Preface

Origin

The majority of this book was initially published as a Ph.D. thesis, a thesis nominated for the *Prix d'excellence de l'Association des Doyens des Études Supérieures au Québec (ADESAQ)* by the Electrical and Computer Engineering department of McGill University.

Scope

Over the last decades we have gradually seen digital circuits take over applications that were traditionally bastions of analog circuits. One of the reasons behind this tendency is our ability to detect and correct errors in digital circuits—circuits making computations with discrete signals as opposed to continuous ones. This ability led to faster and more reliable communication and storage systems. In some cases it enabled things that we thought might have never been possible, e.g., reliable communication with a probe that is located many light years away from our planet.

Right after the Second World War, Claude Shannon created a new field—information theory—in which he defined the limit of reliable communications or storage. In his seminal work, Shannon defined what he calls the channel capacity [60], the bound that many researchers have tried to achieve or even approach ever since. Shannon's work does not tell us how this limit can be reached.

While Reed-Solomon (RS) and Bose-Chaudhuri-Hocquenghem (BCH) codes have good error-correction performance and are in widespread use even today, it's not until the discovery of turbo codes [12] in the 1990s that error-correcting codes approaching the channel capacity were found. Indeed, while Low-Density Parity-Check (LDPC) codes—initially discovered in the 1960s by

Robert Gallager [16]—can also be capacity approaching, their decoding algorithm was too complex for the time and thus were not used until they were independently rediscovered by David McKay in 1997 [39].

The discovery of turbo and LDPC codes greatly rejuvenated the field of error correction. Often used in conjunction with a RS or a BCH code, standards that feature a turbo or a LDPC code are omnipresent. Nowadays, each home contains at least tens of decoders for these codes. They are used in a plethora of applications such as video broadcasting, wireless and wired communications (e.g., WIFI and Ethernet), and data storage.

The latest findings on the road to achieving channel capacity are polar codes. Invented by Arıkan in 2008 [6] and further refined in 2009 [7], this new class of error-correcting codes, contrary to LDPC and turbo codes, has an explicit—nonrandom—construction making the implementation of their encoders and decoders simpler than that of LDPC or turbo codes. Polar codes exploit the channel polarization phenomenon by which the probability of correctly estimating codeword bits tends to either 1 (completely reliable) or 0.5 (completely unreliable). These probabilities get closer to their limit as the code length increases when a recursive construction is used. Under the low-complexity Successive-Cancellation (SC) decoding algorithm, polar codes were shown to achieve the symmetric capacity of memoryless channels as their length tends to infinity. The complexity of the SC algorithm is low but its sequential nature translates in high-latency and low-throughput decoder implementations. To overcome this, new decoding algorithms derived from SC were introduced, most notably [4] and [55]. These algorithms exploit the recursive construction of polar codes along with the a priori knowledge of the code structure. Fast Simplified Successive Cancellation (Fast-SSC), the algorithm described in [55], integrates the Simplified Successive Cancellation (SSC) algorithm described in [4]; thus this book builds upon the former.

Fast-SSC represented a significant improvement over the previous algorithms and led to the first hardware decoder for polar codes achieving a throughput greater than 1 Gbps. However, the optimization presented therein targeted high-rate codes. As low-rate codes are omnipresent in modern wireless communications, it was evident that it would be beneficial to have a closer look at potential improvements for such codes.

In Software-Defined Radio (SDR) applications, researchers and engineers have yet to fully harness the error-correction capability of modern codes. Many are still using classical codes [13, 63] as implementing low-latency high-throughput—exceeding 10 Mbps of information throughput—software decoders for turbo or LDPC codes is very challenging. The irregular data access patterns featured in turbo and LDPC decoders make efficient use of Single-Instruction Multiple-Data (SIMD) extensions present on today's processors difficult. To overcome the difficulty of efficiently accessing memory while decoding one frame and still achieve a good throughput, software decoders resorting to inter-frame parallelism (decoding multiple independent frames at the same time) are often proposed [30, 66, 69]. Inter-frame parallelism comes at the cost of higher latency, as many frames have

to be buffered before decoding can be started. Even with a split layer approach to LDPC decoding where intra-frame parallelism can be applied, the latency remains high at multiple milliseconds on a recent desktop processor [23]. On the other hand, polar codes are well suited for software implementation as their decoding algorithms feature regular memory access patterns.

While the future 5G standards are still in the works, many documents mention the requirement of peak per-user throughput greater than 10 Gbps. Regardless of the algorithm, the state of polar decoder implementations when our research started offered much lower throughput. The fastest SC-based decoder had a throughput of 1.2 Gbps at a clock frequency of 106 MHz [55]. The fastest decoder implementation based on the Belief Propagation (BP) decoding algorithm—an algorithm with higher parallelism than SC—had an average 4.7 Gbps throughput when early termination was used with a clock frequency of 300 MHz [49]. It was evident that a minor improvement over the existing architectures was unlikely to be sufficient to meet the expected throughput requirements of future wireless communication standards.

The book presents a comprehensive evaluation of decoder implementations of polar codes in hardware and in software. In particular, the work exposes new trade-offs in latency, throughput, and complexity, in software implementations for high-performance computing and General-Purpose Graphical Processing Units (GPGPUs), and hardware implementations using custom processing elements, full-custom Application-Specific Integrated Circuits (ASICs), and Field-Programmable Gate Arrays (FPGAs).

The book maintains a tutorial nature clearly articulating the problems that polar decoder implementations are facing, and incrementally develops various novel solutions. Various design approaches and evaluation methodologies are presented and defended. The work advances the state of the art while presenting a good overview of the research area and future directions.

Organization

This book consists of six chapters. Chapter 1 reviews polar codes, their construction, representations, and encoding and decoding algorithms. It also briefly goes over results for the state-of-the-art decoder implementations from the literature.

In Chap. 2, improvements to the state-of-the-art low-complexity decoding algorithm are presented. A code construction alteration method with human-guided criteria is also proposed. Both aim at reducing the latency and increasing the throughput of decoding low-rate polar codes. The effect on various low-rate moderate-length codes and implementation results are discussed.

Algorithm optimization at various levels leading to low-latency high-throughput decoding of polar codes on modern processors is introduced in Chap. 3. Bottom-up optimization and efficient use of SIMD instructions available on both embedded-platform and desktop processors are proposed in order to parallelize the decoding

of a frame, reduce latency, and increase throughput. Strategies for efficient implementation of polar decoders on GPGPU are also presented. Implementation results for all three types of modern processors are discussed.

A family of hardware architectures utilizing unrolling is presented in Chap. 4 showing that polar decoders can achieve extremely high-throughput values and retain moderate complexity. Implementations for various rates and code lengths are presented for FPGA and ASIC. The results are compared with the state of the art.

Expending from the previous chapter, Chap. 5 introduces a method to enable the use of multiple code lengths and rates in a fully unrolled polar decoder architecture. This novel method leads to a length- and rate-flexible decoder while retaining the very high speed typical to those decoders. ASIC results are presented for two versions of a multi-mode decoder and compared against the state-of-the-art decoders.

Lastly, conclusions about this book are drawn in Chap. 6 and a list of suggested future research topics is presented.

Audience

This book is aimed at error-correction researchers who heard about polar codes—a new class of provably capacity achieving error-correction codes—and who would like to learn about practical decoder implementation challenges and trade-offs in either software or hardware. As polar codes just got accepted to protect the control channel in the next-generation mobile communication standard (5G) developed by the 3GPP [40], this includes engineers who will have to implement decoders for such codes. Some prior experience in software or hardware implementation of high performance signal processing systems is an asset but not mandatory. The book can also be used by SDR practitioners looking into implementing efficient decoders for polar codes, or even hardware engineers designing the backbone of communication networks. Additionally, it can serve as reading material in graduate courses notably covering modern error correction.

Lausanne, VD, Switzerland — Pascal Giard
Montreal, QC, Canada — Claude Thibeault
Montreal, QC, Canada — Warren J. Gross

http://www.springer.com/978-3-319-59781-2

High-Speed Decoders for Polar Codes

Giard, P.; Thibeault, C.; Gross, W.J.

2017, XVIII, 98 p. 34 illus., 29 illus. in color., Hardcover
ISBN: 978-3-319-59781-2