Polar Code Decoder Framework

Timo Lehnigk-Emden, Matthias Alles
Creonic GmbH
Kaiserslautern, Germany 67663
{timo.lehnigk-emden, matthias.alles}@creonic.de

Claus Kestel, Norbert Wehn
University of Kaiserslautern
Kaiserslautern, Germany 67663
{kestel, wehn}@eit.uni-kl.de

***Abstract*—Polar codes gained large interest in the last years since they are the first channel codes that are proven to achieve channel capacity. Due to this property, Polar codes were recently adopted for the 5G standard. We present an industrial framework for the generation of Polar code decoders for highest data throughput. The framework automatically generates VHDL models ready for synthesis, placement and routing and corresponding simulation models to assess the communications performance. This framework enables Polar code decoder IP providers to give fast feedback to customers on communications and implementation performance. We demonstrate that this framework outperforms existing manually optimized decoders especially in terms of energy efficiency.**

I. Introduction

Mobile communication plays a central role in our information society and is a key enabler for a connected world. Over the last decades, we have seen an increase in data rates of several orders of magnitude in the various communication standards, e.g., from 10 Kbps data rates in GSM to over 10 Gbps in the newest 5G standard. Beyond 5G, data rates towards 100 Gbit/s are expected. Baseband processing Systems-on-Chips (SoCs) have to provide this large throughput under low-power constraints. Forward error correction (FEC) is an essential IP building block in any baseband processing unit. An emerging FEC scheme are Polar codes that are in the focus of this paper.

Polar codes [3] are linear block codes and belong to the class of multilevel concatenated codes, but in contrast to the similar Reed-Muller Codes, they use the phenomenon of channel polarization to split the channels into reliable and unreliable channels. Information bits are transmitted via the reliable channels, and a-priori fixed bits (called frozen bits) over the unreliable channels. It was proven that Polar codes can exploit the full capacity of binary symmetric channels, however, due to the lack of finite length code construction methods, they had so far no practical relevance in industry standards. The progress in code construction and decoding algorithms brought them into the 3GPP Release 15 standard, also known as 5G. The control channel of the new physical layer uses Polar codes to protect meta information. For this kind of data, existing standards use traditional FEC codes like BCH, Reed Solomon, Hamming or convolutional codes. These schemes have their strengths for hard decoding and mid-size block lengths. Polar codes close the gap for short block lengths with soft decoding because of the availability of nearly optimal decoding algorithms such as list decoding. For logical payload channels the transition in standards to more advanced coding schemes like LDPC and Turbo Codes is in progress for more than 20 years. Polar Codes have the potential of being used in further standards for control or payload channels up to mid-size (~ 1000 bits) block lengths.

It is essential for an IP provider to react fast to new FEC technologies even if a technology is not yet very mature to enlarge its business model. Thus, a short time to market for new IP cores is key. Additionally, an IP provider has to give fast feedback to customers on communications performance, expected silicon area, throughput, latency, power consumption etc. when, e.g., the silicon technology, the code, the decoding algorithm or the quantization changes. However, it is very challenging to estimate implementation metrics on a higher abstraction level. Hence, fast implementations become mandatory. As stated before it is expected that Polar codes will enter into more standards and SoCs with different requirements on communications performance, throughput, clock frequency, latency etc. A sophisticated decoder design framework can efficiently address these challenges.

II. Polar Code Decoder Framework

Our framework puts emphasis on Polar code decoders for a throughput beyond 100 Gbit/s. We do not expect frequencies larger than 1 GHz due to power and design methodology issues. Typical frequencies for these type of SOCs are in the range between 500 MHz and 1 GHz. Targeting 500 Gbit/s throughput with a frequency of 500 MHz requires the decoding of 1000 bits in one clock cycle. Hence, high throughput and energy efficient Polar code decoders demand for extreme parallelism and large data locality.

Encoding and decoding of Polar codes can be performed on a Polar factor graph. A compressed version of this graph is the Polar factor tree (PFT) [2]. Successive Cancellation (SC), Successive Cancellation List (SCL) and Belief-Propagation (BP) are the most prominent decoding algorithms. Decoding corresponds to a traversal of the PFT in which the received log-likelihood ratios (LLR) from the channel are processed by the tree nodes. SC and SCL decoding is a depth-first traversal of the PFT, BP a breadth-first traversal iterating from the leaves to the root and backwards. Different operations have to be performed at the tree nodes visited during traversal depending on the decoding algorithm. Corresponding node results are forwarded to the child/parent nodes, respectively. BP needs a large set of iterations to achieve the error correction performance of SC and is therefore not well suited for very high throughput and low latency demands [1]. SC and SCL,

978-3-9819263-2-3/DATE19/©2019 EDAA

on the other side, have a sequential behavior due to the mandatory depth-first tree traversal. To achieve very high throughput the tree traversal can be unrolled and pipelined [4], this "unrolling" technique was first published for LDPC code decoders [6]. Whenever a node is visited during tree traversal a corresponding pipeline stage is instantiated. In this way, for a block length of N (= number of leaves in the PFT), the maximum number of pipeline stages is $2*(2N-2)+1$. Based on this principle we developed a framework for high throughput and energy efficient decoders. The framework, written in C++, uses the following parameters as input: the Polar code, quantization information, the decoding algorithm (at the moment the framework supports SC, hard and soft-output, and SCL decoding) and technology parameters. It outputs a fully synthesizable VHDL decoder model, the corresponding test bench data and a C++ model for BER/SNR simulations. It is guaranteed by construction that VHDL model and simulation model are bit-level equivalent. So, no further verification is mandatory. The framework includes a set of optimization techniques. Some of them are listed in the following.

Optimization of the PFT: The complexity of the decoding architecture is directly proportional to the size of the PFT. Thus, we implemented a set of transformations to reduce its size. E.g., if a subtree represents a repetition code or a parity-check code, the corresponding subtree can be replaced by a single node. Alike, we can merge rate-0 and rate-1 nodes into their respective parent nodes [5].

Optimization of the memory footprint: Heavily pipelined high throughput architectures require a huge amount of memory that costs area and power. This memory requirement can be reduced on algorithmic level, e.g., by efficient quantization, and on (micro)architectural level. For the latter one we integrated a retiming engine in the framework, clock gating and a latch based design.

Further optimizations: Different data representation are used to further reduce power and fully parallel to partially parallel transformations for large complexity nodes are applied to reduce area complexity.

III. Results

We present a decoder for highest throughput for a 1024/512 Polar code that is decoded with an SC algorithm. A 6 bit quantization for LLRs and internal processing, and a 28 nm FD-SOI technology with worst case PVT (for timing 0.9 V and 1.0 V for power, both 125 °C) conditions are used. The synthesis is performed using the *Design Compiler*, Place & Route with *IC-Compiler*, both from *Synopsys*.

The unoptimized PFT has 2047 nodes, corresponding to 4093 stages. The automatically performed tree optimization yields a reduction to 385 stages. Implementing this tree in a fully pipelined architecture would require 307 KB of memory. Automatic retiming with a frequency constraint of 700 MHz reduces the number of pipeline stages to 105 (corresponding to 85 KB memory). In Table I, the column "Registers" shows the result of the corresponding decoder after placement and routing. More than 70% of the overall power is consumed in registers and clock-tree. In a second generation run, we enabled the latch flag option in the framework. If this flag is set, registers are replaced by latches whenever it is possible together with further power optimizations. Note, that our latch based approach is compatible with a standard VHDL synthesis flow and standard libraries except for the testing issue that can be solved by special scan chains. The corresponding results are shown in the column "Latches". This decoder outperforms the original decoder that used standard registers in all metrics, i.e. area efficiency, throughput and especially energy efficiency that was be improved by a factor of 2x. No manual modification of the automatically generated VHDL code was performed.

Regarding state-of-the-art, the fastest published SC Polar code decoder is from [4] and achieves 1.275 Tbps at a frequency of 1.25 GHz, resulting in an area of 4.627 mm^2 and a power consumption of 8.8 W. This decoder is based on the same Polar code length and rate but has only 5 bit quantization for processing. It also uses a 28 nm FD-SOI technology. The results are shown in Table I. However, no placement and routing was performed, i.e., the above mentioned numbers are after synthesis only and, moreover, only typical case PVT were assumed. Therefore a direct comparison is not possible. No other publications are known achieving more than 500 Gbps.

TABLE I
Implementation results SC 1024/512

Place&Route	Registers	Latches	[4] Synthesis only
Area [mm^2]	3.14	2.79	4.63
- Combinat.	0.96	0.91	-
- Buf/Inv	0.65	0.27	-
- Noncomb	1.55	1.12	-
Area Eff. [Gbps/mm^2]	205	231	276
Utilization	78%	72%	-
Frequency [MHz]	621	629	1245
Throughput [Gbps]	636	644	1275
Power [W]	5.7	2.7	8.8
- Clock	47%	19%	-
- Registers	24%	13%	-
- Combinat.	29%	68%	-
Energy Eff. [pJ/bit]	8.8	4.2	6.9

Acknowledgment

We gratefully acknowledge financial support by the DFG (project-ID: 2442/8-1) and the EU (project-ID: 760150-EPIC).

References

[1] S. M. Abbas, Y. Fan, J. Chen, and C. Y. Tsui. High-Throughput and Energy-Efficient Belief Propagation Polar Code Decoder. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(3):1098–1111, March 2017.

[2] A. Alamdar-Yazdi and F. R. Kschischang. A Simplified Successive-Cancellation Decoder for Polar Codes. *IEEE Communications Letters*, 15(12):1378–1380, December 2011.

[3] Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, 2009.

[4] Pascal Giard, Claude Thibeault, and Warren J. Gross. *High-Speed Decoders for Polar Codes*. Springer, Cham, 2017.

[5] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross. Fast Polar Decoders: Algorithm and Implementation. *IEEE Journal on Selected Areas in Communications*, 32(5):946–957, May 2014.

[6] P. Schläfer, N. Wehn, M. Alles, and T. Lehnigk-Emden. A new dimension of parallelism in ultra high throughput LDPC decoding. In *SiPS 2013 Proceedings*, pages 153–158, October 2013.