An efficient Channel Coding Architecture for 5G Wireless using High-Level Synthesis

K Jaya Sampath[1], N Kiran Kumar[2], K Yeswanth[3], K Snehith[4], Anooja B[5], Bala Tripura Sundari B[6]
Department of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
jayasampath111@gmail.com[1], kirankumarnimakayala@gmail.com[2], ykavuri16@gmail.com[3], kathi.snehith@gmail.com[4],
b_anooja@cb.students.amrita.edu[5], b_bala@cb.amrita.edu[6].

2021 5th International Conference on Trends in Electronics and Informatics (ICOEI) | 978-1-6654-1571-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICOEI51242.2021.9453001

Abstract— **In today's fast paced world, the demand for Mobile Internet is increasing day by day. The fourth generation (4G) systems are now in use world-wide. The present day's 4G LTE has some challenges left such as higher data rates and spectral efficiency due to the tremendous increase in the number of mobile internet users. This led us to a situation of replacing Turbo codes of 4G systems with a channel code that promises higher throughputs. Ever since, the 3GPP had accepted the LDPC codes as a channel coding scheme for 5G wireless communications, a lot of research is going on to optimize the decoder. In 5G, polar codes and LDPC codes are used for error correction for the control channel and data channel respectively. The prime objectives of fifth generation systems are higher data rate, higher spectral efficiency, higher throughput, higher bandwidth, and higher energy efficiency that too at lower latency. There are enormous challenges in the implementation of Channel coding techniques for meeting the requirements of 5G. This highly motivated us to work on the design of efficient LDPC encoder and decoder. Channel coding plays a vital role in any wireless communication system. This work presents a novel efficient high-throughput encoder and decoder for Low Density Parity Check codes for 5th generation wireless Communications. This work proposes strategies to achieve high-throughput Channel coding architecture for LDPC codes using HLS. The proposed design achieves peak throughputs which sufficiently meets the throughput requirement for the 5G NR standard.**

Keywords— 5G, LDPC, Encoding, Decoding, HLS.

I. Introduction

Well, we are all experiencing 4G now. What is 5G? 5G is the fifth generation of wireless network technology and so all the big carriers are working on building out their 5G networks right now all over the world [1]. The main theme of 5G is speed [2]. Every new generation of wireless networks is significantly faster and more capable than the previous generation networks [3]. 5G NR supports enhanced mobile broadband (eMBB), ultra-reliable and low-latency communications (uRLLC) and massive machine-type communication (mMTC). 5G NR is taking a completely different path from LTE in the area of channel coding architecture [4]. For 5G NR Low-Density Parity Check (LDPC) coding is replacing the Turbo coding that was previously used for PDSCH (Physical Downlink Control Channel) coding. LDPC codes and Polar codes are getting considerably more perception in view of their innate benefits of excellent bit error rate execution, quick encoding and decoding process [5]. There are enormous challenges in the implementation of channel coding techniques for meeting the 5G standards which highly motivated us to work on the design of a high throughput decoder for LDPC codes. The 5G promises a throughput of 2GBPS to 200 GBPS in less than 1ms [7]. In this paper, we propose a novel and efficient channel coding scheme for 5G wireless communications implemented using High-Level Synthesis.

High Level Synthesis can synthesize a hardware implementation from a high-level description. The techniques of HLS help in abstracting design to a level higher than RTL. HLS has some constraints that are needed to be taken care of. Data path delay, area utilization and latency. Lower latency leads to better execution speed [8]. Less resource utilization leads to reduced area which in turn leads to reduced power. A design is better if it has lower latency and less resource utilization [13]. HLS techniques are used to abstract the optimal design space exploration.

The objective of this work is to design an efficient High-Level Synthesis framework to utilize in 5G wireless communications. Among the algorithms, the min-sum approximation algorithm promises better results [10]. This work is to design a high-performance Channel coding scheme using LDPC encoding and decoding implemented using High-Level Synthesis. The proposed work exploits the advantages offered by HLS

II. Literature Survey

There are several research works going on the design of high throughput decoder for the 5G standard. Using the local structure for encoding and node processing combined with layered decoding scheduling algorithm for decoding. High throughputs can be achieved using HLS. Serial and layered decoding LDPC scheduling algorithm can be implemented [1]. Smaller amount of hardware resources and power consumption using HSL tool called SDSoC can be designed [2]. The full base matrix can be pruned and only double diagonal sub matrix can be transmitted and used in decoding [3]. The LDPC codes for uRLLC can be generalized using full-precision sum-product and min-sum algorithms [4]. Pipelining while implementing in HLS can be done in multi-threading efficiently [5]. The SNR of the channel also plays an important role in wireless communications. 5G supports various modulation schemes. The choosing of an efficient modulation scheme for a particular weather condition results in lesser BER. Reinforcement learning can be deployed at base station to achieve the same [6]. Partially parallel Low density parity check decoder using min-sum algorithm is presented [7]. WCDMA techniques can be used to achieve maximum power transfer and low energy consumption for wireless systems. [8]. Decoding algorithms are of two kinds. Hard decision and soft decision decoding out of which soft decision decoding presents better results [9]. The Polar codes and LDPC codes are adopted for 5G by 3GPP. The polar codes are used for control channel of 5G wireless [10]. While implementing in HLS, we can use the advantages offered by Vivado HLS like Array Partitioning which helps in optimizing design [11]. HLS has some transformation techniques that help in designing the optimized register transfer level (RTL) code from the user's design [12]. In the field of Wireless Communication the data encoding and decoding schemes are efficient in terms of bandwidth, throughput and latency. That can be made by the efficient use of the available LUTs, FFs, and BRAMs in the FPGA device [13]. FPGA can be used for massively parallel data processing. The analysis of a hardware design can be done using timing summary and hardware resource utilization after implementing it in Xilinx Vivado [14]. The present world has been witnessing a massive 5G new deployments over the past few years [15].

Our goal in this work is to design an efficient LDPC decoding scheme for the 5G wireless. We want to show that this can be achieved by replacing serial hardware design presented in [1] with a parallel and pipelined design. The local structure of the base graphs can be used to completing encoding process in lower complexity [2]. Layered decoding and offset calculations can be deployed for the existing min-sum algorithms [3]. Using HLS transformation techniques for optimizing the hardware design [10]. Among the algorithms available for channel coding, message passing iterative decoder promises better results [11]. This makes us to work on the message passing min-sum iterative decoder, making it more efficient and optimized.

III. 5G NR LDPC Characteristics

As specified above, QC-LDPC codes have been accepted as the channel coding scheme for 5G wireless communications. General structure of the LDPC base graph is depicted in Fig-. The rows are divided into two parts: core check rows and extension check rows [2]. The columns are divided into three categories: information columns, core parity columns and extension parity columns [4]. As shown in the Fig 1, the general structure of a 5G base graph is divided into five different sub categories namely A, D, E, O and I. The submatrix A corresponds to Information bits. D is a square matrix represents the set of parity bits with double diagonal structure [7]. The importance of double diagonal structure is specified in the next chapter. The first column of double diagonal structure is three. Submatrix O is a zero matrix. Submatrix E corresponds to single parity check rows. Submatrix 'I' is an identity matrix which corresponds to the second set of parity bits. The 3GPP considered two different base graphs. Both BG1 and BG2 are rate-compatible. They both have similar structures [15]. Let K be the information bit length in the code word. BG1 supports larger block length i.e. 500 <= K <= 8448. It supports high-rate transmissions i.e. 1/3. While BG2 is most recommended for smaller block lengths (40 <= K <= 2560). It supports lower code rates (1/5 <= R <= 2/3), where R is the code rate of the transmission [16].

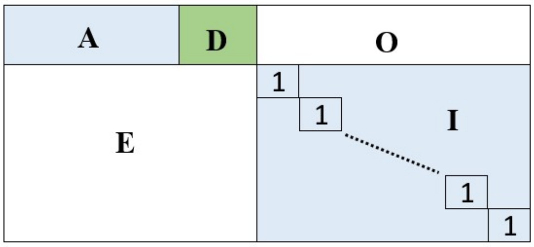

Fig.1 General Structure of 5G Base Graph

IV. Methodology

The LDPC code is a kind of linear block code based on parity check matrix H of size m*n, where m represents the number of check nodes, n represents the number of bit nodes present in the tanner graph of the matrix. The parity check matrix is a sparse matrix which helps in reducing the complexity for computations [10]. There are a few existing methods to construct LDPC codes, such as Progressive Edge-Growth (PEG-LDPC), QC-LDPC and so on [6]. Among them we used, parity check matrix of the QC-LDPC expanded by Z circulants permutation matrix. We devised a new approach to encode LDPC codes using Double Diagonal Structure which is specified clearly in the next section. The parity check matrix has a special structure called Double diagonal structure. The structure is explained in detail in the next section. This new approach executes in lesser complexities than a regular LDPC encoding.

A. Encoding

Encoding LDPC codes in 5G standard is largely different from Hamming Codes. We use parity Check Matrix (PCM) Check Matrix (PCM) instead of Generator matrix for the encoding procedure. Consider a toy-example of encoding using PCM which can be quickly generalized to the encoding of LDPC Codes for 5G communications. Consider an input say, a (6, 3) code with parity check matrix (H) as given below. A message vector say M and the code word C, where p1, p2, p3 are parity bits. These three parity bits are computed using parity check matrix (H) and code word (C) Eqn 1.

$$\text{PCM (H)} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$M = [m1 \quad m2 \quad m3]$$
$$C = [m1 \quad m2 \quad m3 \quad p1 \quad p2 \quad p3]$$
$$H{*}C^T = 0 \qquad \text{-- (1)}$$

Solving the above equation, gives p1 as sum of m1 and m2, p2 as sum of m2 and m3, p3 as the sum of m3 and m1. This is how the encoding of the message bits carried out. Now this parity check matrix is generalized to 5G standard. Encoding to the 5G standard is different from the basic encoding process. As mentioned in the previous chapter, there are two different base matrices of size 46*68 (BG1) and 42*56 (BG2) authorized by 3GPP. Each element of the base matrix is replaced by an identity matrix of size z*z where z is the expansion factor of the transmission. If the expansion factor is 48, the entries of the base matrix are from -1, 0, and 1 to 47. Each entry will expand to a 48*48 identity matrix. Identity matrix is cyclically right shifted by k times where k is the particular entry of the base matrix. If the entry is -1, it is replaced by a zero matrix of size z*z. If the entry is 0, it is replaced by the Identity matrix. If it is greater than 0, it is replaced by an identity matrix that is shifted that many times.

The resultant matrix is the parity check matrix, which is used to find the parity bits in turn leads to determination of code word. The aim of encoding is to generate a code word for the given input bits. Double diagonal structure of the base matrix helps in faster calculation of parity bits for the code word. The double diagonal for a sample H matrix of code (8, 4) is shown in Fig 2. The submatrix D has two diagonals with 'I' as an entry. The double diagonal structure is very interesting and is present in every 5G base matrix. We used this structure to do encoding in lower complexity. This made the encoding ease.

$$H = \begin{bmatrix} I_1 & 0 & I_3 & I_1 & I_2 & I & 0 & 0 \\ I_2 & I & 0 & I_3 & 0 & I & I & 0 \\ 0 & I_4 & I_2 & I & I_1 & 0 & I & I \\ I_4 & I_1 & I & 0 & I_2 & 0 & 0 & I \end{bmatrix}$$

Fig. 2 Double Diagonal Structure

B. Modulation and Transmission

Signal modulation is very essential in wireless communications. Modulation helps signals to travel longer distances. There are different modulation schemes available for wireless communications. The modulation schemes supported by 5G are BPSK, QPSK, 16 QAM, 64 QAM and 256 QAM [16]. For the sake of simplicity BPSK modulation scheme is considered for our proposed design. Binary Phase Shift Keying where binary refers to two phase offsets. If the input digit is 0, it has 0o degree phase shift and 1800 phase shift if it is 1. Hence, if the input binary digit is a then the modulated bit is 2a-1. If 'a' is 1, modulated output is 1. If 'a' is 0, modulated output is -1. The advantages of the BPSK modulation scheme is that they have binary 1 and 0 are separated by 180o phase shift and hence it is the robust modulation. Due to this, the modulated signal can travel longer distances i.e. when travelling from base stations to subscriber stations. When demodulating, the decision must be chosen around only two values. Hence, there is lesser ambiguity while recreating the original binary sequence. Any transmitted signal is high likely to prone to distortion due to noise. So, AWGN channel is considered for this work also monitored the SNR of the transmitted signal. Let the signal power be p. The noise spectral density of an AWGN channel is $^{No}/_{2}$. To calculate the SNR, of the proposed BPSK for this work, consider a simple BPSK signal illustration with only two symbols one at 1 and the other at -1. Energy of the Signal is mean of square of symbols. E_S is 1 and the noise power is σ^2. SNR is the ratio of E_S and E_N. Which is $^{1}/_{\sigma}\,2$. Now that we have SNR, the no of bit errors is calculated using Q function.

$$\text{BER} = Q(1/\sigma) = Q(\text{sqrt(SNR)}) \qquad \text{-- (2)}$$
$$Q(x) = 0.5{*}\text{erfc}(\,^{x}/_{sqrt(2)}) \qquad \text{-- (3)}$$

C. LDPC Decoding

There are various methods of newly discovered algorithms available for decoding of LDPC codes. Efficiency of the LDPC codes majorly depends on decoding. The less the complexity in decoding the more efficient is the approach. Primarily there are two kinds of decoding. They are Hard Decision Decoding and Soft Decision Decoding. Algorithms like sum-product algorithm, belief propagation algorithm, message passing algorithm works on Hard Decision Decoding. However, soft decision decoding is recommended for better results.

The soft decision technique called Min-Sum approximation technique is used for this work. This technique exploits the iterative decomposition of check node and variable node combined processing units which leads to area optimization of lower hardware complexity. The proposed decoding process is shown in Fig-4. An iterative decoder works in multiple rounds of iterations and it actually consists of reasonably simple operations. This decoder of the LDPC is Soft-input, Soft-Output. It deals with the belief propagation

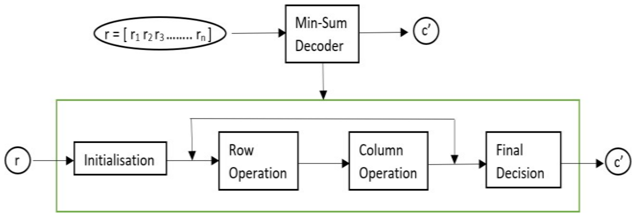

Fig 3 – Decoding Algortihm

The received vector does not have binary values as it is prone to channel noise. Eventually these binary values will all be integer values after suitable scaling and quantitation. LLR comes into picture now. LLR is defined as the log of ratio of probabilities of Ci given a particular bit. For BPSK in AWGN,

$$Li = 2^{*}\mathbf{ri}/\ \sigma * \sigma \qquad \text{-- (4)}$$

Other bits of information are used slowly in an iterative manner and expand the belief, making the belief better and better. Every row of a parity check matrix defines a single parity check code; It gives the parity check constraint that is specified by one subset of the bits of the code word. The estimate is found from the channel itself. The second estimate is formed by the single parity check decoder. Then the min-sum approximation is computed with the log tan hyperbolic function. The extrinsic estimate is computed from min-sum approximation. Likewise, the other estimates are also calculated. Every parity check in the local structure gives more estimates for that particular bit. To do it more efficiently, we used other received values. Using message passing, each code will use the local structure and then pass messages to each other and improve each other in an iterative nice fashion.

If there are two binary random variables XY, modulo 2 addition of Z is X+Y. Then 1-2Z is the product of 1-2X and 1-2Y. This is sort of after BPSK, the XOR becomes multiplication. To take expectations, these binary random variables are assumed to be independent or at least uncorrelated for this proposed, so if the expectation is taken, expected value of binary random variable being equal to 1 is the same as probability that is equal to 1, so that another little quick result is derived. Suppose x, y, z are binary RVs. Consider z is the sum of x and y.

$$(1\text{-}2^{*}z) = (1\text{-}2^{*}x)(1\text{-}2^{*}y) \qquad \text{-- (5)}$$

$$(1\text{-}2^{*}p_z(1) = (1\text{-}2^{*}p_x(1))(1\text{-}2^{*}p_y(1)) \qquad \text{– (6)}$$

$$\tanh(\ I_z/2\ ) = \tanh(\ I_x/2\ )^{*} \tanh(\ I_y/2\ ) \text{ --(7)}$$

$$\mathrm{sgn}(I_z) = \mathrm{sgn}(I_x)^{*}\ \mathrm{sgn}(I_y) \qquad \text{-- (8)}$$

$$\mathrm{abs}(\ \tanh(|\ I_z|/2)) = \mathrm{abs}(\ \tanh(|\ I_x|/2))^{*}\ \mathrm{abs}(\ \tanh(|\ I_y|/2)) \text{ – (9)}$$

To approximate this, there is a useful reason. It is good to have lesser computations and it is close enough to actual the exact completion. So, the min-sum approximation is the best thing to do here. As specified before; the absolute value is dominated by the minimum of these values. So, let us say if the computation of non-linear function has really no loss in practice, so usually we can do a little offset as well, we will implement it in the decoder.

As a part of column operation, the current estimated vector is estimated after every iteration. The entry in the estimated vector is the sum of received entry and the sum of all entries in that particular column. New entry is the difference between estimated entry and the old entry. After a particular number of iterations the estimated code word is passed to the decision device. If the estimated entry is greater than zero, the decision on bit entry is binary 0 and vice versa. It is not necessarily zero all the time; the appropriate threshold is to be chosen based on SNR of the channel.

The working of the proposed LDPC decoder for 5G wireless communications is specified above. We have described how we used the local structure very efficiently to do operations and improve our beliefs and how we combined different messages coming from the local structure iteratively in a very smart way using very extensive information to more and more iterations. So these are the two ideas which give really powerful performance.

D. Layered Decoding

The most important modification is this layered decoding. Before this row and column operations were specified. In layered decoding, we can update the column more aggressively and need not wait for all the rows to finish before doing the column operation. We took a set of rows or a layer of the PCM once when we finished processing the layer, we immediately updated the column. So it turns out that this has faster conversions, lesser iterations. We designed our code in a way which supported layering that is depicted in Fig 15. A small protograph example for the proposed idea is illustrated below.

Layered Decoding works by grouping of the rows of the PCM. Operations are done layer to layer. So this sort of idea : When the column weight of each layer is usually kept as 1 it is possible to increase that also and do some more complicated arithmetic. In the 5G LDPC matrix specifications we think of each layer as 1 block row in the standard in the 5G LDPC codes. For the sake of simplicity, consider a small toy example as shown in Fig 5. Split the PCM into two layers, layer 1and layer 2. The first layer has just a single 1 in each column, the column weight is 1. Similarly, the second layer also has the column weight 1. These two together specify a parity check matrix. Consider a length 7 code for this example. When we actually implement the 5G code, code is much larger, but the algorithm is same.

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

r = [0.2 -0.3 1.2 -0.5 0.8 0.6 -1.1]

When the column weight is assumed to be at most 1, there is no need to update the same bit twice in the same layer. There may be a column with weight 0. But in this particular example it isn't there. The incoming received vector r is r. Easy way to deal with layered decoding is, start with the incoming received vector and start going down the PCM row by row. As we process through each row, we keep updating this received vector. That is the simple of visualizing the decoder. In each Layer some processing occurs and the received vector gets updates. Wherever there are 1's we store something and wherever there are 0's in the PCM we need not store anything, so it is represented by blue solid in the Fig 4. We only use the first layer in the beginning as an initialization step. So the initialization just goes down and every position is initialized with the received vector. In later iterations there are may be already some value stored in L, in that case we did something different. So after we initialize we move to iteration 1 as shown in Fig 5. Now we do row processing. In the first row, mark min 1 and min 2. An absolute value and overall parity is not satisfied, so we replace min 1 with the absolute value of min 2 and then all other positions get replaced by -1, all signs get flipped. The same thing is done in the next row. Then we immediately update r.

Then we move to layer 2 of iteration 1. The same process is repeated and the r is updated as shown in Fig 6. In the next iteration, the first thing we need to do is subtraction, where the incoming belief is subtracted form layer 1. We have an incoming belief and before we start processing layer 1, we have to correct it. We are correcting it because the incoming belief actually had some input from layer 1 itself in the previous iteration. So, when it comes back again, we have to subtract that influence before doing the fresh processing. This is the most important step. If not done correctly, the decoder loses its stability. So the next step of this iteration is min-sum. We have incoming beliefs and layers along with r.

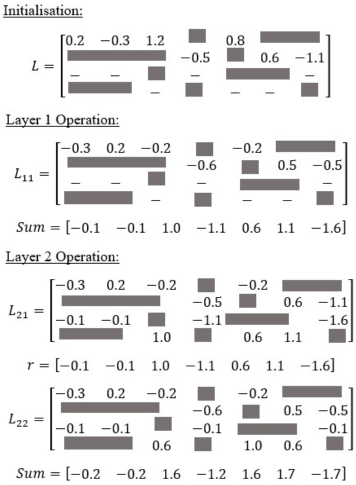
Initialisation:
L =
Layer 1 Operation:
L₁₁ =
Sum = [−0.1 −0.1 1.0 −1.1 0.6 1.1 −1.6]
Layer 2 Operation:
L₂₁ =
r = [−0.1 −0.1 1.0 −1.1 0.6 1.1 −1.6]
L₂₂ =
Sum = [−0.2 −0.2 1.6 −1.2 1.6 1.7 −1.7]

Fig. 4 Layered Decoding Illustration

Min-sum is exactly the same as before, we identify min 1 and min 2 and overall parity is -1. So once you finish processing layer 1, the next step is just updating. Add the row 1 elements with that of elements in sum vector. Repeat the above things for layer 2 as well. Likewise, the loop runs for predefined number of iterations. The r after two iterations is

r = [-0.4 -0.4 1.8 -1.9 1.8 1.9 -1.9]

So, this is the idea where row processing is done first and then the column operation in layer wise fashion. The structure of the storages is very small. To store and read these storages is the most challenging one. We have done it using HLS pragma features. Those features of Vivado HLS made our work easy. To generalize to 5G standard, there is a base matrix which will be expanded to size z*z. This expansion is considered as a single layer of z rows. We did the exact above- mentioned procedure for 5G transmission as well.

E. Introducing Offset

So, it turns out min-sum is pretty useful approximation, now need to improve approximation. Instead of just taking the minimum of two values, we take some constants a and b multiply with that and add b. typically scaling is not preferred. We just used minimum minus some offset. Subtracting with some offset makes it more accurate. Offset min sum approximation is the added modification to the above algorithm. Our implementation used a small offset min-sum with layering in our proposed decoder.

V. IMPLEMENTATION

In the previous section, LDPC encoding and decoding approach illustrates the necessity of high throughput and low latency [11]. Our objective is to design architecture for the 5G channel that promises higher throughputs with lower latency. The proposed architecture reflects the advantages offered by HLS. A partial parallel and full pipelined is used for the proposed decoding technique. This design exploits the advantages of FPGA. Partial parallel means that this design starts updating multiple rows and columns at the same time. Full pipeline refers to the ability of HLS to update a row and a column in a single clock cycle [12]. Vivado HLS offers some features with which we can optimize the design implementation. They are specified in the next section.

VI. TECHNIQUES FOR HIGH THROUGHPUT

A. HLS Pragmas

The hardware kernel must be synthesized from C++, C or OpenCL into RTL which can be embedded into FPGA Device. Vivado HLS offers 'pragmas' which are used to optimize the design improve throughput and reduce latency. Pragmas also helps in reducing the area and resource utilization of the synthesized RTL code. Vivaldi HLS provides various optimization types such as Kernel Optimization, Function In-lining, Interface Synthesis, Loop Optimization etc. We used Interface, Unroll, and Pipeline pragmas for the proposed architecture.

B. Pragma HLS Unroll

Pragma HLS Unroll converts loops by creating multiple copies of data, which allows them to execute multiple loop iterations in parallel. Unroll pragma creates multiple independent operations rather than a single independent operation. We can unroll loops to increase data access and throughput using this pragma.

C. Pragma HLS Pipeline

Pragma HLS Pipeline allows the operations of a loop to be executed in a concurrent manner. Using this pragma, we reduced the initiation interval for a loop and function by allowing the concurrent execution of its operations

D. Pragma HLS Interface

All input and output operations in any RTL design must be performed through a port. The pragma HLS Interface describes how RTL ports are generated from function definition during Interface Synthesis. These ports operate using a specific I/O protocol. Our design uses ports that are derived from Function-Level protocol and Function Arguments.

E. Pipelining

Pipelining is used to split a process into multiple stages and executes the stages in an overlapping fashion. It enables concurrency. Multiple inputs can be processed at the same time. Pipelining increases the throughput of the design. Pipelining is essential for any design which promises lower latency [13]. Pipelining increases the efficiency of resource utilization. Pipeline structure can execute two operations in a single clock cycle.

F. Array Partitioning

When there is a bottleneck in bandwidth Array Partitioning is used. Array partitioning means that a large array is divided into small subarrays in the lane of internal FPGA memory block which is called Block Random Access memory (BRAM). This increases the memory bandwidth. When different resources access a data at the same time, data access conflict occurs. Array Partitioning reduces this conflict.

G. Loop Unrolling

Loop Unrolling reduces the number of loop iterations. In unrolling multiple iterations occur in a single iteration. Loop control overhead is reduced using loop unrolling [14]. Every unrolled loop will be processed in parallel. Complexity caused by control hazards can be resolved using loop unrolling. In the proposed architecture loop unrolling is used in the decoder.
This technique helped us to optimize the execution time of the decoding process. We basically reduced iterations, to increase the decoding speed.

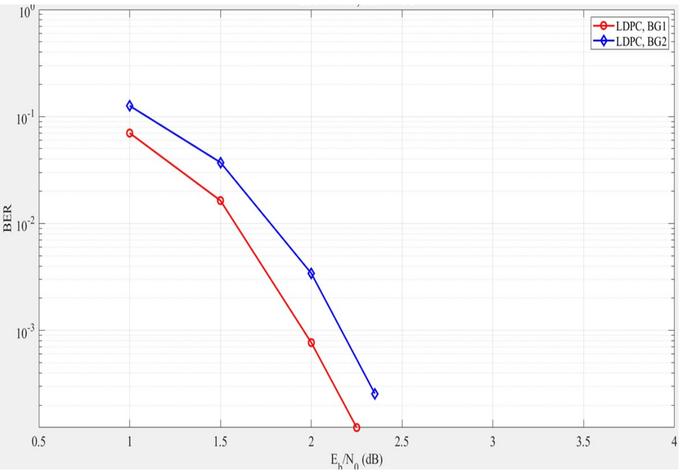

Fig. 7 BER performance of 5G BG1 and BG2.

VII. Results

Fig 5 shows the Bit-Error rate performance of both base graphs BG1 and BG2 with respect to SNR. The size of BG1 matrix is 46*68. The size of BG2 matrix is 42*52. The number of iterations is set to 20. The total number of frames is set to 100, 1000, 5000, 20000 on each SNR. Assumed code rate is ½ and the expansion factor is set to 24. The variations of the design for the above assumed input parameters are shown in Fig 5. We achieved a BER of 10^{-4} where the SNR is greater than 2. So, it shows that the BER performance is absolutely fine.

We have checked our design using 19 different code word sizes for four different code rates (1/2, 2/3, ¾, 5/6), assuming varying input parameters (no of iterations, no of frames or layers, SNR). Synthesis reports are generated for every simulation, waveforms are dumped into Vivado and the waveforms are examined. Table 1 shows a simulation result of the proposed algorithm which is the achieved peak throughput of the design. The assumed input parameters are specified. (N, K, Z) is the code word size, information length and expansion factor respectively. After the simulation the used hardware resources like BRAMs, LUTs and FFs are also tabulated. The resultant throughput and latency is also specified in Table 1.

Attribute	Value
Target Device	xc7k160t fbg484 1
Base Graph	BG1 (46*68)
(N,K,Z)	(576,288,24)
No of Iterations	20
No of Layers	1000
Channel SNR	1.5
BRAMs	104
LUTs	52670
FFs	36640
Latency (Cycles)	4790
Throughput	2.113 GB/s
Latency	19.88 µs

Table 1 Simulation Result.

VIII. CONCLUSION

In this paper, we propose a novel high throughput LDPC encoder and decoder design for the 5G standard. Our novel encoder uses double diagonal structure of base graph to compute parity bits in lower complexity. Our novel decoder uses layered decoding and offset to achieve high throughputs. Based on the proposed algorithm design, obtained results are also specified. The obtained architecture exhibited lower complexity. With optimization techniques such as pipelining, unrolling and array partitioning achieves high throughput in lower latency. The min-sum algorithm using serial and layered implementation achieved a peak throughput of 606 MB/s [1]. We designed a parallel and full pipelined architecture. The proposed architecture demonstrates a superior performance to the existing decoding algorithms. The proposed design achieves a peak throughput of 2.113 GB/s in 19.8 µs. Moreover, the synthesis results of the proposed design satisfy the features offered by 5G wireless communications.

The base station can employ an adaptive modulation scheme for selecting the efficient modulation scheme according to the weather condition [6]. Base Station choosing modulation scheme maximizes the spectral efficiency. The idea is to employ a machine learning algorithm called Reinforcement learning. RL aims to find the best behavior in a particular weather condition in order to maximize the accumulated reward. The future scope of this work is to deploy a RL agent to choose the most efficient modulation scheme.

IX. REFERENCES

[1] Mhaske, Swapnil & Kee, Hojin & Ly, Tai & Aziz, Ahsan & Spasojevic, Predrag.(2017). FPGA-Based Channel Coding Architectures for 5G Wireless Using High-Level Synthesis. International Journal of Reconfigurable Computing. 2017. 1-23.10.1155/2017/3689308.

[2] [2] Choi, Geon & Park, Kyeong-Bin & Chung, Ki. (2018). Optimization of FPGA-based LDPC decoder using high-level synthesis. ICCIP '18: Proceedings of the 4th International Conference on Communication and Information Processing. 256-259. 10.1145/3290420.3290441.

[3] H. Wu and H. Wang, "A High Throughput Implementation of QC-LDPC Codes for 5G NR," in IEEE Access, vol. 7, pp. 185373-185384, 2019, doi: 10.1109/ACCESS.2019.2960839.

[4] Y. Liu, P. M. Olmos and D. G. M. Mitchell, "Generalized LDPC Codes for Ultra Reliable Low Latency Communication in 5G and Beyond," in IEEE Access, vol. 6, pp. 72002-72014, 2018, doi: 10.1109/ACCESS.2018.2880997.

[5] G. Jo, H. Kim, J. Lee and J. Lee, "SOFF: An OpenCL High-Level Synthesis Framework for FPGAs," 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 2020, pp. 295-308, doi: 10.1109/ISCA45697.2020.00034.

[6] Y. Wang, W. Liu and L. Fang, "Adaptive Modulation and Coding Technology in 5G System," 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 2020, pp. 159-164, doi: 10.1109/IWCMC48107.2020.9148457.

[7] A. Verma and R. Shrestha, "A New Partially-Parallel VLSI-Architecture of Quasi-Cyclic LDPC Decoder for 5G New-Radio," 2020 33rd International Conference on VLSI Design and 2020 19th International Conference on Embedded Systems (VLSID), Bangalore, India, 2020, pp. 1-6, doi: 10.1109/VLSID49098.2020.00018

[8] Sivaganesan, D. &quot;IMPROVISATION OF MESH NETWORK WITH WIDEBAND CODE DIVISIONMULTIPLE ACCESS.&quot; Journal: IRO Journal on Sustainable Wireless Systems September 2019, no.03 (2019): 198-205

[9] M. V. Patil, S. Pawar and Z. Saquib, "Coding Techniques for 5G Networks: A Review," 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA), Mumbai, India, 2020, pp. 208-213, doi: 10.1109/CSCITA47329.2020.9137797.

[10] Arora, Komal, Randhawa, Yogeshwar PY - 2020/04/01 SP - 1 EP - 27 T1 - A survey on channel coding techniques for 5G wireless networks VL - 58 DO - 10.1007/s11235-019-00630-3 JO - Telecommunication Systems.

[11] Dr. Bala Tripura Sundari B. and K, R. Krishnanunni, "High Level Synthesis for Design Space Exploration", ARPN Journal of Engineering and Applied Sciences, vol. 11, pp. 1370-1375, 2016.

[12] Krishna Priya PN, Bala Tripura Sundari B , "HLS for retiming Stochastic VLSI Signal Processing Architecture" 8th international Conference on advanced computing, volume 143, 2018.

[13] Dath, Gawtham & Chalil, Anu. (2018). FPGA Implementation of Physical Layer Data Encoding Schemes. 1137-1140. 10.1109/ICICCT.2018.8473041.

[14] Chatterjee, Runa, Rajdeep Chakraborty, and J. K. Mandal. &quot;Design of Cryptographic modelfor End-to-End Encryption in FPGA based systems.&quot; In 2019 3rd International Conference onComputing Methodologies and Communication (ICCMC), pp. 459-465. IEEE, 2019.

[15] Nader Atteya, Sergey Maximov, Mohamed A. El-saidny,"5G NR," Mediatek, 2018.

978-1-6654-1571-2/21/$31.00 ©2021 IEEE