ARCHITECTURE OPTIMIZATIONS FOR BP POLAR DECODERS

Bo Yuan and Keshab K. Parhi
Department of Electrical and Computer Engineering, University of Minnesota Twin Cities

ABSTRACT

Polar codes have emerged as important channel codes because of their capacity-achieving property. For low-complexity polar decoding, hardware architectures for successive cancellation (SC) algorithm have been investigated in prior works. However, belief propagation (BP)-based architectures have not been explored in detail. This paper begins with a review of min-sum (MS) approximated BP algorithm, and then proposes a scaled MS (SMS) algorithm with improved decoding performance. Then, in order to solve long critical path problem in the SMS algorithm, we propose an efficient critical path reduction approach. Due to its generality, this optimization method can be applied to both of SMS and MS algorithms. Compared with the state-of-the-art MS decoder, the proposed (1024, 512) SMS design can lead to 0.5dB extra decoding gain with the same hardware performance. Besides, the proposed optimized MS architecture can also achieve more than 30% and 80% increase in throughput and hardware efficiency, respectively.

Index Terms—Polar codes, VLSI, belief propagation, scaled min-sum, critical path reduction

1. INTRODUCTION

As the first provable capacity-achieving error correction codes (ECC) [1], polar codes have become one of the most favorable topics in information theory. To date, many researchers in information theory community have investigated various aspects of polar codes, ranging from code construction [2-8] to efficient decoding algorithms [9-14]. However, with the exception of [15-19], not many efforts have addressed the hardware design of polar decoders. In [15-18], several successive cancellation (SC)-based architectures were proposed. By applying local optimal decoding schemes, these SC decoders can achieve good error-correcting capability with low complexity. However, for high-speed real-time applications, this serial decoding scheme will become the potential bottleneck. On the other hand, the belief propagation (BP) algorithm [14] has particular advantages with respect to parallelism and low latency. However, current BP-based decoders [19] are still less competitive for practical applications due to their insufficient error-correcting capability.

In this paper, we first review the current non-scaled min-sum (MS) approximated BP algorithm. Then, similar to the approach used in LDPC decoding [20], we propose a performance-improved scaled min-sum (SMS) polar decoding algorithm. Then, to overcome the long critical path problem in the SMS algorithm, we propose an efficient critical path reduction approach at the architecture level. Due to its generality, this optimization method can be applied to both SMS and MS algorithms. Compared with the state-of-the art MS decoder, the proposed (1024, 512) SMS design can achieve 0.5dB decoding gain with the same hardware performance. In addition, the proposed optimized MS architecture can also achieve more than 30% and 80% increase in throughput and hardware efficiency, respectively.

This paper is organized as follows. Section 2 presents a brief review of the current MS algorithm. The proposed SMS algorithm is presented in Section 3. Section 4 proposes an efficient critical path reduction approach, and hardware architectures of optimized SMS and MS decoding based on this approach. Performance characteristics of different polar decoders are compared in Section 5. Section 6 draws conclusions.

2. REVIEW OF BP DECODING FOR POLAR CODES

2.1. Polar codes

Polar codes are derived from the concept of channel polarization. As proved in [1], with some recursive encoding approach, the reliability of decoded bits will be polarized based on their positions at the codeword. Therefore, a good polar code can be constructed by assigning information bits over the reliable positions, and setting "0" bits over the highly unreliable positions. In general, these "0" bits are called "frozen" bits while the bits in the source data are called "free" bits. For the details of polar encoding, the reader is referred to [1].

2.2. Current BP decoding with min-sum approximation

Derived from factor graph theory [21], the belief propagation (BP) algorithm can be applied for polar decoding [14]. Generally, an (n, k) polar code (n=2^m) can be iteratively decoded via an m-stage factor graph network consisting of (m+1)n nodes. Each node (i, j) is associated with two types of likelihood message: left-to-right and right-to-left. In BP decoding procedure, these messages are propagated and updated between adjacent nodes.

Fig. 1(a) shows an example BP factor graph for the case of (8, 4) polar code. Here the graph network has a total of m=$\log_2 8$=3 stages. Each stage consists of n/2=4 processing elements (PEs) (see Fig. 1(b)), which are used for updating the propagating messages. To avoid overflow, these updates are always performed in logarithmic domain. Therefore, the propagating messages are based on logarithmic likelihood ratio (LLR) form, and are updated using equations (1). Here $L_{i,j}^t$ and $R_{i,j}^t$ represent left and right propagating messages, and t is the current iteration index. Notice that at each iteration t=0,1,2,3…, depending on whether i is a frozen position or not, $R_{i,0}^t$ will be set either as a large constant or 0, respectively.

$$
\begin{aligned}
L_{i,j}^t &= sign(L_{i,j+1}^{t-1})sign(L_{i+n/2^j,j+1}^{t-1}+R_{i+n/2^j,j}^t)\min\left(\left|L_{i,j+1}^{t-1}\right|,\left|L_{i+n/2^j,j+1}^{t-1}+R_{i+n/2^j,j}^t\right|\right)\\
L_{i+n/2^j,j}^t &= L_{i+n/2^j,j+1}^{t-1}+sign(L_{i,j+1}^{t-1})sign(R_{i,j}^t)\min\left(\left|L_{i,j+1}^{t-1}\right|,\left|R_{i,j}^t\right|\right)\\
R_{i,j+1}^t &= sign(R_{i,j}^t)sign(L_{i+n/2^j,j+1}^{t-1}+R_{i+n/2^j,j}^t)\min\left(\left|R_{i,j}^t\right|,\left|L_{i+n/2^j,j+1}^{t-1}+R_{i+n/2^j,j}^t\right|\right)\\
R_{i+n/2^j,j+1}^t &= R_{i+n/2^j,j}^t+sign(L_{i,j+1}^{t-1})sign(R_{i,j}^t)\min\left(\left|L_{i,j+1}^{t-1}\right|,\left|R_{i,j}^t\right|\right). \qquad (1)
\end{aligned}
$$

Based on equations in (1), the likelihood messages can be propagated and updated iteratively in the factor graph. After the decoder reaches maximum iteration number (*max_iter*), node (*i*, 1) will output the decoded bits based on hard decision of messages.

It should be noted that the equations in (1) represent the min-sum approximation of the BP algorithm. Compared with the original BP algorithm, this approximated version is more suitable for hardware implementation [19]. However, its error-correcting performance is degraded due to the approximation. In the next section, this problem is addressed further.

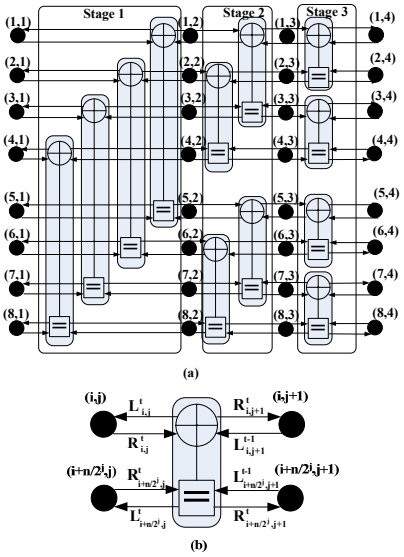

Fig. 1. (a) Factor graph of (8, 4) polar code. (b) Diagram of PE.

3. PROPOSED SCALED MIN-SUM ALGORITHM

As mentioned in Section 2.2, the MS algorithm described by (1) has some inherent performance disadvantages due to the approximation (see Fig. 2). In order to avoid performance loss, similar to the approach used in LDPC decoding [20], we propose to introduce a scaling parameter *s* to offset the approximation error: For each time of min-sum operation, the output will be scaled by *s*. Accordingly, the original non-scaled MS algorithm described by (1) is modified to a scaled MS (SMS) version described by (2).

$$L^{t}_{i,j} = s * sign(L^{t-1}_{i,j+1}) sign(L^{t-1}_{i+n/2^{j},j+1} + R^{t}_{i+n/2^{j},j}) \min\left(\left|L^{t-1}_{i,j+1}\right|, \left|L^{t-1}_{i+n/2^{j},j+1} + R^{t}_{i+n/2^{j},j}\right|\right)$$

$$L^{t}_{i+n/2^{j},j} = L^{t-1}_{i+n/2^{j},j+1} + s * sign(L^{t-1}_{i,j+1}) sign(R^{t}_{i,j}) \min\left(\left|L^{t-1}_{i,j+1}\right|, \left|R^{t}_{i,j}\right|\right)$$

$$R^{t}_{i,j+1} = s * sign(R^{t}_{i,j}) sign(L^{t-1}_{i+n/2^{j},j+1} + R^{t}_{i+n/2^{j},j}) \min\left(\left|R^{t}_{i,j}\right|, \left|L^{t-1}_{i+n/2^{j},j+1} + R^{t}_{i+n/2^{j},j}\right|\right)$$

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

As shown in Fig. 2, the introduction of scaling parameter helps improve the decoding performance greatly. For the example (1024, 512) polar code with *max_iter*=60, the proposed SMS algorithm with *s*=0.9375 can obtain an extra 0.5 dB decoding gain over its non-scaled counterpart. In that case, the SMS algorithm can achieve a similar error-correcting performance with the original BP and SC algorithms. Notice that since $s=0.9375=1-2^{-4}$, the scaling operation can be implemented with a simple shift-addition circuit.

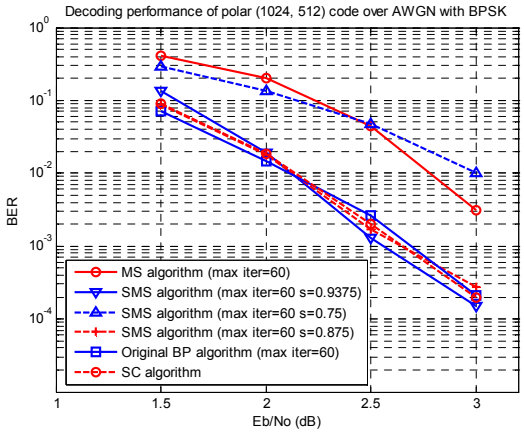

Fig. 2. Performance of different polar decoding algorithms.

4. THE PROPOSED HARDWARE ARCHITECTURES

4.1. Long critical path problem of SMS algorithm

As mentioned in Section III, different from the MS algorithm, the SMS algorithm uses a scaling parameter *s* to offset the error in the min-sum approximation. Although the introduction of *s* can improve error-correcting performance greatly, this extra scaling operation increases the critical path of the SMS decoder. In this subsection, the effect of this increase is analyzed in detail.

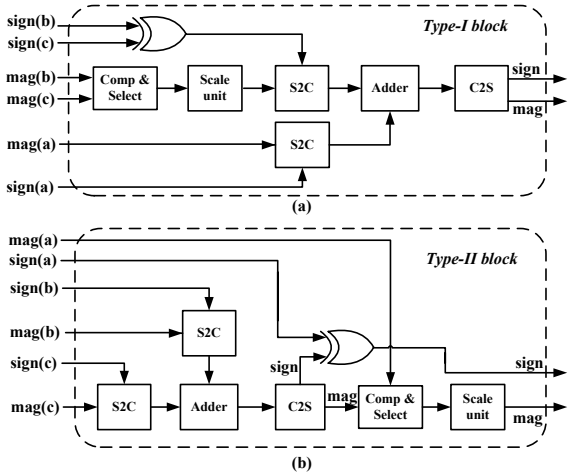

Fig. 3. Architecture of (a) Type-I block (b) Type-II block.

Recall that the LLR computation of the SMS algorithm is described by (2). In general, these four equations can be categorized into two types: Type-I $d=a+s*\mathbf{sign}(b)\mathbf{sign}(c)\mathbf{min}(|b|,|c|)$ and Type-II $d=s*\mathbf{sign}(a)\mathbf{sign}(b+c)\mathbf{min}(|a|,|b+c|)$. Accordingly, the high-level architectures of these two types of computation can be developed and are shown in Fig. 3. Here scale unit is the block that implements the scaling function. Besides, S2C unit carries out the conversion of number representation from sign-magnitude (SM) form to 2's complement form, and C2S performs the inverse conversion. Fig. 4 shows the architectures of these functional units with *q*-bit quantization. It can be seen that the S2C, C2S and scale units have the similar critical path delay ($\approx T_{adder}$). Since the critical

path delay of "comparator & selector" unit is similar to that of an adder, therefore, according to Fig. 3, the critical path delay of Type-I or Type-II block is approximately equal to $5T_{adder}$.

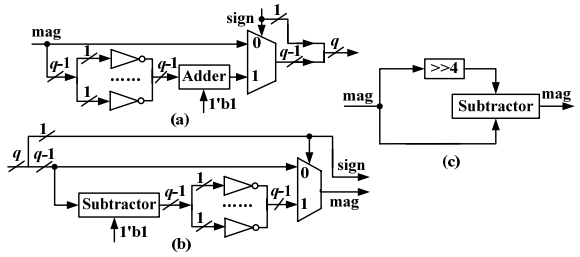

Fig. 4. Inner architecture of (a) S2C unit (b) C2S unit (c) scale unit (s=0.9375=1-2^{-4}).

Notice that for original non-scaled MS algorithm described by (1), its critical path delay is only about 5-1=4T_{adder} since the scale unit in Fig. 3 is not used in this case. Therefore, the critical path delay of SMS polar decoder will be at least 25% larger than the MS decoder. In other words, although the SMS algorithm can provide better error-correcting performance, the penalty on long critical path makes it still less competitive with respect to hardware performance. In the next subsection, we propose an efficient optimization method to reduce its critical path.

4.2. The proposed critical path-reduction approach for SMS

Recall the computation in SMS are $d=a+s*\mathbf{sign}(b)\mathbf{sign}(c)\mathbf{min}(|b|,|c|)$ and $d=s*\mathbf{sign}(a)\mathbf{sign}(b+c)\mathbf{min}(|a|,|b+c|)$, which contain addition, comparison and scaling operations. According to Fig. 3, the critical path of the Type I and II blocks is mainly from the addition operations used in the S2C unit, non-constant adder and C2S unit. Intuitively, the timing cost of this addition is too high since nearly two-third of the critical path is not from actual addition, but from the conversion of number representation. In this subsection, we show how unnecessary latency in the S2C conversion operation can be eliminated leading to a reduced critical path.

Without loss of generality, we denote the targeted addition operation as $z=x+y$, where x, y and z are represented in sign-magnitude (SM) form. Before x and y are input into the non-constant adder in Fig. 3, it is clear that they must first be converted into 2's complement form by S2C unit. With the observation on S2C conversion scheme in Fig. 4(a), it can be discovered that the output from constant adder in S2C unit will be selected only when the input is negative ($sign$=1). For the case of positive input ($sign$=0), the constant adder, which dominates the critical path in the S2C unit, will not affect the final output. This observation suggests that, if we can find a new constant-adder-free S2C conversion scheme for $z=x+y$, the new S2C unit, as well as the overall $z=x+y$ operation, will have a much shorter critical path.

To explore this possibility, we revisit the inner procedure of SM-based $z=x+y$ operation. This operation involves S2C and C2S conversions, and these conversions depend on the sign part of x and y; therefore, four possible cases are discussed.

4.2.1 x is positive and y is positive

When both of x and y are positive, their sign-magnitude forms are just the same with 2's complement forms. As a result, the constant adder in S2C unit will not be used in this case.

4.2.2 x is negative and y is negative

When both of x and y are negative, $z=x+y=-((-x)+(-y))$, thus the magnitude part of z is just the sum of magnitude of x and y, while the sign part of z is always negative(=1). Since $-x$ and $-y$ are positive, according to Section 4.2.1, their sum does not need the constant adder in S2C unit. Therefore, in order to sum negative SM-based x and y, S2C unit only needs to change the sign of x and y and retain the magnitude part. Similarly, C2S unit only needs to change the sign part of the output from a non-constant adder. As a result, the constant adder in S2C unit can also be saved in this case.

4.2.3 x is positive and y is negative

When x is positive and y is negative, because x has the same sign-magnitude and 2's complement form, we only need to perform representation conversion of y. In this case, for functional validity, the constant addition for S2C conversion is required; however, we can merge this constant addition into the outer non-constant adder. Recall that the mission of constant adder in S2C unit is just to sum an LSB-positioned "1" with bit-inversed y (see Fig.4 (a)). Therefore, if we replace 1-bit half-adder (HA) in the original q-bit non-constant adder (Fig. 5(a)) by a 1-bit full-adder (FA) (Fig. 5(b)), the constant "1" can still be summed up when it is selected as the carry input of the full-adder. As a result, the constant adder can now be removed from S2C unit without any functional invalidity.

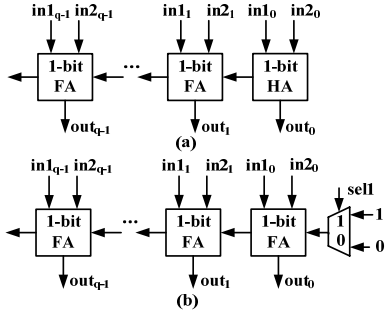

Fig. 5. (a) original non-constant adder (b) modified adder(mAdder).

4.2.4 x is negative and y is positive

Due to the symmetry of addition and generality of x and y, this case is similar to 4.2.3.

Summarizing the above four cases, it can be concluded that, instead of using constant adder in S2C unit, the SM-based addition $z=x+y$, can still be accurately carried out with slight modification of S2C unit, non-constant adder and C2S unit. The architectures of these modified units, denoted as mAdder, mS2C, and mC2S, are shown in Fig. 5(b), Fig. 6(a) and Fig. 6(b), respectively. Then, based on these basic units, the hardware architectures of modified Type-I and Type-II blocks for SMS algorithm can now be developed (see Fig. 7). From Fig. 7 it can be seen that the critical path of modified Type-I or Type-II blocks is about 4T_{adder}, which is the same as that of the non-scaled MS decoder.

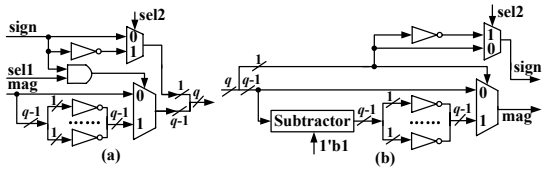

Fig. 6. Architecture of (a) mS2C unit (b) mC2S unit.

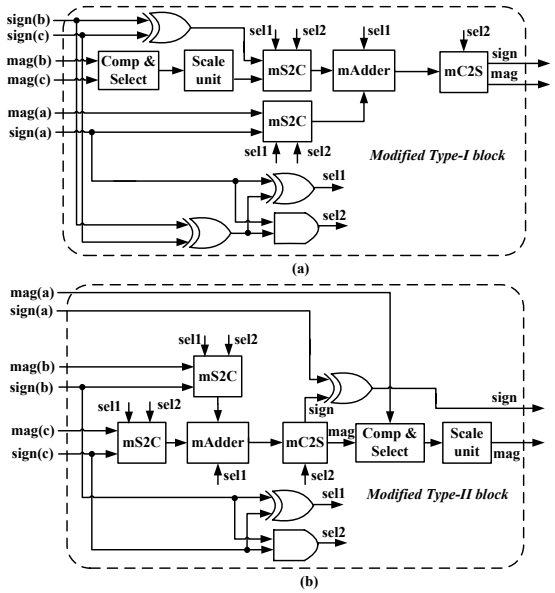

Fig. 7. (a) Modified Type-I block (b) Modified Type-II block.

4.3. Extending the proposed method to MS algorithm

In Section 4.2 we presented a critical path reduction approach for the SMS algorithm. Since this method is a general solution that optimizes SM-based $z=x+y$ operation, it can also be applied to the non-scaled MS algorithm in equations (1). In that case, the corresponding modified Type-I and Type-II blocks for MS algorithm can be easily derived by just removing the scale unit in Fig. 7. As a result, the critical path in this scenario will be reduced from $4T_{adder}$ to $3T_{adder}$.

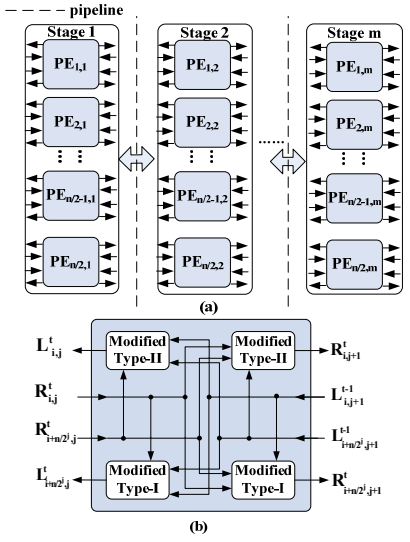

Fig. 8. (a) The overall architecture (b) Architecture of PE.

4.4. Overall architecture

Based on the aforementioned modified Type-I and Type-II blocks, the systolic polar decoding architecture can now be developed. Fig. 8(a) shows the overall architecture for (n, k) polar code, which consists of $m=\log_2 n$ stages. Between adjacent stages register-based pipelines are inserted to store propagating messages. In each stage, there are $n/2$ processing elements (PEs), and the inner architecture of a PE is shown in Fig. 8(b). Here the designs of the modified Type-I and Type-II blocks depend on the choice of algorithm, which has been described in Section 4.2 for the SMS algorithm and in Section 4.3 for the non-scaled MS algorithm.

On the aspect of decoding procedure, in the j-th cycle of each iteration, the PEs of stage-j are activated to update propagating messages. Therefore, a total of m cycles are required for one iteration. Considering maximum iteration number is *max_iter*, the total decoding latency will be *max_iter*m* cycles.

5. PEFORMANCE ANAYLSIS AND COMPARSION

In this section, performance of different BP-based polar decoding architectures is analyzed. Table 1 shows the performance of the proposed scaled-MS (SMS) decoder and reported MS design [19] for (1024, 512) polar code. Because SMS algorithm can obtain extra decoding gain over its non-scaled counterpart, for fair comparison, in this table we also list the MS decoder with the proposed optimization described in Section 4.3.

From Table 1 it can be seen that the proposed SMS decoder can obtain extra 0.5 dB decoding gain over the state-of-the-art MS design with the same hardware performance. In the non-scaled scenario, our optimized MS architecture can also achieve more than 30% and 80% increase in throughput and hardware efficiency, respectively. Therefore, the proposed two architectures are good candidates for high-performance polar decoder design.

Table 1. Performance of different (1024, 512) polar decoders (n=1024, m=10, *max_iter*=60, s=0.9375 with q-bit quantization)

Arch.	Proposed-SMS (Section 4.2)	Proposed-MS (Section 4.3)	MS [19]
Algorithm	Scaled-MS	MS	MS
# of PE	5120	5120	5120
Gate count in 1 PE	~40q	~30q	~46q
# of REG*	~24576q	~24576q	~24576q
Total gate count†	~278528q	~227328q	~309248q
Critical path	~4T_{adder}	~3T_{adder}	~4T_{adder}
Latency	600 cycles	600 cycles	600 cycles
Throughput (normalized)	1	1.33	1
Hardware efficiency (normalized)‡	1.11	1.81	1
Decoding gain over MS	0.5dB	0	0

* Here REG indicates 1-bit register.
† 1-bit REG is converted to three 2-input XOR according to [22].
‡ Hardware efficiency is the ratio of throughput to total gate count.

6. CONCLUSION

This paper presents a novel SMS polar decoding algorithm. With the proposed critical path reduction method, optimized SMS and MS architectures are developed. Comparison results show that the proposed architectures have significant advantages with respect to hardware performance and error-correcting capability.

7. 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, July 2009.

[2] S. B. Korada, E. Sasoglu, and R. Urbanke, "Polar Codes: Characterization of Exponent, Bounds, and Constructions," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6253-6264, Dec. 2010.

[3] R. Mori and T. Tanaka, "Performance of polar codes with the construction using density evolution," *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 519-521, July 2009.

[4] I. Tal and A. Vardy, "How to construct polar codes," arXiv: 1105.6164v1, May 2011.

[5] E. Arikan, and E. Telatar, "On the rate of channel polarization," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pp. 1493-1495, July 2009.

[6] R. Pedarsani, S. H. Hassani, I. Tal, and E. Telatar, "On the construction of polar codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pp. 11-15, July 2011.

[7] S. Cayci, O. Arikan, and E. Arikan, "Polar code construction for non-binary source alphabets," in *Proc. 20th Signal Processing and Communications Applications Conference (SIU)*, pp. 1-4, April 2012.

[8] N. Hussami, S. B. Korada, and R. Urbanke, "Performance of polar codes for channel and source coding," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pp. 1488-1492, July 2009.

[9] A. Eslami and H. Pishro-Nik, "On Bit Error Rate Performance of Polar Codes in Finite Regime," in *Proc. 48th Annual Allerton Conference on Communication, Control, and Computing*, pp. 188-194, Sept. 2010.

[10] 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.

[11] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pp. 1-5, July 2011.

[12] K. Niu and K. Chen, "Stack decoding of polar codes," *Elect. Lett.*, vol. 48, no. 12, pp. 695-696, June 2012.

[13] E. Arikan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447-449, June 2008.

[14] E. Arikan, "Polar codes: A pipelined implementation," in *Proc. 4th Int. Symp. on Broad. Commun. ISBC 2010*, pp. 11-14, July 2010.

[15] C. Leroux, I. Tal, A. Vardy, and W. J. Gross, "Hardware Architectures for Successive Cancellation Decoding of Polar Codes," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1665-1668, May 2011.

[16] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for polar codes," *IEEE Trans. Signal Processing*, vol.61, no.2, pp. 289-299, Jan. 2013.

[17] C. Zhang, B. Yuan, and K. K. Parhi, "Reduced-latency SC polar decoder architectures," in *Proc. IEEE Int. Conf. Commun.*, pp. 3471-3475, June 2012.

[18] C. Zhang and K. K. Parhi, "Low-latency sequential and overlapped architectures for successive cancellation polar decoder," *IEEE Trans. Signal Processing*, vol. 61, 2013.

[19] A. Pamuk, "An FPGA implementation architecture for decoding of polar codes," in *Proc. 8th Int. Symp. on Wireless Commun. Syst. (ICWCS)*, pp. 437-441, Nov. 2011.

[20] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X. Y. Hu, "Reduced complexity decoding of LDPC codes," *IEEE Trans. on Communications*, vol. 53, no. 8, pp. 1288-1299, Aug. 2005.

[21] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498-519, Feb. 2001.

[22] X. Zhang and F. Cai, "Efficient partial-parallel decoder architecture for quasi-cyclic nonbinary LDPC codes," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 58, no. 2, pp. 402-414, Feb. 2011.