ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

High speed Belief Propagation Polar Decoder using Overlapping Scheduling Schemes

[1]Jishor C K, [2]Sindhu T.V
[1]PG Scholar (VLSI Design), [2]Assistant Professor, ECE
[1, 2] IES College of Engineering
Thrissur, India
[1]jishorckishor@gmail.com, [2]sindhugie@gmail.com

***Abstract*—Polar codes have become increasingly popular recently because of their capacity achieving property. Successive cancellation decoding (SCD) and belief propagation decoding (BPD) are two major approaches for decoding polar codes. Due to the serial nature of the algorithm, SCD suffers from long latency, although it requires less computation as compared with BPD.On the other hand, polar BP decoders have the intrinsic advantage of parallel processing. Therefore, compared with their SC counterparts, polar BP decoders are more attractive for low-latency applications. However, due to their iterative nature, the required latency and energy dissipation of BP decoders increase linearly with the number of iterations. To reduce the computation complexity of several methods have been proposed.**

***Index Terms*—Polar code, BP decoder, energy efficient.**

I. INTRODUCTION

Polar code, which is discovered by Arikan recently, is a major breakthrough in coding theory. It is the first known family of error correction codes achieving the Shannon capacity for binary-input discrete memoryless channels. Main attraction polar code is its low coding complexity and channel achieving capacity. Besides achieving the capacity for binary-input symmetric memoryless channels, polar codes were also proved in to be able to achieve the capacity for any discrete and continuous memoryless channel. Moreover, an explicit construction method for polar codes was provided and it was shown that they can be efficiently encoded and decoded with complexity O (n log n), where n is the code length. Since then, polar codes have become one of the most popular topics in information theory and have attracted a lot of attention. Polar code is a best candidate for error correction codes in next generation communication system.

A number of decoding methods have been proposed for polar codes, and among these, successive cancellation decoding (SCD) and belief propagation decoding (BPD) are the two most popular methods. Due to the serial nature of the algorithm, SCD suffers from long latency, although it requires less computation as compared with BPD. Several methods have been proposed to reduce the latency of SC decoders to achieve a high throughput. Moreover, list decoding and stack decoding, which are based on SCD, have been proposed to improve the error-correcting performance for polar codes with short code lengths.

On the other hand, polar BP decoders have the intrinsic advantage of parallel processing. Therefore, compared with their SC counterparts, polar BP decoders are more attractive for low-latency applications. However, due to their iterative nature, the required latency and energy dissipation of BP decoders increase linearly with the number of iterations. The requirement of a large number of iterations results in high computation complexity, and hence makes BPD less attractive than its SC counterpart. To reduce the computation complexity of several methods have been proposed.

II. POLAR CODE OVERVIEW

Polar codes are linear block codes based on the phenomenon of channel polarization, in which

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

individual channels are recursively combined and split, such that their mutual information tends toward either 1 or 0. In other words, some of these channels become completely noise-free, while the others become completely noisy. Furthermore, the fraction of noiseless channels tends toward the capacity of the underlying binary symmetric channels.

Polar codes are constructed based on of the polarization effect to achieve the capacity of symmetric channel. An (n, k) polar code is constructed by assigning k information bits and (n−k)'0's at more reliable positions and unreliable positions, respectively. Those fixed '0' bits are usually referred as frozen bits.

Then message bits including frozen bits and information bits are denoted as u in this paper. An (n,k) polar code can be generated in two steps. First, an n-bit message u is constructed by assigning the k reliable and (n−k) unreliable positions as information bits and frozen bits, respectively. The (n−k) frozen bits are forced to 0 and form the frozen set A^C The n-bit transmitted codeword *x* is the product of u and the generator matrix G, where $G=F^{\oplus m}$. $F^{\oplus m}$ is the *m*-th Kronecker power of $F=\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$ and $m=\log_2 n$. Fig.1 shows the encoding signal flow graph for n = 8 polar codes, where the "⊕" sign represents the XOR operation.

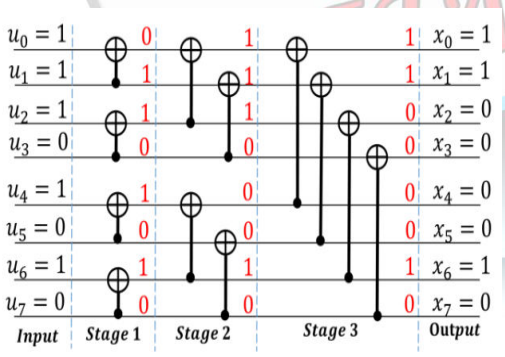

Fig. 1. Encoding signal flow graph of (8.4) polar code

III. BELIEF PROPAGATION DECODING

The BP decoding for polar codes is based on the factor graph representation of the codes. The factor graph for an (n,k) polar code($n=2^m$) is an m-stage network, which consists ofn.(m+1) nodes. Each node is associated with two types of messages: left-to-right messages L and right-to-left messages R. Fig.2 shows the four nodes involved in one basic computation element BE2.

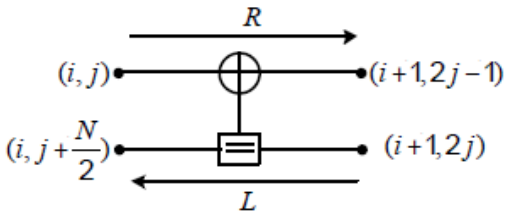

Fig 2.Factor graph of basic computation element BE2 in BP polar decoders.

Firstly, the belief (indicated as log likelihood ratio, LLR) of each node is initiated. The left most source vector nodes in Fig.2 are initiated with zero or positive infinity as Eq.(1) and the right most code word nodes are initiated with channel output LLRs as Eq.(2). Other nodes are initiated with zero

$$R_{i,j} = \begin{cases} 0 \; if \; j \in A \\ \infty \; if \; j \in A^c \end{cases} \quad (1)$$

$$.L_{n+1,j} = ln\frac{P(y_i|x_i=0)}{P(y_i|x_i=1)} \quad (2)$$

In BP decoding, LLRs are passed iteratively from left to right and then from right to left through basic computation elements BE2s to compute the likelihood of information bits. Fig.3 shows an example of a 3-stage factor graph for $n = 8$ polar codes. Here each stage consists of $n/2 = 4$ basic computation elements (BEs). During the BP decoding procedure, these messages are propagated and updated among adjacent nodes using the min-sum updating rule as shown by the following equations

$$L_{i,j} = g(L_{i+1,2j-1}, L_{i+1,2j} + R_{i,j+N/2}) \quad (3)$$

$$L_{i,j+N/2} = g(R_{i,j}, L_{i+1,2j-1}) + L_{i+1,2j} \quad (4)$$

$$R_{i+1,2j-1} = g(R_{i,j}, L_{i+1,2j} + R_{i,j+N/2}) \quad (5)$$

$$R_{i+1,2j} = g(R_{i,j}, L_{i+1,2j-1}) + R_{i,j+N/2} \quad (6)$$

Where

$$g(x,y) = \log(\cosh((x+y)/2)) - \log(\cosh((x-y)/2)) \quad (7)$$

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

it can be simplified as :

$$g(x,y) \simeq 0.9.sign(x)sign(y).min(|x|,|y|) \quad (8)$$

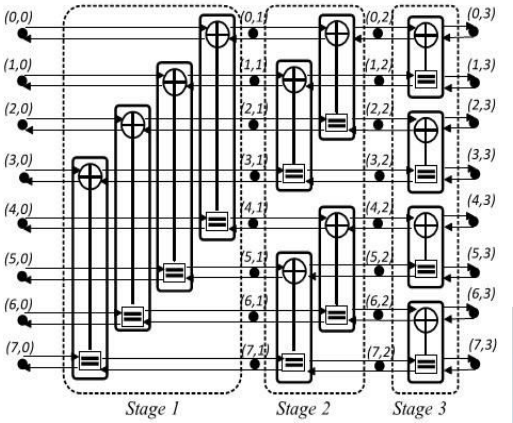

Fig3. Factor graph for n=8 polar code

After t iterations, the decision can be made based on the final LLR results of source vector nodes as

$$u_j = \begin{cases} 0 & if R_{i,j} \geq 0 \\ 1 & else \end{cases} \quad (9)$$

Due to their iterative nature, the required latency and energy dissipation of BP decoders increase linearly with the number of iterations. The requirement of a large number of iterations results in highcomputation complexity, and hence makes BPD less attractive than its SC counterpart. To reduce the computation complexity of several methods have been proposed.

IV. BP DECODER USING OVERLAPPING SCHEDULING SCHEMES

According to the decoding procedure of BP algorithm, PEs are activated stage-by-stage from left to right in each iteration. Fig. 4 shows an example of this decoding scheme of (16, 8) polar code for 3 iterations. Here, $C_{i,j}^{P}$ indicates that, the propagating messages that belong to the i-th received codeword are updated in the j-th stage of PEs during the p-th iteration. For example, $C^{2}_{1,3}$ in clock cycle-7 represents that, in order to decode the 1st received codeword, the stage3 is activated and processes the propagating messages in the 2nd iteration. In addition, the arrow in Fig. 4 describes the data dependency in (5.2). Here the black arrow indicates the dependency within the same iteration, and the red arrow indicates the dependency between consecutive iterations. For example, only after stage 1 and stage 3 finish processing $C^{2}_{1,3}$ and $C^{1}_{1,3}$, respectively, their output is sent to stage 2 as its inputs, and then stage 2 is allowed to process $C^{2}_{1,2}$. From Fig. 4 it can be seen that in each cycle, only one stage is activated while other stages are always idle. For a (n, k) polar BP decoder, this yields a low hardware utilization rate of only 1/m.

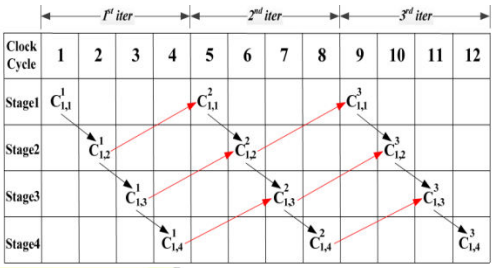

Fig4. Original decoding scheme of n=16 BP decoder.

Fig. 4 shows that m-1 stages of PEs are idle in each cycle of polar BP decoding. In order to avoid this under-utilization, overlapped scheduling technique can be used to fully utilize those idle resources. In this section, we discuss the iteration-level overlapping first. After a careful examination of Fig. 4, we find that in each cycle multiple stages can be activated at the same time. For example, consider $C^{2}_{1,3}$ in cycle-5 at stage 1. This computation is dependent on output from $C^{2}_{1,2}$ C in cycle-2 at stage 2. Since stage 2 can process $C^{2}_{1,2}$ in cycle-2, stage 1 can process $C^{1}_{1,1}$ at the beginning of cycle-3. Therefore, instead of being activated in cycle-5 in the original scheme (see Fig.4), stage1 can now be enabled in cycle-3 to process propagating messages for the 2nd iteration ($C^{2}_{1,1}$) without any timing conflict (see Fig. 5). As a result, one clock cycle can be saved by applying this re-scheduling approach. Similarly, Fig. 5 shows that the other stages can also be activated earlier. Therefore, this re-scheduling approach can lead to reduction of 4 cycles in latency.

By exploiting overlapped-scheduling, from Fig. 5, it can be seen that, from cycle-3 to cycle-6, some computations belong to either 2nd or 3rd iteration. This overlapped-rescheduling approach leads to great reduction in decoding latency. In general, for an (n, k) polar BP decoder with m-stages of PEs, the proposed iteration-level overlapped scheduling

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

approach reduces the decoding latency from *m*max_iter* to *2*max_iter+m-2* clock cycles, where max_*iter* is the preset maximum number of iterations. Considering *m* is usually larger than 10 for practical use, the latency can be reduced by approximately 10/2=5 times.

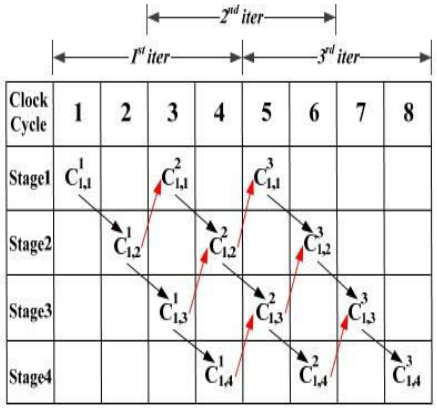

Fig 5. Overlapped-scheduling at iteration level.

Although the above iteration-level overlapped scheduling method can greatly improve the hardware utilization, the schedule in Fig. 5 cannot achieve 100% hardware utilization since some stages still remain idle. In general, due to the data dependency between successive iterations, the maximum hardware utilization rate of iteration-level overlapped scheme is limited to be less than 50%.

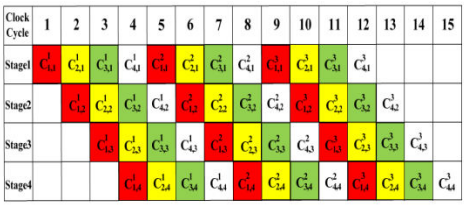

Fig 6. 4-level-overlapping at codeword level.

Different from iteration-level overlapping, codeword-level overlapping, as a common technique used in SC decoder designs, can make hardware utilization close to 100%. In this section, we apply this technique for BP decoder optimization. Fig. 6 shows an example of a 4-level-codeword-overlapping scheme. Here different colors represent propagating messages belonging to different received code words. Compared with original scheme with iteration-level overlapping (see Fig. 4 and Fig. 5), the codeword-level overlapping scheme fully utilizes those idle stages to process multiple independent received code words. As a result, the hardware utilization rate approaches 100%. In general, for an (n, k) polar BP decoder, maximum m independent received code words can be input to the decoder for overlapped processing. As a result, the processing throughput increases by approximately m times at the expense of an extra m-1 clock cycles, and (m-1) times more memory for storing the cod ewords than that in Fig..4

V. RESULTS AND DISCUSSION.

The simulation results for 8 bit polar decoder which is obtained by simulating the Verilog code for the design shown in figure 7. Xilinx software is used to verify the functionality of my design.

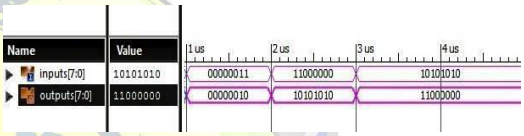

Fig 7. Simulation result of 8 bit Polar decoder

Fig 8. Simulation result of 8 bit Polar decoder using proposed scheme.

Table.1. Timing Analysis

Performance Parameter	Dealy (ns)
Existing Scheme	1.285
Overlapping at iteration level	1.106
Overlapping at code level	1.024

VI. CONCLUSION

In this paper, a novel early stopping method based on overlapping scheduling scheme is proposed for BP polar code decoders. This method not only reduces the average complexity of the BPD, hence reducing the energy consumption during decoding, but also helps to make the decoding faster. This reduces average latency.

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

VII. ACKNOWLEDGMENT

The proposed work was carried out in the VLSI lab of IES College of Engineering Chittilappilly as part of the PG project. This work was greatly supported by all the faculties and staff of IES College of Engineering, Chittilappilly. The authors would like to acknowledge the Management, Principal, Head of the department, all the faculties and staff of Electronics and Communication Department, of IES College of Engineering, Thrissur, Kerala, for their co-operation and technical guidance given during the entire course of the project work to complete it successfully.

VIII. REFERENCE

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 2. May 1993, pp. 1064–1070.

[2] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA, USA: MIT Press, 1963.

[3] E. Arikan, "Channel polarization: A method for constructing capacityachieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[4] E. ¸Sa¸so˘glu, E. Telatar, and E. Arikan, "Polarization for arbitrary discrete memoryless channels," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Oct. 2009, pp. 144–148.

[5] Alamdar-Yazdi and F. R. Kschischang, "A simplified successivecancellation decoder for polar codes," *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1378–1380, Dec. 2011.

[6] 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.

[7] 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. 2,289–299, Jan. 2013.

[8] Y. Fan and C. Y. Tsui, "An efficient partial-sum network architecture for semi-parallel polar codes decoder implementation," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3165–3179, Jun. 2014.

[9] G. Sarkis and W. J. Gross, "Increasing the throughput of polar decoders," *IEEECommun. Lett.*, vol. 17, no. 4, pp. 725–728, Apr. 2013.

[10] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 946–957, May 2014.

[11] Tal and A. Vardy, "List decoding of polar codes," in *Proc. IEEE Int. Symp. Inf.Theory (ISIT)*, Jul./Aug. 2011, pp. 1–5.

[12] Niu and K. Chen, "Stack decoding of polar codes," *Electron. Lett.*, vol. 48, no. 12,695–696, Jul. 2012.