VECTOR OPERATION ON NODES OF PERFECT DIFFERENCE NETWORK USING LOGICAL OPERATORS

Rakesh Kumar Katare¹, Shrinivash Premikar¹, Neha Singh¹, Sunil Tiwar¹, Charvi K²,
Department of Computer Science, Awadhesh Pratap Singh University, Rewa (MP)¹
Jawaharlal Nehru College of Technology, Ratahra, Rewa (MP)²

Abstract: In this paper we are exploring the bitwise connection between the nodes of a interconnection network. We are taking PDN as a model, First of all we are converting the interconnection network into its equivalent connectivity matrix .Then Row/Column vectors of connectivity matrix is used to present the value of a particular node of interconnection network which is shown in Figure 1 as state diagram of PDN which is |=2. Each bitwise vector shows connectivity with another node in position of bit 1.The vectors also shows the mathematical property of PDN, it means the value of vector of a node in the connectivity matrix preserves the mathematical property of the topology. We assume that each node is connected to itself as a self loop in connectivity matrix. Therefore the diagonal matrix is always 1. The presence of 1’s in a vector (Excluding the self loop) shows degree of the node .The connectivity and its complexity will be explored by using logical operators between the nodes of a PDN so that we can develop algorithms for automatic switching between two nodes automatically. In the due course of study we found many patterns of binary/logical relationship between the nodes which will be discussed in our future discussion in this paper.

Keywords: PDN, PDS, Interconnection Network, Connectivity Matrix.

1. INTRODUCTION

The Perfect difference set is discussed by J Singer in 1938 [11]. The formulation was in this terms of points and lines in a finite projective plane [1,2,3]. The Perfect Difference Set (PDS) considered for being develop into a interconnect network mainly through works of Parhami, Behroz and Rakov, M.A [4,5]. In their, Perfect Difference Network (PDN) interconnection, they have shown that PDN interconnection scheme is best possible in the sense that it can cover the nodes with smallest node degree with network diameter 2. They have compared PDNs and some of their derivatives to interconnection networks with similar cost and performance with hypercube and its other variants[4]. Perfect difference networks are a robust high-performance interconnection network for parallel and distributed systems. A more exhaustive comparative study of perfect difference network and hypercube was done by Katare et al.,[6,10], based on topological structured properties. Topological properties of perfect difference network compared with the corresponding properties of hypercube by Katare et. al.[10]. In this technique, sparse linear system was implemented. It was proved that access function or routing function to map data on hypercube contains topological properties. The study of circuits based on the architecture of PDN is further taken forward by Katare et.al, july-25, 2013[12] in their research work on study of link utilization of PDN and Hypercube. They have shown that the circuits formed in PDN are a combination of odd and even length. Adjacency matrix of n x n of PDN presented to study the link utilization and topological Properties[12]. In this paper we are converting PDN architecture in to equivalent Data structure for mapping into itself so that transition between nodes can be determined properly. The row vector which is equivalence to column vector can be used for logical operation for determining the binary relationship between nodes. The fabric nature of architecture can be properly defined for Development of algorithm to study the connectivity and Complexity of the architecture[5].

1.1 Perfect Difference Set

A set \( \{s_0,s_1,\ldots,s_\delta\} \) of \( \delta+1 \) integers having the property that their \( \delta^2+\delta \) differences, \( 0\leq i\neq j\leq \delta \), are congruent modulo \( \delta^2+\delta+1 \), to the integers 1,2,\ldots., \( \delta^2+\delta \) in some order is a perfect difference set of order \( \delta \). Perfect Difference Sets[11] are sometimes also called simple difference sets, given that they correspond to the special \( \delta=1 \) as a case of difference sets for which each of the possible differences is formed in exactly \( \delta \) ways, where \( \delta \) is a prime or power of prime and \( n=\delta^2+\delta+1 \) and \( (S_i-S_j)= (\delta^2+\delta) \mod \delta^2+\delta+1 \).

1.2 Perfect Difference Network

The Perfect Difference set of each node of the PDN can be evaluated by the remainder theorem i.e.

\[ (N= R+ D \times Q) \]

Where N= Numerator, R=Remainder, D= Denominator and Q=Quotient

The above equation can be written as

\[ \text{Integer} = (S_i-S_j) + (\delta^2+\delta+1)^1 \]

Where integer is a member of the set \( \{1, 2, \ldots, \delta^2+\delta\} \) and \( S_i \) is numerator or the difference set. So we can write as-

\[ (S_i-S_j) = (\text{integer}) \mod \delta^2+\delta+1 \]

In the due case of study we are assuming that a node is connected to itself therefore the node is self connected in PDN. The following is the connectivity relation between nodes of a PDN.

\[ i \pm 1 \ (0\leq i < \delta^2+\delta) \]

\[ i \pm j \ (i \mod n) \text{ for } 2 \leq j \leq \delta \]
This formulation is based on the definition of the PDS \( \{S_0, S_1, \ldots, S_\delta\} \) There are \( n = \delta^2 + \delta + 1 \) nodes, numbered 0 to \( n-1 \), the direct mapping between nodes is represented by \( i \pm 1 \), which gives cordial ring pattern in network flow \( (i \pm s_j \mod n) \), for \( 2 \leq j \leq \delta \) means for each link from node \( i \) to node \( j \), the reverse link from node \( j \) to node \( i \) is also exists, hence the network can be drawn as an undirected graph.

![Fig.1: PDN having \( \delta=2 \)](image)

The connectivity matrix derived is always a square matrix since number of nodes for both columns and rows are equal. This defined the relation of nodes to itself. Table 1 represents the connectivity matrix for PDN with \( \delta=2 \).

|     | 0   | 1   | 2   | 3   | 4   | 5   | 6   |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0   | 1   | 1   | 0   | 1   | 1   | 0   | 1   |
| 1   | 1   | 1   | 1   | 0   | 1   | 1   | 0   |
| 2   | 0   | 1   | 1   | 1   | 0   | 1   | 1   |
| 3   | 1   | 0   | 1   | 1   | 1   | 0   | 1   |
| 4   | 1   | 1   | 0   | 1   | 1   | 1   | 0   |
| 5   | 0   | 1   | 1   | 0   | 1   | 1   | 1   |
| 6   | 1   | 0   | 1   | 1   | 0   | 1   | 1   |

**Table 1: Connectivity matrix of nodes in a PDN having \( \delta=2 \)**

In this matrix Zero (0) represent that there is no information flow between the nodes and One (1) represent the information flow between the nodes. Such as node 0 can communicate with node 2 via node 1 or node 3 so in Table 1 (0, 2) contains 0. Similarly node 0 can communicate directly with node 1, so in the connectivity matrix in table (0, 1) contains 1.

### 1.3 Connectivity Matrix

Now the structural relation between nodes of a PDN can be connecting method in the following manner, where we are assuming that self loop for each node is considered for interconnection between processor & Peripheral of one node.

#### 1.3.1 Connectivity Matrix

In connectivity matrix if there is a connection between two nodes then its represented as 1 otherwise it is 0. Symbolically connection matrix for PDN we have

\[
CM_{ij} = \begin{cases} 
1 & \text{if node } i \text{ is connected to itself or to another node} \\
0 & \text{otherwise}
\end{cases}
\]

For example the vector of 0 node is \( (1101101) \) so this value shows the connectivity of nodes of a PDN.

- Node 0 is connected with node \( (0,1,3,4,6) \)
- Node 1 is connected with node \( (0,1,2,4,5) \)
- Node 2 is connected with node \( (1,2,3,4,5) \)
- Node 3 is connected with node \( (1,2,3,4,6) \)
- Node 4 is connected with node \( (0,1,3,4,5) \)
- Node 5 is connected with node \( (1,2,4,5,6) \)
- Node 6 is connected with node \( (0,2,3,5,6) \)

Table 1 Shown the connectivity of nodes of a PDN each row of the matrix where “1” shows the connectivity of nodes in “0” shows no connectivity between nodes is a vector used for logical operation for the investigation of the inter node connectivity of the network. Row and Column vectors are same which shows the symmetry of connectivity between processors. Here we are assuming & considering the vector of a connectivity matrix as the value of a node, for example the vector of 0 node is \( (1101101) \) so this value is the value of node zero.

Now we are explaining the connectivity of each node as follows.

| Node Number | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|-------------|---|---|---|---|---|---|---|
| 0           | 1,1,0,1,1,0,1 |   |   |   |   |   |   |
| 1           | 1,1,0,1,1,0   |   |   |   |   |   |   |
| 2           | 0,1,1,1,1,0   |   |   |   |   |   |   |
| 3           | 1,0,1,1,1,0   |   |   |   |   |   |   |
| 4           | 1,1,0,1,1,0   |   |   |   |   |   |   |
| 5           | 0,1,1,0,1,1   |   |   |   |   |   |   |
| 6           | 1,0,1,1,0,1   |   |   |   |   |   |   |

**Table 2: Connectivity between processors in an interconnection network**

Now we are converting connectivity Matrix of PDN into its equivalent state diagram of PDN, interconnection network have been studied researcher for reduce the connectivity & complexity of a set of nodes with particular architecture (PDN). Study of logical operation helps network flow to reaches from node i to any other
nodes in minimum node connectivity. Here we present the
some relation and show how the information flow may
possible in the PDN when, one of node, connection fails.
The logical AND operation between row vector of adjacent
matrix of PDN gives the set of possible path for information
network flow.

Theorem: Vector of node 0 shows the connectivity between
the nodes of PDN say (1101101) 0 Possible connectivity of
this node is with 0, 1, 3,4,6 if the connected node has value
1.
Proof:-The vector representation of each node shows the
node connectivity as per perfect difference in Prefect
Difference Network. For example vector of node 0 is
(1101101) the presence of 1 shows the connectivity and 0
shows the disconnectivity.

\[
\begin{bmatrix}
0,1,2,3,4,5,6 \\
1,1,0,1,0,1
\end{bmatrix}
\]

(0, 1, 3, 4, 6) nodes are connected in PDS \{0, ±1, ±3\}.
The logical operation AND shows the connectivity of two
nodes for example:

Node 0 & Node 1 are connected or in other words
intersection of node 0 & node 1 is \{0, 1, 4\} both the
operation are same as per the assumption of Discrete math's
between Set Theory and mathematical logic.
Similarly other logical operations are OR, EX-OR, implication, Bi-Implication which can also be performed for
finding the binary relation between the nodes of a PDN.

### 2. Explanation of bit representation of Nodes

In this section we are trying to establish the bitwise
logical operation on the combination of vectors for find out
the binary relation between the nodes, so that the binary
relation can be proved for the study of connectivity and
complexity of the flow of information in this architecture.

#### 2.1 Structural Pattern of architecture

| Node | AND | OR | EX-OR | Implication | Bi-Implication |
|------|-----|----|-------|-------------|----------------|
| 0    | 1101101 | 1111111 | 0011011 | 1111111    | 1111111 |
| 1    | 1110110 |           |        |             |               |

The operation “AND” gives the common nodes
between two nodes where as OR gives the one of the nodes
connectivity or both way connectivity. The equivalence
gives both ways connectivity .on the other hand the
implication gives validity of consequent. The Ex-OR gives
the two way switching.
The following tables show the binary logical
relation between the nodes.
| Node | Value | Connected Nodes | Status |
|------|-------|-----------------|--------|
| 0    | 1101101 | 1,3,6           | all nodes are connected |
| 2    | 0111011 | 0111011         | 0,2,4,5 |
| 1001101 | 1111111 | 1101101         | 4 |
| 0    | 1101101 | 0,3,4,6         | 6 |
| 3    | 1011101 | 1011101         | 1,2        |
| 1001101 | 1111111 | 1111111         | 2 |
| 0    | 1101101 | 0,1,3,4         | 6 |
| 4    | 1101110 | 1101110         | 5,6       |
| 1101100 | 1101111 | 1101111         | 2 |
| 0    | 1101101 | 1,4,6           | 7 |
| 5    | 0110111 | 0110111         | 0,2,3,5 |
| 010101 | 0111011 | 0111011         | 4 |
| 0    | 1101101 | 0,3,6           | 7 |
| 6    | 1011011 | 1011011         | 1,2,4,5 |
| 1001001 | 1111111 | 1111111         | 4 |
| 1    | 1110110 | 0,1,2,4,5       | 5 |
| 1    | 1110110 | 1110110         | 0,1,2,4,5 |
| 1110110 | 1110110 | 1110110         | 4 |
| 1    | 1110110 | 1,2,5           | 7 |
| 2    | 0111011 | 0111011         | 0,3,4,6 |
| 0100100 | 1111111 | 1111111         | 4 |
| 1    | 1110110 | 0,2,4           | 7 |
| 3    | 1011101 | 1011101         | 1,3,5,6 |
| 1010100 | 1111111 | 1111111         | 4 |
| 1    | 1110110 | 0,1,4,5         | 6 |
| 4    | 1101110 | 1101110         | 2,3       |
| 1101100 | 1111110 | 1111110         | 2 |
| 1    | 1110110 | 1,2,4,5         | 6 |
| 5    | 0110111 | 0110111         | 0,6       |
| 0110110 | 1110110 | 1110110         | 2 |
| 1    | 1110110 | 0,2,5           | 7 |
| 6    | 1011011 | 1011011         | 1,3,4,6 |
| 1010010 | 1111111 | 1111111         | 4 |
| 2    | 0111011 | 1,2,3,5,6       | 5 |
| 2    | 0111011 | 0111011         | 1,2,3,5,6 |
| 0111011 | 0111011 | 0111011         | 5 |
| 2    | 0111011 | 0111011         | no nodes are connected |
| 2    | 0111011 | 0111011         | 0 |
| 2    | 0111011 | 0111011         | 0,4       |
| 3    | 1011101 | 1011101         | 0,4,5     |
| 0011001 | 1111111 | 1111111         | 4 |
| 2    | 0111011 | 0111011         | 0,2,4,6   |
| 4    | 1101110 | 1101110         | 4 |
| 0101010 | 1111111 | 1111111         | 4 |
| 2    | 0111011 | 1,2,5,6         | 6 |
| 5    | 0110111 | 0110111         | 3,4       |

Rakesh Kumar Katare et al, International Journal of Advanced Research in Computer Science, 10 (6), Nov-Dec 2019, 29-39

© 2015-19, IJARCS All Rights Reserved
| Node | Implication | Possible Connectivity | Patterns of PDN connectivity | Bi-Implication | Possible Connectivity | Patterns of PDN connectivity |
|------|-------------|-----------------------|-----------------------------|----------------|-----------------------|-----------------------------|
| 0    | 110101     | all nodes are connected | 7                           | 1101101       | all nodes are connected | 7                           |
| 0    | 110101     |                        |                              | 1101101       | all nodes are connected | 7                           |
|      | 1111111    |                        |                              | 1111111       | all nodes are connected | 7                           |
| 0    | 110101     |                        |                              | 1101101       | all nodes are connected | 7                           |
| 1    | 111010     | 0,1,2,4,5             | 5                           | 1101011       | 0,1,4,5              | 4                           |
| 0    | 110101     |                        |                              | 1101101       | 0,1,4,5              | 4                           |
| 2    | 0111011    | 1,2,3,5,6             | 5                           | 0111011       | 1,3,6                | 3                           |
| Node | Binary Code | Adjacent Nodes | Distance |
|------|-------------|----------------|----------|
| 0    | 0111011     | 1101101, 1011110 | 2, 3, 4, 5, 6 |
| 3    | 1011110     | 1101111 | 5 |
| 0    | 1101101     | 1011111 | 3 |
| 4    | 1101110     | 0110111 | 6 |
| 0    | 1101101     | 1111100 | 3 |
| 5    | 0110111     | 0100101 | 5 |
| 0    | 1101101     | 1011011 | 5 |
| 6    | 1011011     | 0101101 | 5 |
| 1    | 1110110     | 1110110 | 5 |
| 1    | 1110110     | 1110110 | 5 |
| 1    | 1110111     | 1111111 | 5 |
| 0    | 1101101     | 1110110 | 5 |
| 2    | 0111011     | 0110011 | 5 |
| 1    | 1101011     | 1011011 | 5 |
| 4    | 1011011     | 1010010 | 5 |
| 1    | 1101011     | 1110101 | 5 |
| 3    | 1101011     | 1110110 | 5 |
| 1    | 1101011     | 1110110 | 5 |
| 5    | 0110111     | 0110111 | 5 |
| 1    | 1101011     | 1011011 | 5 |
| 2    | 0111011     | 0111011 | 5 |
| 4    | 1111011     | 1111011 | 5 |
| 3    | 1111011     | 1111011 | 5 |
| 0    | 1101101     | 1110101 | 5 |
| 1    | 1101011     | 1101011 | 5 |
| 5    | 0110111     | 0110111 | 5 |
| 6    | 1011011     | 1010010 | 5 |
| 1    | 1110111     | 1110111 | 5 |
| 2    | 0111011     | 0111011 | 5 |
| 4    | 1111011     | 1111011 | 5 |
| 3    | 1111011     | 1111011 | 5 |
| 0    | 1101101     | 1110101 | 5 |
| 1    | 1101011     | 1101011 | 5 |
| 5    | 0110111     | 0110111 | 5 |
| 6    | 1011011     | 1010010 | 5 |
| 1    | 1110111     | 1110111 | 5 |

All nodes are connected.
2.2 Finding Connectivity links, Missing link between nodes of PDN as per PDS = {-3, -1, 0, 1, 3} 

| A | B | connected nodes | A+0 | A+1 | A-1 | A+3 | A-3 | B+0 | B+1 | B-1 | B+3 | B-3 |
|---|---|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0,1,3,4,6       | 0   | 1   | 6   | 3   | 4   | 0   | 1   | 6   | 4   | 0   |
| 0 | 1 | 0,1,4           | 0   | 1   | *   | *   | 4   | 1   | *   | 0   | 4   | *   |
| 2 | 0 | 1,3,6           | 0   | 1   | 6   | 3   | *   | *   | 3   | 1   | *   | 6   |
| 0 | 3 | 0,3,4,6         | 0   | 6   | *   | 4   | 3   | 3   | *   | 4   | 6   | 0   |
| 0 | 4 | 0,1,3,4         | 0   | 1   | *   | 3   | 4   | 4   | *   | 4   | 0   | 1   |
| 0 | 5 | 1,4,6           | *   | 1   | 6   | *   | 4   | *   | 6   | 4   | 1   | *   |
| 0 | 6 | 0,3,6           | 0   | 6   | 3   | *   | 6   | 0   | *   | *   | 3   |
| 1 | 1 | 0,1,2,4,5       | 1   | 2   | 0   | 4   | 5   | 1   | 2   | 0   | 4   | 5   |
| 1 | 2 | 1,2,5           | 1   | 2   | *   | *   | 5   | 2   | *   | 1   | 5   | *   |
| 1 | 3 | 0,2,4           | *   | 2   | 0   | 4   | *   | *   | 4   | 2   | *   | 0   |
| 1 | 4 | 0,1,4,5         | 1   | *   | 0   | 4   | 5   | 4   | 5   | *   | *   | 1   |
| 1 | 5 | 1,2,4,5         | 1   | 2   | *   | 4   | 5   | 5   | *   | 4   | 1   | 2   |
| A | B | connected nodes | A+0 | A+1 | A-1 | A+3 | A-3 | B+0 | B+1 | B-1 | B+3 | B-3 |
|---|---|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0,1,3,4,6       | 0   | 1   | 6   | 3   | 4   | 0   | 1   | 6   | 4   | 0   |     |
| 0 | 1 | 0,1,2,3,4,5,6  | 0   | 1   | 6   | 3   | 4   | 1   | 2   | 0   | 4   | 5   |     |
| 0 | 2 | 0,1,2,3,4,5,6  | 0   | 1   | 6   | 3   | 4   | 2   | 3   | 1   | 5   | 6   |     |
| 0 | 3 | 0,1,2,3,4,6    | 0   | 6   | 1   | 4   | 3   | 3   | 2   | 4   | 6   | 0   |     |
| 0 | 4 | 0,1,3,4,5      | 0   | 1   | 6   | 3   | 4   | 5   | 3   | 0   | 1   |     |     |
| 0 | 5 | 0,1,2,3,4,5,6  | 0   | 1   | 6   | 3   | 4   | 5   | 6   | 4   | 1   | 2   |     |
| 0 | 6 | 0,1,2,3,4,6    | 0   | 6   | 3   | 4   | 6   | 0   | 5   | 2   | 3   |     |     |
| 1 | 1 | 0,1,2,4,5      | 1   | 2   | 0   | 4   | 5   | 1   | 2   | 0   | 4   | 5   |     |
| 1 | 2 | 0,1,2,3,4,5,6  | 1   | 2   | 0   | 4   | 5   | 2   | 3   | 1   | 5   | 6   |     |
| 1 | 3 | 0,1,2,3,4,5,6  | 1   | 2   | 0   | 4   | 5   | 3   | 4   | 2   | 6   | 0   |     |
| 1 | 4 | 0,1,2,3,4,5    | 1   | 2   | 0   | 4   | 5   | 4   | 5   | 3   | 0   | 1   |     |
| 1 | 5 | 0,1,2,4,5,6    | 1   | 2   | 0   | 4   | 5   | 5   | 6   | 4   | 1   | 2   |     |
| 1 | 6 | 0,1,2,3,4,5,6  | 1   | 2   | 0   | 4   | 5   | 6   | 0   | 5   | 2   | 3   |     |
| 2 | 2 | 0,1,2,3,4,5,6  | 2   | 3   | 1   | 5   | 6   | 0   | 2   | 3   | 1   | 5   | 6   |
| 2 | 3 | 0,1,2,3,4,5,6  | 2   | 3   | 1   | 5   | 6   | 0   | 4   | 2   | 6   | 0   |     |
| 2 | 4 | 0,1,2,3,4,5,6  | 2   | 3   | 1   | 5   | 6   | 4   | 5   | 3   | 0   | 1   |     |
| 2 | 5 | 0,1,2,3,4,5,6  | 2   | 3   | 1   | 5   | 6   | 5   | 6   | 4   | 1   | 2   |     |
| 2 | 6 | 0,1,2,3,4,5,6  | 2   | 3   | 1   | 5   | 6   | 6   | 0   | 5   | 2   | 3   |     |
| 3 | 3 | 0,2,3,4,6      | 3   | 4   | 2   | 6   | 0   | 3   | 4   | 2   | 6   | 0   |     |
| 3 | 4 | 0,1,2,3,4,5,6  | 3   | 4   | 2   | 6   | 0   | 4   | 5   | 3   | 0   | 1   |     |
| 3 | 5 | 0,1,2,3,4,5,6  | 3   | 4   | 2   | 6   | 0   | 5   | 6   | 4   | 1   | 2   |     |
| 3 | 6 | 0,2,3,4,5,6    | 3   | 4   | 2   | 6   | 0   | 6   | 0   | 5   | 2   | 3   |     |
| 4 | 4 | 0,1,3,4,5      | 4   | 5   | 3   | 0   | 1   | 4   | 5   | 3   | 0   | 1   |     |
| 4 | 5 | 0,1,2,3,4,5,6  | 4   | 5   | 3   | 0   | 1   | 5   | 6   | 4   | 1   | 2   |     |
| 4 | 6 | 0,1,2,3,4,5,6  | 4   | 5   | 3   | 0   | 1   | 6   | 0   | 5   | 2   | 3   |     |
| 5 | 5 | 0,1,2,3,4,5,6  | 5   | 6   | 4   | 1   | 2   | 5   | 6   | 4   | 1   | 2   |     |
| 5 | 6 | 0,1,2,3,4,5,6  | 5   | 6   | 4   | 2   | 1   | 6   | 5   | 0   | 2   | 3   |     |

Based on PDS(0,±1,±3) For OR logical operation
### Based on PDS(0,±1,±3) For EX-OR logical operation

| A | B | connected nodes | A+0 | A+1 | A-1 | A+3 | A-3 | B+0 | B+1 | B-1 | B+3 | B-3 |
|---|---|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | Not connected   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   |
| 0 | 1 | 2,3,5,6         | *   | *   | 6   | 3   | *   | 2   | *   | 5   | *   | *   | *   |
| 0 | 2 | 0,2,4,5         | 0   | *   | *   | 4   | 2   | *   | *   | 5   | *   | *   | *   |
| 0 | 3 | 1,2             | *   | 1   | *   | *   | *   | 2   | *   | *   | *   | *   | *   |
| 0 | 4 | 5,6             | *   | *   | 6   | *   | *   | 5   | *   | *   | *   | *   | *   |
| 0 | 5 | 0,2,3,5         | 0   | *   | *   | 3   | 5   | *   | *   | 2   | *   | *   | *   |
| 0 | 6 | 1,2,4,5         | *   | 1   | *   | 4   | *   | *   | 5   | 2   | *   | *   | *   |
| 1 | 1 | Not connected   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   |
| 1 | 2 | 0,3,4,6         | *   | *   | 0   | 4   | *   | 3   | *   | 6   | *   | *   | *   |
| 1 | 3 | 1,3,5,6         | 1   | *   | *   | 5   | 3   | *   | 6   | *   | *   | *   | *   |
| 1 | 4 | 2,3             | *   | 2   | *   | *   | *   | 3   | *   | *   | *   | *   | *   |
| 1 | 5 | 0,6             | *   | *   | 0   | *   | *   | 6   | *   | *   | *   | *   | *   |
| 1 | 6 | 1,3,4,6         | 1   | *   | *   | 4   | 6   | *   | *   | *   | 3   | *   | *   |
| 2 | 2 | Not connected   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   |
| 2 | 3 | 0,1,4,5         | *   | *   | 1   | 5   | *   | 4   | *   | 0   | *   | *   | *   |
| 2 | 4 | 0,2,4,6         | 2   | *   | *   | 6   | 4   | *   | 0   | *   | *   | *   | *   |
| 2 | 5 | 3,4             | *   | 3   | *   | *   | *   | 4   | *   | *   | *   | *   | *   |
| 2 | 6 | 0,1             | *   | *   | 1   | *   | *   | 0   | *   | *   | *   | *   | *   |
| 3 | 3 | Not connected   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   |
| 3 | 4 | 1,2,5,6         | *   | *   | 2   | 6   | *   | 5   | *   | 1   | *   | *   | *   |
| 3 | 5 | 0,1,4,5         | *   | 4   | *   | 0   | 5   | *   | 4   | 1   | *   | *   | *   |
| 3 | 6 | 4,5             | *   | 4   | *   | *   | *   | 5   | *   | *   | *   | *   | *   |
| 4 | 4 | Not connected   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   |
| 4 | 5 | 0,2,3,6         | *   | *   | 3   | 0   | *   | 6   | *   | 2   | *   | *   | *   |
| 4 | 6 | 1,2,4,6         | 4   | *   | *   | 1   | 6   | 6   | *   | 2   | *   | *   | *   |
| 5 | 5 | Not connected   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   |
| 5 | 6 | 0,1,3,4         | *   | 4   | *   | 1   | *   | 0   | *   | 3   | *   | *   | *   |
| 6 | 6 | Not connected   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   | *   |

### Based on PDS(0,±1,±3) For Bi-implication logical operation

| A | B | connected nodes | A+0 | A+1 | A-1 | A+3 | A-3 | B+0 | B+1 | B-1 | B+3 | B-3 |
|---|---|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0,1,2,3,4,5,6   | 0   | 1   | 6   | 3   | 4   | 0   | 1   | 6   | 3   | 4   | 4   |
| 0 | 1 | 0,1,4           | 0   | 1   | *   | 4   | 1   | *   | 0   | 4   | *   | *   | *   |
| 0 | 2 | 1,3,6           | *   | 1   | 6   | 3   | *   | 2   | 3   | 1   | 5   | 6   | 6   |
| 0 | 3 | 0,3,4,5,6       | 0   | *   | 6   | 3   | 4   | 3   | 4   | *   | 6   | 0   | 0   |
| 0 | 4 | 0,1,2,3,4       | 0   | 1   | *   | 3   | 4   | 4   | 5   | 3   | 0   | 1   | 1   |
| 0 | 5 | 1,4,6           | *   | 1   | 6   | *   | 4   | 5   | 6   | 4   | 1   | 2   | 2   |
| 0 | 6 | 0,3,6           | *   | *   | 6   | 3   | *   | 6   | 0   | 5   | 2   | 3   | 3   |
| 1 | 1 | 0,1,2,3,4,5,6   | 0   | 1   | 6   | 3   | 4   | 0   | 1   | 6   | 3   | 4   | 4   |
| 1 | 2 | 1,2,5           | 1   | 2   | *   | 5   | 2   | *   | 1   | 5   | *   | *   | *   |
| 1 | 3 | 0,2,4           | *   | 2   | 0   | 4   | *   | 4   | 2   | *   | 0   | *   | *   |
3. ALGORITHMIC DEVELOPMENT

3.1 The Algorithm setup for AND logical operation for PDN $δ=2$

Step 1- set node i and node j  
Step 2- for i= 0 to n 
For j=0 to n  
Val= node i $\land$ node j(bitwise)  
j++ 
Print val  
I++ 
Step 3- stop  
The output of the algorithm will generate the following pattern 5,33,44,33. Algorithm for the other logical operation can be given similar way by replacing logical symbol.

4. CONCLUSION

We are using logical operators between vectors of connectivity matrix to study connectivity & complexity of network. We found that binary relation between nodes is well defined with network flow. The principle of Boolean Algebra may holds in interconnection network. The following are the patterns of the bit of the vector of a interconnection network.

1. AND operations of the Vectors of a PDN-5,33,44,33  
2. EX-OR operation of the Vectors of PDN-5,77,66,77  
3. OR operations of the Vectors of a PDN-0,44,22,44  
4. Implication operation of the Vectors of a PDN-7,55,66,55  
5. Bi-Implication operations of the Vectors of a PDN - 7,33,55,33

5. REFERENCES

[1] C. Wu and T. Feng. Tutorial, interconnection networks for parallel and distributed processing. Tutorial Texts Series. IEEE Computer Society Press, 1984.  
[2] www.interconnection of networks, elements of parallel computing and architecture [Last seen 22-11-2018]  
[3] Ms J.Nandagaoli and Dr. J.W. Bakal, “Study of Perfect Difference Network”, International journal of Computer Science”, Vol 3, Issue 6 July 2014.  
[4] Behroz Parhami, Mikhail Rakov “Application of Perfect Difference Sets to the Design of Efficient and Robust Interconnection Networks”;  
[5] S.Tiwari and R.K.Katare, “A Study of fabric of Architecture using Structural Pattern and Relation”, “International Journal of Latest Technology in Engineering and Management and Applied Science”, Vol 4, Issue 09, Sep 2015.  
[6] S.Tiwari, R.K.Katare, V. Sharma and C.M.Tiwari, “ Study of Geometrical Structure of Perfect Difference Network”, “ International Journal of Advanced Research in Computer and Communication Engineering”, Vol5,Issue3,March 2016.  
[7] Ms J.Nandagaoli and Dr. J.W. Bakal, “Study of Perfect Difference Network”, International journal of Computer Science”, Vol 3, Issue 6 July 2014.  
[8] J. Beiriger, W. Johnson, H. Bivens et al., “Constructing the ASCII Grid,” In: 9th IEEE Symposium on High Performance Distributed Computing, IEEE Press, New York, 2000, pp. 193 - 200.  
[9] Agarwal, A. and Agarwal, A. (2011). The Security Risks Associated with Cloud Computing. International Journal of Computer Applications in Engineering Sciences, 1 (Special Issue on CNS), 257-259.  
[10] Katare R K and Chaudhary N S, “Study of topological property of interconnection networks and its mapping to Sparse Matrix model” International journal, 2009.
[11] Singer J. “A theorem in Finite Projective Geometry and Some Applications to Number Theory” Trans. American Math. Soc. Vol. 43, pp. 377-385, 1938.

[12] Katare, R.K., Chaudari, N.S., Mugal, S.A., Verma, S.K., Imran, S. Raina, R.R. “Study of link Utilization of Perfect Difference Network and Hypercube “Conference on”FECS”, the world congress in Computer Science, Computer Engineering and Applied Computing, Las Vegas, Nevada, USA, July 25, 2013.