A High Throughput List Decoder Architecture ForPolar Code Decoders

Mrs.Sk.Nasreen Begam & G. Mahendra
1 PG Student, Dept. Of VLSI and Embedded systems, SKR College Of Engineering & Technology, AP
2 Associate Professor, Dept. Of VLSI and Embedded systems, SKR College Of Engineering &Technology,AP.

***Abstract*—*This paper presents a high-throughput low-complexity decoder architecture and design technique to implement successive-cancellation (SC) polar decoding. A novel merged processing element with aone's complement scheme, a main frame with optimal internal word length, and optimized feedback part architecture are proposed. Generally, a polar decoder uses a two's complement scheme in merged processingelements, in which a conversion between two'scomplement and sign-magnitude requires an adder. However, the novel merged processing elements do not require an adder. Moreover, in order to reduce hardware complexity, optimized main frame and feedback part approaches are also presented. A (1024, SC polar decoder was designed and implemented using 40-nm CMOS standard cell technology. Synthesis results show that the proposed SC polar decoder can lead to a 13% reduction in hardware complexity and a higher clock speed compared to conventional decoders.*

Index Terms—Polar code, successive-cancellation,decoder, one's complement, high-throughput, low-complexity

1.INTRODUCTION

Polar codes, proposed by Arikan in 2009 [1], have attracted a lot of attention because of their excellent capacity-achieving property over a binary-input discrete memoryless channel (B-DMC). Due to their explicit structure and low-complexity encoding/decoding scheme, polar codes have emerged as one of the most important in coding theory. To date, much of the work has addressed several theoretical aspects of polar codes and is aimed at improving the error correction performance of polar codes of moderate lengths [2-7]. However, few publications have reported implementation of polar decoders. Pamuk [8] reported an FPGA implementation of a polar decoder based on the belief-propagation (BP) algorithm. Although a BP decoder has particular advantages in parallel design, due to the requirement for a large number of processing elements (PEs), the BP decoder is not attractive for practical applications. Several researchers have viewed the successive-cancellation (SC) algorithm as a good candidate for hardware design of polar decoders due to its low complexity [10-13]. The *semi-parallel SC decoder* from Leroux et al. [10] has a very low processing complexity, while memory complexity remains similar to previous architectures, also from Leroux et al. [9]. However, due to the inherent serial nature of the SC algorithm, these SC decoders have significant disadvantages with respect to both long latency and low throughput. Since SC decoding has low intrinsic parallelism, look-ahead techniques [11] were proposed to reduce decoding latency and increase throughput of SC decoders, while using limited extra hardware resources. Yuan and Parhi [13] presented a 2b-SC-precomputation decoder that

reduces decoding latency without performance loss. However, these low-latency architectures do not show detailed implementation results and bit error rate (BER) performance. The *list SC decoder* from A. Balatsoukas-Stimming et al. [14] shows higher decoding performance than SC decoder [11], whereas the list SC decoders has significant disadvantages with respect to both high hardware complexity and low throughput. In this paper, we propose a high-throughput low-complexity architecture for SC polar decoding. A novel *merged processing element (M-PE) with a one's complement scheme, a main frame with optimal internal word length*, and an optimized *feedback part* architectureare proposed. Generally, the polar decoder uses a two's complement scheme in the M-PE, in which a conversion between two's complement and sign-magnitude requires an adder [11]. However, our novel M-PEs do not require an adder. Moreover, in order to reduce hardware complexity, optimized *feedback part* approaches are also presented.

2. RELATED WORK

Hereby there are many encoding and decoding process which are related to the model which has been derived at recent times. Polar codes are a significant breakthrough in a coding theory, since they can achieve the channel capacity of binary-input symmetric memoryless channels and arbitrary discrete memoryless channels. Polar codes of block length *N* can be efficiently decoded by a successive-cancellation (SC) algorithm with a complexity of *O(N* log *N)*. While polar codes of a very large block length approach the capacity of underlying channels under theSC algorithm, for short or moderate polar codes, the error performance of the SC algorithm is worse than turbo or low-density parity-check codes. Lots of efforts have already been devoted to the improvement of error performance of polar codes with short or moderate lengths. An SC list (SCL) decoding algorithm performs better than the SC algorithm. In the cyclic redundancy check (CRC) is used to pick the output codeword from *L* candidates, where *L* is the list size. The CRC-aided SCL (CA-SCL) decoding algorithm performs much better than the SCL decoding algorithm at the expense of negligible loss in a code rate. Despite its significantly improved error performance, the hardware implementations of SC-based list decoders still suffer from long decoding latency and limited throughput due to the serial decoding schedule. In order to reduce the decoding latency of an SC-based list decoder, *M (M* > 1*)* bits are decoded in parallel, where the decoding speed can be improved by *M* times ideally. However, for the hardware implementations of the algorithms in the actual decoding speed improvement is less than *M* times due to extra decoding cycles on finding the *L* most reliable paths among 2*M L* candidates, where *L* is the list size. A software adaptive simplified SC (SSC)-list-CRC decoder was proposed. For a (2048, 1723) polar + CRC-32 code, the SSC-list-CRC decoder with *L* = 32 was shown to be about seven times faster than an SC-based list decoder. However, it is unclear whether the list decoder in is suitable for hardware implementation. RLLD algorithm is proposed to reduce the decoding latency of SC list decoding for polar codes. For a node *v*, let *Iv* denote the total number of leaf nodes that are associated with information bits the $^{u}i\text{-}1$, RLLD algorithm performs the SC-based list decoding on *Gn* and follows the node activation schedule, except when a certain type of nodes is activated. These nodes calculate and return the codewords to their parent nodes while updating the decoding paths and their metrics, without activating their child nodes. the channel message memory (CMEM) stores the received channel LLRs, andthe internal LLR message memory (IMEM) stores the LLRs generated during the SC computation process. With the concatenation and split method, the IMEM is implemented with area efficient memories, such as RF or SRAM.

3. SYSTEM OVERVIEW

The system we proposed is a personalized itinerary format to perform the encoding and decoding process. For the given word read from a memory protected with one step MLD EG-LDPC codes, and affected by up to four bit-flips, all errors can be detected upto twenty decoding cycles errors affecting more than five bits were detected with a probability very close to one. The probability of undetected errors was also found to decrease as the code block length increased. This may be sufficient for some application with very little additional circuitry as the decoding circuitry is also used for error detection. low density parity check encoding can be implemented serially with simple hardware but requires a large decoding time. For memory applications, this increases the memory access time. Whenever a data get corrupted or lost due to transient problem using the high efficient LDPC decode algorithm we can retrieve the original information to be stored into the memory using parity bits.

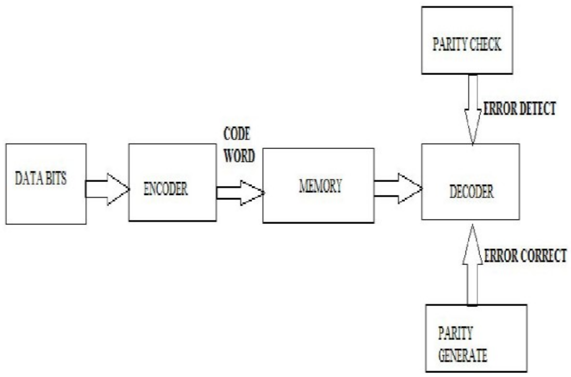

Fig-1: Overall block Diagram

4. LDPC ENCODER

A low-density parity-check (LDPC) code is defined by a parity- check matrix that is sparse. A regular (n,k) LDPC code is defined by an (n- k) × n parity check matrix with n-block length of the code and k information bits generated by the binary source. There are kinds of LDPC codes regular and irregular, irregular performs better than regular but regular codes are easy to implement. Take a special case of LDPC codes as cyclic codes which is used to construct the parity check code and study the behaviour.Construction of parity check matrix is the important part of Encoding process. The block-circulant LDPC code construction is used for creating parity check matrix.

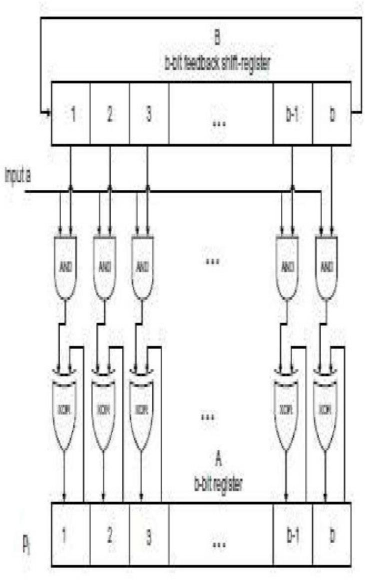

Fig-2: Encoding Process

There are three methods for constructing the generator matrices of QCLDPC codes in systematic-circulant form from their parity-check matrices for two different cases. The first case is that the rank of the parity-check matrix H_{qc}, denoted r, is equal to the number cb of rows of H_{qc}, i.e., r = cb. The second case is that r <cb. encoding process and the expression given by the j^{th} parity check section p_j can be formed with a shift-register-adder-accumulator (SRAA)- circuit. At the beginning of the first step, $g(0)_{1,j} = g_{1,j}$ is stored in the feedback shift register B and the content of register A (accumulator) is set to zero. When the information bit a1 is shifted into the encoder and the channel, the product $a_1g(01_{,j}$ is formed at the outputs of the AND-gates and is added to the content stored in the register A (zero at this time). The sum is then stored in the register A. The feedback register B is shifted once to the right. The new content in B is $g(1)_{1,j}$. When the next information bit a2 is shifted into the encoder, the product $a_2g(1)_{1,j}$ is formed at the outputs of the AND-gates. This product is then added to the sum $a_1g(0)$ 1,j in the accumulator register A. The sum $a_1g(0)_{1,j} + a_2g(1)_{1,j}$ is then stored in A. The above shift-add store process continues. When the information bit ab has been shifted into the encoder, register A stores the partial sum $a_1G_{1,j}$, which is the contribution to the parity section pj from the information section a1. At this time, the generator $g_{2,j}$ of the circulant $G_{2,j}$ is loaded into B. The shift add-store process repeats. When the information section a2 has been completely shifted into the encoder, register A contains the accumulated sum $a_1G_{1,j} + a_2G_{2,j}$, which is the contribution to the parity section pj from the first two information sections, a1 and a2. The above process repeats until the entire information sequence a has been shifted into the encoder. The next stage is to form the c parity sections based on $p_{Tj} = B_jy_T$. This can be done with another c banks of XOR-gates. If the parity-check bits of each parity section are generated serially one bit at a time, simply cyclically shift the buffer registers, $BR_1, . . ., BR_{c,b}$ times (left shift). The parity-check bits are generated in the same manner as the bits of y-vector.

Consider the case for which the rank r of the c×c array D is equal to the rank of H_{qc} given by (1) and r <cb Based on the generator matrix G_{*qc}given by, an encoder with two sub-encoders can be implemented. The first sub-encoder isimplemented based on the submatrixG_{qc} and the second one is implemented based on the submatrix Q. An information sequence a of tb − r bits are divided into two parts, a(1) and a(2), where a(1) consists of the first (t −c)b information bits and a(2) consists of the last cb−r information bits. The first sub encoder

encodes a(1) into a codeword in the subcode generated by G_{qc} and the second sub- encoder encodes a(2) into a codeword in the subcode generated by Q. Adding the outputs of the two sub-encoders, obtain the codeword for the information sequence a. The first sub encoder can be implemented in the same way as described for previous case. The second sub-encoder can be implemented as a conventional encoder for a linear block code.

5.Simulation Results

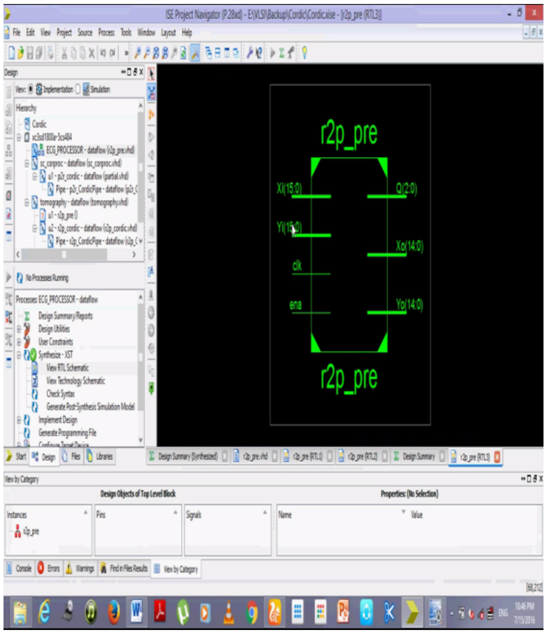
ISE Project Navigator (P.28xd) - E:\VLSI\Backup\Cordic\Cordic.xise - [r2p_pre (RTL3)]
r2p_pre
X(15:0)
Y(15:0)
clk
ena
Q(2:0)
Xo(14:0)
Yo(14:0)
r2p_pre
Design Objects of Top Level Block
Properties: (No Selection)
Instances
Pins
Signals
Name
Value

Fig_5.1

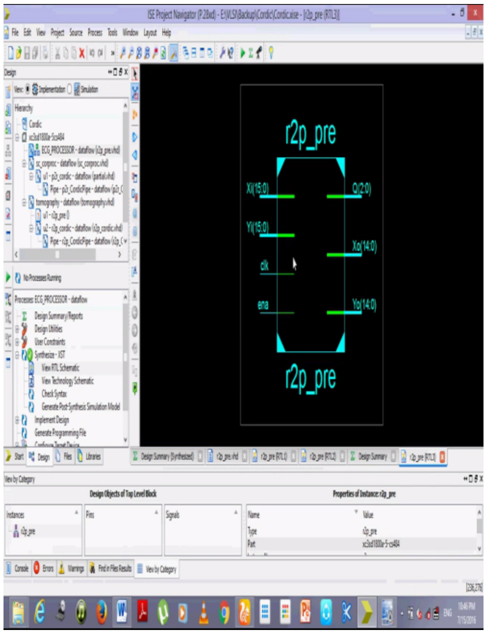
ISE Project Navigator (P.28xd) - E:\VLSI\Backup\Cordic\Cordic.xise - [r2p_pre (RTL3)]
r2p_pre
X(15:0)
Y(15:0)
clk
ena
Q(2:0)
Xo(14:0)
Yo(14:0)
r2p_pre
Design Objects of Top Level Block
Properties of Instance: r2p_pre
Instances
Pins
Signals
Name
Value
Type
Part

Fig_5.2

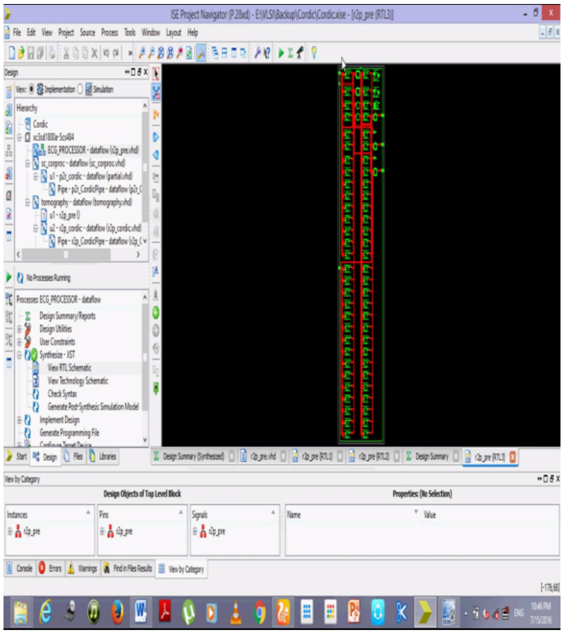
ISE Project Navigator (P.28xd) - E:\VLSI\Backup\Cordic\Cordic.xise - [r2p_pre (RTL3)]
Design Objects of Top Level Block
Properties: (No Selection)
Instances
Pins
Signals
Name
Value

Fig_5.3

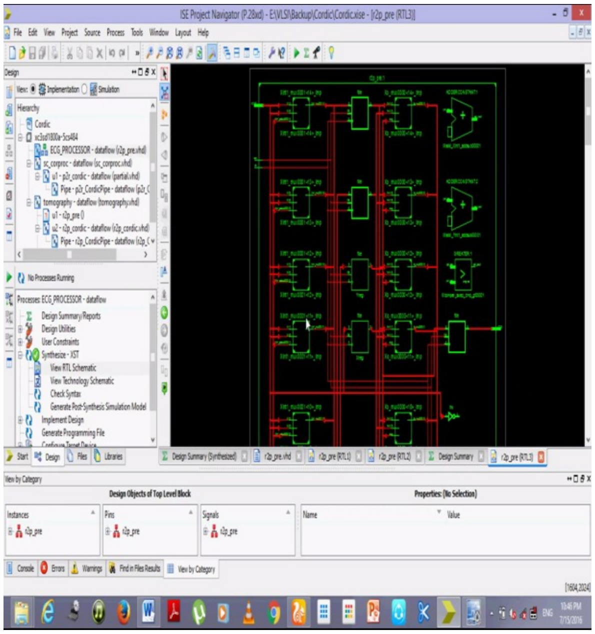

Fig_5.4

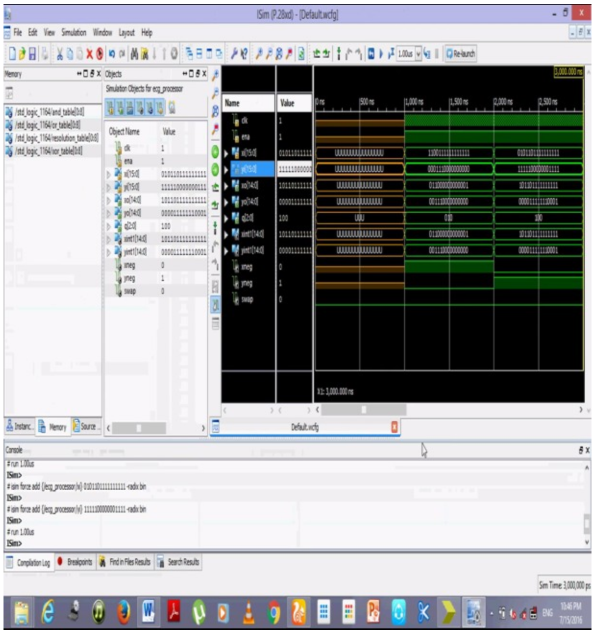

Fig_5.5

6.Conclusion

The iterative decoding approach is already used in turbo codes but the structure of LDPC codes give even better results. In many cases, they allow a higher code rate and also a lower error floor rate. The inherent parallelism in decoding LDPC codes suggests their use in high data rate systems. They provide a performance which is very close to the capacity for a lot of different channels and linear time complex algorithms for decoding. Furthermore, are they suited for implementations that make heavy us of parallelism. Quasi-cyclic (QC) low-density parity-check (LDPC) codes form an important subclass of LDPC codes. These codes have encoding advantage over the other types of LDPC codes. Also, well designed QC-LDPC codes perform as well as computer generated random LDPC codes in terms of bit-error performance, block-error performance, error-floor, and rate of iterative decoding convergence, collectively. Thereby a sustainable amount of throughput can be achieved using LDPC codes when compared with the RLLD algorithm , which can increase the performance of the encoding and decoding process.

REFERENCES

[1]E. Arıkan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” Trans. Inf. Theory, vol. 55, no. 7, pp. 3051–3073, July 2009.

[2]I. Tal and A. Vardy, “List decoding of polar codes,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT 2011), Aug. 2011, pp. 1–5.

[3]C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, “A semi-parallel successive-cancellation decoder for polar codes,” IEEE

Trans. Signal Process., vol. 61, no. 2, pp. 289–299, Jan. 2013.

[4]K. Chen, K. Niu, and J. Lin, "Improved successive cancellation decoding of polar codes," 2013, [Online] Available: http://arxiv.org/abs/1208.3598.

[5]S. Kahraman and M.E. Celebi, "Code based efficient maximum-likelihood decoding of short polar codes," in Proc. IEEE Int. Symp. Inf. Theory (ISIT 2012), Jul. 2012, pp. 1967–1971.

[6]A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive-cancellation decoder for polar codes," IEEE Comm. Letters, vol. 15, no. 12, Dec. 2011, pp. 1378–1380.

[7]G. Sarkis and W. J. Gross, "Increasing the throughput of polar decoders," IEEE Comm. Letters, vol. 17, no. 4, Apr. 2013, pp. 725–728.

[8]C. Zhang and K. K. Parhi, "Low-latency sequential and overlapped architectures for successive cancellation polar decoder," IEEE Trans. Sig. Proc., vol. 61, no. 10, May 2013, 2429–2441.

[9]A. Balatsoukas-Stimming, A. J. Raymond, W. J. Gross, and A. Burg, "Tree search architecture for list successive cancellation decoding of polar codes," submitted to IEEE Trans. Circuits and Systems II, available: http://arxiv.org/abs/1303.7127.

[10] A. Balatsoukas-Stimming, G. Karakonstantis, and A. Burg, "Complexity-performance trade-offs for successive cancellation decoding of polar codes," submitted to IEEE Inf. Theory Workshop, Sept. 2013.

[11] A. Pamuk, "An FPGA implementation architecture for decoding of polar codes," in Proc. 8th Symp. Wireless Commun. Syst. (ISWCS), Nov. 2011, pp. 437–441.

Author's Profile_

Mrs.SK.NASREEN BEGAM received Btech in Electronics and Communication Engineering from SKR College of Engineering, Nellore affiliated to the Jawaharlal Nehru technological university Anatapur in 2011, and pursing M. Tech in VLSI and Embedded systems from SKR College of Engineering affiliated to the Jawaharlal Nehru technological university Anantapur in 2015, respectively.

G.Mahendra Working as Associate Professor Department of ECE.

Qualification: M.Tech
Specialization: VLSI System DesignSKR College of Engineering & Technology
Email ID: Mahendra.goni@gmail.com