Accepted in Electronics Letters, 2015

A 237 Gbps Unrolled Hardware Polar Decoder

Pascal Giard, *Student Member, IEEE*, Gabi Sarkis,
Claude Thibeault, *Senior Member, IEEE*, and Warren J. Gross, *Senior Member, IEEE*

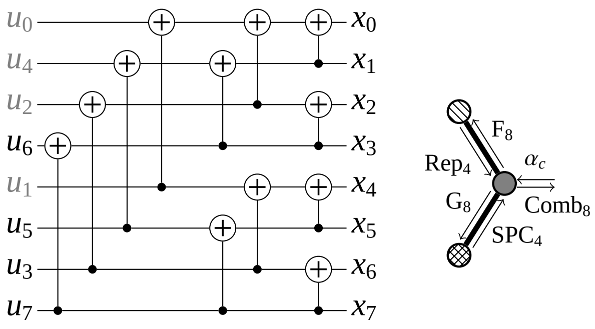
Abstract

In this letter we present a new architecture for a polar decoder using a reduced complexity successive cancellation decoding algorithm. This novel fully-unrolled, deeply-pipelined architecture is capable of achieving a coded throughput of over 237 Gbps for a (1024,512) polar code implemented using an FPGA. This decoder is two orders of magnitude faster than state-of-the-art polar decoders.

I. Introduction

Polar codes provably achieve the symmetric capacity of memoryless channels using the low-complexity successive-cancellation (SC) decoding algorithm [1]. However, the SC algorithm is sequential in nature, leading to low-throughput decoders. In [2], [3], new decoding algorithms with the specific aim of reducing the decoding latency and increasing the throughput were proposed. These algorithms work by decomposing a polar code into its constituent codes and using fast, specialized decoding algorithms on them. They represent polar codes as decoder trees that can be pruned by creating a new node type for each of the recognized constituent code types.

The field-programmable gate-array (FPGA) implementation of the Fast Simplified Successive Cancellation (Fast-SSC) algorithm presented in [3] can achieve an information throughput of 1 Gbps. Fig. 1a is the graph representation for an $(8,4)$ polar code where u_0, u_1, u_2 and u_4 are frozen bits. Fig. 1b shows the decoder tree corresponding to Fast-SSC decoding of that $(8,4)$ polar code after tree pruning is applied. The arrows indicate the data flow whereas the annotations correspond to the channel values (α_c) or functions as defined in the Fast-SSC algorithm [3]. Notably, the striped node corresponds to a Repetition code of length 4 and the cross-hatched one to a single parity check (SPC) code, also of length 4.



(a) Graph

(b) Decoder tree

Fig. 1: From a graph to a Fast-SSC decoder tree.

Currently, the fastest realization of a decoder for polar codes is the belief-propagation (BP) decoder of [4], which achieves a coded throughput of 4.68 Gbps (information throughput of 2.34 Gbps) for a (1024, 512) code on a 65 nm CMOS application-specific integrated-circuit (ASIC) running at 300 MHz.

G. Sarkis, P. Giard, and W. J. Gross are with the Department of Electrical and Computer Engineering, McGill University, Montréal, Québec, Canada (e-mail: {gabi.sarkis, pascal.giard}@mail.mcgill.ca, warren.gross@mcgill.ca).

C. Thibeault is with the Department of Electrical Engineering, École de technologie supérieure, Montréal, Québec, Canada (e-mail: claude.thibeault@etsmtl.ca).

This paper is a postprint of a paper submitted to and accepted for publication in Electronics Letters and is subject to Institution of Engineering and Technology Copyright. The copy of record is available at IET Digital Library. DOI: https://doi.org/10.1049/el.2014.4432

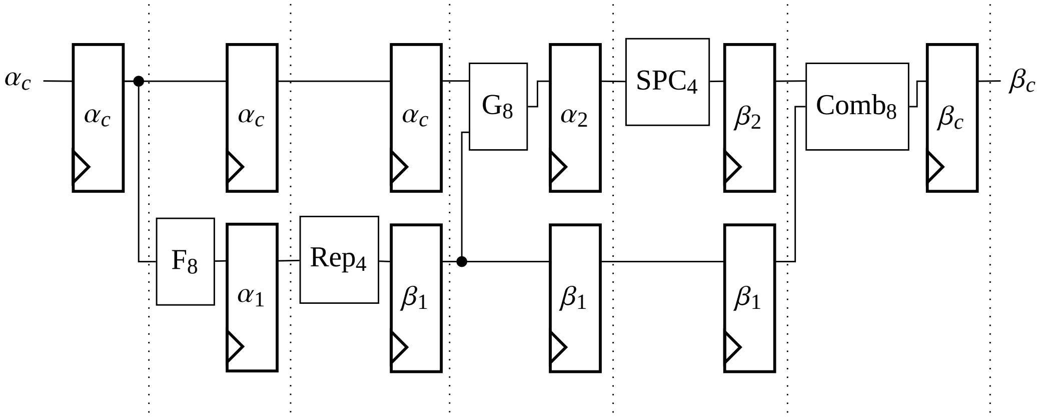


Fig. 2: Implementation for $(8, 4)$ polar code. Clock signal not routed for clarity.

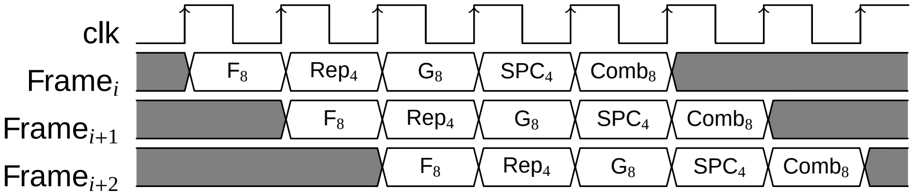


Fig. 3: Timing example to decode 3 frames of a $(8, 4)$ polar code.

In spite of these advances, polar decoders remain slow compared to capacity-approaching codes such as low-density parity-check (LDPC) codes, hampering their adoption for high-speed applications. This work addresses this issue by presenting a new decoder architecture that achieves a coded throughput of 237 Gbps (information throughput of 118.5 Gbps) on an FPGA running at 231 MHz for a (1024, 512) polar code.

II. Architecture

Most existing polar decoders (e.g. [3]–[5]) minimize area and maximize logic utilization by restricting the decoder to decode a single frame. While this approach lowers implementation complexity, it limits decoding throughput. Instead, we propose generating a code-specific unrolled decoder, fully pipelining its execution so that it processes portions of several frames at once, and adding memory registers for the required data persistence.

Fig. 2 shows the decoder architecture for an $(8, 4)$ polar code. The functional units correspond to the operations shown in Fig. 1b, each of which is followed by a pipeline register to store the operation's output. In addition some pipeline stages do not have any processing logic; they are added to ensure that different messages remain synchronized. As a result of the pipelined design, at every clock cycle, a frame is output and a new received frame can be loaded as shown in the timing diagram in Fig. 3. This deeply-pipelined architecture leads to very high-throughput decoders.

Due to the unrolled nature of the architecture, the growth in resources used is quadratic in code length. It is also affected by the code rate and frozen bit locations as both affect the structure of the decoder tree and, in turn, the number of operations performed in a Fast-SSC decoder. The amount of memory used is also quadratic in code length and affected by rate and frozen bit locations. In comparison, the Fast-SSC decoder in [3] requires memory that grows linearly in code length. This growth in resources and memory limits the proposed decoder to codes of moderate lengths when implemented on an FPGA.

III. Implementation Results

The resulting information throughput is PfR bps where P is the width of output bus in bits, f is the execution frequency in Hz and R is the code rate. Latency depends on the frozen bit locations and the constrained maximum width for all modules. In this work, the buses are sized so that all data is transferred simultaneously, i.e. they can carry N log-likelihood ratios (LLRs) and N bit estimates as in [4], [6].

A decoder utilizing the proposed architecture was implemented for a $(1024, 512)$ polar code on an Altera Stratix IV EP4SGX530KH40C2 FPGA. The specialized decoders for repetition and SPC codes were limited to constituent codes of length ≤ 4, all others were limited a maximum of 1024. Table I presents results for two different execution frequencies. It can be observed that, at the cost of some register duplication, the coded (information) throughput can be increased from 210 Gbps (105 Gbps) to 237 Gbps (118.5 Gbps). The latency also decreases from 2.7 μs to 2.4 μs at 231 MHz. It can also be noted that, in both cases, register chains are implemented using SRAM blocks.

TABLE I: Post-fitting results for a $(1024, 512)$ polar code on the Altera Stratix IV EP4SGX530KH40C2 FPGA.

LUTs	Registers	RAM (bits)	f (MHz)	Info. T/P (Gbps)	Latency (CC)
156,450	152,124	285,120	206	105.3	559
155,858	158,185	285,120	231	118.5	559

Table II compares the proposed decoder with others from the literature. Notably, the unrolled decoder has 50.7 times the throughput of the BP decoder of [4], with the latter implemented as a 65 nm CMOS ASIC clocked at 300 MHz. With its maximum of 15 iterations, the BP decoder has a latency that is 21 times higher than the proposed decoder. The Altera Stratix IV FPGA is built using the more recent 40 nm technology. The delay gain between 65 nm and 40 nm CMOS technology is little over 1.23 as this corresponds to the gain between 65 nm and 45 nm [7]. However, the speed gain of building an ASIC instead of using an FPGA was shown to be from 3.4 to 4.6 [8].

TABLE II: Comparison with state-of-the-art polar decoders.

	This work	[4]	[6]	[3]
Dec. Algo.	Fast-SSC	BP	SC	Fast-SSC
Code	$(1024, 512)$	$(1024, 512)$	$(512, k)$	$(1024, 512)$
IC Type	FPGA	ASIC	ASIC	FPGA
Tech.	40 nm	65 nm	90 nm	40 nm
f (MHz)	231	300	6	108
Latency (μs)	2.4	50	0.2	2
T/P (Gbps)	237	4.7	2.9	0.5

Recently, another fully unrolled polar decoder based on the less efficient SC algorithm has been presented in [6]. That work is fully combinational with the exception of its input and output interfaces and as a result has a much lower frequency. The proposed decoder has a 14 times higher latency but is over 81 times faster than the 90 nm CMOS implementation of [6]. The delay gain between 90 nm and 45 nm CMOS technology is 1.58 [7], still lower than the 3.4 to 4.6 factor between FPGA and ASIC. It should be noted that [6] implemented a smaller polar code of length $N = 512$ instead of $N = 1024$.

Table II also presents results for a $(1024, 512)$ polar code decoded using the implementation of [3]. Our fully-unrolled, deeply-pipelined decoder has a throughput that is over 474 times greater than that previous Fast-SSC decoder implementation; while the latency is similar.

The proposed decoder has a throughput that is two orders of magnitude greater than that of state-of-the-art polar decoders.

IV. Conclusion

In this Letter we presented a new architecture for a fully-unrolled, deeply-pipelined polar decoder. We showed that a decoder for a $(1024, 512)$ polar code implemented on an FPGA can achieve a coded throughput that is two orders of magnitude faster than state-of-the-art polar decoders. At 237 Gbps, it is 51 to 81 times faster than the state-of-the-art ASIC implementations.

Acknowledgement

Claude Thibeault is a member of ReSMiQ. Warren J. Gross is a member of ReSMiQ and SYTACom.

References

[1] E. Arıkan, “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] 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.

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

[4] Y. S. Park, Y. Tao, S. Sun, and Z. Zhang, “A 4.68Gb/s belief propagation polar decoder with bit-splitting register file,” in *Symp. on VLSI Circuits Digest of Technical Papers*, June 2014, pp. 1–2.

[5] A. Raymond and W. Gross, “A scalable successive-cancellation decoder for polar codes,” *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5339–5347, Oct. 2014.

[6] O. Dizdar and E. Arıkan, “A high-throughput energy-efficient implementation of successive-cancellation decoder for polar codes using combinational logic,” *CoRR*, vol. abs/1412.3829, Dec. 2014. [Online]. Available: http://arxiv.org/abs/1412.3829

[7] H. Wong, V. Betz, and J. Rose, “Comparing FPGA vs. custom CMOS and the impact on processor microarchitecture,” in *ACM/SIGDA Int. Symp. on Field Programmable Gate Arrays*, 2011, pp. 5–14.

[8] I. Kuon and J. Rose, “Measuring the gap between FPGAs and ASICs,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 2, pp. 203–215, 2007.