International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 10, 2020
ISSN: 1475-7192

Optimized and Approximate Belief Propagation Decoder for Polar Codes

[1]V. Hari Krishna Prasad, [2]Mamatha Samson

Abstract--- *The polar code is one among the most effective error correcting code, attributable to the channel achieving property. In this paper, we propose an optimized and approximate belief propagation (BP) decoder for polar code for the first time with efficiency in delay. BP decoder is parallel in nature and is more attractive for low-latency applications. Adder is one of the key hardware blocks in BP decoder. By utilizing the approximate computation schemes and by using adders such as ripple carry adder and parallel self timed adder, delay can be reduced. The design is verified and synthesized using Xilinx ISE 14.7.*

Keywords--- *Polar codes, belief propagation decoder, approximate computation, Ripple Carry Adder, PASTA.*

I INTRODUCTION

The polar code is a new class of error correcting codes that demonstrably achieves the capacity of the underlying channels [2]. The result of a superb errors correcting performance is noninheritable when the code is satisfactorily long [3]. Among some manuscript's handling the hardware implementation [2] provided a coding structure that arranges all of the message bits in an absolutely parallel method. The absolutely parallel design is attractive butit is not acceptable for long polar codes because of the hardware complexity.

The polar codes are provably ability accomplishing and their regular structure promises energy-efficient codec designs. Different decoding concepts are introduced for polar codes[4-13]. Two main decoding concepts are successive cancellation(SC) and belief propagation(BP). SC decoding is serial and BP decoding is parallel nature. Belief propagation (BP) decoding had drawn lots of interest when creation of the polar codes took place. Based at the component graph representation of the codes [6], authors in [7], [8] had investigated the performance of the BP interpreting. Results showed that BP decoding had particular benefits with regard to the decoding latency and throughput.

The belief propagation(BP) decoding algorithm for polar codes is based on the factor graph representation of the code [7]. Each node is associated with two types of messages: left-to-right messages and right-to-left messages. In every new release, the message replacement will start from the leftmost column to rightmost column, then from the rightmost to leftmost. Adders are primary circuits to implement any mathematical operationin hardware form.Rapid and correct operation of virtual system depends on the performance of adders. The core of each microprocessor, digital signal processor (DSP), and information processing application specificintegrated circuit (ASIC) is its data path.

1,2, Department Electronics & Communication Engineering, GRIET Hyderabad,Telangana, India.

Received: 27 Feb 2019 | Revised: 20 Mar 2019 | Accepted: 30 Apr 2020

It is regularly the vital circuit thing if hardware cost and power dissipation is concerned. So here in this paper the belief propagation decoder (BPD) of polar codes is designed to offer approximated values with high performance in speed by usingoptimized adders.

II RELATED WORK

II.I. Approximate BP Polar Decoder

Before giving information of BP polar decoders, [1] discussed about the division theme first. For (64; 32) polar code and discovered that the values of LLR specifically distributed within the interval of [-35, 35], this implies that one sign bit and 5 integer bits are needed a minimum for quantizationto get satisfying performance [9]-[10].

Figure 1 shows factor graph of belief propagation decoding for (8, 4) polar code. Polar BPD illustrated for n-level factor graph[15]. For (8, 4) polar code, 8 represents information and 4 represents frozen bits. 8*(3+1)=24 nodes are present in 3 level factor graph.

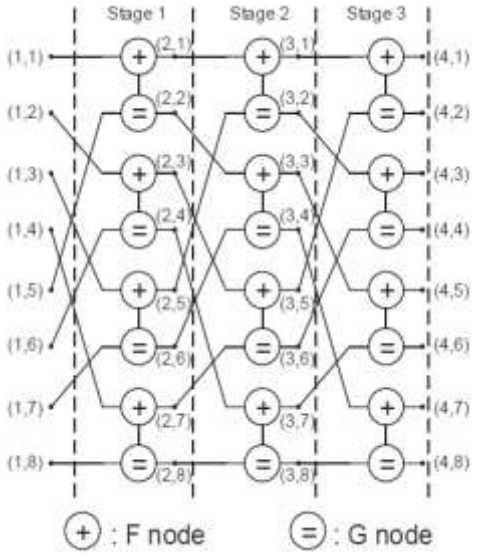

Fig. 1. Factor graph of BP decoding with N = 8

$$F(x, y) = x + y \quad (1)$$

$$G(x, y) = \text{sign}(x)\,\text{sign}(y)\,\min(|x|, |y|) \quad (2)$$

$$d_j = \begin{cases} 0 & if\ R_{4,j} \geq 0, \\ 1 & else \end{cases} \quad (3)$$

If $(p_{[n-1:k]}) \geq (q_{[n-1:k]})$,then the approximate G node predicts $(s_{[n-1:0]}) = (q_{[n-1:0]})$. Otherwise we have $(s_{[n-1:0]}) = (q_{[n-1:0]})$. However, while $(p_{[n-1:k]}) = (q_{[n-1:k]})$ and the neglected k bits of 'p' is small that those of q, got a wrong result. Supposing that p and q are random numbers with uniform distribution, then the chance of $(p_{[n-1:k]}) = (q_{[n-1:k]})$ and the possibility of the $(p_{[k-1:0]}) < (q_{[k-1:0]})$ are derived as follows:

$$P\left(p_{[n-1:k]} = q_{[n-1:k]}\right) = \frac{1}{2}^{n-k} \quad (4)$$

$$P\left(p_{[k-1:0]} = qb_{[k-1:0]}\right) = \frac{2}{2^{n+1}}^{k-1} \quad (5)$$

II.II. Approximate Architecture for G Node

Generally, LLR messages are in the form of sign magnitude. The purpose of G-node is magnitude comparison for finding minimum number from the inputs. In conventional G-node, almost all bits from MSB to LSB are need to be compared to find minimum number.There might be a chance of delay with conventional G-node to find which issmall. Conventional G-node result in a long time as well as high power consumption. The error rate(ER) for the approximated G-node is represented in equation (6).

$$ER=\frac{1}{2}^{n-k} \cdot \frac{2^k-1}{2^{k+1}} = \frac{2^k-1}{2^{k+1}} \quad (6)$$

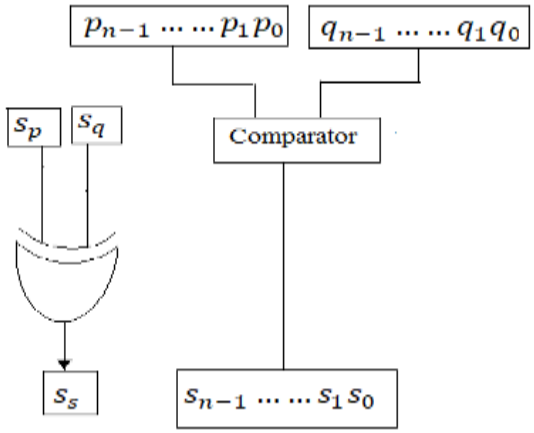

Fig 2: Conventional G node.

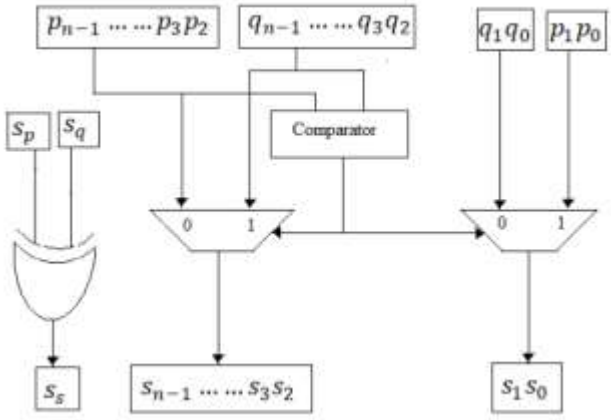

Fig. 3 Approximate architecture for G-node

According to (6) for a specific n, a bigger k will purpose more overall performance loss and less hardware in take [12]. Half of-rate polar codes with code period N = 64 for simulationused in [1]. The quantization scheme includes 1 sign bit, 5 integer bits, and three fractional bits.

II.III. Approximate Architecture for F Node

Notice that the conventional F-node suffers from long delay because of change in information format. Also, delay and hardware utilization increased due to AOU and comparator in traditional structure.So, in this phase, an efficient approximate structure for F node is shown in Fig. 4. For this approximate architecture, subtraction is at once executed rather than doing data conversion before computation. As a result, we only want one data format conversion for F node.$m_a + m_b$ &$m_a - m_b$ are computed at atime and the value of m_s is selected from those values. When S_a ^S_b = zero, the F node puts in force $m_a + m_b$. Otherwise, it implements$m_a - m_b$. When $m_a \geq m_b$, S_s=S_a. Otherwise, $S_s = S_b$.Approximate computation schemes are added to alleviate the contradiction between higher throughput and hardware consumption. Add one unit (AOU) plays most important role in very last computation. In Fig. 5 shown below, only add 1 to the m low-order bits of the input data. In addition, the carry out generated when including one is dropped without being propagated to MSB bits. These m low-order bits are set to 1, if the carry out bit is equal to 1.

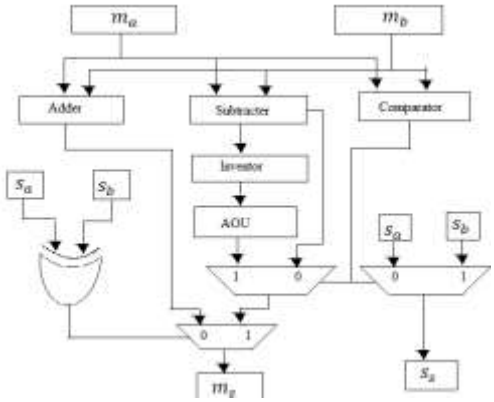

Fig. 4. Approximate architecture for F-node

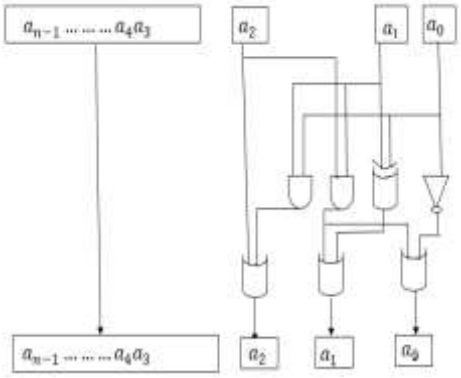

Fig. 5 Block Diagram of Approximate AOU

III PROPOSED WORK

The approximation-based circuits ofF-node is modified to attain delay efficiency; by changing various types of adders and their variation in terms of delay are observed.Fig. 2 gives the conventional architecture of G node and approximate circuit is shown in the Fig. 3.

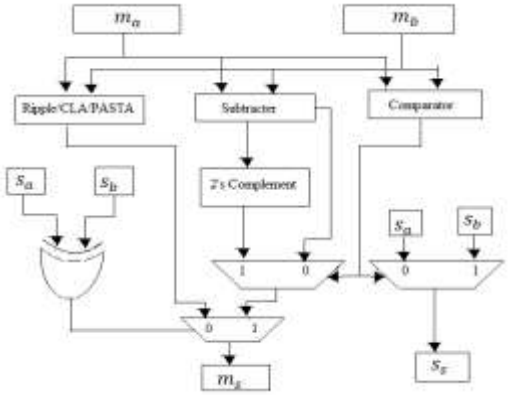

Fig. 6: Proposed F node.

Figure 6 shows proposed F-node. When compared to existing architecture of F-node, the proposed architecture shows the reduction in terms of area. Here m_a and m_b are the inputs and m_s is output. Here the adder which plays a major role in computation in F-node is modified. The add one circuit and inventor is replaced by 2's complement block. The performance of the F-node is verified by trying different types of adders in it. The adders considered are ripple carry adder, and PASTA adder.

III.I. Ripple Carry Adder

This method is an asynchronous addition networks. Linking the N full adders printed material N bit Binary adder. In this perform of going before entire adder turns into the enter bring for the following complete adder. It ascertains total and convey as indicated by the accompanying conditions.

$$S_i = A_i \hat{} B_i \hat{} C_i \tag{7}$$

$$C_{i+1} = A_i.B_i + B_i.C_i + C_i.A_i \tag{8}$$

Where i = 0, 1, 2, 3 …n-1

RCA is the slowest in all adders yet anyway exceptionally smaller in size. If the ripple carry adder(RCA) is completed with the useful resource of concatenating N complete adders, the delay of such an adder is 2N gate delays from C_{in} toC_{out}.

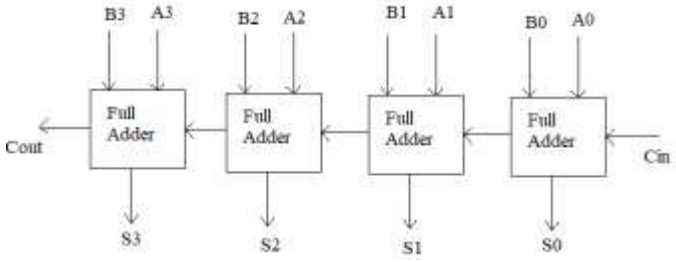

Fig. 7: Block Diagram of RCA

III.II. PASTA(Parallel Self-Timed Adder)

Another adder is Pasta adder, the general block figure of the Parallel Self-Timed Adder (PASTA) is exhibited in Fig.8. Multi bit adders frequently developed from single piece adders utilizing combinational and successive circuits for offbeat or synchronous structure.

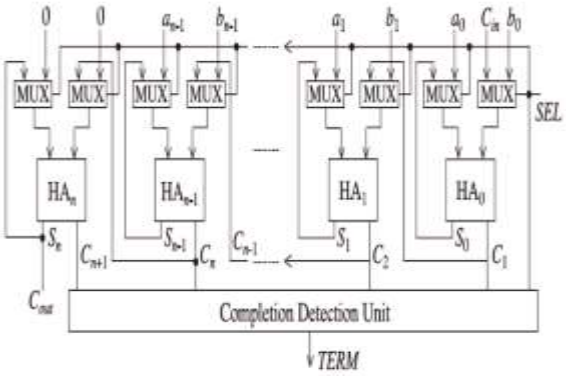

Fig. 8: Parallel self- Timed adder.

The contribution for two-input multiplexers relates to the request acknowledgement flag and can be a zero to one amendment signification by SEL. It will initially choose the real operands among SEL equal to zero and can change to input/convey ways that for ensuing cycles utilizing SEL equal to one. The adder ab initio acknowledges two operands to perform half-additions for every bit. during this manner, it emphasizes utilizing previous created convey and aggregates to perform half-adder over and over till all convey bits are eaten and settled at zero dimension. Ripple carry adder is just arrangement of full adders associated consecutive wherever carry propagates from 1^{st} full adder to last one.Delay is most extreme for this situation on the grounds that the yield won't get produced until the point when convey has proliferated until the last full adder. Convey look forward contains combinational circuit which computes already. Area is to a great extent expanded for this situation. Obviously, delay is substantially less.

IV RESULTS AND DISCUSSIONS

To show the advantage of the proposed approximate F-node over F-node in BP decoder, the relating execution results distinctive types of adders in F-node design is explained. It has shown that the planned approximate BP decoder shows sensible hardware reduction than the correct one. Compared with the prevailing F-node in decoder, the reduction in delay is discovered. The Comparison results are shown in below mentioned table I.

Here the below table presents the results (in terms of area in LUT's and delay in nanoseconds) with comparison. Which shows the comparisons between F nodes with different adders.

TABLE I: F-NODE Comparison With Differentadders

Adders in F-node	Area(LUT's)	Delay(ns)
RCA	51	12.093
PASTA	46	10.431

Received: 27 Feb 2019 | Revised: 20 Mar 2019 | Accepted: 30 Apr 2020

Figure 9 shows simulation results of F-node. The first waveform represents 8bit output, which is indicated by 'm'.Second waveform represents sign bit output, which is indicated by's'. Third and fourth waveforms represent 8bit inputs, indicated by 'ma', 'mb' respectively. Remaining two waveforms represents sign bits of inputs, indicated by 'sa' & 'sb'.

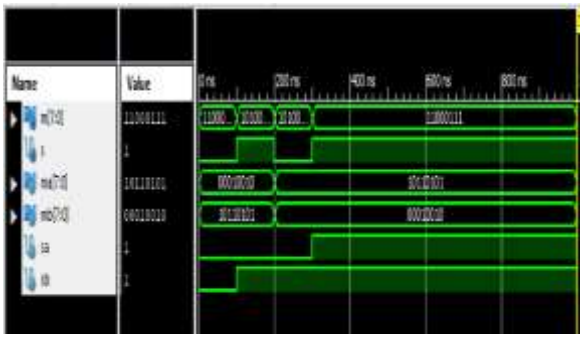

Fig. 9: F-node simulation result.

Figure 10 shows simulation results of G-node. The first waveform represents sign bit output, which is indicated by 'sz'. Second waveform represents 8-bitoutput, which is indicated by 'Z'. Third and fourth waveforms are represents two sign bit inputs, indicated by 'sx', 'sy' respectively. Remaining two waveforms represents 8-bit inputs, indicated by 'X' & 'Y'.

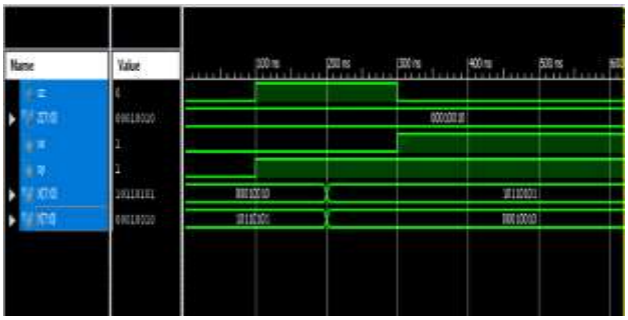

Fig 10: G-node simulation result.

Fig 11: BPD simulation result.

Figure 11 shows simulation results for BPD. The first eight waveforms are represent final output, which is indicated by d1, d2, d3, d4, d5, d6, d7 & d8 respectively. Next eight waveforms represent inputs, which is indicated by u1, u2, u3, u4, u5, u6, u7 & u8 respectively.

V CONCLUSION

In this paper, optimized and approximation supported BP polar code decoder is implemented. The results shows that an approximate computation method causes negligible performance degradation compared with the standard one in addition to reducing the delay. Delay seems to improve even more (13.74% for 8 bits) usingPASTA adders/subtractors in F-node of BPD. This approximation of architecture can be applied for other polar decoders in future.

REFERENCES

[1] Manoharan, Rajesh, et al. "Selection of Intermediate Routes for Secure Data Communication Systems using Graph Theory Application and Grey Wolf Optimization Algorithm in MANETs." *IET Networks* (2020).
[2] Rajesh, M., Gnanasekar, J.M. Path Observation Based Physical Routing Protocol for Wireless Ad Hoc Networks. *Wireless Pers Commun* **97,** 1267–1289 (2017). https://doi.org/10.1007/s11277-017-4565-9
[3] Rajesh, M. Streamlining Radio Network Organizing Enlargement Towards Microcellular Frameworks. *Wireless Pers Commun* (2020). https://doi.org/10.1007/s11277-020-07336-9
[4] Menghui Xu, Shusen Jing, "approximate belief propagation decoder for polar codes" ICASSP 2018, pp. 1169–1173.
[5] Erdal Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," IEEE Transactions on Information Theory, vol. 55, no. 7, pp. 3051–3073, 2009.
[6] Franc¸ois Leduc-Primeau, Saied Hemati, Warren J Gross, and Shie Mannor, "A relaxed half-stochastic iterative decoder for LDPC codes," in Proc. IEEE Global Telecommunications Conference (GLOBECOM), 2009, pp. 1–6.
[7] Chuan Zhang, Bo Yuan, and Keshab K Parhi, "Reduced- latency SC polar decoder architectures," in Proc. IEEE International Conference on Communications (ICC), June 2012, pp. 3471–3475.
[8] Chuan Zhang and Keshab Parhi, "Low-latency sequential and overlapped architectures for successive cancellation polar de- coder," IEEE Transactions on Signal Processing, vol. 61, no. 10, pp. 2429–2441, 2013.
[9] Xiao Liang, Chuan Zhang, Menghui Xu, Shunqing Zhang, and Xiaohu You, "Efficient stochastic list successive cancellation decoder for polar codes," in Proc. IEEE International System- on Chip Conference (SOCC), 2015, pp. 421–426.
[10] Erdal Arıkan, Haesik Kim, Garik Markarian, U Ozgur, and Efecan Poyraz, "Performance of short polar codes under ML decoding," in Proc. IEEE ICT-Mobile Summit, Santander, Spain, Jun. 2009.
[11] Yuanrui Ren, Chuan Zhang, Xing Liu, and Xiaohu You, "Efficient early termination schemes for belief-propagation decoding of polar codes," in ASIC (ASICON), 2015 IEEE 11th Inter- national Conference on. IEEE, 2015, pp. 1–4.
[12] Junmei Yang, Chuan Zhang, Huayi Zhou, and Xiaohu You, "Pipelined belief propagation polar decoders," in Proc. IEEE International Symposium on Circuits and Systems (ISCAS), May 2016, pp. 413–416.
[13] Menghui Xu, Xiao Liang, Chuan Zhang, Zhizhen Wu, and Xi- aohu You, "Stochastic bp polar decoding and architecture with efficient re randomization and directive register," in Signal Processing Systems (SiPS), 2016 IEEE International Workshop on. IEEE, 2016, pp. 315–320.
[14] Chu Hsiang Huang, Yao Li, and Lara Dolecek, "Belief propagation algorithms on noisy hardware," IEEE Transactions on Communications, vol. 63, no. 1, pp. 11–24, 2015.
[15] Yangcan Zhou, Jun Lin, and Zhongfeng Wang, "Efficient approximate layered LDPC decoder," in Proc. IEEE Internation- al Symposium on Circuits and Systems, 2017, pp. 1–4.
[16] Pascal Giard, Gabi Sarkis, Claude Thibeault, and Warren J Gross, "A 237 Gbps unrolled hardware polar decoder," Electronics Letters, vol. 51, no. 10, 2014.
[17] G. Sarkis and W. J. Gross, "Increasing the throughput of polar decoders," IEEE Commun. Lett., vol. 17, no. 4, pp. 725–728, Apr. 2013.
[18] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, W. J. Gross, "Fast polar decoders: Algorithm and implementation," IEEE J. Sel. Areas Commun., vol. 32, no. 5, pp. 946–957, May 2014.
[19] I. Tal and A. Vardy, "List decoding of polar codes," in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jul./Aug. 2011, pp. 1–5.
[20] K. Niu and K. Chen, "Stack decoding of polar codes," Electron. Lett., vol. 48, no. 12, pp. 695–696, Jul. 2012.