Performance Analysis of Successive Cancellation Decoder for Polar Codes with Two Stage Channel Estimator

J. Varsha[1], D. Bakialakshmi[2]

[1,2]Assistant professor, Department of ECE, Unnamalai Institute of Technology, Kovilpatti

ABSTRACT

Nowadays polar codes are becoming one of the most favorable capacity achieving Error Correction Codes (ECC) for their simple low encoding and decoding complexity, have received much attention in recent years; they probably achieve the theoretical capacity of discrete memoryless channels using the low complexity successive cancellation (SC) decoding algorithm. However, among the very few prior successive cancellation polar decoder designs, the required long code length makes the decoding latency high. As a new polar decoder, referred to as 2b-SC-Precomputation decoder, reduce the latency from (2n-1) to ((3n/4)-1) without performance loss, which can reduce the hardware complexity by two stage channel estimators. Furthermore, in our method, the decoding latency decreases rapidly with increasing throughput. Significant latency and throughput reductions are shown by simulation results and report synthesis results for ASIC.

KEYWORDS: Error Correction Codes (ECC), Successive Cancellation (SC), Polar decoder, Precomputation, Channel Estimator.

I. INTRODUCTION

Polar codes, as the first provable capacity-achieving codes over binary-input discrete memoryless channel (B-DMC) [1], have received significant attention among various forward error correction (FEC) codes. Due to their explicit structure and low complexity encoding/decoding scheme, polar codes have emerged as one of the most important codes in coding theory. To date, many efforts have addressed several theoretical aspects of polar codes. However, apart from not many publications have considered the VLSI design of polar decoders. In FPGA implementation of polar decoder based on the Belief-propagation (BP) algorithm was reported. Although BP decoder has advantages in parallel design, due to the requirement of large number of processing elements (PEs), the BP decoder is not attractive for practical applications. In successive cancellation (SC) polar decoders were presented. These low-complexity architectures are suitable for area-stringent applications; however, due to the inherent serial nature of the SC algorithm, these SC decoders fall short due to long latency and low throughput. In a pre-computation scheme was applied to the SC algorithm, which succeeded in reducing the overall latency from (2n-2) to (n-1). However, considering the penalty of increased hardware, the SC-Precomputation decoder does not show significant improvement with respect to hardware efficiency.

II. POLAR DECODER ARCHITECTURE

In this section, three hardware architectures of the new 2b-SC algorithm are presented. According to Fig. 2.1 the overall 2b-SC decoder mainly consists of three types of processing nodes: f, g and P nodes. Besides these nodes, a simple partial sum generator (PSG) is also needed to generate partial sum $\hat{u}_{sum}$. Since PSG block is similar to polar encoder with simple architecture, therefore in this section we focus on the architectures of and P nodes. Fig. 2.2 the block diagram of successive cancellation polar decoder architecture consists of two stage channel estimator and path decorrelator.

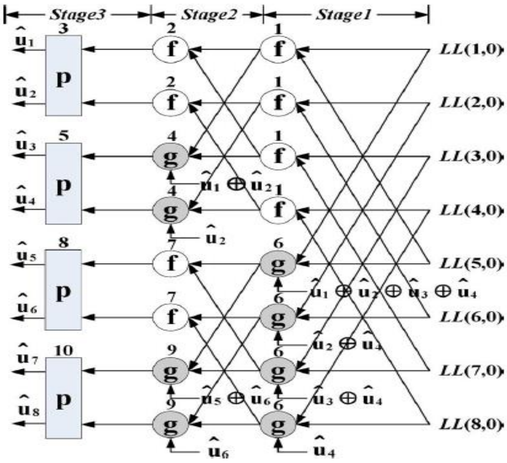


Fig 2.1: Polar Decoder Architecture

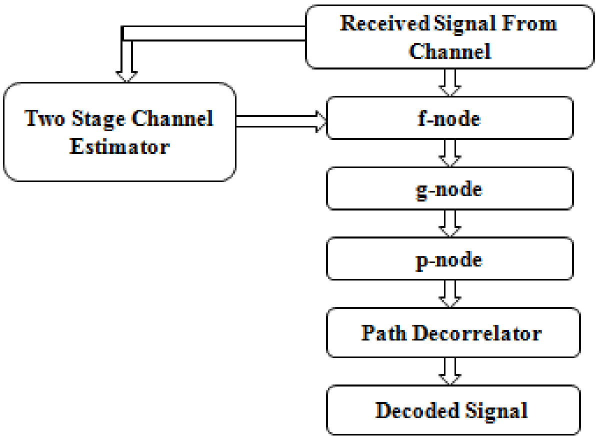


Fig 2.2: Block Diagram of Polar Encoder

2.1 f and g Node

The F node and g node are used to calculate the propagated LLR values. Here S2C is the block that performs the conversion from sign-magnitude form to 2's complement form, while C2S unit carries out the inverse conversion. Additionally, adder and subtractor are employed to carry out addition and subtraction between the two inputs. Finally, at the output end of the PE, control signal is used to determine the output as LLR(a) or LLR(b) ,which is propagated to the next stage.

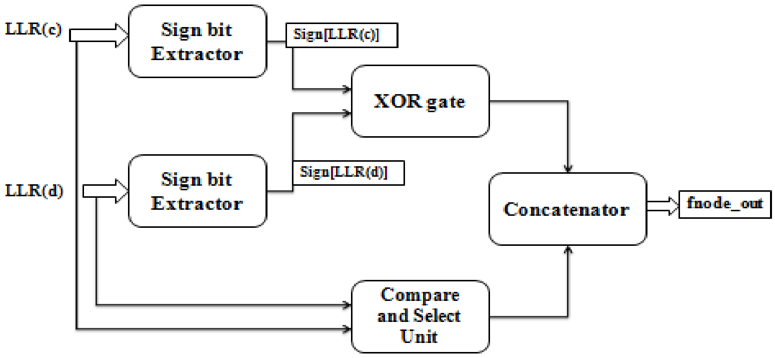


Fig 2.3: f-node architecture

As shown in Fig. 2.1, nodes are used in stage-1 and nodes are used in other stages to calculate the propagated LLR values. For simplicity of hardware design, the functions of and nodes are always implemented by unified processing elements (PEs) Fig. 2.3 and 2.4 shows the architecture of this PE based on the LLR version of and with min-sum approximation. Here S2C is the block that performs the conversion from sign-magnitude form to 2's complement form, while C2S unit carries out the inverse conversion. Additionally, adder and subtractor are employed to carry out addition and subtraction between the two inputs.

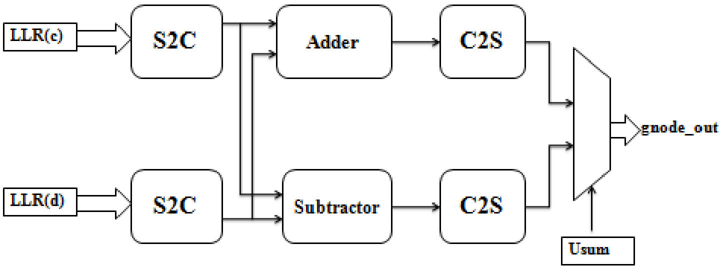


Fig 2.4: g-node architecture

The decision scheme in node has been described based on the LLR representation. To implement its function, a straightforward approach is to employ a sorting circuit and a signed adder. However, this method is too complex and is not hardware-efficient. After careful examination of Scheme-A, we observe that the node can be implemented with a very simple method, which is described as below.

2.2 p node

First, since the function of p node depends on the frozen conditions of $\hat{u}_{2i-1}$ *and* $\hat{u}_{2i}$, signals *frozen1* and *frozen2* are introduced to indicate whether $\hat{u}_{2i-1}$ and $\hat{u}_{2i}$ are frozen bits or not. If is frozen, *frozen1* will be 1, otherwise 0. Similarly, frozen2 will be 1 or 0 when $\hat{u}_{2i}$ is frozen or not. Secondly, the sign bits of *LLR* $(\hat{c})$ *and LLR* $(\hat{a})$ are employed for simplifying computations. Denoted as *sign (LLR* $(\hat{c})$*)* and *sign (LLR* $(\hat{a})$*)* these sign bits will be, respectively, 0 or 1 when the corresponding LLR values are non-negative or negative. Furthermore, the *comp* signal, which is the result of comparison between absolute value of *LLR* $(\hat{c})$ *and LLR* $(\hat{a})$ is also employed. When $|LLR(\hat{c})| \geq |LLR(\hat{a})|$, *comp* will be 1, otherwise 0. Accordingly, with the above five signals, we can obtain the truth table shown in Table 3.1 for $\hat{u}_{2i-1}$ or $\hat{u}_{2i}$ based on Scheme-A.

Then, with the help of above truth table, Boolean expression of $\hat{u}_{2i-1}$ or $\hat{u}_{2i}$ can be derived as follows:

$$\hat{u}_{2i-1} = \overline{frozen1}(sign(LLR(\hat{c}) \oplus sign(LLR(\hat{d}))$$

$$\begin{aligned}\hat{u}_{2i} = {} & \overline{comp}\;\overline{frozen2}\;sign(LLR(\hat{d})) \\ & + comp\;\overline{frozen1}\;\overline{frozen2}\;sign(LLR(\hat{d})) \\ & + comp\;frozen1\;\overline{frozen2}\;sign(LLR(\hat{c})).\end{aligned}$$

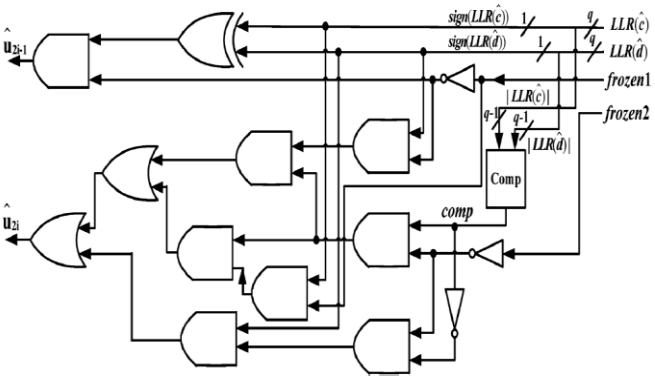


Fig 2.5: P-node architecture

2.3 Two Stage Channel Estimator

The proposed channel estimator has significant performance improvements, especially when it is applied in fast and selective fading channels. The proposed channel estimator includes the following features:

- provision of an efficient channel estimator architecture for low-complexity hardware implementation while keeping the high performance.

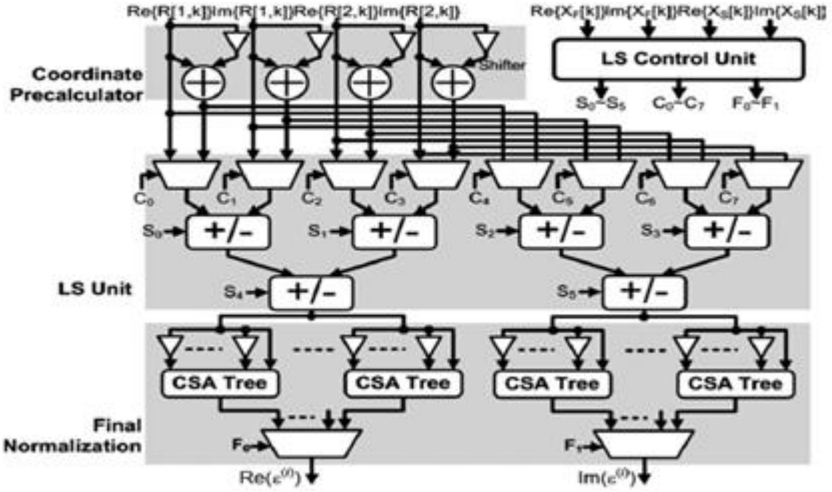


Fig 2.6: Two Stage Channel Estimator Architecture

Fig. 2.6 shows the design of the LS estimator which is composed of coordinate precalculators, LS units, an LS control unit and a final normalization. The coordinate precalculators are designed to generate the partial products of R[t,k] multiplied by the coordinate values. The coordinate precalculators support the modulations of BPSK, QPSK, and 16QAM, and the multiples should be Ex-OR gates.

III. EXPERIMENTAL RESULTS

3.1 Simulation Results

ModelSim tool is used to simulate the polar codes whether it is suitable or not then this tool is used to all the digital circuits. Run the polar codes and then finalize we check the operation of polar encoder. In this section, we consider the computational complexity of the SC decoding algorithm. In this section, three hardware architectures of the new 2b-SC algorithm are presented. According to Fig 5, the overall 2b-SC decoder mainly consists of three types of processing nodes: ***f, g*** and ***p*** nodes. Besides these nodes, a simple partial sum generator (PSG) is also needed to generate partial sum $\hat{u}_{sum}$. Since PSG block is similar to polar encoder with simple architecture, therefore in this section we focus on the architectures of ***f, g*** and ***p*** nodes. Here, u_{final1} and u_{final2} are output data.

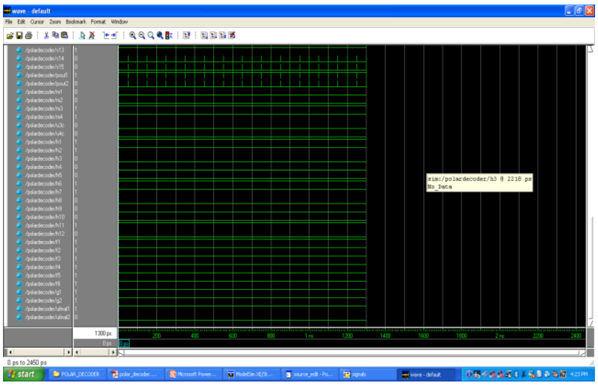
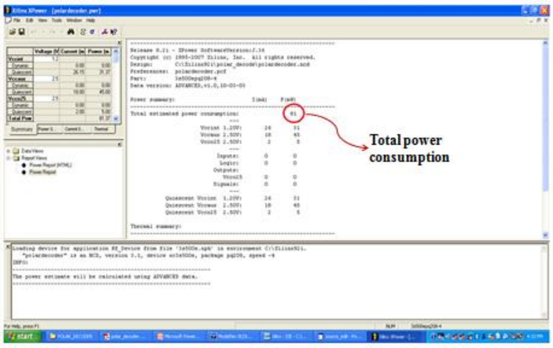


Fig 3.1: Simulation screenshot of polar decoder

3.2 Synthesized Report

Synthesis report gives the efficient usage of number of flip-flops used and bonded IOBs, number of slices used GCLKs and register used. Additionally, in order to demonstrate the advantage of the proposed architectures, we have implemented our designs for polar (1024, 512) code with Verilog HDL. Here tree-based 2b- SC- Precomputation architecture is selected for implementation.


Xilinx XPower - [polardecoder.pwr]
File Edit View Tools Window Help
Total power
consumption

```
Release 8.2i - XPower SoftwareVersion:I.34
Copyright (c) 1995-2007 Xilinx, Inc.  All rights reserved.
Design:       C:\Xilinx92i\polar_decode\polardecoder.ncd
Preferences:  polardecoder.pcf
Part:         3s500epq208-4
Data version: ADVANCED,v1.0,10-03-03

Power summary:                        I(mA)    P(mW)
----------------------------------------------------
Total estimated power consumption:              81
                                 ---
                   Vccint 1.20V:       24       31
                   Vccaux 2.50V:       18       45
                   Vcco25 2.50V:        2        5
                                 ---
                        Inputs:         0        0
                         Logic:         0        0
                       Outputs:
                         Vcco25         0        0
                       Signals:         0        0
                                 ---
           Quiescent Vccint  1.20V:    24       31
           Quiescent Vccaux  2.50V:    18       45
           Quiescent Vcco25  2.50V:     2        5

Thermal summary:
----------------------------------------------------
```

Fig 3.2: Power Report

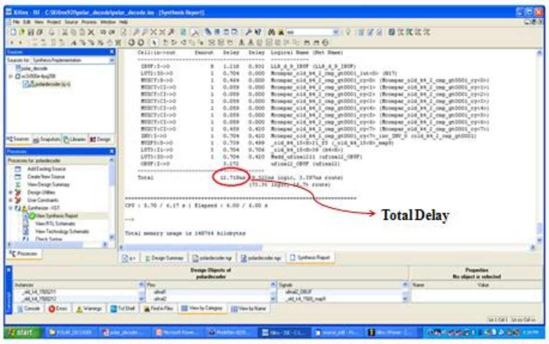


Fig 3.3: Latency Report

3.3 Performance Analysis

In this section, we analyze hardware performance of the proposed 2b- SC architectures and compare them with the state-of-the-art designs. Table 1 lists the synthesis results of reported polar (1024, 512) SC decoders.

Performance Parameters	Proposed	Bo Yuan et al.,[2014]	C. Leroux et al.,[2013]	A. Mishra et al.,[2012]
Latency	12.719ns	76.7ns	208ns	156ns
Throughput	1.5Mb/sec	1.48Mb/sec	0.83Mb/sec	1.07Mb/sec
Power Consumption	81mW	-	-	-

Table 3.1 : Performance analysis

IV CONCLUSION

In this paper, a novel reformulation for the last stage of the SC decoding is proposed. Based on this reformulation, a reduced- latency 2b-SC decoding algorithm is presented. In addition, with the use of overlapped scheduling and precomputation approaches, the decoding latency of 2b-SC design is further reduced and also increased the throughput. It is shown that by this modification, significant latency and complexity reductions are achieved with no sacrifice in error performance. Comparison results have shown that the proposed architectures turn out to be very attractive for real-time applications. Polar coding is a recently discovered coding technique which is capable of achieving the symmetric capacity of binary discrete memoryless channels. This paper focuses on developing approaches for designing low-latency low-complexity channel decoder architectures for modern communication systems. Decoding latency (or throughput) and hardware complexity are the two main focuses of polar decoder design. The two-stage estimator architecture introduced in polar decoder architecture which cancels the multi user interference, hardware complexity and reduced the time consumption. In the future, it could like to extend my work in the following blocks in polar decoder architecture. The path decorrelator architecture will be designing, that is introducing between the p node and destination of polar decoder in communication system, which can be used to reduce the error in signal, power consumption and latency.

V. REFERENCES

1. E. Arıkan, “Channel polarization: A method for constructing capacityachieving codes for symmetric binary-input memoryless channels,” *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
2. S. B. Korada, E. Sasoglu, and R. Urbanke, “Polar codes: Characterization of exponent, bounds, and constructions,” *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6253–6264, 2010.
3. R.Mori and T. Tanaka, “Performance of polar codes with the construction using density evolution,” *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 519–521, Jul. 2009.
4. I. Tal and A.Vardy, “How to construct polar codes,”May 2011, arXiv: 1105.6164v1.
5. A. Alamdar-Yazdi and F. R. Kschischang, “A simplified successivecancellation decoder for polar codes,” *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1378–1380, 2011.
6. I. Tal and A. Vardy, “List decoding of polar codes,” in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2011, pp. 1–5.
7. K. Niu and K. Chen, “Stack decoding of polar codes,” *Elect. Lett.*, vol. 48, no. 12, pp. 695–696, 2012.
8. E. Arıkan, “Systematic polar coding,” *IEEE Commun. Lett.*, vol. 15, no. 8, pp. 860–862, 2011.
9. E. Arıkan, “Polar codes: A pipelined implementation,” in *Proc. 4th Int. Symp. on Broad. Commun. ISBC 2010*, Jul. 2010, pp. 11–14.
10. A. Pamuk, “An FPGA implementation architecture for decoding of polar codes,” in *Proc. 8th Int. Symp. on Wireless Commun. Syst. (ICWCS)*, Nov. 2011, pp. 437–441.
11. C. Leroux, I. Tal, A. Vardy, andW. J. Gross, “Hardware architectures for successive cancellation decoding of polar codes,” in *Proc. IEEE ICASSP*, May 2011, pp. 1665–1668.
12. C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, “A semi-parallel successive-cancellation decoder for polar codes,” *IEEE Trans. Signal Processing*, vol. 61, pp. 289–299, Jan. 2013.
13. C. Zhang, B. Yuan, and K. K. Parhi, “Reduced-latency SC polar decoder architectures,” in *Proc. Int. Conf. Commun.*, June 2012, pp. 3471–3475.
14. C. Zhang and K. K. Parhi, “Low-latency sequential and overlapped architectures for successive cancellation polar decoder,” *IEEE Trans. Signal Processing*, vol. 61, no. 10, pp. 2429–2441, May 2013.
15. K. K. Parhi and D. G. Messerschmitt, “Static rate-optimal scheduling of iterative data flow programs via optimum unfolding,” *IEEE Trans. Comput.*, vol. 40, pp. 178–195, Feb. 1991.