

# Low Density Parity Check Codes Based on Finite Geometries and Balanced Incomplete Block Design

Saad Bin Qaisar  
Electrical and Computer Engineering Department,  
Michigan State University.  
[saadq@ieee.org](mailto:saadq@ieee.org)

## I. INTRODUCTION

Low Density Parity Check (LDPC) Codes are the class of linear block codes which provide near capacity performance on large collection of data transmission channels while simultaneously feasible for implementable decoders. LDPC codes were first proposed by Gallger in 1967 [1] and were rarely used until their rediscovery by Mackay, Luby and others [9][10][11]. Much research is devoted to characterize LDPC codes and enhancing their performance on different channels.

Tanner formulated a bipartite graph representation of low density codes now known as Tanner graphs [12]. We can associate a graph  $T$  to the LDPC code using basic graph theory. Let  $T = \{(V, \varepsilon)\}$ , with  $V$  being a set of vertices or nodes  $V$  and  $\varepsilon$  is a set of edges  $E$  connecting the vertices. A cycle of a graph  $T$ , sometimes also called a circuit, is a subset of the edge set  $E$  of  $T$  that forms a path such that the first node of the path corresponds to the last. The length of cycle is the number of edges associated with it. The girth of a graph is the length of shortest cycle. A bipartite graph is one in which the nodes can be partitioned into two disjoint classes,  $V_1$  and  $V_2$ . An edge of the graph may connect a node of one class  $V_1$  to a node of the other class  $V_2$ , but there are no edges connecting nodes of the same class [12]. In Tanner graphs, one class of nodes of bipartite graph is associated to bits whereas other class is associated with check equations. Although tanner graphs are a good visual tool for studying the decoding algorithms for LDPC codes, they cannot be regarded as good design tools.

Example: Let  $C$  be a (10,5) linear block code, with column weight 2, and row weight 4 for the associated parity check matrix  $H$ , given as:

$$H = \begin{bmatrix} 1111000000 \\ 1000111000 \\ 0100100110 \\ 0010010101 \\ 0001001011 \end{bmatrix}$$

Then the associated tanner graph is shown in figure 1:

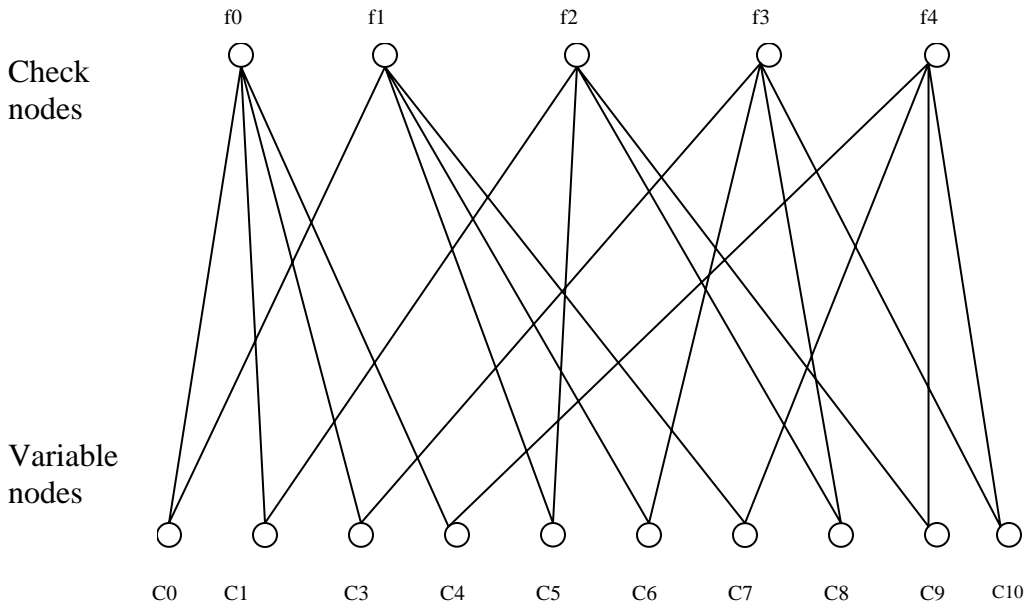


Figure 1: Tanner graph for (10,5) example code

Though LDPC codes are shown to achieve remarkable error correcting capabilities [9], not many algebraic or geometric methods have been found for construction of these codes. Most of the LDPC codes in use today are computer generated. Encoding of these codes is quite complex due to lack of a code structure such as cyclic or quasi cyclic structure. In this paper, we study construction of LDPC codes based on finite geometries and balanced incomplete block design, two methodologies to give structure to LDPC codes. In section II, we study finite geometries and four classes of LDPC codes based on Projective and Euclidean geometries. Section III discusses decoding of LDPC codes with emphasis on Sum Product Algorithm, and section IV briefly touches upon design of LDPC codes based upon Balanced Incomplete Block Design. Before proceeding further, we define few properties of codes. For a code to be quasi-cyclic, a cyclic shift of a codeword of that code by  $p$  positions results in another codeword. Cyclic codes are a subclass of quasi cyclic codes with  $p = 1$ . An LDPC code is regarded as regular if all the columns and rows of parity check matrix of the code have equal number of ones, and irregular, otherwise.

## II. FINITE GEOMETRIES

Let  $G$  be a finite geometry with  $n$  points and  $J$  lines which has the following fundamental structural properties: 1) every line consists of  $\rho$  points; 2) any two points are connected by one and only one line; 3) every point is lies on  $\gamma$  lines and 4) two lines are either parallel or they intersect at only one point [13]. There are two families of finite geometries which have the above fundamental structural properties, namely, Euclidean and Projective geometries over finite fields.

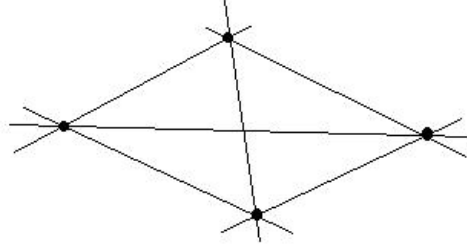


Figure2: A Finite geometry with  $\rho = 2, \gamma = 3$

Figure 2 gives an example of a finite geometry with  $n=4, J=6, \gamma = 3$  and  $\rho = 2$ . A  $J \times n$  incidence matrix  $H_G^{(1)} = h_{i,j}$  over Galois Field  $GF(2)$  can be associated with a finite geometry  $G$ . Each row of the incidence matrix corresponds to a line, and each column correspond to a point.  $h_{i,j} = 1$  if  $j$ th point is on the  $i$ th line in  $G$ , zero otherwise. For  $\rho \ll n$  and  $\gamma \ll J$  the incidence matrix  $H_G^{(1)}$  can be regarded as low density parity check matrix. The null space of  $H_G^{(1)}$  defines LDPC codes.  $H_G^{(1)}$  corresponds to type-I geometry  $G$  LDPC codes with block length  $n$ . The code has a minimum distance of at least  $\gamma + 1$  and is capable of correcting  $\gamma/2$  or fewer errors [14]. If we interchange the rows and columns of incidence matrix, the resulting matrix corresponds to type-II geometry  $G$  LDPC codes given as:  $H_G^{(2)} = H_G^{(1)}$  [14]. The code has a minimum distance of at least  $\rho + 1$ .

### II.1 Euclidean and Projective Geometry LDPC Codes

Euclidean and Projective geometries are two large families of finite geometries. The structure of these families has been well studied and researched.

A geometry in which Euclid's parallel postulate holds is called Euclidean Geometry. The parallel postulate states that if two lines are drawn which intersect a third in such a way that the sum of the inner angles on one side is less than two right angles, then the two lines inevitably must intersect each other on that side if extended far enough. Its sometimes also called parabolic geometry [16].

A projection is the transformation of points and lines in one plane onto another plane by connecting corresponding points on the two planes with parallel lines. This can be visualized as shining a (point) light source (located at infinity) through a translucent sheet of paper and making an image of anything drawn on it on a second sheet of paper. The branch of geometry dealing with the properties and invariants of geometric figures under projection is called projective geometry [17].

Lin et al. [14] [15] form four classes of finite geometry LDPC codes based on these two geometries. We summarize here construction and properties of these four classes based on their work.

**a) Type I Euclidean Geometry (EG) LDPC Codes**

Consider  $EG(m, 2^s)$  as an  $m$ -dimensional Euclidean Geometry over the Galois Field  $GF(2^s)$  where  $s$  and  $m$  are two positive integers. Total number of points in this geometry is  $2^{ms}$  with each point being  $m$ -tuple over  $GF(2^s)$ . Origin corresponds to all zero  $m$ -tuple  $\mathbf{0}=(0,0,0,\dots,0,0)$ .  $EG(m, 2^s)$  can be seen as an  $m$ -dimensional vector space over  $GF(2^s)$ . A line in  $EG(m, 2^s)$  is either a one dimensional subspace of  $EG(m, 2^s)$  or a coset of a one dimensional subspace. Thus, a line consists of  $2^s$  points. Hence, total number of lines in  $EG(m, 2^s)$  is

$$2^{(m-1)s} (2^{ms} - 1) / (2^s - 1)$$

Each line has  $2^{(m-1)s} - 1$  lines parallel to it. For a point in  $EG(m, 2^s)$  the number of lines intersecting at this point is:

$$(2^{ms} - 1) / (2^s - 1)$$

Considering  $GF(2^{ms})$  be the extension field of  $GF(2^s)$ . Thus,  $2^{ms}$  elements in  $GF(2^{ms})$  can be regarded as  $2^{ms}$  elements in  $EG(m, 2^s)$ . If  $\alpha$  is the primitive element of  $GF(2^{ms})$ , then  $0 = \alpha^\infty, \alpha^0, \alpha^1, \dots, \alpha^{2^{ms}-2}$  form the  $2^{ms}$  points of  $EG(m, 2^s)$ . Now consider  $H_{EG}^{(1)}(m, s)$  a matrix over  $GF(2)$  whose columns correspond to  $n=2^{ms} - 1$  non origin points in  $EG(m, 2^s)$  and whose rows correspond to incidence vectors of all the lines in  $EG(m, 2^s)$  that do not pass through origin. Then number of rows  $J$  is given by

$$J = (2^{(m-1)s} - 1)(2^{ms} - 1) / (2^s - 1) \quad (1)$$

$H_{EG}^{(1)}(m, s)$  has following structural properties: 1) Each row has weight  $\rho = 2^s$ ; 2) Each column has weight  $\gamma = (2^{ms} - 1) / (2^s - 1) - 1$ ; 3) Any two columns have at most one "1-component" in common 4) Any two rows have at most one "1-component" in common. The density of the matrix is

$$\gamma = 2^s / (2^{ms} - 1) \quad (2)$$

which is small for  $m \geq 2, s \geq 2$  implying  $H_{EG}^{(1)}(m, s)$  to be a low density matrix. If  $C_{EG}^{(1)}(m, s)$  is the null space of  $H_{EG}^{(1)}(m, s)$ , then  $C_{EG}^{(1)}(m, s)$  is the regular LDPC code of length  $n = 2^{ms} - 1$  termed as type-I Euclidean Geometry LDPC code. It is one step majority logic decodable code based on  $EG(m, 2^s)$  and is the dual code of polynomial code [14]. Thus, it is cyclic with its generator polynomial completely characterized by its roots in  $GF(2^{ms})$ . Tanner graphs of these codes do not contain any 4-cycles, though there are many cycles of length 6 [14].

### ***b) Type II Euclidean Geometry LDPC Codes***

Let us have  $H_{EG}^{(2)}(m, s) = (H_{EG}^{(1)}(m, s))^T$ , then  $H_{EG}^{(2)}(m, s)$  has both values of  $J$  and  $n$ , and  $\rho$  and  $\gamma$  interchanged as compared to  $H_{EG}^{(1)}(m, s)$ . Also, any two rows have exactly “1-component” common, and any two columns have exactly “1-component” common. Then null space of  $H_{EG}^{(2)}(m, s)$  gives an LDPC code  $C_{EG}^{(2)}(m, s)$ . Since Tanner graphs for  $C_{EG}^{(2)}(m, s)$  and  $C_{EG}^{(1)}(m, s)$  are dual, both the codes have identical cycle distribution. Also,  $C_{EG}^{(2)}(m, s)$  is not cyclic, but it can be put into quasi cyclic form [14].

### ***c) Type I Projective Geometry LDPC Codes***

Let us assume  $GF(2^{(m+1)s})$  to be extension field of  $GF(2^s)$  and  $\alpha$  be a primitive element of  $GF(2^{(m+1)s})$ . Taking  $n = (2^{(m+1)s} - 1)/(2^s - 1)$  and  $\beta = \alpha^n$ , implying order of  $\beta$  to be  $2^s - 1$ . The  $GF(2^s)$  is formed by  $2^s$  elements  $0, 1, \beta, \dots, \beta^{2^s-2}$ . Partitioning the non zero elements of  $GF(2^{(m+1)s})$  into  $n$  disjoint subsets as follows:  $\{\alpha^i, \beta\alpha^i, \dots, \beta^{2^s-2}\alpha^i\}$  We define  $\alpha^{(i)}$  as:

$$(\alpha^{(i)})^\Delta = \{\alpha^i, \beta\alpha^i, \dots, \beta^{2^s-2}\alpha^i\} \quad (3)$$

with  $0 \leq i \leq n$ . If each element in  $GF(2^{(m+1)s})$  is represented by an  $(m+1)$ -tuple over  $GF(2^s)$ , then  $(\alpha^{(i)})$  consists of  $2^s - 1$   $(m+1)$ -tuples over  $GF(2^s)$ . The  $(m+1)$  tuple in  $\alpha^i$  represents  $2^s - 1$   $(m+1)$ -tuples in  $(\alpha^{(i)})$ . The  $(m+1)$ -tuple over  $GF(2^s)$  that represents  $\alpha^i$  may be regarded as a point in a finite geometry over  $GF(2^s)$ . The points  $(\alpha^0), (\alpha^1), \dots, (\alpha^{(n-1)})$  thus form a projective geometry  $(m, 2^s)$  over  $GF(2^s)$ . Each line in  $PG(m, 2^s)$  consists of  $2^s + 1$  points and number of lines that intersect a given point is  $(2^{ms} - 1)/(2^s - 1)$  [14]. There are  $J = (2^{ms} + \dots + 2^s + 1)(2^{(m-1)s} + \dots + 2^s + 1)/(2^s + 1)$  lines in  $PG(m, 2^s)$ . Considering  $H_{PG}^{(1)}(m, s)$  a matrix with  $J$  rows, corresponding to lines in  $PG(m, 2^s)$  and  $n$  columns corresponding to points, with columns arranged in ascending powers of  $\alpha$ .  $H_{PG}^{(1)}(m, s)$  has following structural properties: 1) Each row has weight  $2^s + 1$  2) Weight of each column is  $\gamma = (2^{ms} - 1)/(2^s - 1)$  3) Any two columns have exactly one “1-component” common 4) Any two rows have at most one “1-component” in common. The density of  $H_{PG}^{(1)}(m, s)$  is given as  $r = (2^{2s} - 1)/(2^{(m+1)s} - 1)$  which is quite small when  $m \geq 2, s \geq 2$ . If  $C_{PG}^{(1)}(m, s)$  denotes the null space of  $H_{PG}^{(1)}(m, s)$ , then  $H_{PG}^{(1)}(m, s)$  corresponds to type-I  $m$ -dimensional Projective Geometry LDPC code. This code is one step majority logic decodable projective geometry code, and is the dual of non primitive polynomial. It is cyclic, and thus, simpler to encode using linear feedback shift registers.

#### d) Type-II Projective Geometry LDPC Codes

Consider

$$H_{PG}^{(2)}(m, s) = (H_{PG}^{(1)}(m, s))^T \quad (4)$$

$H_{PG}^{(2)}(m, s)$  has row weight  $\rho = (2^{ms} - 1)/(2^s - 1)$  and column weight  $\gamma = 2^s + 1$ . The null space of  $H_{PG}^{(2)}(m, s)$  corresponds to a regular LDPC code  $C_{PG}^{(2)}(m, s)$ , called the type-II m-dimensional Projective Geometry LDPC code.  $C_{PG}^{(2)}(m, s)$  is one step majority logic decodable and can be put into quasi cyclic form using similar techniques as for type-II Euclidean Geometry LDPC codes.

### III. DECODING OF FINITE GEOMETRY LDPC CODES

There exist various algorithms and techniques that can be employed in order to decode finite geometry LDPC codes. Some of these include one-step majority logic decoding (MLG), weighted majority logic decoding, Gallager's bit flipping (BF) decoding, weighted bit flipping (BF) decoding, a posteriori probability decoding (APP) and iterative decoding based upon belief propagation, also termed as Sum Product Algorithm. Finite geometry LDPC codes are known to perform better using SPA decoding. Details of SPA algorithm are described here, though niceties of other decoding algorithms can be found in [14], [18], [19], [1], [2] and [15].

#### III.1 Sum Product Decoding algorithm (SPA)

Before going into details of SPA, let us consider a code  $C$  used for channel coding of binary phase shift keying modulated signals. We have length  $n$  codeword,  $v = (v_0, v_1, \dots, v_{n-1})$  which is mapped into a bipolar sequence  $x = (x_0, x_1, \dots, x_{n-1})$  where  $x_l = (2v_l - 1) = \pm 1$  for  $0 \leq l \leq n-1$ . Considering transmit channel to be AWGN, let soft-decision based received sequence at the output of matched filter be  $y = (y_0, y_1, \dots, y_{n-1})$ . If  $n_l$  is a Gaussian random variable having zero mean and variance  $N_0/2$ , then we have:  $y_l = \pm 1 + n_l$ . Let  $z = (z_0, z_1, \dots, z_{n-1})$  be the binary hard decision received sequence taken as follows:  $z_l = 1$  for  $y_l > 0$  and  $z_l = 0$  for  $y_l \leq 0$ . If  $H$  is the  $J \times n$  parity check matrix for a LDPC code  $C$ , the syndrome of the received sequence is given as:  $s = z.H^T$ . The received vector  $z$  is a codeword if and only if  $s = 0$ , otherwise, an error is detected.

The sum product algorithm operates via a sequence of local computations. These computations follow a single conceptually simple rule, combining the product and summarization operations. The results of these computations are passed as messages along the edges of the tanner graph [20]. The sum product algorithm can be best described by imagining that there is a processor at each node of the underlying graph, and the edges represent channels by which the processors may communicate by sending messages.

Sum product algorithm is a soft in/soft out decoding algorithm providing high level of decoding performance for LDPC codes. The received symbols are processed

iteratively to improve the reliability of each decoded code symbol based on the parity-check sums computed from the hard decisions of the received symbols and the parity-check matrix  $H$  that specifies the code. Marginal posteriori probability, log likelihood ratio (LLR) or the value of corresponding received symbol can be used to find out the reliability of decoded symbol. The reliability measures computed at end of one iteration are fed to the next iteration as input. Once a certain criteria is met, hard decisions are made based upon the computed reliability measures.

Let  $h_1, h_2, \dots, h_j$  be the rows of parity check matrix  $H$  described earlier. We define the support of  $h_j$  as:  $B(h_j) = \{l : h_{j,l} = 1, 0 \leq l < n\}$ , where  $1 \leq j \leq J$ , calculate the marginal posteriori probabilities  $P(v_l | y)$  for  $0 \leq l \leq n$ . The log likelihood ratio for each bit is thus given by:

$$L(v_l) = \log \frac{P(v_l = 1 | y)}{P(v_l = 0 | y)} \quad (5)$$

Consider  $p_l^0 = P(v_l = 0)$  and  $p_l^1 = P(v_l = 1)$ . Let for each  $h_j \in A_l$ , the conditional probability that  $v_l$  has a value  $x$ , given the check sum is calculated based on the check vectors in  $A_l \setminus h_j$  at the  $i$ th decoding iteration be  $q_{j,l}^{x,(i)}$  [14]. Let  $\sigma_{j,l}^{x,(i)}$  be the conditional probability that check sum  $s_j$  is satisfied, given  $v_l = x$  and we have a separable distribution for other code bits in  $B(h_j)$ ,  $\{q_{j,t}^{v_t,(i)} : t \in B(h_j) \setminus l\}$ , i.e.,

$$\sigma_{j,l}^{x,(i)} = \sum_{\{v_t : t \in B(h_j) \setminus l\}} P(s_j | v_l = x, \{v_t : t \in B(h_j) \setminus l\}) \prod_{t \in B(h_j) \setminus l} q_{j,t}^{v_t,(i)} \quad (6)$$

which is used to obtain:

$$q_{j,l}^{x,(i+1)} = \alpha_{j,l}^{(i+1)} p_l^x \prod_{h_j \in A_l \setminus h_j} \sigma_{j,l}^{x,(i)} \quad (7)$$

where  $\alpha_{j,l}^{(i+1)}$  is chosen such that  $q_{j,l}^{0,(i+1)} + q_{j,l}^{1,(i+1)} = 1$ . Therefore, the posteriori probability at  $i$ th iteration is given as:

$$P^{(i)}(v_l = x | y) = \alpha_l^{(i)} p_l^x \prod_{h_j \in A_l} \sigma_{j,l}^{x,(i-1)} \quad (8)$$

where,  $\alpha_l^{(i)}$  is chosen such that  $P^{(i)}(v_l = 0 | y) + P^{(i)}(v_l = 1 | y) = 1$ . Thus, the decoding vector at  $i$ th iteration is given as:

$$z_l^{(i)} = \begin{cases} 1, & \text{for } P^{(i)}(v_l = 1 | y) > 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

which is used to get syndrome  $z^{(i)} H^T$ . If the syndrome is zero, decoding is stopped and  $z^{(i)}$  is the decoded codeword. Otherwise, another iteration is performed. This process is repeated until either syndrome is equal to zero, or maximum number of iterations is reached.

Though sum product algorithm gives very good error performance, a drawback is its decoding complexity. Each iteration requires divisions and multiplications of the order

of  $O(2\gamma J + 4n\gamma)$ , and number of exponentials and logarithm operations for each iteration is of the order of  $O(n)$ [14]. A balance may be achieved depending upon the application, by using two-stage hybrid decoding, which first uses smaller number of SPA iterations, and then, one-step majority logic decoding to reach to the final codeword. A trade off is maintained between computational complexity and error performance depending upon the application specific requirements.

#### IV. BALANCED INCOMPLETE BLOCK DESIGN

In combinatorics, a *design* is a pair  $(V, B)$ , where  $V$  is a  $v$ -set of elements called points, and  $B$  is a collection of  $k$ -subsets of  $V$ , called blocks.  $v = |V|$  is the number of points whereas,  $b = |B|$  is the number of blocks.. If  $t \leq v$  is an integer parameter, such that any  $t$ -subset of points from  $V$  is contained in exactly  $\lambda$  blocks, we deal with a  $t$ -*design*. A Balanced Incomplete Block Design (*BIBD*) is a  $t$ -design for which each block contains the same number of points  $k$ , and every point is contained in the same number of blocks  $r$  [5]. Considering a 2-design, BIBD satisfies following two relations:

$$r(k - 1) = \lambda(v - 1) \quad (10)$$

$$bk = vr \quad (11)$$

Therefore, a  $(v, k, \lambda, r, b)$  BIBD can be represented as  $(v, k, \lambda)$  design by exploiting dependency in (10) and (11). For  $\lambda = 1$ , the  $(v, k, 1)$  BIBD is called *Steiner system*, and with  $k=3$ ,  $(v, 3, 1)$  is called a *Steiner Triple system*. A balanced incomplete block design  $(B, V)$  is resolvable if there exists a partition  $R$  of its set of blocks  $B$  into parallel classes each of which in turn partitions the set  $V$ . The partition  $R$  is called a resolution. The resolvable Steiner triple systems are termed as Kirkman systems [3].

Example: Let  $V$  be a set of seven points  $0, 1, 2, 3, 4, 5, 6$  and  $B = \{B_1, B_2, B_3, B_4, B_5, B_6, B_7\}$  be a collection of blocks with:  $B_1 = \{0, 1, 3\}, B_2 = \{1, 2, 4\}, B_3 = \{2, 3, 5\}, B_4 = \{3, 4, 6\}, B_5 = \{0, 4, 5\}, B_6 = \{1, 5, 6\}, B_7 = \{0, 2, 6\}$

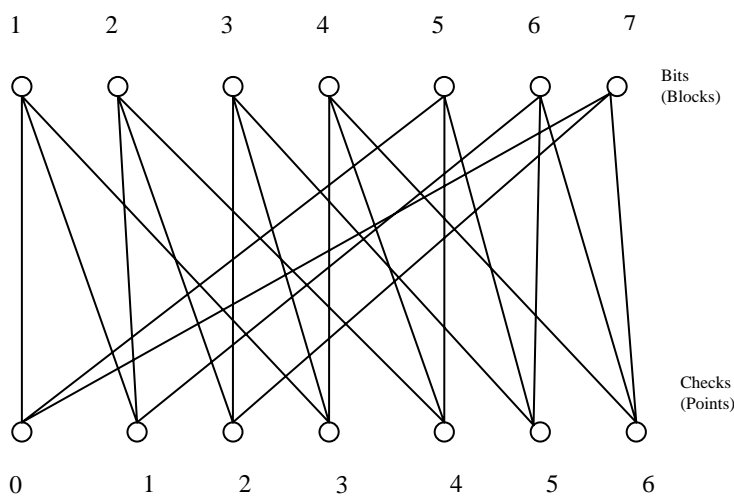


Figure 3: Bipartite graph representation for Kirkman System  $(7, 3, 1)$

The block point incidence matrix for (V,B) is defined as  $C = c_{ij}$  in which  $c_{ij} = 1$  if the i-th element of V occurs in j-th block of B and 0, otherwise. The block point incidence matrix C for the above stated example is given by:

$$C = \begin{bmatrix} 1101000 \\ 0110100 \\ 0011010 \\ 0001101 \\ 1000110 \\ 0100011 \\ 1010001 \end{bmatrix}$$

$C^T$  is the point block incidence matrix. If points are considered parity check equations and blocks as bits, then  $C^T$  can be regarded as parity check matrix H for LDPC codes. The codes based on BIBDs are a super-class of projective and affine geometry codes [5]. They can be decoded using similar methodologies as described earlier.

## V. CONCLUSION

Good structured approaches can be devised for design of low density parity check codes. The works of Lin, Vasic and Fossorier [4],[5],[8] provide interesting insights into architecture of low density parity check codes based upon finite geometries and balanced incomplete block design, providing simplicity in encoding using simple linear feedback shift registers. Combinatorics and graph theory go hand in hand acting as powerful tools for design, analysis and implementation of capacity achieving low density codes.

## VI. REFERENCES

- [1] R.G. Gallager, “*Low-Density Parity Check Codes*. Cambridge,” MA: MIT Press, 1963.
- [2] S.J. Johnson and S.R. Weller, “Construction of low-density parity-check codes from Kirkman triple systems,” in *Proc. IEEE Globecom 2001*, San Antonio, TX, Nov. 2001, vol. 2, pp. 970–974.
- [3] B. Vasic, “Structured iteratively decodable codes based on Steiner systems and their application in magnetic recording,” in *Proc. IEEE Globecom 2001*, San Antonio, TX, Nov. 2001, vol. 5, pp. 2954–2960.
- [4] B. Vasic, E.M. Kurtas, and A.V. Kuznetsov, “Kirkman systems and their application in perpendicular magnetic recording,” *IEEE Trans. Magn.*, vol. 38, no. 4, pp. 1705–1710, July 2002.
- [5] B. Vasic, E.M. Kurtas, and A.V. Kuznetsov, “LDPC codes based on mutually orthogonal Latin rectangles and their application in perpendicular magnetic recording,” *IEEE Trans. Magnetics*, vol. 38, no. 5, pp. 2346–2348, Sept. 2002.
- [6] B. Vasic, “High-rate low-density parity check codes based on anti-Pasch affine geometries,” in *Proc. ICC 2002*, Washington, DC, Apr. 28–May 2 2002, vol. 3, pp. 1332–1336.
- [7] S.J. Johnson and S.R. Weller, “Regular low-density parity-check codes from combinatorial design,” in *Proc. Inf. Tech. Workshop (ITW) 2001*, Cairns, Australia, Sept. 2001.
- [8] Y. Kou, S. Lin, and M.P.C. Fossorier, “Low density parity check codes: Construction based on finite geometries,” in *Proc. IEEE Globecom 2000*, San Francisco, CA, Nov. 2000, vol. 2, pp. 825–829.
- [9] D. MacKay and R. Neal, “Good codes based on very sparse matrices,” in *Cryptography and Coding, 5<sup>th</sup> IMA Conf.*, C. Boyd, Ed., Lecture Notes in Computer Science, pp. 100–111, Berlin, Germany: Springer, 1995.
- [10] D. MacKay, “Good error correcting codes based on very sparse matrices,” *IEEE Transactions on Information Theory*, pp.399–431, March,1999.
- [11] N. Alon and M. Luby, “A linear time erasure-resilient code with nearly optimal recovery,” *IEEE Transactions on Information Theory*, pp. 1732–1736, Nov, 1996.
- [12] R. Michael Tanner, “A recursive approach to low complexity codes,” *IEEE Transactions on Information Theory*, Volume IT-27, September, 1981

[13] José M.F. Moura, Jin Lu and Haotian Zhang, “Structured low density parity-check codes,” *IEEE Signal Processing Magazine*, January, 2004

[14] Yu Kou, Shu Lin, and Marc P. C. Fossorier, “Low-Density Parity-Check Codes Based on Finite Geometries: A Rediscovery and New Results,” *IEEE Transactions on Information Theory*, November, 2001.

[15] Heng Tang, Jun Xu, Shu Lin, and Khaled A. S. Abdel-Ghaffar, “Codes on Finite Geometries,” *IEEE Transactions on Information Theory*, February, 2005

[16] Eric W. Weisstein. "Euclidean Geometry." From [MathWorld](http://mathworld.wolfram.com/EuclideanGeometry.html)--A Wolfram Web Resource. <http://mathworld.wolfram.com/EuclideanGeometry.html>

[17] Eric W. Weisstein. "Projection." From [MathWorld](http://mathworld.wolfram.com/Projection.html)--A Wolfram Web Resource. <http://mathworld.wolfram.com/Projection.html>

[18] J. L. Massey, “*Threshold Decoding*,” Cambridge, MA: MIT Press, 1963.

[19] S. Lin and D. J. Costello, Jr., “Error Control Coding: Fundamentals and Applications,” Englewood Cliffs, NJ: Prentice-Hall, 1983.

[20] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.