High-Throughput and Energy-Efficient Belief Propagation Polar Code Decoder

Abstract:

Owing to their capacity-achieving performance and low encoding and decoding complexity, polar codes have received significant attention recently. Successive cancellation decoding (SCD) and belief propagation decoding (BPD) are two popular approaches for decoding polar codes. SCD, despite having less computational complexity when compared with BPD, suffers from long latency due to the serial nature of the SC algorithm. BPD, on the other hand, is parallel in nature and is more attractive for low-latency applications. However, due to the iterative nature of BPD, the required latency and energy dissipation increase linearly with the number of iterations. In this paper, we propose a novel scheme based on sub-factor graph freezing to reduce the average number of computations as well as the average number of iterations required by BPD, which directly translates into lower latency and energy dissipation. Simulation results show that the proposed scheme has no performance degradation and achieves significant reduction in computation complexity over the existing methods. Moreover, the hardware architecture for the proposed scheme is developed and compared with the state-of-the-art BPD implementations for (1024, 512) polar codes. A decoding throughput of 13.9 Gb/s is achieved along with a 60%–73% improvement in energy reduction and two times increase in hardware efficiency when compared with the existing BPD implementations. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

Existing 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 area cost, recently, a memory efficient BP decoder was proposed by Sha et al. based on the idea of combining two adjacent stages of a BPD factor graph. To reduce the computation complexity, another decoding method, called soft cancellation (SCAN) decoding, was proposed. By restricting the soft information propagation schedule in the decoding process, the computational complexity of SCAN is much lower than that of BPD. However, different from BPD, the SCAN operation is serial in nature, leading to much longer decoding latency. Based on the SCAN decoder, another decoding algorithm, called reduced complexity soft cancellation (RCSC), was proposed to reduce the BPD complexity. RCSC requires significantly less memory for storing log-likelihood ratios (LLRs) as compared with the SCAN decoder, and by eliminating the unnecessary additions, it also requires fewer computations. However, similar to the SCAN decoder, RCSC suffers from long decoding latency. In this paper, we aim at the design of a low-latency polar codes decoder, and hence, we concentrate on the parallel BPD implementation.

To further reduce the computation complexity of BP decoders, in our previous work, we proposed a method based on the convergence of the sub factor-graphs, which is reached at a much earlier stage. Borrowing an idea from SCD, some of the sub factor-graphs are checked during each iteration, and if they have converged, they are frozen and do not need to be computed in the subsequent iterations. In addition, the freezing of these sub factor-graphs will help to improve the convergence of the overall decoding process over the rest of the complete factor graph. As a result, both the computation complexity and the average number of iterations are reduced. In this paper, different scheduling schemes (specific methods of information dissemination across the factor graph) for the BP algorithms are analyzed in detail from the aspects of latency and convergence speed. Moreover, two new scheduling schemes, which result in huge reduction in the average number of iterations required and, hence, the latency for BPD as compared with the existing scheduling schemes, are introduced in this paper. Using the proposed scheduling schemes, the average latency of BPD for (1024, 512) polar codes is reduced to just 37.8 cycles, which is, to the best of our knowledge, the lowest latency achieved so far. In addition, the hardware architecture of the proposed BPD scheme is developed and implemented in a 45-nm CMOS process technology. Synthesis results show that our proposed architecture can achieve a coded throughput of 13.9 Gb/s. Moreover, 60%–73% improvement in energy efficiency
and two times improvement in hardware efficiency are achieved, when compared with the existing hardware implementations.

Disadvantages:

- Energy consumption is high
- Hardware efficiency is low

Proposed System:

Polar Codes and Belief Propagation Decoding:

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

CSFG Freezing Concept:

The proposed BP decoding scheme is based upon a CSFG freezing concept to achieve lower complexity. At a particular iteration $t$, if a CSFG at stage $j$ can correctly decode its corresponding constituent code, it is frozen and no message passing or updating within the CSFG will be needed in the subsequent iterations. The details of how to check whether a CSFG can be frozen will be presented later. When the decoding reaches a certain stage, its CSFGs are checked for freezing. If the CSFG cannot correctly decode its constituent code, then it cannot be frozen and the message passing and updating are executed by the PEs at that stage. After that, we move to the next stage and check the convergence of the corresponding CSFGs. When we move to the next stage, the number of CSFGs is doubled. This freezing checking procedure continues from stage to stage until the end of the BPD factor graph is reached.

Checking Order for CSFGs:

The order of checking the freezing of CSFGs at a stage is important to the operation of the proposed scheme. Here, we borrow an idea from SCD, where the results of the previously decoded bits are used for the decoding of the current bit. Similar to the SCD operation, the decoded result of a constituent code associated with a frozen CSFG will be useful in decoding the
Do Your Projects With Technology Experts…

constituent codes of the subsequent CSFGs in the next few iterations. Thus, the freezing order of the CSFG has to follow the decoded bit order, and the top CSFGs at each stage will be frozen first. A CSFG can only be checked for freezing if all the previous CSFGs (in the order of the decoded bits) at that stage have already been frozen.
Do Your Projects With Technology Experts…

LeMeniz Infotech
36, 100 Feet Road, Natesan Nagar, Near Indira Gandhi Statue, Pondicherry-605 005.
Call: 0413-4205444, +91 9566355386, 99625 88976.
Mail: projects@lemenizinfotech.com
Fig. 1 shows the SCD scheduling tree and the factor graph of the \(n=8\) polar code. We can see that the top CSFGs at the different stages correspond to the first few subtrees that resulted from the depth-first traversal of the SCD scheduling tree.

CSFG Freezing Criterion:

MLD is based on exhaustive search, and hence has a huge computational complexity. To reduce the complexity, a novel checking criterion is proposed to efficiently find the MLD result of the constituent code.

Advantages:

- Energy consumption is low
- Hardware efficiency is high

Software implementation:

- Modelsim
- Xilinx ISE