IJIEMR Transactions, online available on 30th Jan 2024. Link :http://www.ijiemr.org/downloads.php?vol=Volume-13&issue=Issue 01

10.48047/IJIEMR/V13/ISSUE 01/08

Title LDPC Encoder Based on Reduced Complexity of XOR Trees

Volume 13, ISSUE 01, Pages: 63-71

Paper Authors **N. MOUNIKA, MR. R. RAMPRAKASH**

International Journal for Innovative Engineering and Management Research

PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL

www.ijiemr.org

LDPC Encoder Based on Reduced Complexity of XOR Trees

1. N. MOUNIKA, MTECH, Department of Electronics and Communication Engineering, Anurag University, Hyderabad.

2. MR. R. RAMPRAKASH, Assistant Professor, Anurag University, Hyderabad.

Abstract: A two-step encoding technique that was described in this article is utilised in this article to encode the 12 quasi-cyclic (QC)- low-density parity-check (LDPC) (QC-LDPC) codes that are required by the IEEE 802.11n/ac/ax standards. This strategy was presented in this article. These codes must be used in order to comply with the requirements of the IEEE 802.11n/ac/ax standards. The strategy that has been suggested addresses the collection as a whole rather than focusing on individually addressing each code that is included inside the collection. In the proposed methodology, the operation of multiplication is carried out by using inverse matrices. The new way of encoding makes the procedures of multiplying and dividing numbers quite a bit simpler and more streamlined. It makes it possible to design completely parallel architectures that can decode in a single clock cycle, or even faster with pipelined implementations, for any of the available encoding formats. This is made possible thanks to the fact that it makes it possible to design completely parallel architectures. These architectures may be created for any of the supported encoding formats. While this is going on, we will propose a VLSI encoding architecture that makes use of trees of XOR gates. CSs may be extracted using the recommended technique by capitalising on the structure and characteristics of the related matrices. This is accomplished via the use of common subexpression sharing mechanisms (CSST). These types of expressions are a direct result of the similarities that exist between the original matrices and their inverses, both of which are covered in further detail in the aforementioned article. In this article, we provide innovative methodologies for the extraction of subexpressions that simultaneously target certain codes. Throughputs of up to 1.62 Tbps are attainable by integrating single-clock hardware encoders manufactured using the method described into technologies operating at 1 GHz and requiring a minimum of 125 or 107 KGates, respectively, respectively. These technologies have a 90-nm and 45-nm technology node size.

Index Terms— *The terms "common subexpression" (CS), "complexity reduction," "encoding complexity," "low-density parity-check" (LDPC) encoding," "matrix inversion"; "multi-Gbps throughput rate;" "quasi-cyclic" (QC) LDPC; and "very large scale integration" (VLSI) architecture are all terms associated with these concepts.*

1. INTRODUCTION

Gallager [1] was the one who pioneered the use of LOW-DENSITY parity-check (LDPC) codes, and ever since their creation, these codes have seen significant improvement in both their structural makeup and their overall performance. The standard for digital video broadcasting (DVB-S2), IEEE 802.16e, IEEE 802.11n/ac/ax, 10Gb Ethernet, magnetic storage, Consultative Committee on Space Data Systems (CCSDS) standard, GB20600 standard, and 5G New Radio (NR) are just a few examples of the many protocols that have gradually adopted LDPC codes due to their excellent error-correcting performance and suitability for highly parallel decoders. Other protocols include the standard for Other protocols that have been implemented more often It has been shown that LDPC codes provide higher error-correction performance when compared to other options; yet, it is still difficult to successfully implement them in hardware. Because of the unpredictable nature of the codes, the hardware blocks have to be coupled in intricate ways, and the codewords have to be rather lengthy. Additionally, one of the most challenging difficulties in real-world applications, particularly in ultra-reliable and high-speed communication systems, is the realisation of cost- and time-effective implementations of numerous LDPC codes with varying properties into a single core. This is one of the most difficult challenges that can be found in real-world applications [2, 3]. Real-world applications are one of the most difficult issues to address, and this is one of the obstacles that contributes to their difficulty.

Multiplying the information bits by the dense generator matrix that is computed based on the sparse parity check matrix is all that is required to employ LDPC encoding (PCM). It is not possible to implement this straightforward approach due to the dense structure of the generator matrix as well as the often long character of the code. If the PCM is substantially lower (or upper) triangular, as shown by Richardson and Urbanke (RU) [4, then the complexity of encoding may be considerably decreased by doing the encoding directly making advantage of the sparse PCM. This is because Richardson and Urbanke established that this is the case. This is due to the fact that Richardson and Urbanke (RU) [4] provided evidence to support the assertion that this is the case. This concept is often used in many of the low-complexity encoder hardware design solutions that have been made available. [5]-[7].

Structured codes, which simplify hardware by imposing limits on the code structures themselves in exchange for better implementation efficiency, have been employed. A quasi-cyclic (QC)-LDPC code is the name given to this specific kind of LDPC code. This particular group of LDPC codes has been approved for usage in almost all of the international standards that use LDPC into their error correcting procedures. The key benefits provided by QC-LDPC codes are linear encoding time and minimal decoding complexity [8, 12]. These are only two of the advantages given by these codes. In a QC-LDPC PCM, each circulant submatrix (SM) has the dimension Z Z and is characterised by a different shifting factor. These kinds of codes use shift registers [13], [15] as their encoders, and corresponding decoder architectures [16] that call for straightforward methods of address creation and restricted memory access.

Li et al. [15] proposed encoding schemes that may be implemented by making advantage of the circular structure of the generator matrix G. Yasotharan and Carusone [18] whereas Andrews et al. [17] suggest an encoding technique for block-circulant codes that is based on the structure of the PCM and the generator matrix, describe a low-complexity encoding architecture that is based on simple multiplication with the generator matrix, and describe an encoding architecture that is based on simple multiplication with the generator matrix. Encoding block-circulant codes is accomplished with the help of this method.

In this research, an approach to the construction of a full-parallel encoder as well as an architecture for said encoder are provided; together, the encoding process takes place during the course of two stages. The design may be capable of calculating the parity bits for each of the 12 QC-LDPC codes that are needed by the IEEE 802.11n/ac/ax standard within the limitations of a single clock cycle. These codes are required by the standard.In addition to this, it is easy to pipeline. It is dependent on components of the implemented hardware that are both shared and reused, and hence the structure of the XOR trees upon which it is formed is determined by these components. Utilizing a number of different methods for recycling sub-expressions is one way to approach the problem of determining the structure of the XOR trees. The examination of matrices in this article shows distinctive features that may be used to make the encoding process less expensive and more straightforward. MCM circuits, which stand for multiple constant multiplication, often take use of the removal or sharing of similar sub-expressions [19–21]. When you multiply a variable by a set of constants, you get what are known as common subexpressions (CS) [22]. These CS are the same as the typical partial products in their corresponding form.

When working with several QC-LDPC codes that have distinct PCMs but are structurally comparable in terms of value and row/column index, it might be helpful to group these codes according to their similarities.We use the concept of subexpression elimination to get rid of redundant expressions. In order to do this, we engage in the practise of subexpression sharing, which is sometimes referred to as vector-matrix multiplications (VMM). This occurs when certain columns or rows in a matrix are the same as sections of columns or rows in either the same matrix or a different matrix.

2. LITERTAURE SURVEY

The decision to use LDPC codes as the data channel coding method for 5G New Radio was made during the 2016 3GPP Conference [1]. Since then, there has been a rise in research on the use of 5G LDPC codes in the real world. One has the potential to increase both the efficiency of encoding and decoding as well as the throughput of the system by splitting the base matrix of the original code rate in [2]. The smaller sub-base matrix is used in place of the whole base matrix. A number of different ideas have been made in order to enhance the performance of LDPC codes in three distinct types of 5G use scenarios ([3,4,5]).

Finding methods to implement LDPC encoding with the least amount of delay possible has always been the primary focus of research into applications of LDPC. If the technique of multiplying the generator matrix G is directly applied in the building of the encoder, then the cost of data storage as well as the computational cost are both quadratic in the code length[6]. In order to efficiently compute the parity bits, this approach includes changing the sparse parity check matrix H into an approximate lower triangular form. There are two RU-based encoders that were built in [7], but the enormous increases in the amount of data storage required and the processing time make it difficult to employ this method. Following that, a quasi-cyclic structure was established via the structural design of the LDPC codes in order to considerably minimise the complexity of the encoding process as well as the requirements for the quantity of data storage space that was required.

Recent research has produced a number of studies that investigate the possibility that QC-LDPC codes are encoded in hardware. The RU approach serves as the foundation for the structural optimization of a number of different encoder designs. This is as a result of the fact that the encoding complexity of the RU approach is much less complicated than that of the direct encoding method. The most revolutionary and ground-breaking invention that has been produced is known as the parallelized encoding structure. According to the information provided in reference number 8, QC-LDPC codes may have something to gain from an approach to parallel LDPC encoding that makes effective use of the space that is now available. By using a bit selection algorithm and a multi-parallel cyclic shift network, this design is able to keep the degree of complexity in the hardware to a minimum, which is one of the primary goals of the design. We demonstrate in reference [9] a QC-LDPC encoding scheme that is appropriate for data rates of many gigabits per second. This architecture takes use of the inherent parallelism of the QC structure in order to process multiple bits in parallel by making effective use of scheduling, and it does this by taking use of the inherent parallelism of the QC structure. A high-efficiency multi-rate encoding for IEEE 802.16e QC-LDPC codes is presented in the publication [10], which can be found here. This encoding makes advantage of the double-diagonal character of the parity matrix to avoid executing the time-consuming inverse matrix operation. This is done so that the matrix may be read in either direction. The encoding rate may be increased by using a structure that incorporates parallel matrix vector multiplication, and the number of storage bits that are needed can be decreased by utilising storage compression. A novel codec that employs a totally parallel QC-LDPC encoder that is based on a reduced complexity XOR tree was proposed in [11] as a means of conforming to the specifications established by IEEE 802.11n. A concept for a QC-LDPC encoder pipeline was given in [12]. The architecture may easily be redesigned via the use of parameters in order to provide a diverse range of possible values for both the code rate and the code length. The matrix vector in [13] is maintained by the encoder via the use of random-access memory, abbreviated as RAM. (RAM). The row index of the non-zero element in each column of the sparse check matrix is used as the write address of the RAM in order to simplify storage and calculation. This is done in order to save space.

For the purpose of the channel coding approach in 5G NR, QC-LDPC codes are used. The 5G standard has established two basis graphs, BG1 and BG2, which map to two base matrices, HBG1 and HBG2, in order to offer interoperability across a broad variety of use cases. These base graphs and base matrices are referred to collectively as HBG1 and HBG2. The HBG matrix is equivalent to a total of 16 parity check matrices (PCM) for the 5G LDPC coding schemes [14], as determined by the lifting sizes of the corresponding 5G QC-LDPC codes. It is necessary for the hardware that supports 5G NR codes to have a high degree of adaptability in order for it to be able to function with a wide range of PCMs.

The acronyms eMBB (Improved Mobile Broadband), URLLC (Ultra-Reliable Low-Latency Communication), and HTM (Huge Machine-to-Machine) refer to the three different application cases for 5G New Radio (mMTC). To provide more clarity, eMBB necessitates a user-plane delay of 4 milliseconds, but URLLC requires just 1 millisecond. According to the investigation done by 3GPP, the LDPC encoding method that was designed for eMBB circumstances is used in URLLC settings (mainly low latency) [15]. This was done since LDPC was developed for eMBB situations. The system performance of 5G NR was measured in [16] by testing a physical downlink shared channel (PDSCH) transmitter prototype on a software-defined radio (SDR), with channel coding tests including the entire processing flow of data transmission in TS38.212. This was done so that the results could be compared to the requirements of 5G NR. Researchers [17,18] have created 5G LDPC encoders that take into consideration the full of the encoding chain for both the uplink and the downlink channels in the wireless communication system. This includes everything from cyclic redundancy check (CRC) encoding to code block segmentation to LDPC encoding to rate matching and bit interleaving. When all of the components necessary for the encoding process have been put together, the finished product may then be sent to the client in its fully operational state. The channel coding activity that takes place at the base station transmitter is the most significant activity to focus on when considering how long it takes to process bits at the physical layer. As a consequence of this, a more sophisticated method of parallel encoding as well as a hardware design for 5G QC-LDPC need to be provided.

Several pieces of scholarly writing have provided references to the hardware architecture of a 5G LDPC encoder. In [19], an efficient LPDC encoding approach combined with an encoding architecture that has both high throughput and low latency was discussed. The synthesis results, which were produced by employing TSMC's 65 nm CMOS technology, made use of a variety of submatrix sizes. Using the method described in [19], the authors of [20] built a flexible and high-throughput 5G LDPC encoder on the CUDA platform. On a single GPU, the encoder was able to reach a throughput of 38-62 Gbps at rates ranging from 1/2 to 8/9. A parallel encoder and pipeline operator are both presented as potential solutions in the research study referred to as [21]. The former is fabricated using CMOS technology with a resolution of 65 nanometers, while the latter claims enhanced parallelism in comparison to [19]. The CMOS technology used to create both of these devices has a resolution of 65 nm. In the article referred to as reference number 22, it is suggested that a genetic algorithm be used in order to make gradual improvements to a QC-LDPC encoder. When working with short codes, it is feasible for many check matrix sub-blocks to be partly processed in parallel. This is made possible by the use of short codes. The level of parallelism is identical to that of the longer code in every respect.

3. METHODOLOGY

Encoder Design Details

The encoding process of 5G QC-LDPC codes involves cyclic shifts and XOR operations, both implemented as straightforward logical processes. As code length increases, handling the complexity of encoder technology becomes more challenging, especially regarding CRC computation. To optimize hardware configurability and resource utilization while maximizing encoding calculation parallelism, the study designed an architecture suitable for adaptive configuration. In the context of 5G NR, where substantial data flow is essential for enhanced Mobile Broadband (eMBB) and low latency and high reliability are critical for Ultra-Reliable Low Latency Communication (URLLC), an encoder must operate within acceptable delay parameters. Therefore, the study focused on increasing the parallelism of CRC and QC-LDPC encoding to its maximum potential.

Traditionally, CRC calculation involves serial computation using Linear Feedback Shift Register (LFSR), which is inefficient for prolonged transmission blocks in 5G. Instead, the study employs a highly parallel hardware architecture utilizing lookup table (LUT) structures and XOR gate circuits. Each byte's CRC is stored in an SRAM-based LUT, partitioning the input bit stream into multiple lookup tables to avoid RAM exhaustion. By pre-calculating CRC values for each data byte, the LUT structure enables fast CRC computation, significantly reducing processing time. This approach, coupled with low-latency CRC calculation, optimizes encoding efficiency, contributing to improved overall coding performance in 5G communication scenarios.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

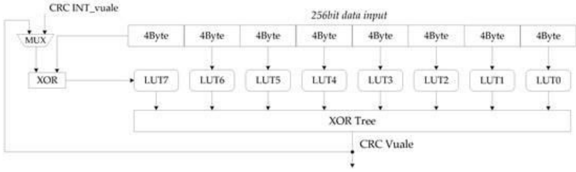


Fig 1 CRC module architecture for 256bits parallel computing.

The encoder architecture designed for 5G QC-LDPC codes features a flexible lookup table (LUT) that can handle 256-bit CRC values divided into 32 bytes across 32 LUTs. If the input data is shorter or longer than 256 bits, the system efficiently handles zero initialization or truncation of high significant bits to ensure proper CRC calculation. Utilizing a parallel CRC computation method, the design retrieves 32 lookup database values per clock cycle, significantly reducing processing time. The architecture also accommodates dynamic code length and rate adjustments, enabling compatibility with all eight permitted code speeds and various code lengths. This flexibility is achieved through a configurable circuit that sets static parameters based on dynamic inputs, ensuring optimal performance and adaptability for diverse 5G communication scenarios. By dynamically adjusting parameters and employing parallel processing techniques, the encoder achieves improved performance in terms of latency and throughput while maintaining hardware complexity within acceptable limits. This comprehensive approach ensures compatibility with all 5G configurations, enhancing the encoder's utility and versatility in next-generation communication systems.

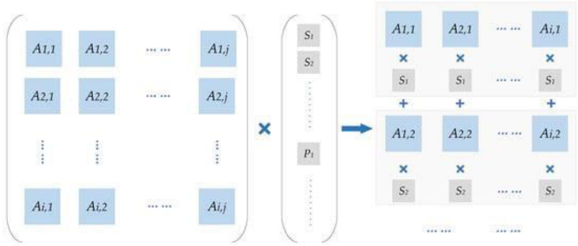


Fig 2 Parallelism improvement of Parities P2 calculation.

Representation of the IEEE 802.11n/ac/ax QC-LDPC Codes

To begin, the circularly shifted identity SMs (I), which are also known as monomials [23], and the zero identity SMs make up the prime factorization group (PCM) of a QC-LDPC code (Z). A clear and straightforward representation of a PCM may be obtained by the use of shift base matrices. These matrices contain either the shifting values of the monomials or the number 1, which is a value that represents a zero submatrix. Shift matrices with a base of Z have an order of mbnb, where mb is defined as equal to m/Z and nb is defined as equal to n/Z. Because of this, the base shift matrices that correspond with H1 and H2 have the orders mbkb and mbmb in their respective H1b and H2b entries. Figure 1 illustrates one example of this kind of thing. Within the IEEE 802.11n/ac/ax code family, there are a total of 12 QC-LDPC codes, each of which is available in one of three unique codeword lengths (648, 1296, 1944). There are three unique code speeds, with Z1 equal to 27, Z2 equal to 54, and Z3 equal to 81, for codes of length 648, 1296, and 1944, respectively, with the same size of permuted and zero SMs. Because the four codes in this family all have the same codeword length and Z, we may divide them into three subfamilies as follows: 1, 648, Z1, 2, 1296, Z2, and 3, 1944, Z3.

$$H_{2_b} = \begin{bmatrix} a & b & -1 & -1 & -1 & -1 \\ -1 & b & b & -1 & -1 & -1 \\ -1 & -1 & b & b & -1 & -1 \\ b & -1 & -1 & b & b & -1 \\ -1 & -1 & -1 & -1 & b & b \\ a & -1 & -1 & -1 & -1 & b \end{bmatrix}$$

Standard components of WiFi QC-LDPC coding structures are called base shift matrices H2 (see Fig. 3). (a) a shift of binary SMs by one place in a cyclical fashion. b is the case for the identity submatrix. A negative entry does not indicate the presence of any SM (1).

The usage of the Almost Lower Triangular (ALT) form for the PCMs is something that is shared by all of the IEEE 802.11n/ac/ax QC-LDPC codes. Additionally, the H2 matrices are stair matrices that include identical SMs along the two diagonals. One other characteristic that is shared by all H2 matrices is the fact that the nonzero Z Z SMs are made up of only two shifting components, denoted by the letters a and b and shown in figure 3.

Full-parallel MVM by H1 and H1 2 multiply the input vector by all columns of the corresponding matrix simultaneously. The above-described procedure was used. The just-described strategy, unlike others based on Forward/Backward Substitution, does not confine the input data. At this stage, you simply need to evaluate qT before computing pT. Multiplying by the nonzero components uses the PCM's sparse structure. Let Nz(A,ri) be a function that, for each row ri in matrix A, gives the indices of the nonzero bits, and let's assume that this function already exists (aces). After that, the steps that were just discussed are written down as

$$q_i^T = \sum_{i=1}^{k} s^T_{\mathrm{Nz}(H_1, r_i)} \pmod 2$$

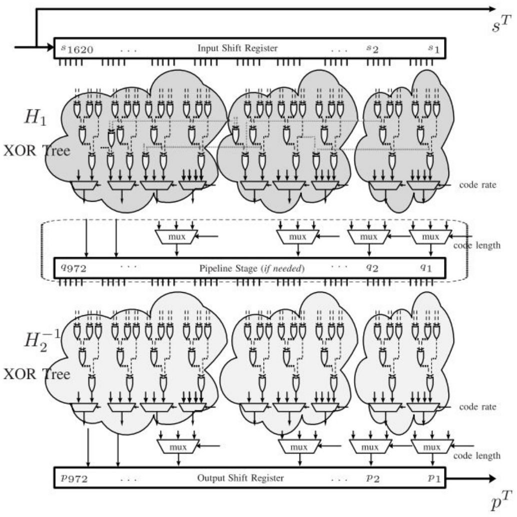
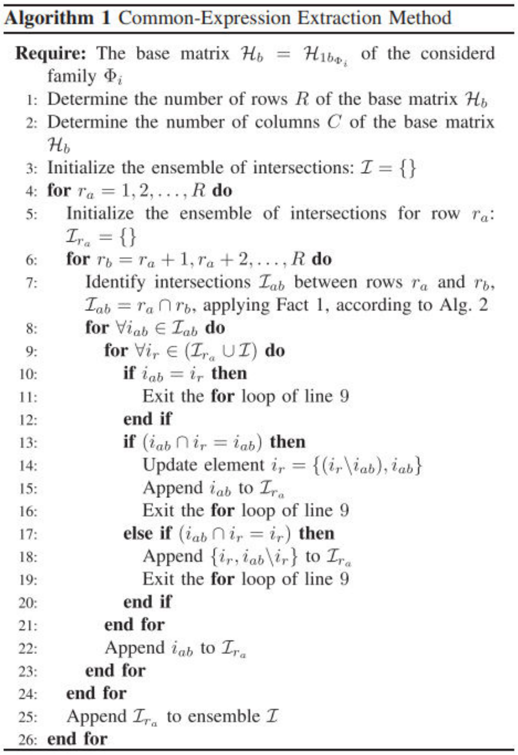


Fig 4 Overview of the full-parallel encoder architecture

$$p_j^T = \sum_{j=1}^{m} q^T_{\mathrm{Nz}\left(H_2^{-1}, r_j\right)} \pmod 2.$$

CS elimination may be used to LDPC encoding, which is a finite-field multiplication that involves a constant matrix and a data vector. This can help reduce the total number of operations that need to be performed. To do this, it is sufficient to concentrate on the set of elements in the relevant matrices that are not zero and to make advantage of the common digit patterns that may be found in those matrices. In the case of the IEEE 802.11n/ac/ax LDPC encoder, there are 12 constant matrices, and CSST may be applied to every one of them. Additionally, it can be applied to each subfamily of WiFi QC-LDCP codes as well as the whole family of these codes. Fig. 4 depicts the proposed architecture; the lengths of the input and output registers are 1620 and 972 bits, respectively, such that they may store the greatest information and parity-bit vectors. Calculations of the intermediate results and parity bits for each of the 12 matrices are performed using XOR trees, which correspond to multiplications by H1 and H 1 2. Specifications regarding code length and code rate are used by the multiplexers at each stage to ensure that only the required results are sent on.



Algorithm 1 Common-Expression Extraction Method

```
Require: The base matrix $\mathcal{H}_b = \mathcal{H}_{1b_{\Phi_i}}$ of the considerd family $\Phi_i$
 1: Determine the number of rows $R$ of the base matrix $\mathcal{H}_b$
 2: Determine the number of columns $C$ of the base matrix $\mathcal{H}_b$
 3: Initialize the ensemble of intersections: $\mathcal{I} = \{\}$
 4: for $r_a = 1, 2, \ldots, R$ do
 5:   Initialize the ensemble of intersections for row $r_a$: $\mathcal{I}_{r_a} = \{\}$
 6:   for $r_b = r_a + 1, r_a + 2, \ldots, R$ do
 7:     Identify intersections $\mathcal{I}_{ab}$ between rows $r_a$ and $r_b$, $\mathcal{I}_{ab} = r_a \cap r_b$, applying Fact 1, according to Alg. 2
 8:     for $\forall i_{ab} \in \mathcal{I}_{ab}$ do
 9:       for $\forall i_r \in (\mathcal{I}_{r_a} \cup \mathcal{I})$ do
10:         if $i_{ab} = i_r$ then
11:           Exit the for loop of line 9
12:         end if
13:         if $(i_{ab} \cap i_r = i_{ab})$ then
14:           Update element $i_r = \{(i_r \backslash i_{ab}), i_{ab}\}$
15:           Append $i_{ab}$ to $\mathcal{I}_{r_a}$
16:           Exit the for loop of line 9
17:         else if $(i_{ab} \cap i_r = i_r)$ then
18:           Append $\{i_r, i_{ab} \backslash i_r\}$ to $\mathcal{I}_{r_a}$
19:           Exit the for loop of line 9
20:         end if
21:       end for
22:       Append $i_{ab}$ to $\mathcal{I}_{r_a}$
23:     end for
24:   end for
25:   Append $\mathcal{I}_{r_a}$ to ensemble $\mathcal{I}$
26: end for
```

It has been shown that the proposed encoder is efficient as a result of its astute use of CSST to generate a low-complexity encoding core. We describe a method for extracting the CS that is based on a few newly found algebraic facts and features of the necessary matrices. This method may be used to extract the CS and build the encoding core. The initial step in the process of encoding involves multiplying the information word sT by a certain matrix called H1. When it comes to the design of electronic circuits, the quantity of two-input XOR gates required is fully determined by the number of aces present in each row, as seen in (4). If you count aces with the same index in rows of a single matrix, you may reduce the number of XOR gates needed. This decrease may be enhanced by considering all H1 rows for each I. By summing I's four H1b matrices, we obtain H1bi. This matrix's rows show all i-related H1 intersections.

Algorithm 2 Identify Intersections Between Two Base-Matrix Rows

```
Require: Two rows $r_a$ and $r_b$ of the base matrix $\mathcal{H}_{1b_{\Phi_i}}$
Require: The number of columns $C$ of the base matrix $\mathcal{H}_b$
Initialize the ensemble of intersection groups $\mathcal{I}_{ab} = \{\}$
Initialize the ensemble of temporary intersection groups $\mathcal{I}_{tmp} = \{\}$
Initialize the ensemble of shifting-factors differences, $\mathcal{I}_d = \{\}$. The $k$-th element of $\mathcal{I}_d$ corresponds to the $k$-th element of $\mathcal{I}_{tmp}$.
Initialize the counter $c_{tmp}$ of the temporary intersections: $c_{tmp} = 0$
for $i = 1, 2, \ldots, C$ do
    if $(\sigma_{(r_a,i)} \neq -1) \wedge (\sigma_{(r_b,i)} \neq -1)$ then
        Compute the difference of shifting factors $d = [\sigma_{(r_a,i)} - \sigma_{(r_b,i)}]_Z$
        if $d \in \mathcal{I}_d$ then
            Find the index $k$ of that element of $\mathcal{I}_d$ that equals to d, $k : \mathcal{I}_d\{k\} = d$
            Evolving Fact 1, column $i$ participates on the intersection described by group of columns $\mathcal{I}_{tmp}\{k\}$, since all any pair of these columns meets (8). Thus, append $i$ to the $k$-th group of $\mathcal{I}_{tmp}$, $\mathcal{I}_{tmp}\{k\} = \{\mathcal{I}_{tmp}\{k\}, i\}$
        end if
        if $d \notin \mathcal{I}_d$ then
            Count in a new temporary intersection group, $c_{tmp} = c_{tmp} + 1$.
            Append the new group, consisted of $i$, to $\mathcal{I}_{tmp}$ $\mathcal{I}_{tmp}\{c_{tmp}\} = i$
            Append $d$ to $\mathcal{I}_d$, $\mathcal{I}_d\{c_{tmp}\} = \{d\}$
        end if
    end if
end for
for $i = 1, 2, \ldots, c_{tmp}$ do
    if $size(\mathcal{I}_{tmp}\{i\}) \geq 2$ then
        Append $\mathcal{I}_{tmp}\{i\}$ to $\mathcal{I}_{ab}$
    end if
end for
```

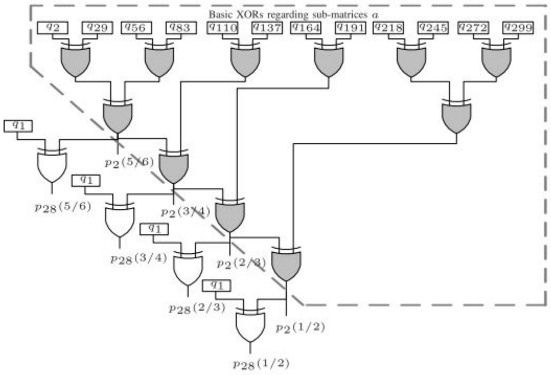
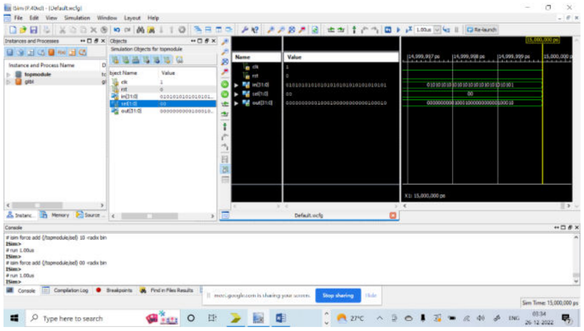


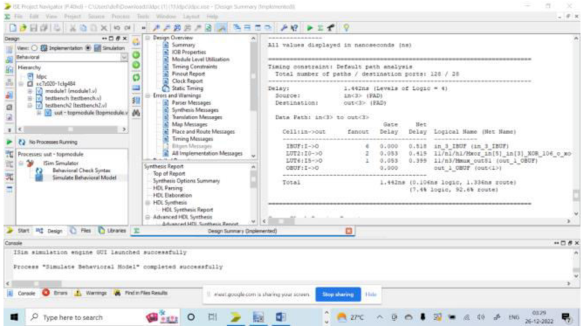
Fig 5 Illustrative utilization of the Basic XORs subtrees to compute the parity bit pT Z+1, where Z = 27.

4. EXPERIMENTAL RESULTS



```
# isim force add {/topmodule/sel} 10 -radix bin
ISim>
# run 1.00us
ISim>
# isim force add {/topmodule/sel} 00 -radix bin
ISim>
# run 1.00us
ISim>
```

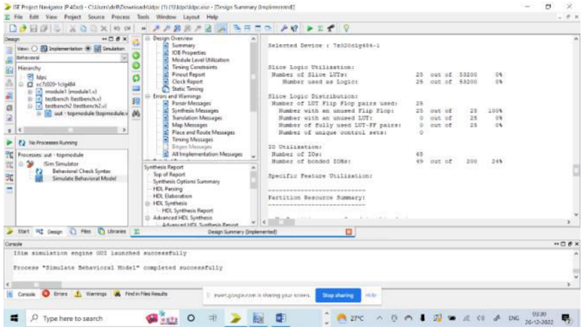
Fig 6 Wave form Results



```
All values displayed in nanoseconds (ns)

================================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 128 / 28
--------------------------------------------------------------------------------
Delay:               1.442ns (Levels of Logic = 4)
  Source:            in<3> (PAD)
  Destination:       out<3> (PAD)

  Data Path: in<3> to out<3>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             6   0.000   0.518  in_3_IBUF (in_3_IBUF)
     LUT2:I0->O            2   0.053   0.415  l1/n1/h1/Mxor_in[5]_in[3]_XOR_106_o_xo
     LUT6:I5->O            1   0.053   0.399  l1/n3/Mmux_out81 (out_1_OBUF)
     OBUF:I->O                 0.000          out_1_OBUF (out<1>)
    ----------------------------------------
    Total                      1.442ns (0.106ns logic, 1.336ns route)
                                       (7.4% logic, 92.6% route)

================================================================================
```

```
ISim simulation engine GUI launched successfully

Process "Simulate Behavioral Model" completed successfully
```

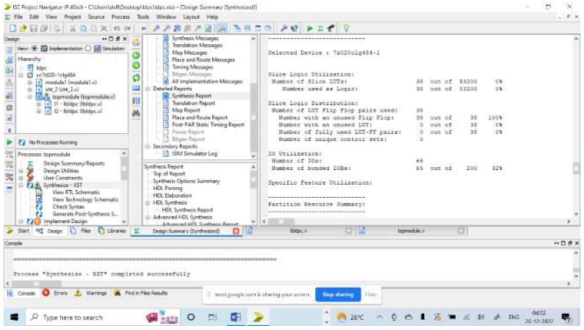
Fig 7 Delay Results of proposed 32bit encoder



```
Selected Device : 7z020clg484-1


Slice Logic Utilization:
 Number of Slice LUTs:                   25  out of  53200     0%
    Number used as Logic:                25  out of  53200     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     25
   Number with an unused Flip Flop:      25  out of     25   100%
   Number with an unused LUT:             0  out of     25     0%
   Number of fully used LUT-FF pairs:     0  out of     25     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                          65
 Number of bonded IOBs:                  49  out of    200    24%

Specific Feature Utilization:

---------------------------
Partition Resource Summary:
---------------------------
```

```
ISim simulation engine GUI launched successfully

Process "Simulate Behavioral Model" completed successfully
```

Fig 8 Area complexity Results of Existing 32bit encoder



```
Selected Device : 7z020clg484-1

Slice Logic Utilization:
 Number of Slice LUTs:                    38  out of  53200     0%
    Number used as Logic:                 38  out of  53200     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:      38
   Number with an unused Flip Flop:       38  out of     38   100%
   Number with an unused LUT:              0  out of     38     0%
   Number of fully used LUT-FF pairs:      0  out of     38     0%
   Number of unique control sets:          0

IO Utilization:
 Number of IOs:                           66
 Number of bonded IOBs:                   65  out of    200    32%

Specific Feature Utilization:

---------------------------
Partition Resource Summary:
---------------------------
```

```
Process "Synthesize - XST" completed successfully
```

Fig 9 Delay Results of Existing 32bit encoder

Table1: Comparison results for both existing and proposed method with respective parameters

32BITLPDCENCODERRESULTS		
Parameter	Existing Method	Proposed Method
Delay	1.762ns	1.442ns
Complexity	25 LUT's	38 LUT's

The aforementioned findings address the project summary report of a 32-bit LDPC encoder with regard to latency and complexity, as shown in table.

5. CONCLUSION

Existing WiFi device designs need to be capable of being upgraded in order to keep pace with the development of IEEE standards such as 802.11n/ac, 802.11ax, and 802.11h. In addition, there is a strong need for consumer electronics that provide a high throughput while while maintaining a small footprint. It has been shown that the proposed Wi-Fi LDPC encoder is capable of satisfying the stringent criteria that are imposed by the most recent versions of Wi-Fi Gigabit Ethernet. The high operating frequency and Gbps processing speed of the encoder that was exhibited make it a potential competitor for the implementation of the 802.11n/ac/ax physical layer over the complete spectrum of channel bandwidths that are enabled by the underlying CMOS technology (at a cost of roughly 100 Kgates). The Common Sub-expression Sharing Approach is the foundation for both the encoding technique and the architecture that is recommended for use when constructing small VMMs. We effectively leverage the algebraic properties of the concerned matrices by following appropriate propositions in order to discover the common patterns and combine them in order to save money on hardware by sharing computations and eliminating the expense of implementing multiple copies of identical expressions. This allows us to save money on hardware by sharing the work that needs to be done. Because of the high density, multiplications by inverted matrices have traditionally been avoided. However, it has been shown in this article that dense multiplications by such matrices can be converted into low-complexity multiplications if the structure of the matrices is taken into account in a strategic manner. This is a significant advance from previous research. It has been established that the encoder has a relatively low level of complexity overall. In conclusion, the suggested approach may be used with a variety of encoding methods. This is possible due to the fact that various QC-LDPC codes all utilise VMM to encode messages. To apply it to different families of QC-LDPC codes, all that is required is a few simple tweaks to accommodate for the particular properties of the matrices that are in issue.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," IRE Trans. Inf. Theory, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[2] R. El Hattachi and J. Erfanian, Eds., "5G white paper, v1.0," NGMN Alliance, Feb. 2015. Accessed: Sep. 7, 2020. [Online]. Available: https://www.ngmn.org/wp-content/uploads/NGMN_5G_White_Paper_V1_0.pdf

[3] IMT Vision–Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond, document ITU-R Rec. M.2083-0, Sep. 2015.

[4] T. J. Richardson and R. L. Urbanke, "Efficient encoding of lowdensity parity-check codes," IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 638–656, Feb. 2001.

[5] E. Eleftheriou and S. Olcer, "Low-density parity-check codes for digital subscriber lines," in Proc. IEEE Int. Conf. Commun. Conf., vol. 3, Apr./May 2002, pp. 1752–1757.

[6] T. Zhang and K. Parhi, "Joint (3,k)-regular LDPC code and decoder/encoder design," IEEE Trans. Signal Process., vol. 52, no. 4, pp. 1065–1079, Apr. 2004.

[7] S. Kopparthi and D. Gruenbacher, "Implementation of a flexible encoder for structured

low-density parity-check codes," in Proc. IEEE Pacific Rim Conf. Commun., Comput. Signal Process. (PacRim), Aug. 2007, pp. 438–441.

[8] I. Tsatsaragkos and V. Paliouras, "A reconfigurable LDPC decoder optimized for 802.11n/ac applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 1, pp. 182–195, Jan. 2018.

[9] A. Verma and R. Shrestha, "A new partially-parallel VLSI-architecture of quasi-cyclic LDPC decoder for 5G new-radio," in Proc. 33rd Int. Conf. VLSI Design 19th Int. Conf. Embedded Syst. (VLSID), Jan. 2020, pp. 1–6.

[10] O. Boncalo and A. Amaricai, "Ultra high throughput unrolled layered architecture for QC-LDPC decoders," in Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI), Jul. 2017, pp. 225–230.

[11] T. T. B. Nguyen and H. Lee, "Low-complexity multi-mode multi-way split-row layered LDPC decoder for gigabit wireless communications," Integration, vol. 65, pp. 189–200, Mar. 2019.

[12] M. Li et al., "An area and energy efficient half-row-paralleled layer LDPC decoder for the 802.11AD standard," in Proc. SiPS Proc., Oct. 2013, pp. 112–117.

[13] A. E. Cohen and K. K. Parhi, "A low-complexity hybrid LDPC code encoder for IEEE 802.3an (10GBase-T) Ethernet," IEEE Trans. Signal Process., vol. 57, no. 10, pp. 4085–4094, Oct. 2009.

[14] H. Fujita and K. Sakaniwa, "Some classes of quasi-cyclic LDPC codes: Properties and efficient encoding method," IEICE Trans. Fundam. Electron., Commun. Comput. Sci., vol. 88, no. 12, pp. 3627–3635, Jan. 2005.

[15] Z. Li, L. Chen, L. Zeng, S. Lin, and W. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," IEEE Trans. Commun., vol. 53, no. 11, pp. 71–81, Jan. 2006.

[16] Y. Chen and K. Parhi, "Overlapped message passing for quasi-cyclic low-density parity check codes," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 6, pp. 1106–1113, Jun. 2004.

[17] K. Andrews, S. Dolinar, and J. Thorpe, "Encoders for block-circulant LDPC codes," in Proc. Int. Symp. Inf. Theory (ISIT), Sep. 2005, pp. 2300–2304.

[18] H. Yasotharan and A. C. Carusone, "A flexible hardware encoder for systematic low-density parity-check codes," in Proc. 52nd IEEE Int. Midwest Symp. Circuits Syst., Aug. 2009, pp. 54–57.

[19] R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 43, no. 10, pp. 677–688, Oct. 1996.

[20] I.-C. Park and H.-J. Kang, "Digital filter synthesis based on minimal signed digit representation," in Proc. 38th Conf. Design Autom. (DAC), 2001, pp. 468–473.

[21] C.-Y. Yao, H.-H. Chen, T.-F. Lin, C.-J. Chien, and C.-T. Hsu, "A novel common-subexpression-elimination method for synthesizing fixed-point FIR filters," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 11, pp. 2215–2221, Nov. 2004.

[22] P. Flores, J. Monteiro, and E. Costa, "An exact algorithm for the maximal sharing of partial terms in multiple constant multiplications," in Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD), Nov. 2005, pp. 13–16.

[23] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," IEEE Trans. Inf. Theory, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[24] A. Mahdi and V. Paliouras, "Simplified multi-level quasi-cyclic LDPC codes for low-complexity encoders," in Proc. IEEE Workshop Signal Process. Syst., Oct. 2012, pp. 1–6.

[25] A. Mahdi and V. Paliouras, "A low complexity-high throughput QC-LDPC encoder," IEEE Trans. Signal Process., vol. 62, no. 10, pp. 2696–2708, May 2014.