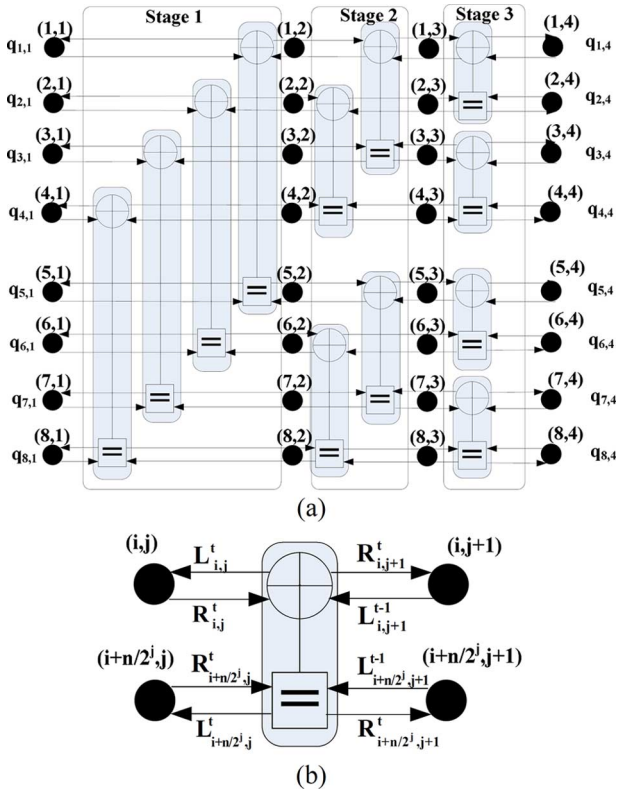


Fig. 3. (a) Factor graph of (8, 4) polar code. (b) PE. (cited from [19]).

a result, in our previous work [19] we had proposed use of a scaling parameter α to offset the approximation error. This scaled min-sum (SMS) algorithm is described by:

$$\begin{aligned} L^t_{i,j} &= \alpha^* sign(L^{t-1}_{i,j+1}) sign(L^{t-1}_{i+n/2^j,j+1} + R^t_{i+n/2^j,j}) \\ &\quad * \min(\left|L^{t-1}_{i,j+1}\right|, \left|L^{t-1}_{i+n/2^j,j+1} + R^t_{i+n/2^j,j}\right|) \\ L^t_{i+n/2^j,j} &= L^{t-1}_{i+n/2^j,j+1} + \alpha^* sign(L^{t-1}_{i,j+1}) sign(R^t_{i,j}) \\ &\quad * \min(\left|L^{t-1}_{i,j+1}\right|, \left|R^t_{i,j}\right|) \\ R^t_{i,j+1} &= \alpha^* sign(R^t_{i,j}) sign(L^{t-1}_{i+n/2^j,j+1} + R^t_{i+n/2^j,j}) \\ &\quad * \min(\left|R^t_{i,j}\right|, \left|L^{t-1}_{i+n/2^j,j+1} + R^t_{i+n/2^j,j}\right|) \\ R^t_{i+n/2^j,j+1} &= R^t_{i+n/2^j,j} + \alpha^* sign(L^{t-1}_{i,j+1}) sign(R^t_{i,j}) \\ &\quad * \min(\left|L^{t-1}_{i,j+1}\right|, \left|R^t_{i,j}\right|). \end{aligned} \quad (1)$$

Here $L^t_{i,j}$ and $R^t_{i,j}$ represent logarithmic likelihood ratio (LLR)-based right-to-left and left-to-right propagating messages in the t-th iteration, respectively. Based on (1), adjacent nodes can iteratively propagate their associated messages and these messages can be updated by the PEs. After the BP decoder reaches a pre-set maximum iteration ($\max\_iter$), the i-th decoded bit ($\hat{u}_i$) is determined and the output based on the hard-decision of the overall LLR values of node $(i, 1)$:

$$\hat{u}_i = \mathbf{sign}(LLR^{\max\_iter}_{i,1}) \triangleq \mathbf{sign}(\boldsymbol{L}^{\max\_iter}_{i,1} + \boldsymbol{R}^{\max\_iter}_{i,1}) \quad (2)$$

where $\mathbf{sign}(y) = 0$ when $y \geq 0$, and otherwise is 1.

As shown in Fig. 4, the use of scaling parameter can completely compensate the performance loss caused by the min-sum approximation. For the example (1024, 512) polar code with

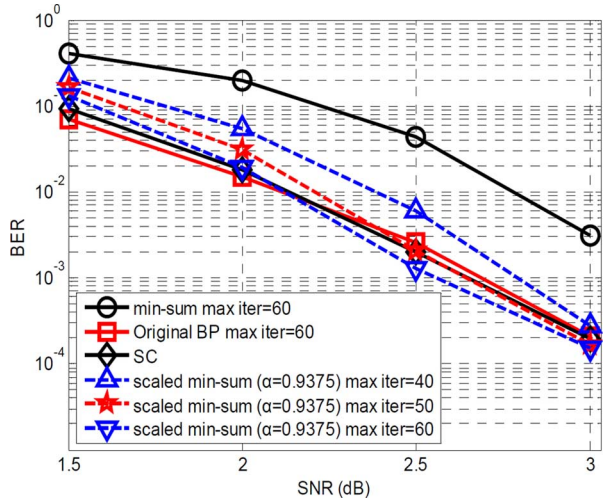


Fig. 4. Performance of scaled and non-scaled min-sum approximation of BP decoding for (1024, 512) polar codes.

$\max\_iter = 60$ and $\alpha = 0.9375$, the scaled min-sum approximation can achieve decoding performance that is similar to that of the original BP and SC algorithms.

III. The Proposed Detection-Type Stopping Criteria

Fig. 4 shows that with proper scaling parameter and large number of iterations, the polar BP decoder can perform similar to the SC decoder. However, as mentioned in Section I, when the noise level of the channel is low (high SNR scenario), the iterative decoders can always decode the valid output much earlier than reaching the fixed maximum number of iterations. As a result, in order to avoid unnecessary iterations, an efficient method that can detect a valid output is useful and important for the design of iterative decoders. For most block codes, this detection process can be performed with the use of $\boldsymbol{H}$ matrix. However, as analyzed below in Section III-A, the traditional $\boldsymbol{H}$ matrix-based detection approach cannot work for the polar codes. Therefore, in order to solve this problem, we propose two novel detection-type early stopping criteria in Sections III-B and III-C, respectively.

A. Inability of Traditional H-Matrix-Based Detection Approach for Polar Decoding

For most block codes (such as LDPC codes), $\boldsymbol{H}$ matrix is commonly used for codeword detection. According to coding theory [27], given generator matrix $\boldsymbol{G}$ and parity matrix $\boldsymbol{H}$, $\boldsymbol{x}\boldsymbol{H}^{\mathbf{T}} = \mathbf{0}$ always holds for any codeword $\boldsymbol{x}$. Therefore, if the output of the decoder, referred as $\hat{\boldsymbol{x}}$ (the estimate of $\boldsymbol{x}$), satisfies $\hat{\boldsymbol{x}}\boldsymbol{H}^{\boldsymbol{T}} = \mathbf{0}$, then $\hat{\boldsymbol{x}}$ is also the codeword generated by $\boldsymbol{G}$. In that case, $\hat{\boldsymbol{x}}$ is equivalent to $\boldsymbol{x}$ with high probability. Therefore, the decoder can immediately terminate the iterations and output $\hat{\boldsymbol{x}}$ as the valid estimate of $\boldsymbol{x}$ (see Fig. 5(a)). In general, the above $\boldsymbol{H}$-matrix-based approach has very high detection accuracy with very small hardware overhead.

As a type of block code, polar codes have the similar property of $\boldsymbol{x}\boldsymbol{H}^{\mathbf{T}} = \mathbf{0}$ [29]. However, the $\boldsymbol{H}$ matrix of polar codes cannot be used for codeword detection. This is because the output after each iteration of polar decoder is always $\hat{\boldsymbol{u}}$(the estimate of $\boldsymbol{u}$)

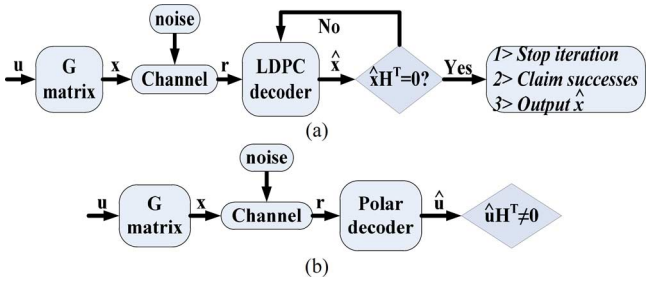


Fig. 5. (a) $\boldsymbol{H}$-matrix can detect valid output of LDPC decoder. (b) $\boldsymbol{H}$-matrix cannot detect valid output of polar decoder.

instead of $\hat{\boldsymbol{x}}$ (see Fig. 5(b)). Next we prove that $\hat{\boldsymbol{u}}\boldsymbol{H}^{\boldsymbol{T}}$ is generally not equivalent to 0 even for decodable cases. Assume $\hat{\boldsymbol{x}}$ is the valid codeword $\boldsymbol{x}$, then $\hat{\boldsymbol{x}}\boldsymbol{H}^{\boldsymbol{T}} = 0$. Because $\hat{\boldsymbol{u}} = \hat{\boldsymbol{x}}\boldsymbol{G}$, we then have $\hat{\boldsymbol{u}}\boldsymbol{H}^{\boldsymbol{T}} = \hat{\boldsymbol{x}}\boldsymbol{G}\boldsymbol{H}^{\boldsymbol{T}} = \hat{\boldsymbol{x}}(\boldsymbol{G}\boldsymbol{H}^{\boldsymbol{T}})$. Consider in general $\boldsymbol{G}\boldsymbol{H}^{\boldsymbol{T}} \neq \boldsymbol{H}^{\boldsymbol{T}}$, then $\hat{\boldsymbol{u}}\boldsymbol{H}^{\boldsymbol{T}} = \hat{\boldsymbol{x}}(\boldsymbol{G}\boldsymbol{H}^{\boldsymbol{T}}) \neq \hat{\boldsymbol{x}}\boldsymbol{H}^{\boldsymbol{T}}$. Hence $\hat{\boldsymbol{u}}\boldsymbol{H}^{\boldsymbol{T}} \neq 0$ even if $\hat{\boldsymbol{x}} = \boldsymbol{x}$. As a result, $\boldsymbol{H}$ matrix cannot be used to examine the validity of the output of polar decoder.[1]

B. The Proposed G-Matrix-Based Detection-Type Early Stopping Criterion

Fig. 5 shows that the absence of $\hat{\boldsymbol{x}}$ causes $\boldsymbol{H}$-matrix-based approach to fail in detecting a valid output for polar decoders. In this subsection, we propose a novel $\boldsymbol{G}$-matrix-based approach to solve this problem. By utilizing $\hat{\boldsymbol{u}}$ and $\boldsymbol{G}$ matrix, this $\boldsymbol{G}$-matrix-based approach achieves good performance in detecting a valid output. As a result, this method can be used as an efficient detection-type early stopping criterion for polar codes[2].

As introduced in Section II-B, the BP algorithm is performed over the factor graph of polar codes. Due to the special auto-duality property of the generator matrix $\boldsymbol{G} = \boldsymbol{F}^{\otimes \boldsymbol{m}}$, the factor graph of BP decoder is just the edge-to-edge estimation of the encoder ($\boldsymbol{G}$ matrix). As a result, the node $q_{i,j}$ in factor graph (see Fig. 3(a)) is the estimation of $b_{i,j}$ in the encoder (see Fig. 2), where $q_{i,j} \triangleq \mathbf{sign}(L^t_{i,j} + R^t_{i,j})$. Consider $b_{i,1} = u_i$ and $b_{i,m+1} = x_i$, hence for decodable cases, after certain rounds of iterations, $q_{i,1}$ and $q_{i,m+1}$ are very likely equal to u_i and x_i, respectively. As a result, the two vectors which consist of $q_{i,1}$ and $q_{i,m+1}$ can be viewed as the estimates of $\boldsymbol{u}$ and $\boldsymbol{x}$, referred as $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{x}}$, respectively. Here $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{x}}$ are defined as:

$$\begin{aligned} \hat{\boldsymbol{u}} = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_n) &\triangleq (q_{1,1}, q_{2,1}, \ldots, q_{n,1}) \\ \text{and } \hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) &\triangleq (q_{1,m+1}, q_{2,m+1}, \ldots, q_{n,m+1}). \end{aligned} \tag{3}$$

[1]Notice that even we re-encode $\hat{\boldsymbol{u}}$ to $\hat{\boldsymbol{y}} = \hat{\boldsymbol{u}}\boldsymbol{G}$, we still cannot detect whether $\hat{\boldsymbol{x}}$ is valid codeword $\boldsymbol{x}$ or not from $\hat{\boldsymbol{y}}\boldsymbol{H}^{\boldsymbol{T}} = 0$ or not. This is because due to the decoding principle of BP algorithm, the frozen positions of output $\hat{\boldsymbol{u}}$ are always filled as "0" as well. Therefore, even $\hat{\boldsymbol{u}} \neq \boldsymbol{u}$ (this corresponds to $\hat{\boldsymbol{x}} \neq \boldsymbol{x}$), $\hat{\boldsymbol{y}} = \hat{\boldsymbol{u}}\boldsymbol{G}$ is still a codeword with $\hat{\boldsymbol{y}}\boldsymbol{H}^{\boldsymbol{T}} = 0$; however, in that case $\hat{\boldsymbol{x}}$ is not the valid codeword $\boldsymbol{x}$ that we want to transmit. As a result, even using the re-encoding method $\boldsymbol{H}$ matrix still cannot be used for valid codeword detection.

[2]In [9] an approach using $\mathbf{G}$ matrix was proposed to reduce the average decoding latency of SC algorithm. The use of $\mathbf{G}$ matrix in BP (this work) and SC [9] algorithms shows its potential more general property than $\mathbf{H}$ matrix in polar decoding procedure. As shown in Section IV, G matrix can be also used to estimate channel noise. In addition, our ongoing research shows G matrix can be even used to predict the decodability of input codewords.

Recall that for any polar codes $\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}$, hence if $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{x}}$ are the valid estimates, $\hat{\boldsymbol{x}} = \hat{\boldsymbol{u}}\boldsymbol{G}$ must also hold. Therefore, $\hat{\boldsymbol{x}} = \hat{\boldsymbol{u}}\boldsymbol{G}$ can be used to detect valid $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{x}}$ as follows:

1) G-Matrix-Based Stopping Criterion for Detecting Valid Output (G-Matrix Criterion): If $\hat{\boldsymbol{x}} = \hat{\boldsymbol{u}}\boldsymbol{G}$, then the $\hat{\boldsymbol{u}}$ is assumed as a valid estimate of $\boldsymbol{u}$. The decoder will output $\hat{\boldsymbol{u}}$ and stop further iterations.

Accordingly, a polar BP decoder with the above ***G-matrix*** early stopping criterion is developed as shown in Scheme-A.

Scheme-A (n, k) Polar BP decoder with G-matrix criterion:

1: Input:
2: Log-Likelihood ratios $LLR(r_i)$ from channel output
3: Frozen positions set: $\mathbf{frz} = \{frz_1, frz_2, \ldots frz_{n-k}\}$
4: Initialization:
5: Set scale value α maximum iteration number $\max\_iter$
6: For the propagating messages $L^t_{i,j}$ and $R^t_{i,j}$ of each node (i, j):
7: if $(j == 1)$ & $(i \in \mathbf{frz})$ $R^t_{i,l} = \infty$ for $t = 0, 1 \ldots \max\_iter$
8: else if $(j == 1 + m)$ $L^t_{i,m+1} = LLR(r_i)$ for $t = 0, 1 \ldots \max\_iter$
9: else $L^0_{i,j} = R^0_{i,j} = 0$
10: Iteration process:
11: While $t < \max\_iter$ do
12: Update $L^t_{i,j}$ and $R^t_{i,j}$ for each node based on **(3)**
13: Update $\hat{\mathbf{u}} = (\hat{u}_1, \hat{u}_2 \ldots, \hat{u}_n)$ and $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2 \ldots, \hat{x}_n)$:
14: if $(L^t_{i,m+1} + R^t_{i,m+1} \geq 0)$ $\hat{x}_i = 0$ else $\hat{x}_i = 1$
15: if $(L^t_{i,1} + R^t_{i,1} \geq 0)$ $\hat{u}_i = 0$ else $\hat{u}_i = 1$
16: G-matrix-I early stopping criterion for decodable cases:
17: if $(\hat{\mathbf{u}}\mathbf{G} == \hat{\mathbf{x}})$ $1 >$ Decoding is assumed to be successful
18: $2 >$ Stop iteration & Output $\hat{\mathbf{u}}$
19: else $t = t + 1$ & Begin next iteration
20: Output : $\hat{\mathbf{u}} = (\hat{u}_1, \hat{u}_2 \ldots, \hat{u}_n)$

Fig. 6 shows the failure rate of the proposed ***G-matrix*** stopping criterion for polar (1024, 512) code. Here the failure is counted when the stopping criterion causes an invalid decoded output. From this figure it can be seen that the proposed method shows very low failure rate ($< 0.1\%$) at both low and high SNR regions. As a result, compared to the BP decoder with constant number of iterations, the BP decoder with the ***G-matrix*** stopping criterion can achieve the same decoding performance (see Fig. 7) with less number of iterations (see Fig. 8). In particular, as seen in Fig. 8, in high SNR region the benefit of this approach

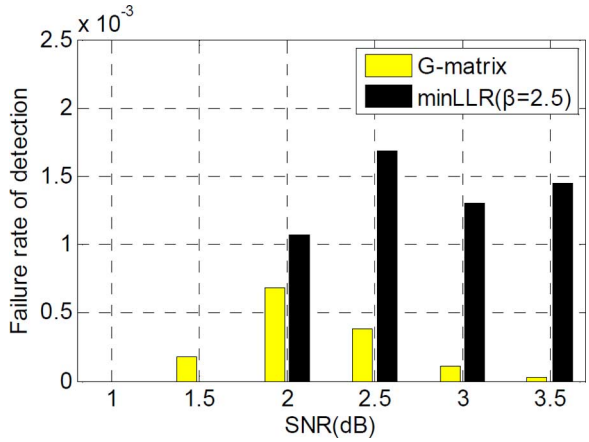


Fig. 6. Failure rate of stopping criteria for (1024, 512) BP decoding with $\max\_iter = 40$.

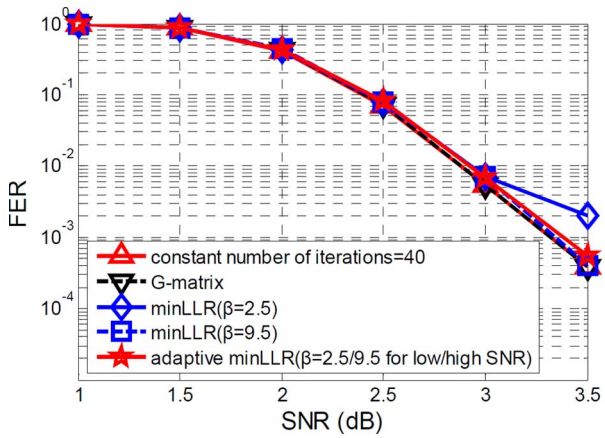


Fig. 7. Decoding performance with the proposed stopping criteria for (1024, 512) polar BP decoding.

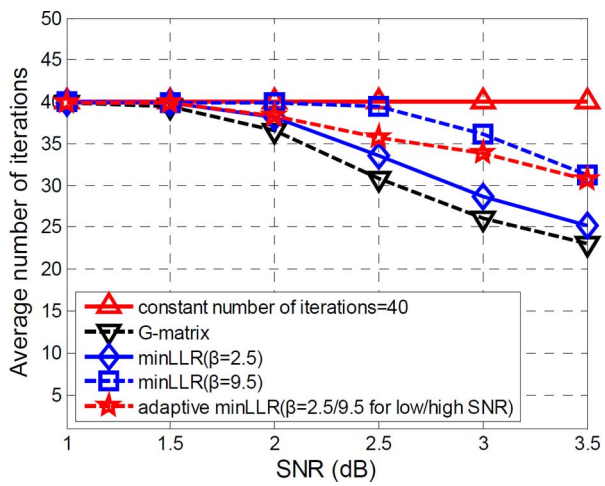


Fig. 8. Average number of iterations with the proposed stopping criteria for (1024, 512) polar BP decoding.

on reducing the number of iterations is very significant. For example, when SNR is 3.5 dB, the average number of required iterations can be reduced by 42.5%.

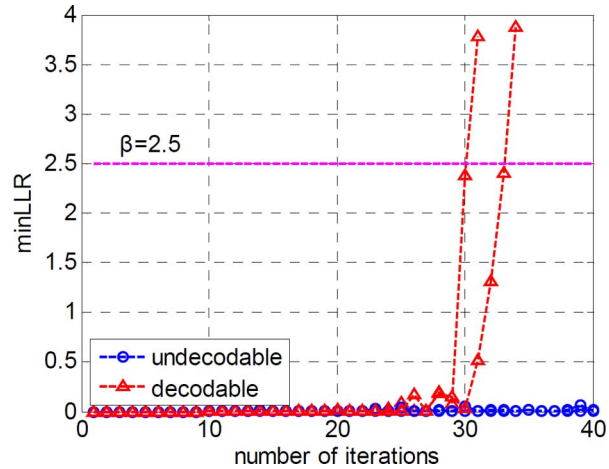


Fig. 9. *minLLR* trend with (1024, 512) polar BP decoding at SNR $= 2.5$ dB for two undecodable cases and two decodable cases.

C. The Proposed Minimum Magnitude LLR (minLLR)-based Detection-type Early Stopping Criterion

Section III-B introduced a $\boldsymbol{G}$-matrix-based early stopping criterion, which utilizes $q_{i,1}$ and $q_{i,m+1}$, as the hard-decision values of $LLR^t_{i,1}$ and $LLR^t_{i,m+1}$, to detect the valid $\hat{\boldsymbol{u}}$. In this subsection, we present another efficient detection-type stopping criterion, which only uses the information of $LLR^t_{i,1}$.

Recall that $\hat{u}_i$ is determined as $\hat{u}_i = \mathbf{sign}(LLR^t_{i,1})$. This hard-decision procedure only utilizes the information in sign part of $LLR^t_{i,1}$, while the information in magnitude part, denoted as $|LLR^t_{i,1}|$, is not used at all. However, since $LLR^t_{i,1}$ is related to the probability of $\hat{u}_i$ being 0 or 1 (where $LLR^t_{i,1} = \ln(\Pr(\hat{u}_i = 0)/\Pr(\hat{u}_i = 1))$), intuitively the larger $|LLR^t_{i,1}|$ means the corresponding hard-decision value is more reliable. As a result, $|LLR^t_{i,1}|$ can be used to measure the reliability of $\hat{u}_i$. Therefore, we propose to use the minimum $|LLR^t_{i,1}|$ (*minLLR*) for all $i = 1, 2, \ldots n$ to detect the valid $\hat{\boldsymbol{u}}$:

minLLR-based stopping criterion for detecting valid output (minLLR criterion): if minLLR is larger than threshold value β, then the corresponding $\hat{\boldsymbol{u}}$ can be assumed as a valid estimate of $\boldsymbol{u}$. The decoder will output $\hat{\boldsymbol{u}}$ and stop further iterations.

This criterion can be understood in an intuitive way. When *minLLR* is larger than β, that means all $|LLR^t_{i,1}|$ values are larger than β. Considering β is typically larger than 2.5, it indicates that the probability of each corresponding hard-decision $\hat{u}_i$ being 0 or 1 is at least $\mathrm{e}^{2.5} \approx 12$ times larger than that of it being 1 or 0. In that case, all the decoded $\hat{u}_i$ are highly reliable. Hence, $\hat{\boldsymbol{u}}$ is most likely a valid estimate of $\boldsymbol{u}$.

The validity of the proposed approach is illustrated in Fig. 9. Fig. 9 show the *minLLR*-vs-number of iterations curve for two undecodable and two decodable cases with SNR $= 2.5$ dB. Here polar (1024, 512) scaled min-sum decoder with $\alpha = 0.9375$ is used. From this figure, it can be seen that, for undecodable cases *minLLR* is always smaller than the threshold value $\beta = 2.5$ during the whole iteration process. On the other hand, for decodable cases *minLLR* always exceeds β at an earlier time.

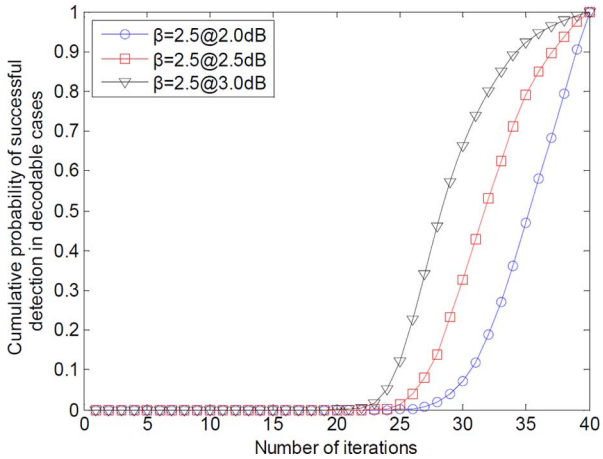


Fig. 10. Cumulative distribution function of successful detection for decodable cases with $\beta = 2.5$ and (1024, 512) polar codes.

Fig. 10 shows the cumulative distribution function of successful detection for decodable cases when using *minLLR* criterion. With $\beta = 2.5$, the (1024, 512) BP decoder can always decode the valid codewords earlier than reaching the pre-set maximum iteration number. From this figure it is also found that with higher SNR value the probability that finding valid codewords with less number of iterations become higher.

Accordingly, a polar BP decoder with the *minLLR*-based early stopping criterion can be summarized in Scheme-B.

Scheme-B (n, k) Polar BP decoder with minLLR criterion:

1: Input: Same with Scheme-A, threshold parameter β

2: Initialization: Same with Scheme-A

3: Iteration process:

4: While $t < \max\_iter$ do

5: Update $L_{i,j}^t$ and $R_{i,j}^t$ for each node based on **(3)**

6: Update $\hat{\mathbf{u}} = (\hat{u}_1, \hat{u}_2 \ldots, \hat{u}_n)$: if $(L_{i,1}^t + R_{i,1}^t \geq 0)$ $\hat{u}_i = 0$ else $\hat{u}_i = 1$

7: Find minimum magnitude LLR ($\min$ *LLR*) among all $|L_{i,1}^t + R_{i,1}^t|$

8: minLLR early stopping criterion for decodable cases

9: if $(\min LLR \geq \beta)$ $1 >$ Decoding is assumed to be successful

10: $2 >$ Output $\hat{\mathbf{u}}$ & Stop iteration

11: else $t = t + 1$ & Begin next iteration

12: Output: $\hat{\mathbf{u}} = (\hat{u}_1, \hat{u}_2 \ldots, \hat{u}_n)$

The failure rate of *minLLR* criterion for polar (1024, 512) codes is shown in Fig. 6. From this figure it can be seen that, for low and medium SNR scenarios that allow higher FER rate $(10^{-1} \sim 10^{-2})$, the failure rate of *minLLR* criterion is low enough to guarantee decoding performance. However, for high SNR regions that require lower FER rate (such as 10^{-3} for SNR $= 3.5$ dB), the failure rate is relatively large to cause performance loss (see Fig. 7). In that case, a larger β is needed to avoid performance degradation. For example, simulation results show that the choice of $\beta = 9.5$ can totally avoid the performance loss for polar (1024, 512) code (see Fig. 7).

The decoding performance and average number of iterations for (1024, 512) BP decoder with the *minLLR* stopping criterion are show in Figs. 7 and 8, respectively. As seen from these two figures, after using *minLLR* criterion the required number of iterations for the BP decoder at low and medium SNR regions can be reduced by $17.5\% \sim 37.5\%$ without performance loss. For the performance loss at high SNR region, since it is caused by the relatively high failure rate with smaller β, the use of larger β (for example $\beta = 9.5$) can avoid this performance degradation in high SNR regions (see Fig. 7). In that case, the average number of iterations at 3.5dB can be reduced by 32.5%. Notice that compared to using a smaller β, the use of larger β leads to a relatively larger number of iterations (see Fig. 8). This is the penalty for avoiding performance loss.

IV. The Proposed Channel Condition Estimation Method and Adaptive Early Stopping Criterion

A. The Proposed Channel Condition Estimation Method

For practical applications, the SNR information from channel output is very useful for efficient polar decoding. Recall that in Section III-C, the required value of threshold in the high SNR regions (for example $\beta = 9.5$ at SNR $= 3.5$ dB) is always larger than that in low SNR regions ($\beta = 2.5$ when SNR ≤ 3 dB). This difference of β in different SNR scenarios leads to the dilemma of choosing β. If smaller β is selected, decoding performance in high SNR scenario cannot be guaranteed (see Fig. 7); however, if a larger β is chosen, it leads to extra iterations in low SNR cases since smaller β is sufficient for these cases (see Fig. 8). Fortunately, this dilemma can be easily solved if the channel SNR level is known. In that case, an adaptive strategy can be applied by selecting different values of β based on different channel SNR regions.

Although knowing SNR information can offer great benefit to polar decoding, in many practical applications, channel SNR information is unknown at the decoder end; therefore, a simple channel condition estimation approach, which can roughly estimate the SNR level of channel, is useful and needs to be explored for efficient polar decoding.

In this subsection, we propose a novel channel condition estimation approach. This method is based on measuring λ, which is defined as the Hamming distance between $\hat{\boldsymbol{u}}\boldsymbol{G}$ and $\hat{\boldsymbol{x}}$. Recall that in Section III-B $\hat{\boldsymbol{u}}\boldsymbol{G} = \hat{\boldsymbol{x}}$, which corresponds to $\lambda = 0$, indicates that it is very likely that the valid output $\hat{\boldsymbol{u}}$ has been found. Therefore, λ, as the difference between $\hat{\boldsymbol{u}}\boldsymbol{G}$ and $\hat{\boldsymbol{x}}$, can be viewed as *the number of unsatisfied constraints for valid polar decoding*. Based on this property, λ can be further used as the *metric to measure channel noise*. The relationship between λ and channel noise can be understood in an intuitive way. For

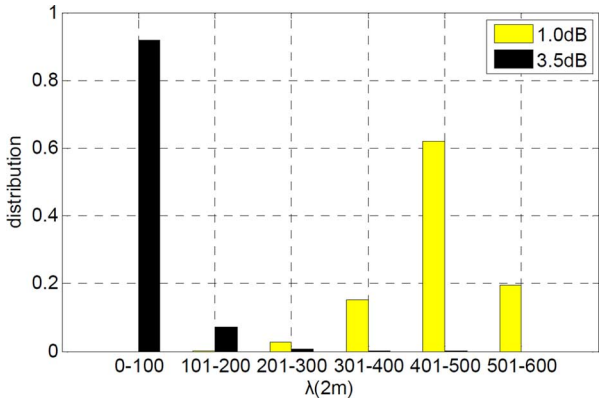


Fig. 11. The distribution of $\lambda(2\mathrm{m})$ of different channel SNR values for (1024, 512) polar BP decoding.

low SNR case, because a large portion of transmitted bits is corrupted due to the stronger noise, many unsatisfied constraints still remain even after certain iterations of decoding; for high SNR case, because many transmitted bits are not corrupted due to good channel condition, all the constraints can be quickly satisfied after few iterations. As a result, the value of λ after certain rounds of iterations can be used to roughly estimate channel noise condition. This leads to the following channel condition estimation approach described next.

1) Channel Condition Estimation Approach for Polar Codes: If λ at the 2m-th iteration, denoted as $\lambda(2m)$, is larger than a threshold value μ, then the channel condition can be assumed to be in low SNR region, otherwise in high SNR region.

Note that in the above approach λ is selected to be measured at the $2m$-th iteration. This is because the polar BP decoder typically needs $2m$ iterations to fully propagate initial channel LLRs in the decoder.

Simulation results verify the different distributions of $\lambda(2m)$ in low and high SNR regions. Fig. 11 shows the distributions of $\lambda(2m)$ at $\mathrm{SNR} = 1.0$ dB and 3.5 dB for (1024, 512) polar codes, respectively. Here $m = \log_2 1024 = 10$. From this figure it can be seen that $\mu = 100 \sim 200$ can distinguish low and high SNR regions very well. As a result, the different distributions of λ in different SNR regions show that $\lambda(2m)$ is a good metric to roughly estimate channel SNR level.

B. The Proposed Adaptive Early Stopping Criterion

Based on the proposed channel condition estimation approach, a *minLLR*-based adaptive stopping criterion can be developed. In the proposed adaptive strategy, the selection of β depends on the channel SNR condition, which is determined by the value of $\lambda(2m)$. As a result, a polar BP decoder with the adaptive stopping criterion is summarized in Scheme-C.

The decoding performance for polar (1024, 512) codes after using the proposed adaptive stopping criterion is shown in Fig. 7. It can be seen that, compared to the *minLLR* stopping criterion with fixed β, the adaptive minLLR-based stopping criterion, referred as *adaptive minLLR*, can achieve the same decoding performance with an additional 10% reduction in the number of iterations in low SNR regions (see Fig. 8).

Scheme-C (n, k) Polar BP decoder with adaptive stopping criterion:

1: Input: Same with Scheme-A, threshold parameter μ
2: Iteration process:
3: While $t < \max\_iter$ do
4: Update $L_{i,j}^t$ and $R_{i,j}^t$ for each node based on (4)
5: Update $\hat{\boldsymbol{u}}$, $\hat{\boldsymbol{x}}$, $\min$ *LLR*
6: if $t = 2m$ Calculate Hamming distance $\lambda(2m)$ between $\hat{\mathbf{u}}\mathbf{G}$ and $\hat{\mathbf{x}}$.
7: if $\lambda(2m) < \mu$ (high SNR) larger β is chosen for minLLR criterion
8: else (low SNR) smaller β is chosen for minLLR criterion
9: Check minLLR criterion(Scheme-B)
10: if (Satisified) $1 >$ Decoding is assumed to be successful
11: $2 >$ Output $\hat{\mathbf{u}}$ & Stop iteration
12: else $t = t + 1$ & Begin next iteration
13: Output : $\hat{\mathbf{u}}$

V. Hardware Architectures of Polar BP Decoder With Early Stopping Criteria

In this section, hardware architectures of BP decoders with stopping criteria are presented. Due to the generality of stopping criteria, they can be applied to any BP decoders. In this paper we use folded iteration-level overlapping decoders in our prior work [20] as the reference design.

A. Architecture of Folded Iteration-Level Overlapping Decoder

1) Processing Element (PE): As introduced in Section II-B, the LLR-based SMS algorithm is described by (1). In general, the four equations in (1) can be generalized as: $d = a + s^*\mathbf{sign}(b)\mathbf{sign}(c)\mathbf{min}(|b|,|c|)$ and $d = s^*\mathbf{sign}(a)\mathbf{sign}(b+c)\mathbf{min}(|a|,|b+c|)$. Based on these two kinds of computation, the basic processing element (PE) of polar BP decoder can be developed. For more details of the architecture of PE, the reader is referred to [20].

2) Folded Iteration-Level Overlapped Decoding Scheme: In the conventional polar BP decoder [19], $m = \log_2 n$ stages of PEs are used to develop the overall architecture. Fig. 12(a) shows the conventional BP decoding scheme for $n = 4$ polar code with $\max\_iter = 3$. Here C_j^p means the propagating messages are updated in the j-th stage of PEs during the p-th iteration. In addition, the arrows in Fig. 12(a) show the inherent data dependency in (1). Here the black and red arrows show the dependency within the same iteration and between consecutive iterations, respectively. From this example it can be seen that in

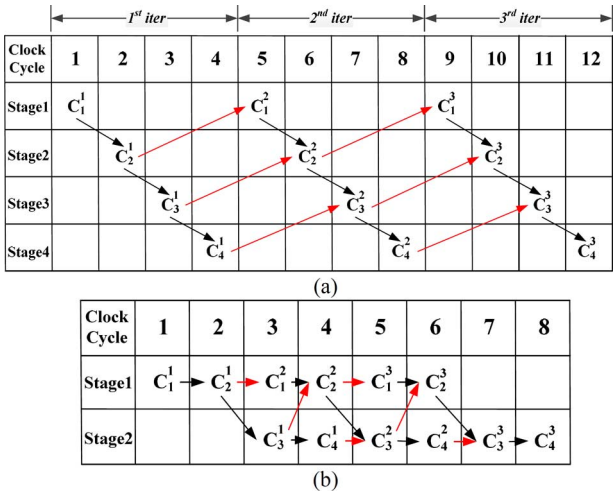


Fig. 12. (a) Original decoding scheme for $n = 8$ polar codes with $\max\_iter = 3$. (b) Folded iteration-level overlapping scheme. Cited from [20].

each cycle only one stage is activated while other $m-1$ stages are idle. This yields a hardware utilization rate of only $1/m$.

In our prior work [20], we showed that the low hardware utilization rate can be linearly improved by using folding technique to fully utilize idle hardware. In addition, multiple stages can be further activated in the same cycle without any violation of data dependency. This rescheduling approach is referred as iteration-level overlapping. Fig. 12(b) shows the new decoding scheme with 2-level folding and iteration-level overlapping techniques. As analyzed in [20], in general the iteration-level overlapping technique can reduce latency from $m^* \max\_iter$ cycles to $2^* \max\_iter + m - 2$ cycles. Meanwhile the folding technique can reduce number of stages of PEs from m to $m/2$. For the details of folded iteration-level overlapping scheme, the reader is referred to [20].

B. Hardware Architectures of the Proposed Stopping Criteria

1) Architecture of G-Matrix Stopping Criterion: The lines $13 \sim 19$ in Scheme-A describe ***G-matrix*** stopping criterion. Fig. 13 shows its corresponding hardware architecture. Here $\boldsymbol{G}$ matrix block performs multiplication with $\boldsymbol{G}$ matrix. Besides, the equality detector, which contains XNORs and AND trees, is used to determine whether $\widehat{\boldsymbol{x}'} = \hat{\boldsymbol{u}}\boldsymbol{G}$ is equivalent to $\hat{\boldsymbol{x}}$ or not.

2) Timing Analysis of G-Matrix Stopping Criterion: Fig. 14 illustrates the worst-case decoding scheme after applying ***G-matrix*** stopping criterion to Fig. 12(b). Here the dashed arrows in Fig. 14 represent the data dependencies between stages of PEs and components of stopping criterion. In the first cycle of the t-th iteration, stage 1 updates the propagating messages C_1^t and outputs $L_{i,1}^t$ and $R_{i,1}^t$. In the next cycle, $L_{i,1}^t$ and $R_{i,1}^t$ are added by the adder array and then $\hat{u}_i$ is determined. Next, $\hat{\boldsymbol{u}} = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_{16})$ is sent to $\boldsymbol{G}$ matrix block to calculate $\widehat{\boldsymbol{x}'} = \hat{\boldsymbol{u}}\boldsymbol{G}$. Similarly, after the last cycle of the t-th iteration, since C_4^t have been processed by stage 2, $L_{i,m+1}^t$ and $R_{i,m+1}^t$, as the left-to-right and right-to-left LLR values of $\hat{x}_i$, are output. Then in the next two cycles, $\hat{x}_i$ is calculated and $\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{16})$ is sent to an equality detector.

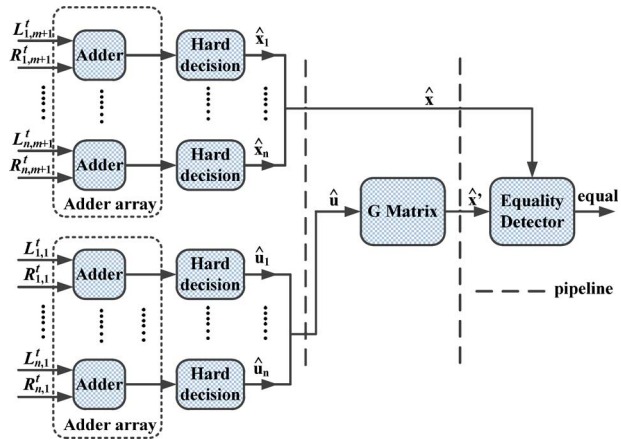


Fig. 13. Hardware architecture of ***G-matrix*** stopping criterion.

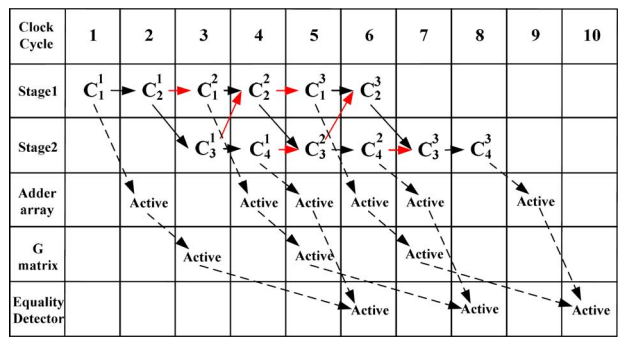


Fig. 14. Worst-case decoding scheme after applying ***G-matrix*** to Fig. 12(b).

For most polar codes with practical length ($n \leq 10000$), the critical path delays of the adder array, $\boldsymbol{G}$ matrix block and equality detector are always less than that of PE. Therefore, the use of ***G-matrix*** stopping criterion does not increase the critical path delay of the whole decoder. In addition, as illustrated in Fig. 14, compared to the case in Fig. 12(b), the use of ***G-matrix*** criterion only leads to an additional latency of two clock cycles. If we assume the whole decoding procedure is terminated at the v-th iteration, then the overall latency after using ***G-matrix*** criterion is $2v + m - 2 + 2 = 2v + m$ cycles.

3) Architecture of minLLR Stopping Criterion: The lines $6 \sim 11$ in Scheme-B describe *minLLR* stopping criterion. Fig. 15 shows the corresponding hardware architecture. Here the ABS block is used to obtain the absolute value of input, and threshold comparator, which contains comparators and AND tree, is used to compare the values of *minLLR* and threshold β.

4) Timing Analysis of minLLR Stopping Criterion: Fig. 16 illustrates the worst-case decoding scheme after applying *minLLR* stopping criterion to Fig. 12(b). Since the critical path delays of adder array and threshold comparator are also less than that of PE, the use of *minLLR* stopping criterion does not change the overall critical path delay. Furthermore, as illustrated in Fig. 16, the use of *minLLR* criterion does not change the overall latency as well. In general, if the decoding procedure is terminated at the v-th iteration, then the overall latency after using the *minLLR* criterion is $2v + m - 2$ cycles.

TABLE I
AVERAGE NUMBER OF ITERATIONS AND PERFORMANCE LOSS OF STOPPING CRITERIA FOR POLAR (1024, 512) BP DECODER WITH $\max\_iter = 40$

Stopping criterion	The proposed *G-matrix*			The proposed adaptive *minLLR (β=2.5/9.5 for low/high SNR)*		
SNR	Average No. of iterations	Iteration reduction	Perform degradation	Average No. of iterations	Iteration reduction	Perform degradation
2.5dB	30.8	23.0%	No	35.7	10.7%	No
3.0dB	26.1	34.7%	No	33.9	15.2%	No
3.5dB	23.0	42.5%	No	30.7	23.2%	<0.05dB

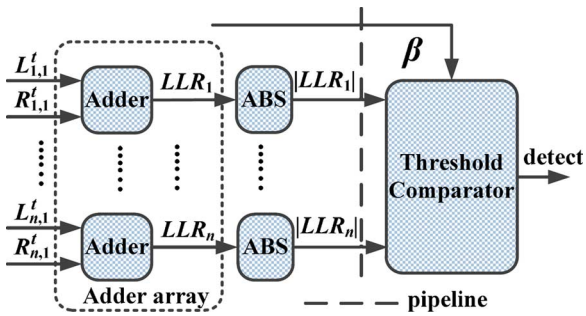


Fig. 15. Hardware architecture of *minLLR* stopping criterion.

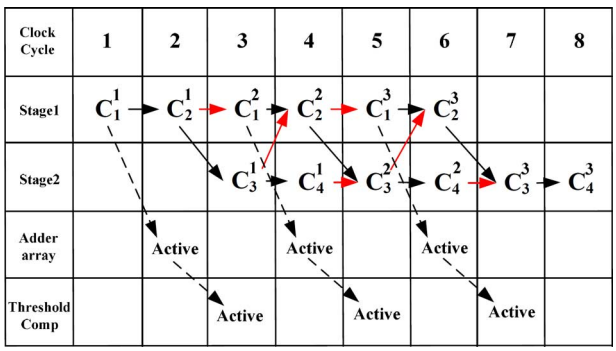


Fig. 16. Worst-case decoding scheme after applying *minLLR* to Fig. 12(b).

5) Architecture of Channel Condition Estimation Approach: In Section IV-A, channel condition estimation approach calculates the Hamming distance (HD) between $\hat{\boldsymbol{u}}\boldsymbol{G}$ and $\hat{\boldsymbol{x}}$. This computation is very similar to that in ***G-matrix*** criterion. As a result, the architecture of channel condition estimator can be directly developed by just replacing equality detector in Fig. 13 with a new Hamming distance (HD) measurement block. Notice that because the HD block contains XORs and adder tree, a one-stage pipeline is needed for high-speed design, which leads to extra one cycle to the entire computation procedure (see Fig. 17).

6) Timing Analysis for the Case With Very Large n: It should be noted that with n increases, the critical path delay of PE will not increase but all the other components of stopping criteria will. That means for a very large n, it is possible that the critical path delay of components of stopping criteria, such as $\boldsymbol{G}$ matrix, equality detector, threshold comparator and Hamming distance measure blocks, would be larger than that of PE. However, this problem can be solved by performing 1 or 2 stages pipelining to those components with only 1 or 2 extra clock cycles for the overall latency. Notice that in current design, 1-stage pipelining has been applied to Hamming distance measure block to reduce its critical path delay, and it shows that the use of pipelining only leads to 1 extra clock cycle to the overall latency (see Fig. 17). In summary, for the large n case, 1 or 2 stage pipelining to the key components of stopping criteria can guarantee the critical path of the overall decoder always locates in the PE. As a result, the use of stopping criterion will not affect the maximum clock frequency, but will only slightly increase the latency.

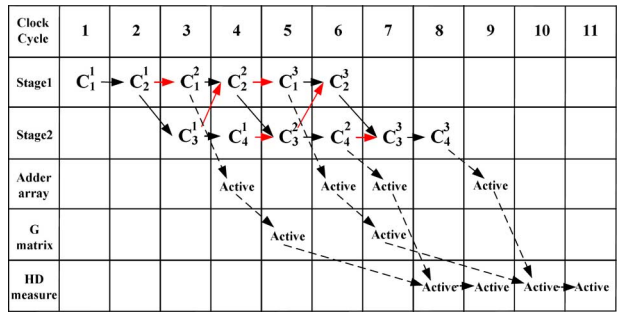


Fig. 17. Computing scheme of 1-stage pipelined channel condition estimator.

VI. HARDWARE ANALYSIS AND COMPARISON

In this section, we analyze the improvement on hardware performance of polar BP decoders with the use of the proposed early stopping criteria. Table I shows the reduction in the average number of iterations and performance loss for different early stopping criteria in different SNR regions. Here (1024, 512) polar BP decoder with $\max\_iter = 40$ is used as the original referenced decoder. From this table it can be seen that, the proposed criteria are very useful for reducing the number of iterations with keeping the same decoding performance. For ***G-matrix*** criterion it can save $23.0\% \sim 42.5\%$ iterations for SNR in the range of 2.5dB to 3.5dB. For adaptive *minLLR* criterion, it can save $10.7\% \sim 23.2\%$ iterations with negligible performance loss.

The RTL models of the proposed polar (1024, 512) BP decoders are developed with Verilog HDL. Here the I/O buffers and memories that store channel LLRs, decoded bits and hard/soft information are included in the hardware design. Then the designs are synthesized by Synopsys Design Compiler with FreePDK CMOS 45nm library. The supply voltage is 1.1 volts with typical timing model at 27 C. Table II list the critical path delays of key components in the proposed designs. From this table it can be seen that the longest path is still in the PE, hence the use of stopping criterion does not affect the maximum clock frequency.

Table III compares the hardware performance of the polar (1024, 512) BP decoder before and after using the proposed early stopping criteria. It can be seen that, after using the stopping criteria, the average decoding latency is reduced greatly

TABLE II
THE CRITICAL PATH DELAY OF KEY COMPONENTS IN STOPPING CRITERIA AND PE

Block	PE	G matrix	Equality detector	Threshold Comparator	Hamming Distance Measure (pipelined)
Location	PE	*G-matrix* *Channel condition estimator*	*G-matrix*	*minLLR*	*Channel condition estimator*
Critical path delay(*ns*)	1.9337	1.3868	0.7666	1.8381	0.9795

TABLE III
HARDWARE PERFORMANCE ANALYSIS OF (1024, 512) POLAR BP DECODER BEFORE AND AFTER USING STOPPING CRITERIA ($\max\_iter = 40$)

Design	Decoder without use of stopping criteria (fixed number of iter=40)	Decoder with *G-matrix*	Decoder with adaptive *minLLR*
Hardware Architecture	5-stage folded iteration-level overlapping architecture		
CMOS Technology	45nm		
Maximum Clock Frequency (MHz)	500		
Total Gate Counts	1920500	1961584	2018993
Average number of iterations @3.5dB	40	23.0	30.7
Average Latency (cycles) @3.5dB	88	56	73
Energy per bit (pJ/bit) @3.5dB	328	220	291
Average Throughput (Gbps) @3.5dB	2.9	4.5	3.5

with very small overhead (2% and 5% for ***G-matrix*** and adaptive *minLLR*, respectively). More importantly, the use of stopping criteria leads to great reduction in energy dissipation. Here energy per bit (EPB) [25] is used as the metric to evaluate required energy consumption for decoding process. The value of EPB is calculated as:

$$Energy\ per\ bit(EPB) = \frac{power \times decoding\ latency}{clock\ frequency \times n},$$

where the power of each design is reported by Design Compiler and n is 1024. Notice the unit of latency is clock cycle.

From Table III it can be seen that, the proposed ***G-matrix*** and adaptive *minLLR* stopping criteria can reduce EPB at 3.5dB by 32.9% and 11.2%, respectively. This means for decoding the same codeword, the decoder with the stopping criteria can save $11\% \sim 30\%$ energy without performance loss, as compared to the decoder with fixed number of iterations. As a result, the use of stopping criteria is a powerful and low-complexity solution for saving the energy of polar BP decoders.

In addition, because the average decoding latency is reduced significantly, the average decoding throughput increases. Here similar to the case for iterative LDPC decoder, the decoding throughput for iterative polar decoder is calculated as:

$$Decoding\ throughput = \frac{clock\ frequency \times k}{decoding\ latency}.$$

Compared to the original decoder with fixed number of iterations, the ***G-matrix*** and adaptive *minLLR* stopping criteria can improve the average throughput of polar BP decoder by 55.1% and 20.6% at 3.5dB, respectively.

In addition, Fig. 18 shows the distribution of throughput of BP decoders operated at 3.5dB. Here the distribution of throughput is derived from the distribution of number of iterations. It can be seen that, for the decoder with adaptive *minLLR* stopping criterion, the distribution on the range of $3.3 \sim 4.4$ Gbps is much larger than other cases, therefore the actual throughput is located in this range, which is consistent with Table III. For the decoders with ***G-matrix*** stopping criterion, the probability of being larger than 4.4Gbps is much larger than other cases. This indicates that the codewords are always decoded in a short time. As a result, the actual throughput is rarely degraded by the long decoding time of the successive codewords. In general, for the BP decoders with the stopping criteria, their high throughput can be guaranteed.

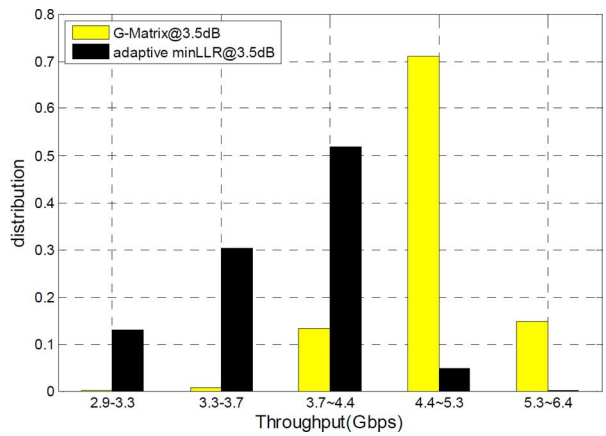


Fig. 18. Distribution of throughput with the use of stopping criteria for (1024, 512) polar codes.

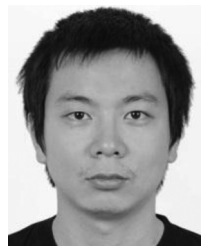
VII. CONCLUSION

This paper presents several novel stopping criteria and a channel condition estimation approach for polar BP decoders. To the best of our knowledge, this is the first work on stopping criteria for polar codes. Analysis shows that the use of stopping criteria can lead to significant improvements in hardware performance with small overhead.

REFERENCES

[1] E. Arikan, "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, 2009.

[2] E. Arikan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447–449, Jun. 2008.
[3] E. Arikan, "Polar codes: A pipelined implementation," in *Proc. 4th Int. Symp. Broadcast. Commun. (ISBC)*, Jul. 2010, pp. 11–14.
[4] C. Leroux, I. Tal, A. Vardy, and W. J. Gross, "Hardware architectures for successive cancellation decoding of polar codes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (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] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive cancellation decoder for polar codes," *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1378–1380, Dec. 2011.
[7] A. Mishra, A. Raymond, L. Amaru, G. Sarkis, C. Leroux, P. Meinerzhagen, A. Burg, and W. J. Gross, "A successive cancellation decoder ASIC for a 1024-bit polar code in 180 nm CMOS," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, 2012.
[8] 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.
[9] Z. Huang, C. Diao, J. Dai, C. Duanmu, X. Wu, and M. Chen, "An improvement of modified successive-cancellation decoder for polar codes," *IEEE Commun. Lett.*, vol. 7, no. 12, pp. 957–961, Jul. 2013.
[10] A. Pamuk and E. Arikan, "A two phase successive cancellation decoder architecture for polar codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2013, pp. 7–12.
[11] C. Zhang and K. K. Parhi, "Latency analysis and architecture design of simplified SC polar decoders," *IEEE Trans. Circuits Syst. II, Trans. Briefs*, vol. 61, no. 2, pp. 115–119, Feb. 2014.
[12] B. Yuan and K. K. Parhi, "Low-latency successive-cancellation polar decoder architectures using 2-bit decoding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 4, pp. 1241–1254, Apr. 2014.
[13] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE J. Sel. Areas Coummun.*, vol. 32, no. 5, pp. 946–957, 2014.
[14] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2011, pp. 1–5.
[15] K. Niu and K. Chen, "Stack decoding of polar codes," *Electron. Lett.*, vol. 48, no. 12, pp. 695–696, 2012.
[16] B. Li, H. Shen, and D. Tse, "An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2044–2047, Dec. 2012.
[17] A. Eslami and H. Pishro-Nik, "On finite-length performance of polar codes: stopping sets, error floor, and concatenated design," *IEEE Trans. Commun.*, vol. 61, no. 3, pp. 919–929, Mar. 2013.
[18] A. Pamuk, "An FPGA implementation architecture for decoding of polar codes," in *Proc. 8th Int. Symp. Wireless Commun. Syst. (ICWCS)*, Nov. 2011, pp. 437–441.
[19] B. Yuan and K. K. Parhi, "Architecture optimizations for BP polar decoders," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2013.
[20] B. Yuan and K. K. Parhi, "Architectures for polar BP decoder using folding," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2014.
[21] R. Y. Shao, S. Lin, and M. P. C. Fossorier, "Two simple stopping criteria for turbo decoding," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1117–1120, Aug. 1999.
[22] J. Li, X.-H. You, and J. Li, "Early stopping for LDPC decoding: convergence of mean magnitude (CMM)," *IEEE Commun. Lett.*, vol. 10, no. 9, pp. 667–669, Sep. 2006.
[23] W. J. Ebel, Y. Wu, and B. D. Woerner, "A simple stopping criterion for turbo decoding," *IEEE Commun. Lett.*, vol. 4, no. 8, pp. 258–260, 2000.
[24] Z. Wang and K. K. Parhi, "On-line extraction of soft decoding information and applications in VLSI turbo decoding," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 49, no. 12, pp. 760–769, Dec. 2002.
[25] T. Mohsenin, H. Shirani-mehr, and B. Baas, "Low power LDPC decoder with efficient stopping scheme for undecodable block," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 1780–1783.
[26] Z. Cui, L. Chen, and Z. Wang, "An efficient early stopping scheme for LDPC decoding," in *Proc. 13th NASA Symp. VLSI Des.*, 2007.
[27] R. E. Blahut, *Theory and Practice of Error-Control Codes*. Reading, MA, USA: Addison-Wesley, 1983.
[28] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
[29] N. Goela, S. B. Korada, and M. Gastpar, "On LP decoding of polar codes," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2010, pp. 1–5.
[30] A. Balatsoukas-Stimming, A. J. Raymond, W. J. Gross, and A. Burg, "Hardware architecture for list SC decoding of polar codes," *IEEE Trans. Circuits Syst. II, Trans. Briefs*, vol. 61, no. 8, pp. 609–613, Aug. 2014.
[31] B. Yuan and K. K. Parhi, "Low-latency successive-cancellation list decoders for polar codes with multi-bit decision," *IEEE Trans. VLSI Syst.*, 2014, doi: 10.1109/TVLSI.2014.2359793, preprint.



Bo Yuan received the B.S. degree in physics and M.S. degree in microelectronics from Nanjing University, Nanjing, China, in 2007 and 2010, respectively. He is now working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities. His research interests include VLSI architecture and algorithm design for high-speed low-power communication and digital signal processing systems.

Keshab K. Parhi (S'85–M'88–SM'91–F'96) received the B.Tech. degree from the Indian Institute of Technology, Kharagpur, in 1982, the M.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 1984, and the Ph.D. degree from the University of California, Berkeley, in 1988. He has been with the University of Minnesota, Minneapolis, since 1988, where he is currently Distinguished McKnight University Professor and Edgar F. Johnson Professor in the Department of Electrical and Computer Engineering. He has published over 500 papers, has authored the textbook *VLSI Digital Signal Processing Systems* (Wiley, 1999) and coedited the reference book *Digital Signal Processing for Multimedia Systems* (Marcel Dekker, 1999). His research addresses VLSI architecture design and implementation of signal processing, communications and biomedical systems, error control coders and cryptography architectures, high-speed transceivers, stochastic computing, secure computing, and molecular/DNA computing. He is also working on intelligent classification of biomedical signals and images, for applications such as seizure prediction and detection, schizophrenia classification, biomarkers for mental disorders, brain connectivity, and screening of fundus and optical coherence tomography (OCT) images for ophthalmic abnormalities.

Dr. Parhi is the recipient of numerous awards including the 2013 Distinguished Alumnus Award from IIT, Kharagpur, India, the 2013 Graduate/Professional Teaching Award from the University of Minnesota, the 2012 Charles A. Desoer Technical Achievement award from the IEEE Circuits and Systems Society, the 2004 F. E. Terman award from the American Society of Engineering Education, the 2003 IEEE Kiyo Tomiyasu Technical Field Award, the 2001 IEEE W. R. G. Baker prize paper award, and a Golden Jubilee medal from the IEEE Circuits and Systems Society in 2000. He has served on the editorial boards of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I and PART II, *VLSI Systems, Signal Processing, Signal Processing Letters,* and *Signal Processing Magazine*, and served as the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I (2004–2005 term), and currently serves on the Editorial Board of the *Journal of VLSI Signal Processing*. He has served as technical program co-chair of the 1995 IEEE VLSI Signal Processing workshop and the 1996 ASAP conference, and as the general chair of the 2002 IEEE Workshop on Signal Processing Systems. He was a distinguished lecturer for the IEEE Circuits and Systems society during 1996–1998. He served as an elected member of the Board of Governors of the IEEE Circuits and Systems society from 2005 to 2007.