A POLAR CODE SUCCESSIVE CANCELLATION DECODER IMPLIMENTATION THROUGH A COMBINATIONAL LOGIC CIRCUIT

Mr. A.Chakradhar[1], Mrs.K.Rajyalakshmi[2]
1. Associate Professor, Department of ECE ,Jayamukhi Institute of Technogical Sciences , Warangal, India
2. Department of ECE, Jayamukhi Institute of Technological Sciences, Warangal, India.

Abstract:

This paper proposes a high-throughput energy efficient Successive Cancellation (SC) decoder architecture for polar codes based on combinational logic. The proposed combinational architecture operates at relatively low clock frequencies compared to sequential circuits, but takes advantage of the high degree of parallelism inherent in such architectures to provide a favourabletrade-off between throughput and energy efficiency at short to medium block lengths. At longer block lengths, the paper proposes a hybrid-logic SC decoder that combines the advantageous aspects of the combinational decoder with the low complexity nature of sequential-logic decoders. Performance characteristics on ASIC and FPGA are presented with a detailed power consumption analysis for combinational decoders. Finally, the paper presents an analysis of the complexity and delay of combinational decoders, and of the throughput gains obtained by hybridlogic decoders with respect to purely synchronous architectures.

1. INTRODUCTION

Shannon defines channel capacity as the maximum rate of information, which can be reliably transmitted over a communication channel. He also shows that the channel capacity can be achieved by a random code construction method. With a code rate smaller than the channel capacity, a communication system encounters negligible errors. It has always been a challenge to achieve channel capacity with low complexity algorithms. Polar coding is a method that achieves channel capacity with low complexity encoding and decoding.

Polar Codes:

Polar codes are a class of capacity-achieving linear forward error correction (FEC) block codes. The complexity of both encoding and successive cancellation (SC) decoding of polar codes are O(N log N), where N is the code block length. The recursive construction of both encoder and the SC decoder enables neat processing structures, component sharing and an efficient utilization of the limited sources. Polar codes provide a flexible selection of the code rate with 1/N precision such that an arbitrary code rate can be used without reconstructing the code. Polar codes are channel specific codes that means a polar code designed for a particular channel might not has a good performance for other channels. The important properties of polar codes are:

- Capacity achieving error correction performance.
- O(N log N) encoding and SC decoding complexity.
- Enhanced bit error rate (BER) with systematic polar codes.
- Adjustable code rate with 1 N precision without generating the code again.
- Channel specific recursive code construction.
- Achieves maximum likelihood (ML) performance by successive cancellation list (SCL) decoding with a sufficiently large list size.
- Block error probability of the SC decoder is asymptotically smaller than $2 \wedge \sqrt{N}$.

Decoding of polar codes is an active research problem. There are several decoding methods in literature such that these methods are SC, SCL, SC stack and belief propagation. The scope of this thesis includes SC and SCL methods. Due to low complexity encoding and decoding methods, implementation of polar codes at long block lengths is feasible for practical communication systems. In contrast, the

noticeable concern for polar codes at long block lengths is the decoding latency due to strong data dependencies. In this thesis, we consider moderate code block lengths such as 1024 and 256 in order to enhance finite length performance with limited latency and resource usage. Although the SC decoder asymptotically achieves channel capacity as N increases, the superior performance of the SC decoder decays at short and moderate code block lengths due to poor polarization. For this reason, the SC decoder does not provide as good performance as a maximum likelihood (ML) decoder at short and moderate block lengths. To overcome this issue, it is necessary to add more features to the algorithm such as tracking multiple possible decision paths instead of one such that the SC decoder does.

This algorithm uses beam search method for exploring the possible decoding paths efficiently. It is considerable as a greedy algorithm that approaches ML performance with sufficiently large list size, L. Since considering all 2NR possible decoding paths is impractical and too complex, the SCL algorithm has a restricted complexity such that at most L best possible paths can be traced. In that way, the algorithm operates with O(L N log N) computational complexity. The error correction performance is further improved by combining the SCL algorithm with a cyclic redundancy check (CRC) code. At the end of decoding, the SCL decoder selects a CRC valid path from among L surviving paths.

The adaptive SCL decoder:

The adaptive SCL decoder has three main components, SC, SCL and CRC decoders. Initially, the SC decoder is activated and a hard decision estimate vector is calculated. After that, the CRC decoder controls whether the hard decision vector is correct. If the CRC is valid, the hard decision vector is quitelikely to be the correct information vector.

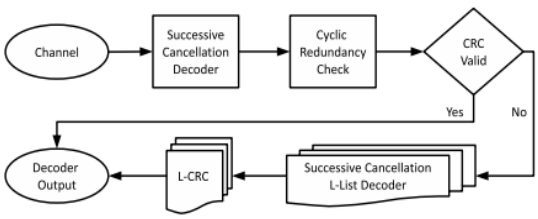

Figure 1.Data flow of the adaptive decoder.

In this case, the adaptive decoder is immediately terminated without the activation of the SCL decoder. In other case, when the CRC is invalid, the SCL decoder is activated and L information vector candidates are generated. Among these candidates, the CRC decoder selects a candidate, which has a valid CRC vector. If more than one CRC vector candidate is valid, the most probable one among these candidates is selected. Lastly, when none of the candidates has a valid CRC vector, the CRC decoder selects the most probable decision estimation vector to reduce BER.

Polar codes provably achieve the symmetric capacity of memoryless channels using the low-complexity successive-cancellation (SC) decoding algorithm. However, the SC algorithm is sequential in nature, leading to low-throughput decoders. 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 can achieve an information throughput of 1 GBPS. Fig. 2(a) is the graph representation for an (8, 4) polar code where u_0, u_1, u_2 and u_4 are frozen bits. Fig. 2(b) 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 or functions as defined in the Fast-SSC algorithm. 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.

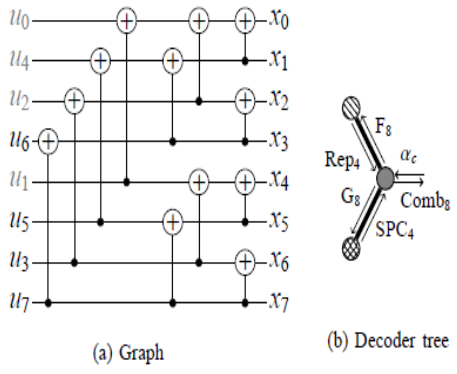

Fig. 2: From a graph to a Fast-SSC decoder tree.

II. POLAR CODES

A polar code is a linear block error-correcting code designed for a specific discrete

input, memoryless channel. From here on, we will assume that the channel has a binary input alphabetand is symmetric as well. Let n = 2m be the code length and let u = (u0, u1, . . . , un−1) and c = (c0, c1, . . . , cn−1) denote the input bits and the corresponding codeword1, respectively. The encoding operation has a Fast-Fourier-Transform like butterfly structure depicted in Figure 1 for n = 8. Note that the ordering of the ui bits in Figure 1 is according to the bit-reversed order: if we reverse the order of the bits in the binary representation of i, we then get the natural ordering. After u is encoded into c, the codeword c is sent over the underlying channel (the channel is used n times). Denote by y = (y0, y1, . . . , yn−1) the corresponding channel output. We now wish to decode y. This is done in terms of a *successivecancellation* decoder. That is, given y, we first try to deduce the value of u0, then that of u1, and so forth up until un−1.

Code Construction:

The aim of the polar code construction is to determine A and Ac sets according to the capacity of individual channels. Since polar codes are channel specific codes, the code construction may differ from channel to channel. Channel parameters, such as σ for BAWGNC and € for binary erasure channel (BEC) are an input to a code construction method. For BEC W, code construction for (N = 8, K = 4) polar code with an erasure probability € = 0.3 is shown in Figure 2.2.

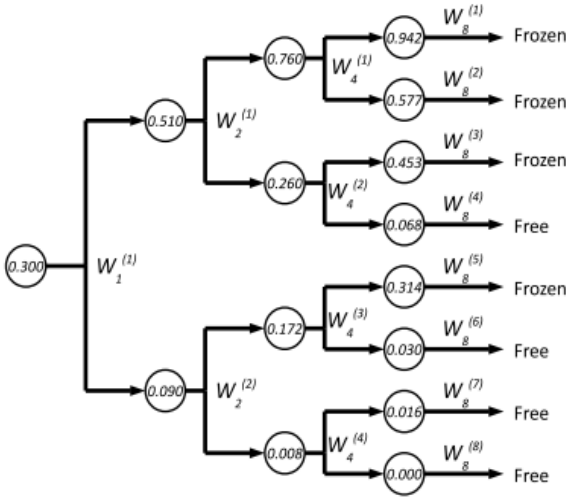

Figure 3: Code construction for BEC with N = 8, K = 4 and € = 0.3

Implementation methods such as pre computations, pipe-lined, and unrolled designs, have also been proposed to improve the throughput of SC decoders. These methods trade hardware complexity for gains in throughput. For example, it has been shown that the decoding latency may be reduced to *N* by doubling the number of adders in a SC decoder circuit. A similar approach has been used in a first ASIC implementation of a SC decoder to reduce the latency at the decision-level LLR calculations by *N*/2 clock cycles and provide a throughput of 49 Mb/s with 150 MHz clock frequency for a rate-1/2 code . In contrast, pipelined and unrolled designs do not affect the latency of the decoder; the increase in throughput is obtained by decoding multiple code words simultaneously without resource sharing. A recent study exhibits a SC decoder achieving 254 Gb/s throughput with a fully-unrolled and deeply-pipelined architecture using component code properties for a rate-1/2 code.

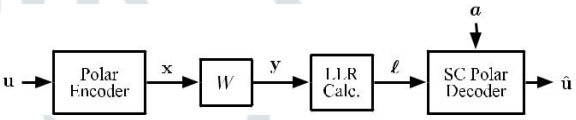

Fig. 4. Communication scheme with polar coding.

The present work is motivated by the desire to obtain high-throughput SC decoders with low power consumption, which has not been a main concern in literature so far. These desired properties are attained by designing completely combinational decoder architectures, which is possible thanks to the recursive and feed-forward (non-iterative) structure of the SC algorithm. Combinational decoders operate at lower clock frequencies compared to ordinary synchronous (sequential logic) decoders. However, in a combinational decoder an entire codeword is de-coded in one clock cycle.

III.Proposed method:

Pipelining can be applied to combinational decoders at any depth Pipelining can be applied to combinational decoders at any depth to adjust their throughput, hardware usage, and power consumption characteristics. Therefore, we also investigate the performance of pipelined combinational decoders. We do not use any of the multi-bit decision shortcuts in the architectures we propose. Thus, for a given block length, the combinational decoders that we propose retain the inherent flexibility of polar coding to operate at any desired code rate between zero and one. Retaining such flexibility is important since one of the main motivations behind the combinational decoder is to use it as an “accelerator” module as part of a hybrid decoder that combinesa

synchronous SC decoder with a combinational decoder to take advantage of the best characteristics of the two types of decoders.We give an analytical discussion of the throughput ofhybrid-logic decoders to quantify the advantages of the hybrid decoder.

The encoder input vector $\mathbf{u} \in F^N2$ consists of a *data* part $\mathbf{u}A$ and a *frozen* part $\mathbf{u}Ac$, where A is chosen in accordance with polar code design rules, We fix the frozen part $\mathbf{u}Ac$to zero in this study. We define a *frozen-bit indicatorvector* $\boldsymbol{a}$ so that $\boldsymbol{a}$ is a 0–1 vector of length N with

ai=0, if $i \in Ac$ =1,if $i \in A$.

The frozen-bit indicator vector is made available to the decoder in the system. The channel W in the system is an arbitrary discrete memoryless channel with input alphabet $X = \{0, 1\}$, output alphabet Y and transition probabilities $\{W(y|x) : x \in X, y \in Y\}$. In each use of the system, a codeword $\mathbf{x} \in F^N/2$ is transmitted,

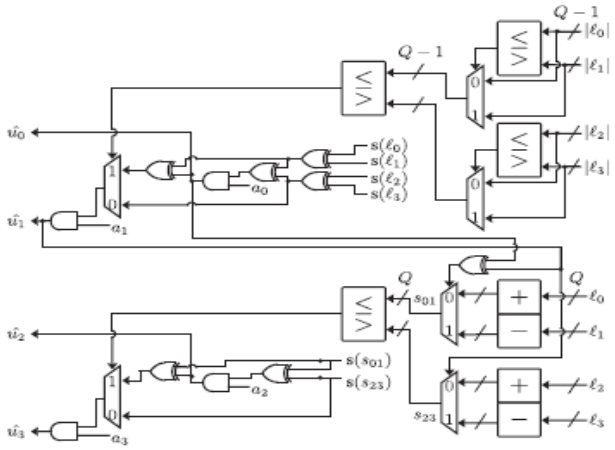

Fig. 5. Combinational decoder for N = 4.

SC DECODER USINGCOMBINATIONAL LOGIC

The pseudocode in Algorithm shows that the logic of SC decoding contains no loops, hence it can be implemented using only combinational logic. The potential benefits of a combinational implementation are high throughput and low power consumption, which we show are feasible goals. In this section, we first describe a combinational SC decoder for length N = 4 to explain the basic idea. Then, we describe the three architectures that we propose. Finally, we give an analysis of complexity and latency characteristics of the proposed architectures.

Combinational Logic for SC Decoding

In a combinational SC decoder the decoder outputs are expressed directly in terms of decoder inputs, without any registers or memory elements in between the input and output stages. Below we give the combinational logic expressions for a decoder of size N = 4, for which the signal flow graph (trellis) is depicted in Fig.
At stage 0 the LLR functions is given by

$$\ell'_0 = f(\ell_0, \ell_1), \quad \ell'_1 = f(\ell_2, \ell_3)$$

$$\ell''_0 = g(\ell_0, \ell_1, \hat{u}_0 \oplus \hat{u}_1), \quad \ell''_1 = g(\ell_2, \ell_3, \hat{u}_1).$$

At stage 1

$$\hat{u}_0 = s\left[f\left(f(\ell_0, \ell_1), f(\ell_2, \ell_3)\right)\right] \cdot a_0$$
$$\hat{u}_1 = s\left[g\left(f(\ell_0, \ell_1), f(\ell_2, \ell_3), \hat{u}_0\right)\right] \cdot a_1$$
$$\hat{u}_2 = s\left[f\left(g(\ell_0, \ell_1, \hat{u}_0 \oplus \hat{u}_1), g(\ell_2, \ell_3, \hat{u}_1)\right)\right] \cdot a_2$$
$$\hat{u}_3 = s\left[g\left(g(\ell_0, \ell_1, \hat{u}_0 \oplus \hat{u}_1), g(\ell_2, \ell_3, \hat{u}_1), \hat{u}_2\right)\right] \cdot a_3$$

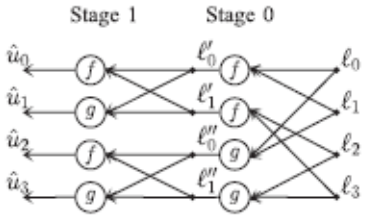

Fig. 6. SC decoding trellis for N = 4.

Architectures:

- *CombinationalDecoder:* A combinational decoder architecture for any block length N using the recursive algorithm in Fig. 4. This architecture uses two combinational decoders of size $N/2$, with glue logic consisting of one $fN/2$ block, one $gN/2$ block, and one size- $N/2$ encoder block.

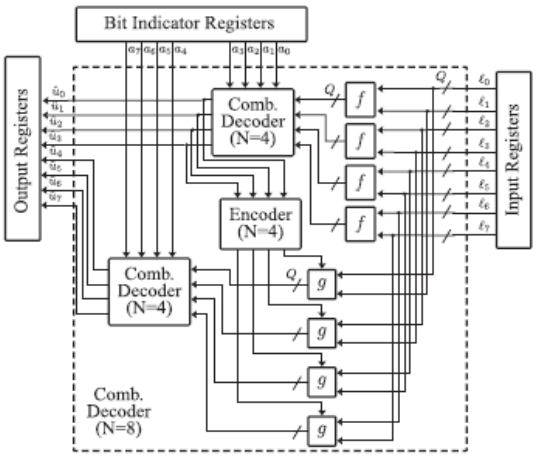

Fig. 7. RTL schematic for combinational decoder (N = 8).

- **Pipelined Combinational Decoder***:*

Unlike sequential circuits, the combinational architecture explained above

has no need for any internal storage elements. The longest path delay determines the clock period in such a circuit. This saves hardware by avoiding usage of memory, but slows down the decoder. In this subsection, we introduce pipelining in order to increase the throughput at the expense of some extra hardware utilization.

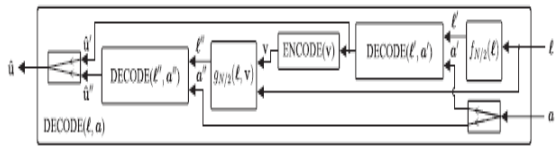

Fig. 8. Recursive architecture of polar decoders for block length *N*.

Hybrid-Logic Decoder:

In this part, we give an architecture that combines synchronous decoders with combinational decoders to carry out the decoding operations for component codes. In sequential SC decoding of polar codes, the decoder slows down every time it approaches the decision level (where decisions are made sequentially and number of parallel calculations decrease). In a hybrid-logic SC decoder, the combinational decoder is used near the decision level to speed upthe SC decoder by taking advantage of the GCC structure of polar code. The GCC structure is illustrated in Fig. 9.

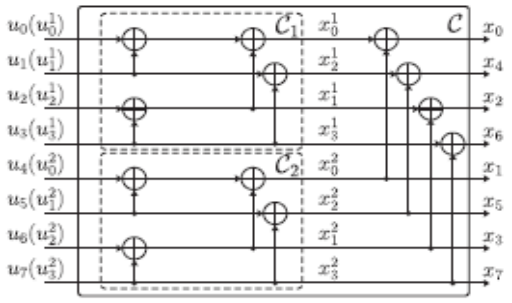

Fig. 9. Encoding circuit of *C* withcomponent codes *C*1 and *C*2 (*N* = 8 and *N*_ = 4).

IV.Simulation results

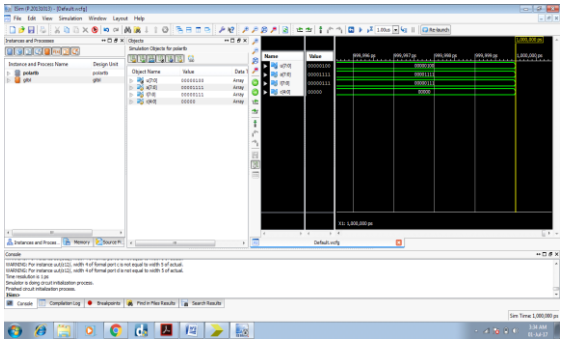

Fig.10.Simulated output of polar codes

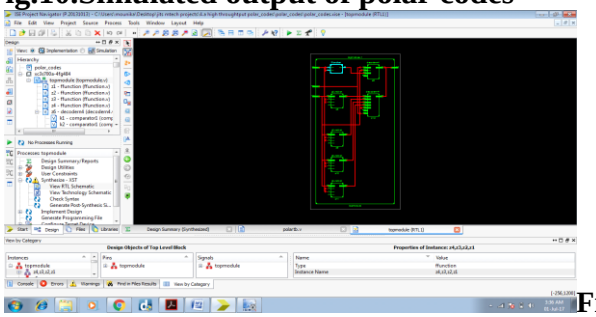

Fig .11. RTL schematic of proposed method

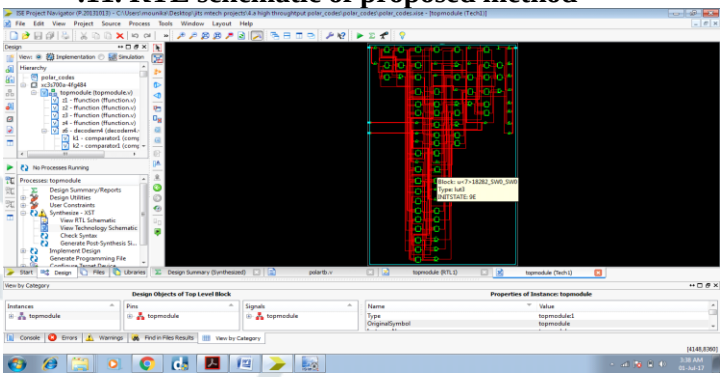

Fig.12. Technology schematic ofproposed method

V.Conclusion

An adaptive SCL algorithm was developed by combining the SC, the SCL and the CRC decoders. The implementation of the adaptive SCL decoder on FPGA was challenging for large list sizes due to increasing complexity and the excessive demand on routing paths. Thus, we achieved an implementation for the code block length N = 256 up to list size L = 8 and for N = 1024 up to list size L = 4. Although we had tried to implement with longer list sizes, the implementation failed due to routing congestion in FPGA. As a result, we achieved an adaptive SCL decoder, which has a throughput up to 225 Mb/s.

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, Jul. 2009.

[2] E. Arıkan, "Polar codes: A pipelined implementation," in *Proc. Int. Symp. Broadband Commun. (ISBC2010)*,Melaka, Malaysia, 2010, pp. 11–14.

[3] C. Leroux, I. Tal, A. Vardy, and W. J. Gross, "Hardware architectures for successive cancellation decoding of polar codes," 2010. [Online]. Available: http://arxiv.org/abs/1011.2919.

[4] A. Pamuk, "An FPGA implementation architecture for decoding of polar codes," in *Proc. 8th Int. Symp. Wireless Commun. (ISWCS)*, 2011, pp. 437–441.
[5] A. Mishra, A. Raymond, L. Amaru, G. Sarkis, C. Leroux, P. Meinerzhagen, A. Burg, and W. Gross, "A successive cancellation decoder ASIC for a 1024-bit polar code in 180 nm CMOS," in *Proc.IEEE Asian Solid State Circuits Conf. (A-SSCC)*, 2012, pp. 205–208.
[6] Y. Fan and C.-Y. Tsui, "An efficient partial-sum network architecture for semi-parallel polar codes decoder implementation," *IEEE Trans. SignalProcess.*, vol. 62, no. 12, pp. 3165–3179, Jun. 2014.