PARTIALLY PARALLEL ENCODER DECODER ARCHITECTURE FOR LONG POLAR CODES

Vasamsetti Ramoji[1], R Satish Kumar[2], Addagalla Srinivasarao[3]
[1,2,3]Department of Electronics and Communication Engineering,
Bonam Venkata Chalamayya Institute of Technology & Science,
Amalapuram, Andhra Pradesh, India.

ABSTRACT:

Shannon proved the achievability part of his noisy channel coding theorem using a random-coding argument which showed the existence of capacity-achieving code sequences without exhibiting any specific sequence . Polar codes are an explicit construction that provably achieves channel capacity with low-complexity encoding, decoding, and code construction algorithms. The most favourable error correcting code in use, today is the polar code, due to its channel achieving property. This property is achieved by the polar code in a asymptotic manner, even though, in order to have a good performance over error correcting, it must be longer. We use a partially parallel encoder and partially parallel decoder in this paper since the already available fully parallel encoder, fully parallel decoder does not suit for long coded due to its complexity in the hardware. Here, the process of encoding and decoding has been analyzed for VLSI implementation and an advanced architecture required for long polar codes has been proposed with reduced hardware complexity. This proposed architecture has high throughput so it can be designed for higher levels of parallelism too.

KEYWORDS: Polar, Encoder, Decoder, Partial, Parallel, High Throughput, Latency, Density, Combinatinal path delay.

INTRODUCTION:

Nearly seventy years ago, Claude Shannon set a limit: no matter how advanced our technology becomes, how sophisticated our algorithms, we cannot reliably transmit information at rates, or efficiency, higher than what he called the channel capacity [5]. However, he left the question of how to achieve this channel capacity unanswered. For decades, researchers tried to find methods to achieve, or even approach, the channel capacity. These efforts bore fruit in 1993 when turbo codes were discovered [6]. These codes came close to capacity in certain cases and rejuvenated the field of error correcting codes. Low-density parity-check (LDPC) codes, originally discovered by Gallager in the 1960s [7], were rediscovered and found to also have excellent error-correction performance. These two codes, along with Reed-Solomon codes, are largely responsible for our ability to communicate with probes on the edge of our solar system [8], to store data at unprecedented densities [9], and to exchange 167 terabits of data per second via wired and wireless internet devices [10]. While the error-correction performance of turbo and LDPC codes under iterative decoding comes within fractions of a dB from the Shannon limit in certain cases, they do not achieve the channel capacity. Turbo codes in particular exhibit severe error floor; where the error-correction performance of the decoder almost stops improving with improving channel conditions. Moreover, both codes require randomness in their construction, complicating that process and increasing routing complexity in hardware implementations. In 2008, Arıkan introduced polar codes, which asymptotically achieve the symmetric capacity of memoryless channels [11, 12].

Polar code is a linear block error correcting code. The code construction is based on a multiple recursive concatenation of a short kernel code which transforms the physical channel into virtual outer channels. When the number of recursions becomes large, the virtual channels tend to either have high reliability or low reliability (in other words, they polarize), and the data bits are allocated to the most reliable channels.

Polar-coding is a capacity achieving code setting up method mainly for binary-input discrete memory much less channels. This can be done by the phenomenon of channel-polarization that every channel processes a flawlessly secure else a fully noisy channel as the code-length drives beyond over a collective channel built using a suite of N same sub channels1. 50% of power consumption can be reduced by parallel processing of two-input samples which reduces the frequency of operation by. For small or adequate polar code, fault performance by the Cyclic Redundancy Check (CRC) supported Successive Cancelation List (SCL) decoding procedure is improved than the Successive Cancelation (SC) decoding process3, which isn't appropriate for lengthy polar-codes owing to extreme hardware density. Linear block code with appropriate parameters can be used to perform block wise decoding of outer codes if there is not take into account, not necessary polar, as C 4. Path-search techniques for coding tree polar-codes are given as combined depiction of the SC, SCL, and SCS decoding algorithm. Integration of SCL and SCS, a new decoding process called the Successive Cancellation Hybrid (SCH). A semi-parallel- encoder based partial-sum update features as accessible architecture for SC decoding of polar-codes. This module uses Static Random Access Memory (SRAM) for storing, and uses a fixed data path. This design influences a multi-level quantization structure for Limb Lengthening and Reconstruction society (LLRs), reducing the memory usage and area. More importantly, they accomplish this with an explicit, nonrandom construction, using the low-complexity successive-cancellation decoding algorithm. Polar codes exploit the channel polarization phenomenon, in which certain bits are always estimated reliably while others are completely unreliable when decoding using successive cancellation.

POLAR CODES:

Polar codes were introduced by Erdal Arıkan in 2009 and they provide the first deterministic construction of capacity-achieving codes for binary memoryless symmetric (BMS) channels [1]. They are the culmination of Arıkan's research (e.g., more than 20 years) into the computational cutoff rate of sequential decoding.

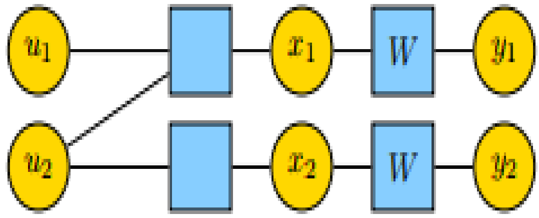


Fig: Factor Graph

In particular, consider a setup where (U1, U2) are two equiprobable bits that are encoded into (X1, X2) = (U1 ⊕ U2, U2). Then, (X1, X2) are mapped to (Y1, Y2) by two independent BMS channels with transition probabilities P(Y1 = y | X1 = x) = P(Y2 = y | X2 = x) = W(y|x). The factor graph for this setup is shown in Figure 1. Since the mapping from (U1, U2) to (X1, X2) is invertible, one finds that I(U1, U2; Y1, Y2) = I(X1, X2; Y1, Y2) = I(X1; Y1) + I(X2; Y2) = 2I(W), where C = I(X1; Y1) = I(W) is the capacity of symmetric BMS channel because X1 is equiprobable and I(W) , X y W(y|0) log2 W(y|0)/ 1 2W(y|0) + 1 2W(y|1) = I(X1; Y1). Thus, the transformation (X1, X2) = (U1 ⊕ U2, U2) preserves the sum capacity of the system. Also, the chain rule decomposition I

I(U1, U2; Y1, Y2) = I(U1; Y1, Y2) + I(U2; Y1, Y2|U1) = 2I(W)

implies that one can also achieve the rate 2I(W) using two steps. First, information is transmitted through the virtual channel W− : U1 → (Y1, Y2) at a rate I(U1; Y1, Y2) and decoded to Uˆ 1. Then, information is transmitted through the virtual channel W+ : U2 → (Y1, Y2, U1) at a rate I(U2; Y1, Y2|U1) and decoded to Uˆ 2 based on the side information Uˆ 1. If one uses convolutional codes with sequential decoding, however, the expected decoding complexity per bit becomes unbounded for rates above the computational cutoff rate

R0(W) = 1 − log2 (1 + Z(W)),

where Z(W) , P y p W(y|0)W(y|1) is the Bhattacharyya parameter of the channel. Arıkan's key observation was that, while the sum capacity of the two virtual channels is preserved, the sum cutoff rate satisfies R0(W+) + R0(W−) ≥ 2R0(W), with equality iff R0(W) ∈ {0, 1}. Thus, repeated transformations of this type cause the implied virtual channels to polarize into external channels whose capacities approach 0 or 1. But, for these external channels, coding is trivial and one either sends an information bit or a dummy bit. From a theoretical point of view, polar codes are beautifully simple. Practically, they approach capacity rather slowly as their block length increases and, thus, are not yet competitive with good photograph LDPC codes for moderate block lengths.

POLAR ENCODER ARCHITECTURE:

A polar code with a length of 16 bits is implemented with 32 XOR gates and the processed with 4 stages as shown in below figure. The fully parallel encoder is intuitively designed based on the generator matrix, but implementing such an encoder becomes a significant burden when a long polar code is used to achieve a good error correcting performance. The memory size and the number of XOR gates increase as the code length increases. The fully parallel encoder is intuitively designed based on the generator matrix, but implementing such an encoder becomes a significant burden when a long polar code is used to achieve a good error-correcting performance. In practical implementations, the memory size and the number of XOR gates increase as the code length increases. None of the previous works has deeply studied how to encode the polar code efficiently, although various tradeoffs are possible between the latency and the hardware complexity.

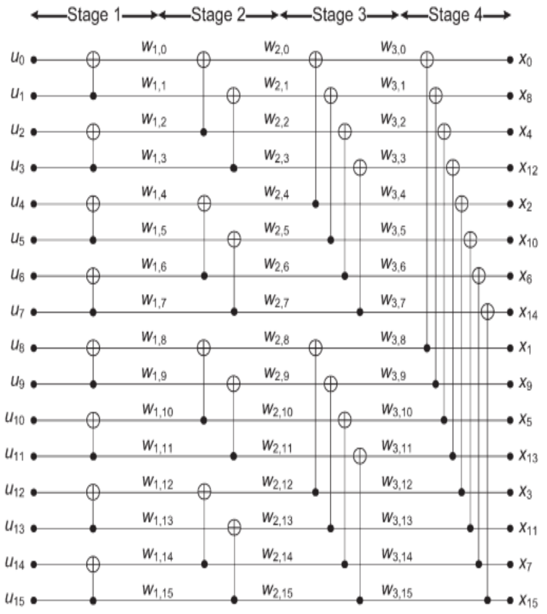

Stage 1
Stage 2
Stage 3
Stage 4
u_0 … u_{15}
x_0, x_8, x_4, x_{12}, x_2, x_{10}, x_6, x_{14}, x_1, x_9, x_5, x_{13}, x_3, x_{11}, x_7, x_{15}

Fig: Fully parallel architecture for encoding a 16-bit polar code.

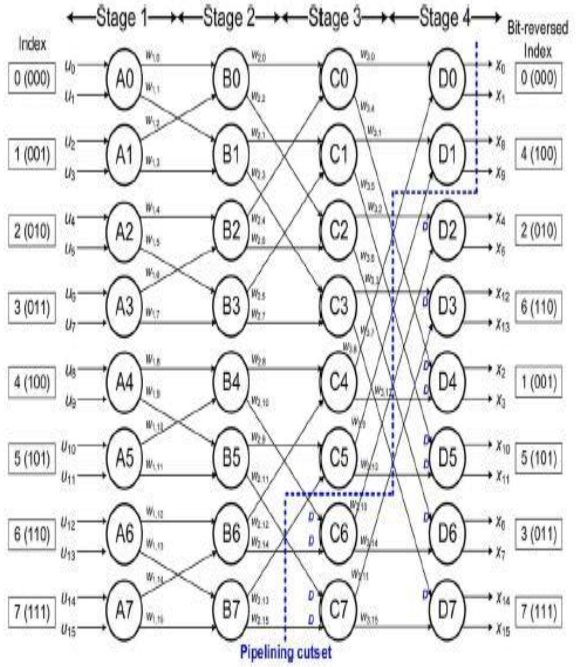


Fig. DFG of 16-bit polar encoding

In this section, we propose a partially parallel structure to encode long polar codes efficiently. To clearly show the proposed approach and how to transform the architecture, a 4-parallel encoding architecture for the 16-bit polar code is exemplified in depth. The fully parallel encoding architecture is first transformed to a folded form [5], [8], and then the lifetime analysis [6] and register allocation [7] are applied to the folded architecture. Lastly, the proposed parallel architecture for long polar codes is described.

POLAR DECODER:

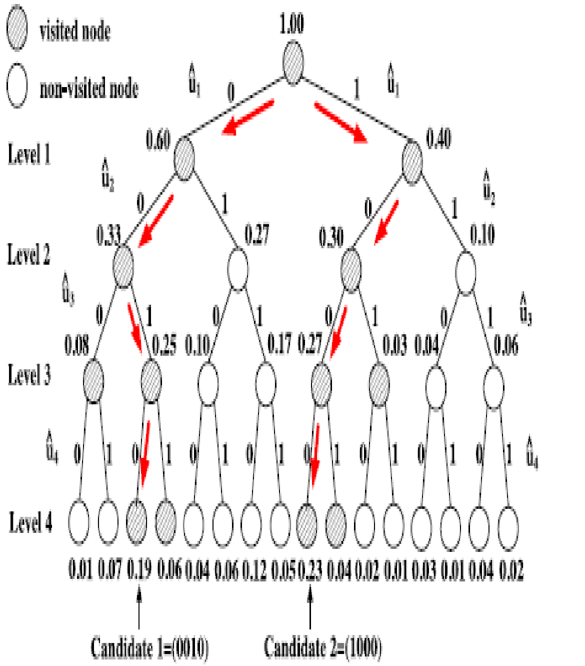
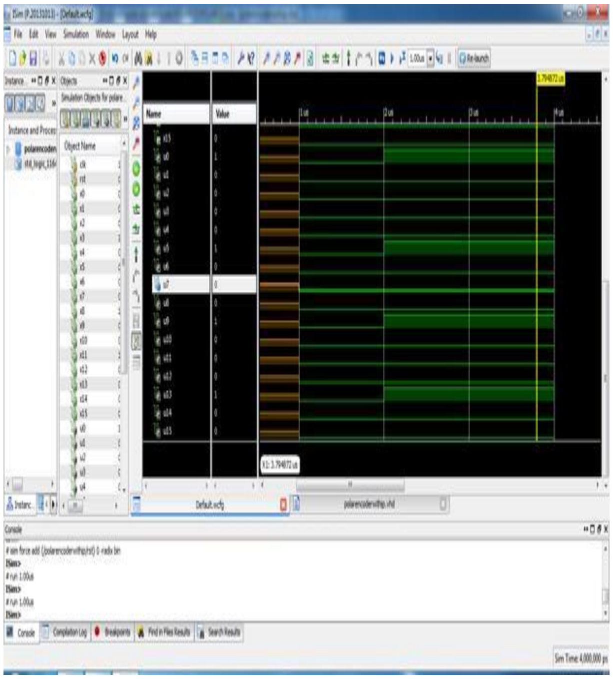


Figure. Searching of SCL decoder with n=4, p=4, and L=2, cited from

The decoding process of polar codes can also be interpreted from the view of code tree. Fig. 3 shows an example code tree for n = 4 polar codes. Here, each level represents a decoded bit. The value that is associated with each node represents the probability (metric) of the corresponding decoding path. For example, 0.23 at level 4 represents that the probability for the length-4 path (1000) is Pr (_u1 = 1, _u2 = 0, _u3 = 0, _u4 = 0) = 0.23. Based on this representation, the objective of a successful decoding procedure is to find the length-n path that is the corrected codeword. To achieve this goal, the SC decoder first visits the children nodes that are associated with the current survival path at each level. Then, it selects the new path that has the larger metric as the updated survival path. Because this searching strategy is only locally optimal, the performance of the SC decoder is limited.. Different from the SC decoder that only selects a single path, the L-size SCL algorithm utilizes L different searching paths. Therefore, it is more likely for the SCL algorithm to find the desired path than the SC algorithm. For example, the SCL decoder with L = 2is able to trace the valid path (1000) in Fig., while the SC decoder fails to find it. From above Figure, it can be seen that the SCL algorithm needs to visit the nodes at each level of the code tree.

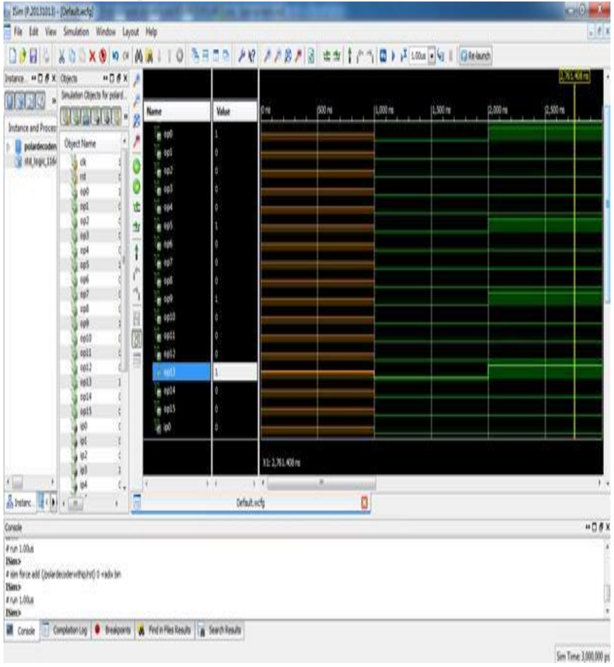
RESULT:

ENCODER INPUT:



```
# isim force add {/polarencoderwithp/rst} 0 -radix bin
ISim>
# run 1.00us
ISim>
# run 1.00us
ISim>
```

Decoder output:



```
# run 1.00us
ISim>
# isim force add {/polardecoderwithp/rst} 0 -radix bin
ISim>
# run 1.00us
ISim>
```

CONCLUSION:

As the number and variety of connected devices grow; so does the need for robust and efficient error-correcting codes. Polar codes offer a new avenue to pursue with promising results. The first implementation of polar decoders and encoders suffered from issues relating to speed and finite length performance. This thesis offered solutions to these problems, enabling high-speed, robust communications using polar codes. Finally, this brief has presented a new partially parallel encoder and decoder architecture developed for long polar codes .Many optimization techniques have been applied to derive the proposed architecture. Experimental results show that the proposed architecture

REFERENCES:

[1] 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 Circ. Dig. of Tech. Papers, Jun 2014, pp. 1–2.

[2] X. Han, K. Niu, and Z. He, "Implementation of IEEE 802.11n LDPC codes based on general purpose processors," in IEEE Int. Conf. Commun. Technol. (ICCT), on, Nov2013, pp. 218– 222.

[3] H. Yoo and I.-C. Park, "Partially parallel encoder architecture for long polar codes," IEEE Trans. Circuits Syst. II, vol. 62, no. 3, pp. 306–310, March 2015.

[4] P. Jouguet and S. Kunz-Jacques, "High performance error correction for quantum key distribution using polar codes," Quantum Inf. & Computation, vol. 14, no. 3-4, pp. 329338, 2014.

[5] C. Shannon, "A mathematical theory of communication," Bell Syst. Tech. Journal, vol. 27, no. 3, pp. 379–423, July 1948.

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

[7] R. Gallager, "Low-density parity-check codes," IRE Trans. Inf. Theory, vol. 8, no. 1, pp. 21–28, January 1962.

[8] K. S. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. R. Jones, and F. Pollara, "The development of turbo and ldpc codes for deep-space applications," Proceedings of the IEEE, vol. 95, no. 11, pp. 2142–2156, 2007.

[9] L. Kong, L. He, P. Chen, G. Han, and Y. Fang, "Protograph-based quasi cyclic ldpc coding for ultrahigh density magnetic recording channels," Magnetics, IEEE Transactions on, vol. 51, no. 11, pp. 1–4, 2015.