RECONFIGURABLE DECODER FOR LDPC AND POLAR CODES

Devisri Snigdha Kaduluri
Assistant Professor, Dept. of ECE, S.V. College of Engineering

ABSTRACT: With low-density parity-check (LDPC) code and polar code designated because the commonplace codes for 5G eMBB state of affairs, one challenge is the way to improve the hardware potency once both decoders are needed by one system. Since LDPC and polar codes will be decoded with belief propagation (BP) algorithms, this similarity permits U.S. to style a reconfigurable decoder, which can decipher each code at the value of only 1 decoder. Numerical and implementation results also are given during this paper to point out that the planned decoder achieves higher hardware potency than complete LDPC or polar decoder, while not harming the error performance.

Index Terms: LDPC codes, polar codes, belief propagation, reconfigurable decoder.

1. INTRODUCTION

Low-density parity-check (LDPC) code was initial projected by Gallager in 1961, and have become illustrious for its close to Shannon-capacity performance and high secret writing similarity. The standard secret writing algorithmic program for LDPC is belief propagation (BP) that takes advantage of the code poorness and reiterative secret writing. For higher balance of performance and complexity, variants of BP like min-sum and bedded minsum were projected. Nowadays, LDPC code has been widely adopted for applications and become the quality code for eMBB knowledge channel by 3GPP. Polar code is another breakthrough in channel writing once LDPC code. fictitious by Arıkan in 2008, it's the primary code proved to attain data rate of radial binary-input discrete memoryless channels (B-DMCs). Recently, polar code has been hand-picked because the code for eMBB management channel by 3GPP. For polar secret writing, consecutive cancellation (SC) was projected along with polar code. Then, SC list (SCL)

was projected to any improve the error performance of SC. though SCL considerably improves the performance, it suffers from high secret writing complexness. What is more, since SC and SCL decoders are tree-searching primarily based, they suffer from low similarity and high delay. To handle the matter, BP decoding, which might be applied in parallel, was utilized for top turnout applications. Since each LDPC and polar codes are set as commonplace codes for eMBB, one amongst the utmost challenges is to implement each decoder expeditiously at the identical time. One straightforward resolution is to own 2 separated decoders which be sure of each codes, severally. Admittedly, high level transforms like folding and fine-grain optimization can facilitate to boost the potency.

BP LDPC Decoding Algorithm

An (m, k) LDPC code has an (m − k) × m parity-check matrix H, where m denotes code word length and k denotes data word length (m > k). Its BP decoding is based on a factor graph which is illustrated in Fig. 1. A '1' located at the j_{th} row and the i_{th} column of H indicates an edge between the j_{th} check node (CN) and the i_{th} variable node (VN).

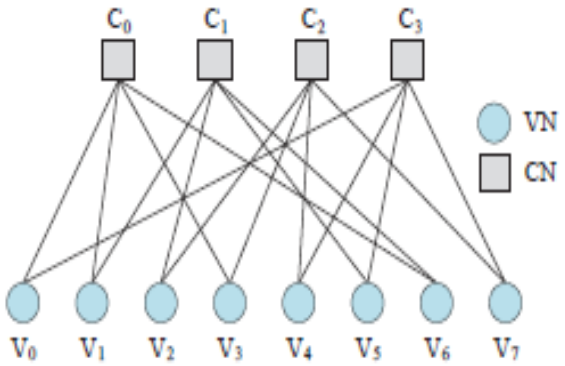

Fig.1: Factor graph of a regular (8; 4) LDPC code.

In each pair of nodes, CN-to-VN messages $L(r_{ji})$ and VNto- CN messages $L(q_{ij})$ are propagated and updated iteratively according to the following equations:

$$L(r_{ji}) = \left(\prod_{i' \in R_j / i} \operatorname{sgn}(L(q_{i'j})) \right) \times \min_{i' \in R_j / i} |L(q_{i'j})|, \quad (1)$$

$$L(q_{ij}) = L(c_i) + \sum_{j' \in C_i / j} L(r_{j'i}), \quad (2)$$

where R_j denotes the set of column locations of the 1's in the j_{th} row and C_i denotes the set of row locations of the 1's in the i_{th} column, respectively. Besides, $L(c_i)$ is used for initialization, while $L(Q_i)$ is iteratively computed for c_i.

BP Polar Decoding Algorithm

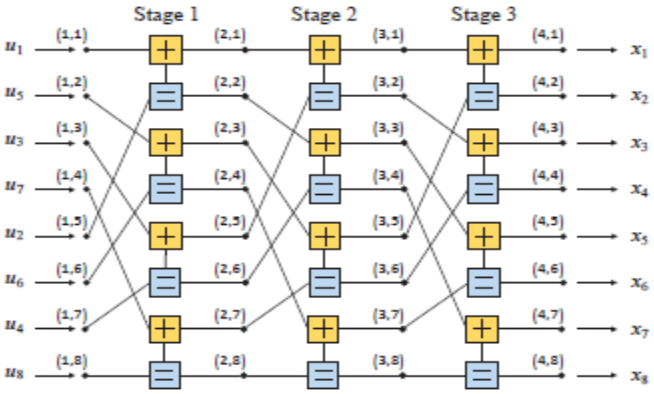

Fig.2: Factor graph of a polar code with N = 8.

BP coding for a polar code with code length $N = 2^n$ is also supported an element graph. Fig.2 a pair of illustrates the issue graph of associate 8-bit polar code. The issue graph has n+1 node stages, and each stage consists of N nodes. The label for the j_{th} node in the ith stage is denoted by (i; j). There are n×N/2 basic computational blocks (BCBs), and every of them is connected with 4 nodes. In Fig. 3, every node is related to 2 forms of LLRs: right-to-left messages L and left-to-right messages R. During the BP coding procedure, 2 kinds of messages are passed and updated iteratively between adjacent nodes according to the subsequent equations:

$$\begin{cases} L_{i,j} = g(L_{i+1,2j-1}, L_{i+1,2j} + R_{i,j+N/2}), \\ L_{i,j+N/2} = g(R_{i,j}, L_{i+1,2j-1}) + L_{i+1,2j}, \\ R_{i+1,2j-1} = g(R_{i,j}, L_{i+1,2j} + R_{i,j+N/2}), \\ R_{i+1,2j} = g(R_{i,j}, L_{i+1,2j-1}) + R_{i,j+N/2}. \end{cases} \quad (3)$$

Using a min-sum algorithm, the function g(x; y) is approximated by Eq. (4) in implementation.

$$g(x, y) \approx \operatorname{sgn}(x)\operatorname{sgn}(y) \min(|x|, |y|). \quad (4)$$

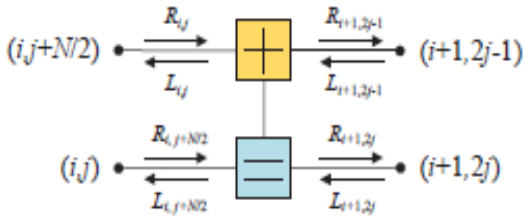

Fig.3: A BCB of polar codes

2. PROPOSED RECONFIGURABLE DECODER

In this section, a hardware design for the reconfigurable BP decoder is planned. We'll use a selected example to describe it very well.

Basic Modules

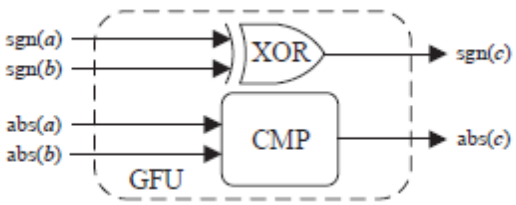

Fig.4: Hardware design for a GFU.

Two vital factors are concerned in Eqs. (1) and (4): the sign and also the minimum. As shown in Fig. 4, we have a tendency to style a g- function unit (GFU), that is created of a comparison (CMP) and associate exclusive-OR (XOR) gate. Given two inputs a and b, the CMP and also the XOR is meant to get the minimum of 2 absolute values and also the XOR of 2 signs, severally.

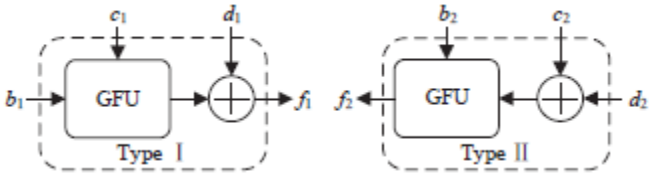

Fig.5: Two forms of design for computation

Architecture for Polar Decoder

For polar decoding, 2-input GFUs are used to compute Eq. (3), which can be classified into two forms:

$$\begin{cases} f_1 = g(b_1, c_1) + d_1, \\ f_2 = g(b_2, c_2 + d_2). \end{cases} \qquad (5)$$

As shown in Fig. 5, the corresponding structure for every type is each designed with a GFU associate degreed an adder. Fig.6 illustrates a parallel design for a BCB. gray squares represent memories, that are used for storing L and R messages.

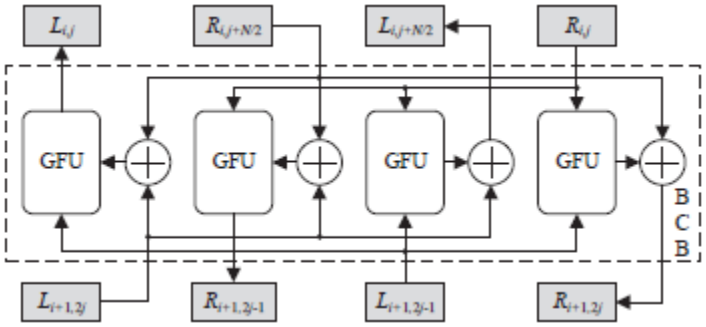

Fig.6: Parallel architecture for a BCB.

Architecture for LDPC Decoder

Since there are more than two $L(q_{ij})$s on the right side of Eq. (1), several GFUs are combined to carry out LDPC decoding process. We can design a CN unit (CU) by observing the recursive property of Eq. (6):

$$\left(\prod_{i=1}^{3} \operatorname{sgn}(a_i)\right) \times \min_{1 \le i \le 3} |a_i| = g(g(a_1, a_2), a_3). \qquad (6)$$

Fig. 7 illustrates the architecture for a 3-input CU and a 4-input CU. Generally, a CU with t(t > 3) inputs and 1 output is made up of t -1 GFUs.

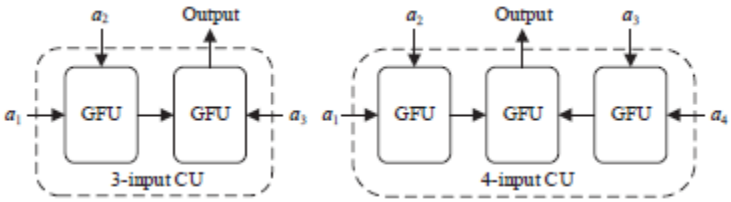

Fig.7: Architecture for 3-input and 4-input CUs.

Then, consider a regular (12; 6) LDPC code with $H_{6\times12}$ listed in Eq. (7) as an example.

$$\mathbf{H}_{6\times12} = \begin{bmatrix} 1&1&1&1&0&0&0&0&0&0&0&0 \\ 1&0&0&0&1&1&1&0&0&0&0&0 \\ 0&1&0&0&1&0&0&1&1&0&0&0 \\ 0&0&1&0&0&1&0&0&0&1&1&0 \\ 0&0&0&1&0&0&0&1&0&0&1&1 \\ 0&0&0&0&0&0&1&0&1&1&0&1 \end{bmatrix}. \qquad (7)$$

Notice that each CN has 4 inputs and 4 outputs, while each VN has 3 inputs and 3 outputs. Fig. 8 presents an architecture for the 0th CN. In the CN, 4 CUs are used for calculating 4 groups of Eq. (1) simultaneously. Besides, 3 adders are needed to construct a VN. After 2 $L(q_{ij})$s and 1 $L(c_i)$ are put in, 2 $L(r_{ji})$s and 1 $L(Q_i)$ are calculated by Eq. (2) and Eq. (8), respectively.

$$L(Q_i) = L(c_i) + \sum_{j \in C_i} L(r_{ji}). \qquad (8)$$

In total, 48 GFUs and 36 adders are used to construct CNs and VNs.

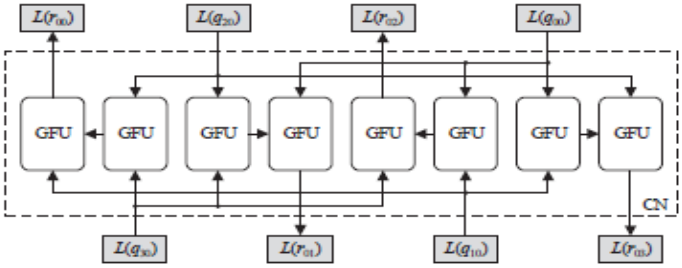

Fig.8: Architecture for the 0_{th} CN

Reconfigurable BP Decoder

To implement each cryptography processes, we have a tendency to style the merged BCB (MBCB). Fig. nine illustrates associate degree design for the 9-input and 6-output MBCB, that is various to hold out a BCB's computation or a part of the 0_{th} CN's computation. There are 2 modes of inner association within the MBCB. Black lines and red lines represent the mode for polar cryptography and LDPC cryptography, severally. The input S is employed for switching the 2 association modes.

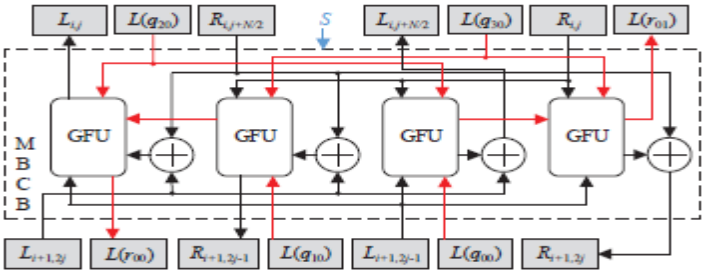

Fig.9: Architecture for an MBCB

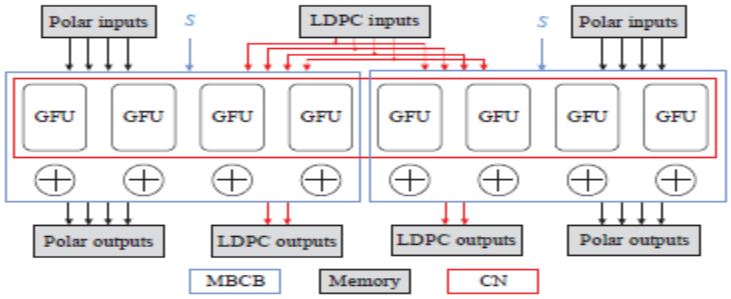

Fig.10: Architecture for a pair of MBCBs.

For polar secret writing, the MBCB's perform is that the same to the BCB's. For LDPC secret writing, a pair of $L(r_{ji})$s are calculated by 2 CUs. As a result, eight GFUs in an exceedingly try of MBCBs are reused to hold out a CN's computation. Fig. ten illustrates the corresponding design for a try of MBCBs. GFUs and adders are connected per Fig. 9. Besides, adders in MBCBs are reused to hold out VNs' computation. Fig. 11 illustrates the design for the projected reconfigurable BP decoder. it's created from forty eight GFUs and 48 adders, and its connection modes are shown in previous figures. Blue squares representing three stages of MBCBs are connected to implement polar secret writing. In an exceedingly try of MBCBs, 8 GFUs and 6 adders are reused to create up a CN and a pair of VNs, severally. For LDPC secret writing, half-dozen system and twelve VNs are connected according to the issue graph. The input S determines that connection mode to be switched. If each secret writing processes are enforced severally individual basis} by a polar decoder and an LDPC decoder, 96 CMPs, ninety six XORs and eighty four adders are required in total. As a result, CMPs, XORs and adders are reduced by
50%, 50%, and 43%, severally.

3. HARDWARE IMPLEMENTATION

Complexity Analysis

Generally, the reconfigurable BP decoder is composed of 2N×log2 N CMPs, XORs and adders, respectively. For the LDPC code, we denote the number of '1's in the j_{th} row by a_j ($a_j \geq 3$) and the number of '1's in the i_{th} column by b_i ($b_i \geq 2$). By counting the number of '1's, we obtain the equation:

$$\sum_{j=0}^{m-k-1} a_j = \sum_{i=0}^{m-1} b_i. \qquad (9)$$

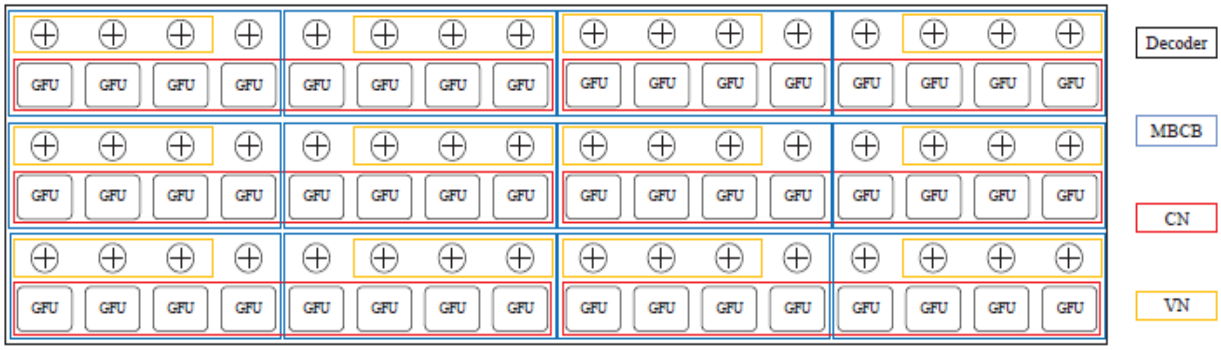

Fig.11: Architecture for the reconfigurable BP decoder.

Since each output in the j_{th} CN is figured out by aj-1 inputs, the CN consists of $a_j(a_j\text{-}2)$ GFUs. In the i_{th} VN, bi-1 adders are needed when adding bi variables, and an extra adder is used for calculating $L(Q_i)$. As a result, the VN consists of $b_i(b_i\text{-}1) + 1$ adders. Totally, $\sum_{j=0}^{m-k-1} a_j(a_j - 2)$ GFUs and $\sum_{i=0}^{m-1}(b_i(b_i - 1) + 1)$ adders are reused for LDPC decoding. It can be implemented if both numbers are less than or equal to $2N\times\log_2 N$. Table I compares the reconfigurable decoder's hardware complexity and the total of two separated decoders.

Table.1: Comparison of Decoders' Complexity.

Decoder Type	Reconfigurable	The total of Polar and LDPC
CMPs	$2N \times \log_2 N$	$2N \times \log_2 N + \sum_{j=0}^{m-k-1} a_j(a_j - 2)$
XORs	$2N \times \log_2 N$	$2N \times \log_2 N + \sum_{j=0}^{m-k-1} a_j(a_j - 2)$
ADDs	$2N \times \log_2 N$	$2N \times \log_2 N + \sum_{i=0}^{m-1} (b_i(b_i - 1) + 1)$

B. Numerical Results

Fig. 12 compares the bit error rate (BER) performance of the float theme and also the fastened scheme for each coding processes. The fastened theme employs one sign bit, four number bits and a pair of decimal bits. Once BER = 10^{-2}, compared with the float scheme, the fastened theme has 0.2dB and 0.5 dB loss for polar coding and LDPC decoding, severally. Hardware's performance loss is appropriate below this condition. FPGA implementation results are given in Table II.

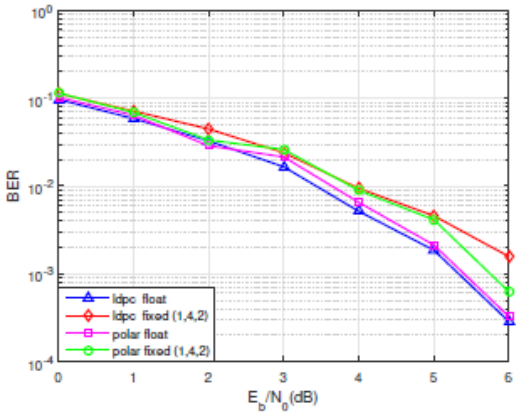

Fig.12: BER performance with N = 8, m = 12, k = 6.

Table.2: FPGA Implementation Results of the Decoder

Items	This Work
Registers	384
LUTS	3223
Max Freq. (MHz)	120.5

Decoding Process	LDPC	Polar
Iteration Number	10	10
Code Length	12	8
Code Rate	0.5	0.5
Throughput (Mbps)	144.6	15.3
Latency (clocks)	10	63

4. CONCLUSION

In this paper, we tend to propose a reconfigurable BP decoder, which can flexibly decrypt polar code or LDPC code. Theoretical analysis shows that the decoder needs lower hardware quality, whereas numerical results indicate negligible performance loss. FPGA implementation results also are given Future work are going to be directed towards additional improvement of performance and design, and its application in our 5G Cloud Platform.

REFERENCES

1. Y. Hawwar, E. Farag, S. Vanakayala, R. Pauls, X. Yang, S. Subramanian, P. Sadhanala, L. Yang, B. Wang, Z. Li, H. Chen, Z. Lu, D. Clark, T. Fosket, P. Mallela, M. Shelton,D . Laurens, T. Salaun, L. Gougeon, N. Aubourg, H. Morvan,N. L. Henaff, G. Prat, F. Charles, C. Creach, Y. Calvez,nd P. Butel, "3G UMTS Wireless System Physical Layer: Baseband Processing Hardware Implementation Perspective," IEEE Communications Magazine, vol. 44, no. 9, pp. 52–58, Sept 2006.
2. Z. Shen, A. Papasakellariou, J. Montojo, D. Gerstenberger, and F. Xu, "Overview of 3GPP LTE-Advanced Carrier Aggregation for 4G Wireless Communications," IEEE Communications Magazine, vol. 50, no. 2, pp. 122–130, February 2012.
3. R. Gallager, "Low-density parity-check codes," IEEE Trans. Inf. Theory, vol. 8, no. 1, pp. 21–28, Jan. 1962.
4. C. Zhang, X. You, and Z. Wang, "Efficient column-layered decoders for single block-row quasi-cyclic LDPC codes," in Proc. IEEE International Symposium on Circuits and Systems (ISCAS), Jun. 2014, pp. 413–416.
5. F. Zarkeshvari and A. H. Banihashemi, "On implementation of minsum algorithm for decoding low-density parity-check (LDPC) codes," in Proc. IEEE Global Telecommunications Conference (GLOBECOM), vol. 2, Nov. 2002, pp. 1349–1353.