International Journal of Basic and Applied Sciences, 14 (4) (2025) 320-329

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/m7a8kz38
Research paper

Study of Structural Relationship between Vectors Using Hypercube Interconnection Network

Rakesh Kumar Katare [1] *, Surya Prakash Pandey [1], Charvi Katare [2]

[1] *Department of Computer Science, Awadhesh Pratap Singh University, Rewa (MP) 486003, India*
[2] *Velocis Systems Pvt. Ltd. A-25, Sector-67, Noida-201301, Uttar Pradesh, India*
**Corresponding author E-mail: katare1962@gmail.com*

Received: May 14, 2025, Accepted: July 7, 2025, Published: August 11, 2025

Abstract

In this paper, we have studied the structural relationship of nodes and edges of a Hypercube of degree *n*. By using different combinations of vectors of the connectivity matrix, we derived properties of binary relations of sets of vertices and edges, and developed an algebraic system of switching for the development of efficient communication algorithms.

Keywords: *Binary Relation; Connectivity Matrix; Interconnection Network; Structural Relationship; Tripod; Vector.*

1. Introduction

An interconnection network is a system that connects processors and memories to enable efficient data transfer and communication. The modelling of interconnection involved various methods of linking processors. The linking between processors in a different mathematical way is a topology, therefore topology of an interconnection network refers to the physical arrangement of nodes and connections within a network, determining how data is routed between them [1] [2] [26].
Recent progress highlights [27] [28] [29] the development of hypercube variants with increased fault tolerance, Hamiltonicity, diagnostic capability, and optimization of resources. These developments meet the burgeoning demands of high-performance computing and quantum communication models [8].
There are various methods of modelling the interconnection network. We are using one of the mo;delling of the interconnection network by converting the network into its equivalent connectivity matrix [3]. The different connectivity matrices show nodes-to-nodes connectivity, nodes-to-edge connectivity, and edge-to-edge connectivity [4]. In this way, a connectivity matrix shows the structural relationship between the processors [5] [6]. A row or column vector of a connectivity matrix shows one node's connectivity to other nodes, and a vector of edge-to-edge connectivity matrix shows one edge's connectivity to other edges of the interconnection network [7]. In the same manner, a vector of node-to-edge connectivity matrix shows node connectivity with edges of the interconnection network. In this paper, we used switching between the vectors of the interconnection network by using the mathematical logical operators and the operators of set-theoretic notations. We have shown that all the binary relations between the vectors work properly and are very useful for the development of routing and communication algorithms [8] [19]. This study will help in the development of an efficient algorithm for connectivity and complexity of the interconnection networks [9].
A vector is a row or a column in a connectivity matrix. Each vector in a connectivity matrix shows the connectivity [6] with other processors through edges in the interconnection network [1] [7] [8]. In processor-to-edge connectivity, the number of row vectors is the number of edges shown in Table 2, and the column vector shows the number of edges connected to one processor. In edge-to-edge connectivity [10] [11], the number of columns or row vectors is the number of edges. The edge-to-edge connectivity matrix shows the connectivity and complexity of the interconnection network. Our approach is to apply mathematical logical operators and set-theoretic notations between vectors to validate binary relations and lattice formatting by using vectors of a connectivity matrix to analyze the connectivity and complexity of the interconnection network. This algorithmic analysis will develop efficient communication between the processors. Here we are using the set-theoretic notations [12] and equivalent statements of statement calculus for the formation of an algebraic system so that efficient communication algorithms can be developed.

2. Review of literature and research gaps

- Early models such as the generalized hypercube proposed by Bhuyan and Agrawal (1984) [3] and the symmetric tree structure proposed by Wu and Agrawal (1985) [16] form a strong foundation for the structural flexibility of hypercubes in various dimensions and processor communication patterns. Subsequently, restricted connectivity and super-connectivity discussed by Chen and Tan (2007) [6] and Chen et al. (2003) [11] have enriched the structural insights of flexible communication topologies. Recent research efforts have focused on topological models for estimating various hypercube versions by Imran et al. (2014) [2], while Patnaik and Tripathi (2014) [1] have presented innovative hybrid models such as the star-Mobius cube in the context of massively parallel processing. Tiwari et al. (2021) [5] introduced a well-organized communication structure using Perfect Difference Networks (PDNs), while Katare et al. (2019) [7] further elaborated on its structural rationality. This discussion is further refined by the novel vector-based communication models of Singh et al. (2024) [8].

- Formal exploration of connectivity and Hamiltonian properties through cycle embeddings and Cartesian products further clarifies the concept of hypercube structure and highlights the importance of graph-based analysis methods, as reflected in the work of Liu (2024) [14], Chiu and Shih (1999) [10], and Yang et al. (2023) [13]. Innovative approaches such as fuzzy sets and interconnection network embeddings reveal new data flow upsurge dimensions by Kong et al. (2009) [12], Chaudhari et al. (2011) [20], and Chaudhari et al. (2013) [21].
- The poor diagnostic capability of folded hypercube networks, which ensures enhanced resilience in the face of multiple failures, has been analyzed in mission-critical parallel computing systems by Chang et al. (2025) [28]. Hamiltonian cycles in balanced hypercubes with faulty edges provide unprecedented contributions to routing and scheduling by refining the sensitivity of cyclic fault tolerance by Lan et al. (2025) [27].
- In 2025, Mary and Rajasingh (2025) uncovered vertex cover problems in extended Fibonacci cubes derived from faulty hypercubes and highlighted its potential applications in decryption and monitoring of interconnection networks [30]. Also, Noguez et al. (2024) applied hypercube principles in digital integration initiatives and demonstrated the versatility of this model in education and policy making [29].

Despite the many advances mentioned above, several research gaps remain:

- Non-tripod-based switching: No tripod-based studies have been found yet in the entire literature, so our study intends to explore tripod-based switching methods.
- Lack of Boolean Algebraic and Lattice Theory Applications: There is no use of lattice and Boolean algebra in the stated literature. Studies such as those in Shmatkov (1992) also suggest unexplored links with Boolean algebras in topology formation.
- Non-binary Communication Methods: There is limited or no exploration using set-theoretic and mathematical logical operations between vectors or quantum-inspired communication in hypercube variants.
- Limited application or use of vectors in the studies: In earlier studies, there was no application or use of vectors in studying the structural relationships of interconnection networks. To overcome these limitations, our study demonstrated the vector-based structural relationship between vectors of an interconnection network.

3. Vector representation of nodes in the connectivity matrix for an interconnection network

The diagram given below shows the chordal addressing of the nodes of a hypercube. In Table 1, the connectivity matrix between processors is shown, while in Table 2, the vertex-to-edge connectivity matrix is presented. Also in Table 3, we have shown the edge-to-edge connectivity. Here, we assume that a processor is connected to itself; hence, self-loops are presented in these matrices.

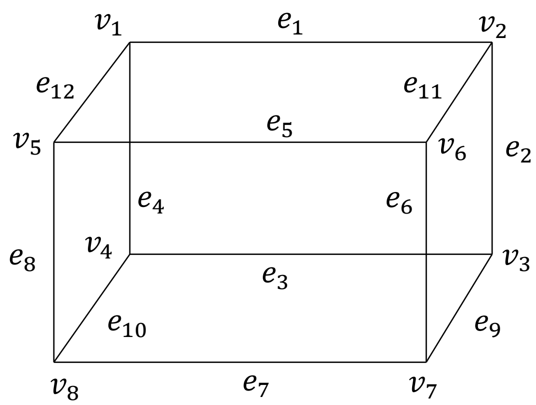

Fig. 1: Hypercube Connectivity of Dimension 3 with Choral Address Representation of Nodes.

3.1. Node-to-node square connectivity matrix or processor-to-processor square connectivity matrix

Table 1 shows the node-to-node structural representation of a 3-dimensional hypercube. The density of the matrix of Table 1 is $(\frac{8*4}{64} * 100) = 50\%$. Total edges are $\frac{32-\text{self\_loop}}{2}$ is 12. Each vector in the node-to-node connectivity matrix has a binary value. For example, a vertex $v_1 = \{11011000\}$. The other node vector also contains a binary value shown in the connectivity matrix.

Table 1: The Vertex-to-Vertex Connectivity Matrix of the Hypercube

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	1	0	1	1	0	0	0
v_2	1	1	1	0	0	1	0	0
v_3	0	1	1	1	0	0	1	0
v_4	1	0	1	1	0	0	0	1
v_5	1	0	0	0	1	1	0	1
v_6	0	1	0	0	1	1	1	0
v_7	0	0	1	0	0	1	1	1
v_8	0	0	0	1	1	0	1	1

3.2. Node-to-edge connectivity matrix or structural representation of the processor-to-link connectivity matrix

The matrix shown in Table 2 is the structural representation between two nodes. Each row vector shows connectivity between two processors/nodes. There are twelve edges between 8 nodes; therefore, the density of the matrix is $(\frac{12*2}{96} * 100) = 25\%$. Here we have the edge vector as e_1 to e_{12} and the binary values of the set $e_1 = \{11000000\}$ and vector v_1 to v_8 and the binary value of the set $v_1 = \{100100000001\}$. Similarly, all nodes have their binary value shown in the connectivity matrix.

Table 2: The Vertex-to-Edge Connectivity Matrix of the Hypercube

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
e_1	1	1	0	0	0	0	0	0
e_2	0	1	1	0	0	0	0	0
e_3	0	0	1	1	0	0	0	0
e_4	1	0	0	1	0	0	0	0
e_5	0	0	0	0	1	1	0	0
e_6	0	0	0	0	0	1	1	0
e_7	0	0	0	0	0	0	1	1
e_8	0	0	0	0	1	0	0	1
e_9	0	0	1	0	0	0	1	0
e_{10}	0	0	0	1	0	0	0	1
e_{11}	0	1	0	0	0	1	0	0
e_{12}	1	0	0	0	1	0	0	0

3.3. Edge-to-edge square connectivity matrix through processors or the link connectivity matrix

In Table 3, we have shown the edge-to-edge square connectivity matrix through processors or the link connectivity matrix. Each edge has five connections with a self-loop, then the density of the matrix is $(\frac{(12*5)-12}{12*12} * 100) = 33.33\%$. Here in the connectivity matrix, each vector has a binary value, i.e. $e_1 = \{110100000011\}$.

Table 3: The Edge-to-Edge Connectivity Matrix of the Hypercube

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}
e_1	1	1	0	1	0	0	0	0	0	0	1	1
e_2	1	1	1	0	0	0	0	0	1	0	1	0
e_3	0	1	1	1	0	0	0	0	1	1	0	0
e_4	1	0	1	1	0	0	0	0	0	1	0	1
e_5	0	0	0	0	1	1	0	1	0	0	1	1
e_6	0	0	0	0	1	1	1	0	1	0	1	0
e_7	0	0	0	0	0	1	1	1	1	1	0	0
e_8	0	0	0	0	1	0	1	1	0	1	0	1
e_9	0	1	1	0	0	1	1	0	1	0	0	0
e_{10}	0	0	1	1	0	0	1	1	0	1	0	0
e_{11}	1	1	0	0	1	1	0	0	0	0	1	0
e_{12}	1	0	0	1	1	0	0	1	0	0	0	1

Table 1 and Table 3 are square matrices, and the diagonals of both matrices are {11111111} for Table 1, and {111111111111} for Table 3; each row vector and its corresponding column vector are the same. In this manner, both connectivity matrices are symmetric in their upper triangular off-diagonal matrix with their lower triangular off-diagonal matrix. Therefore, we can call Table 1 and Table 3 the connectivity matrix or the butterfly matrix. The value of binary vectors of all the connectivity will be used to explore the properties of binary relations like reflexive, symmetric, anti-symmetric, and transitive, and we will also use the properties of lattice and equivalence relation to explore the efficiency of the interconnection network for communication between the nodes. We are using the mathematical models of Boolean algebra and lattices because when we use mathematical logical operations, we get structural relationships [4].

4. Use of set-theoretic notation and mathematical logical relations between the vectors of an interconnection network

The identities of statement algebra, which include idempotent law, associative law, distributive law, absorption law, and De Morgan's law concerning 'AND' and 'OR' operations, behave the same as stated in statement calculus in set algebra [12] concerning Union and Intersection [10]. Based on the above statement, we are using identities of statement algebra that will be applied to vectors of the connectivity matrix of the hypercube [5]. We have used properties of binary relations between the binary vectors of an interconnection network for the development of efficient algorithms.

4.1. The behaviour of set-theoretic notations and mathematical logical functions

Theorem 1: *The behaviour of set-theoretic notation and mathematical logical functions remains the same in the study of the interconnection network.*

Proof: Applying the assumptions of set theory and statement calculus on the set of edges defined in section 3.2.

1) $(v_1 \cap v_2) = e_1$, where e_1 is connected to v_1 and v_2 as shown in Table 2.
2) $(v_1 \wedge v_2) = e_1$
3) $v_1 \cap v_4 = e_4$, where e_4 is connected to v_1 and v_4 as shown in Table 2.
4) $(v_1 \wedge v_4) = e_4$

5) $(v_1 \cap v_5) = e_{12}$, where e_{12} is connected to v_1 and v_5 as shown in Table 2.
6) $(v_1 \wedge v_5) = e_{12}$
Where $e_1 = \{v_1, v_2\}$, $e_4 = \{v_1, v_4\}$, $e_{12} = \{v_1, v_5\}$ as shown in Table 2.
The above points can be represented as follows:

$$(v_1 \wedge v_2) \vee (v_1 \wedge v_4) \vee (v_1 \wedge v_5) = (e_1 \vee e_4 \vee e_{12}) = v_1 \tag{1}$$

And similarly, applying set-theoretic operations in (1)We found the following:

$$(v_1 \cap v_2) \cup (v_1 \cap v_4) \cup (v_1 \cap v_5) = (e_1 \cup e_4 \cup e_{12}) = v_1 \tag{2}$$

The above presentation (1) and (2) show that v_2, v_4, v_5 is routed through e_1, e_4 and e_{12} to v_1.

- Proving the $(v_1 \wedge v_2) = e_1$

$$v_1 = \{e_1, e_4, e_{12}\} \tag{3}$$

$$v_2 = \{e_1, e_2, e_{11}\} \tag{4}$$

$$v_1 \wedge v_2 = \{(e_1 \wedge e_1) \vee (e_4 \wedge e_2) \vee (e_{12} \wedge e_{11})\} \tag{5}$$

$v_1 \wedge v_2$ =((11000000 ∧ 11000000) ∨ (10010000 ∧ 01100000) ∨ (10001000 ∧ 01000100))
=11000000 is equivalent to e_1

- Proving the $(v_1 \wedge v_4) = e_4$

$$v_1 = \{e_1, e_4, e_{12}\} \tag{6}$$

$$v_4 = \{e_3, e_4, e_{10}\} \tag{7}$$

$$v_1 \wedge v_4 = \{(e_1 \wedge e_3) \vee (e_4 \wedge e_4) \vee (e_{12} \wedge e_{10})\} \tag{8}$$

$v_1 \wedge v_4$ =((11000000 ∧ 00110000) ∨ (10010000 ∧ 10010000) ∨ (10001000 ∧ 00010001))
=10010000 is equivalent to e_4

- Proving the $(v_1 \wedge v_5) = e_{12}$

$$v_1 = \{e_1, e_4, e_{12}\} \tag{9}$$

$$v_5 = \{e_5, e_8, e_{12}\} \tag{10}$$

$$v_1 \wedge v_5 = \{(e_1 \wedge e_5) \vee (e_4 \wedge e_8) \vee (e_{12} \wedge e_{12})\} \tag{11}$$

$v_1 \wedge v_5$ =((11000000 ∧ 00001100) ∨ (10010000 ∧ 00001001) ∨ (10001000 ∧ 10001000))
=10001000 is equivalent to e_{12}

Applying mathematical logical operations on edges to find a tripod vertex v_1.

$$v_1 = \{e_1, e_4, e_{12}\} \tag{12}$$

$$v_1 = \{(e_1 \vee e_4) \vee e_{12}\} = \{(11000000 \vee 10010000) \vee 10001000\} \tag{13}$$
= 11011000 which is equivalent to v_1.

It is clear from the above that with operations on vectors, we can derive edges; on the other hand, applying operations on edges, we can derive vertices. The data flow and switching between the above tripod are shown in Figures 2, 3, and 4 as an example that we have simulated through the Python programming using the networking "NetworkX Python package" and other mathematical libraries of Python. NetworkX is used for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

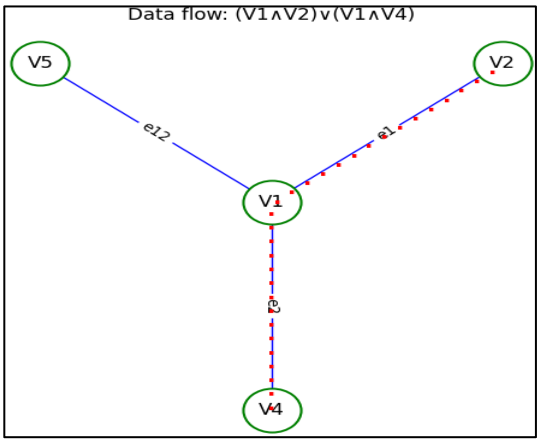

Fig. 2: Data flow and switching between the sets $(v_1 \wedge v_2) \vee (v_1 \wedge v_4)$ using mathematical logical operations

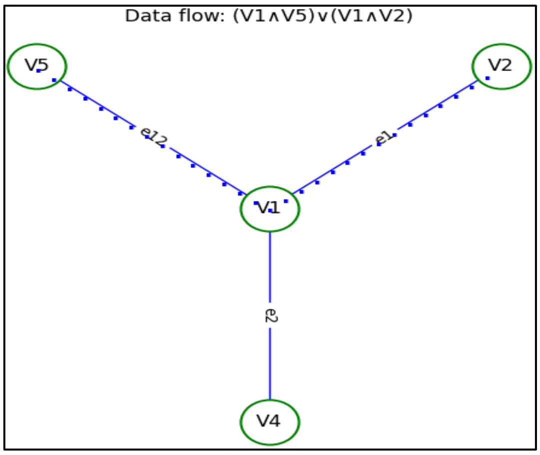

Fig. 3: Data flow and switching between the sets $(v_1 \wedge v_5) \vee (v_1 \wedge v_2)$ using mathematical logical operations

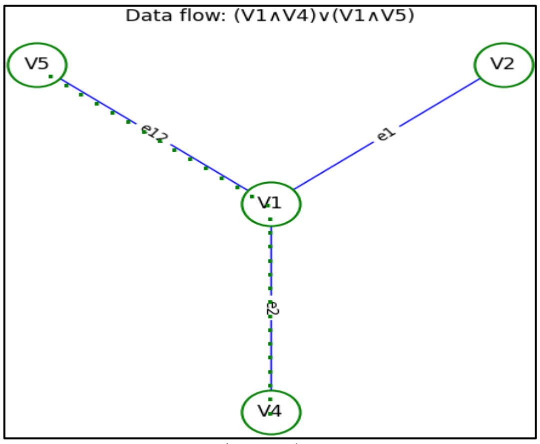

Fig. 4: Data flow and switching between the sets $(v_1 \wedge v_4) \vee (v_1 \wedge v_5)$ using mathematical logical operations

4.2. A tripod of hypercube follows the properties of a binary relation

There are 2^n nodes in a hypercube where $n = 3$ as shown in Figure 1. We say that a tripod is a lattice and also contains the properties of a Connex and Equivalence Relation. We know that a lattice is a partially ordered set with a Least Upper Bound or Greatest Lower Bound. The connectivity values for the tripod vertices are shown below.

Tripod v_1
$v_1 = 11011000$
$v_2 = 11100100$
$v_4 = 10110001$
$v_5 = 10001101$

Tripod v_2
$v_2 = 11100100$
$v_1 = 11011000$
$v_3 = 01110010$
$v_6 = 01001110$

Tripodv_3
$v_3 = 01110010$
$v_2 = 11100100$
$v_4 = 10110001$
$v_7 = 00100111$

Tripod v_4
$v_4 = 10110001$
$v_3 = 01110010$
$v_1 = 11011000$
$v_8 = 00011011$

Tripod v_5
$v_5 = 10001101$
$v_1 = 11011000$
$v_6 = 01001110$
$v_8 = 00011011$

Tripod v_6
$v_6 = 01001110$
$v_2 = 11100100$
$v_7 = 00100111$
$v_5 = 10001101$

Tripod v_7
$v_7 = 00100111$
$v_3 = 01110010$
$v_6 = 01001110$
$v_8 = 00011011$

Tripod v_8
$v_8 = 00011011$
$v_4 = 10110001$
$v_5 = 10001101$
$v_7 = 00100111$

Applying the properties of binary relations on the above tripods and applying binary relations on the vectors of the connectivity matrix, we will get a tautology, which is shown in Case I, Case II, and Case III.

4.3. Case I

1) Reflexive property
   a) Vertex-to-vertex and vertex-to-edge relation

$$(v_i \wedge v_i) \to v_i \quad (14)$$

   b) Edge-to-edge relation

$$(e_i \wedge e_i) \to e_i \quad (15)$$

A reflexive relation is a type of binary relation on a set where every element in the set is related to itself. If vertex v_1 is reflexive and a vertex v_2 is reflexive, then we can presume that $e_1$is also reflexive because a reflexive relation is a type of binary relation in a set where every element in the set is related to itself [19].

2) Symmetry Property
   a) Vertex-to-vertex and edge-to-edge relation

$$(v_i \wedge v_{i+1}) \to (v_{i+1} \wedge v_i) \quad (16)$$

   b) Edge-to-edge relation

$$(e_i \wedge e_{i+1}) \to (e_{i+1} \wedge e_i) \quad (17)$$

In an interconnection network a "Symmetric node" refers to a node that has the same connectivity pattern as all other nodes in the network, meaning it is connected to the same number of neighbouring nodes with identical links characteristics, essentially exhibiting a balanced and consistent structure throughout the network, essentially, no single node has a different connection pattern compared to others [13] [16] [17].

- Symmetry in topology

The overall network structure is symmetrical, allowing for efficient routing and data transfer due to predictable connectivity patterns, as shown in Table 1 [22].

Why is symmetry important in interconnection networks [17]

i) Uniform Connectivity: Each node has the same degree (number of connections) to other nodes.
ii) No special positions: No node has a unique placement or role compared to others.
iii) Benefits of symmetry: Symmetric networks often facilitate simpler routing algorithms, better load balancing, and improved fault tolerance.
iv) Simplified routing: With symmetric nodes, finding the shorter path between any two nodes becomes easier due to the predictable structure.
v) Efficient load balancing: Symmetrical networks can distribute workload evenly across nodes, improving overall performance.
vi) Fault tolerance: If one node fails, the network can often reroute data through other nodes due to the consistent connectivity.

3) Antisymmetric property
a) Vertex-to-vertex and vertex-to-edge relation

$$(v_i \wedge v_{i+1}) \wedge (v_{i+1} \wedge v_i) \to (v_i \wedge v_{i+1}) \quad (18)$$

b) Edge-to-edge relation

$$(e_i \wedge e_{i+1}) \wedge (e_{i+1} \wedge e_{i+2}) \to (e_i \wedge v_{i+2}) \quad (19)$$

Anti-symmetric means relating to a relation that implies equality of any two quantities for which it holds in both directions. It means the communication between v_1 and v_2 holds in both directions [24].

4) Transitive property
a) Vertex-to-vertex and vertex-to-edge relation

$$(v_i \wedge v_{i+1}) \wedge (v_{i+1} \wedge v_{i+2}) \to (v_i \wedge v_{i+2}) \quad (20)$$

b) Edge-to-edge relation

$$(e_i \wedge e_{i+1}) \wedge (e_{i+1} \wedge e_{i+2}) \to (e_i \wedge v_{i+2}) \quad (21)$$

In an interconnection network, a "transitive node" refers to a node that, if connected to another node which is further connected to a third node, implies a connection between the first and third node as well. The node v_1, v_2 and v_4 are transitive means the tripod is transitive. The above explanation with reflexive, symmetric, transitive, and anti-symmetric properties and the property of Greatest Lower Bound (GLB), which is and Least Upper Bound (LUB) between nodes, leads to the formation of a lattice and an equivalence relation [14] [15] [25].

Now we can summarize the above explanation shows that the tripod is a lattice because the architecture shows the properties of a binary relation in the following manner:

1) The tripods are lattice because the properties of reflexivity, symmetry, and transitivity hold.
2) The tripods also follow the properties of an equivalence relation because the properties of reflexivity, Anti-symmetry, and transitivity hold in the same way.
3) The property of connex is also followed by tripods.

The verification of the above relation shows that the binary relation between vectors of the connectivity matrix proves that the architecture contains properties of a lattice, equivalence relation, and connex.

4.4. Case II

In modern times, the modelling and algorithmic development of interconnection networks for the study of connectivity and complexity requires many mathematical theories, like geometrical and algebraic structures, such as lattices and the properties of binary relations. This fills the gap between the old and modern concepts of modelling the interconnection networks [18]. The lattice and finite projective geometry [20] [21] give the properties of modularity, the principle of duality, and incidence relationships.

a) Modularity: In discrete mathematics, a modular lattice is a lattice where every pair of elements is modular. This means that for any elements or nodes v_1, v_2 and v_4 in the lattice, if $v_4 \leq v_1$ then $v_1 \vee (v_2 \wedge v_4) = (v_1 \vee v_2) \wedge v_4)$.
i.e. $(v_4 \leq v_1) \Rightarrow (v_1 \vee (v_2 \wedge v_4) \Leftrightarrow (v_1 \vee v_2) \wedge v_4)$ which can also be represented as:

$$(v_4 \rightarrow v_1) \Rightarrow (v_1 \vee (v_2 \wedge v_4) \Leftrightarrow (v_1 \wedge v_2) \wedge v_4) \quad (22)$$

- Every sublattice is a modular lattice that is also modular.
- If a lattice is distributive and $v_4 \leq v_1$, then $v_1 \vee (v_2 \wedge v_4) = (v_1 \wedge v_2) \vee (v_2 \wedge v_4) = v_1 \vee (v_2 \wedge v_4)$

$$\text{i.e.}\ v_4 \leq v_1 \Rightarrow v_1 \vee (v_2 \vee v_4) \Leftrightarrow (v_1 \vee v_2) \wedge v_4 \Leftrightarrow (v_1 \wedge v_4) \vee (v_2 \wedge v_4) \quad (23)$$

- A lattice is modular if and only if all pairs of elements are modular.
- A lattice is m-symmetric if (v_1, v_2) is a modular pair, then (v_2, v_1) is also a modular pair.

In the hypercube, Mono-pod, di-pod, tripod, and quarter-pod are modular, and a lattice is responsible for the construction of an n-cube. The modularity of the n-cube is dual because one module is like an n-pod, which is the "building block" of an n-cube. The tripod has three connections, the monopod has one connection, the di-pod has two connections, and the quad-pod has four connectivity lines. When two tripods are connected, we assume that the communication will take place on both sides in one line that connects the two tripods.

Theorem 2: *In the hypercube interconnection network, every tripod is also modular.*

Proof: Let's consider the vertex-to-vertex relation in Table 1 and define a tripod in the following way:

$$v_4 \leq v_1 \Longrightarrow (v_1 \wedge (v_2 \vee v_4) \Leftrightarrow (v_1 \wedge v_2) \vee v_4) \quad (24)$$

Using the binary value of the vector's v_1, v_2 and v_4 from Table 1.

$(10110001 \Longrightarrow 11011000) \Rightarrow (11011000 \wedge (11100100 \vee 1011000) \Leftrightarrow (11011000 \wedge 11100100) \vee 10110001)$. We get 11111111, which is a tautology that fulfils the properties of modularity.

Theorem 3. *A tripod is a modular lattice in the structural representation of an interconnection network.*

Proof: Let's consider the edge-to-edge relation in Table 3.

$$e_{11} \leq e_1 \Rightarrow e_1 \wedge (e_2 \vee e_{11}) \Leftrightarrow (e_1 \wedge e_2) \vee e_{11} \quad (25)$$

i.e. $(e_{11} \Longrightarrow e_1) \Rightarrow (e_1 \wedge (e_2 \vee e_{11}) \Leftrightarrow (e_1 \wedge e_2) \vee e_{11})$. It is a tautology.
Using the binary value of edge vectors e_1, e_2 and e_{11} from Table 1, we get the following:

$$(110011000010 \Rightarrow 110100000011) \Rightarrow (110100000011 \wedge (110011000010 \vee 110011000010)) \Leftrightarrow ((110100000011 \wedge 111000001010) \vee 110011000010) \quad (26)$$

We get the tautology 111111111111 as a result that verifies the tripod is a modular lattice.
In an interconnection network a "node as modular lattice" refers to a concept where each node within the network can be considered as a part of a larger lattice structure that adheres to the "Modular Law" meaning that certain relationships between nodes within the network follow specific rules regarding their connection and hierarchy, allowing for efficient data routing and processing based on this structure organization.

Theorem 4: *The Modular inequalities in the structural representation of the interconnection network hold.*

Proof: Let's define the Modular inequalities in the following way:

$$(v_1 \wedge v_2) \vee (v_1 \wedge v_4) \leq v_1 \wedge [v_2 \vee (v_1 \wedge v_4)] \quad (27)$$

$$(v_1 \vee v_2) \wedge (v_1 \vee v_4) \geq v_1 \vee [v_2 \wedge (v_1 \vee v_4)] \quad (28)$$

Using the value of vertex-to-vertex relations in Table 1 for (27)
$(11011000 \wedge 11100100) \vee (11011000 \wedge 10110001) \Longrightarrow (11011000 \wedge [11100100 \vee (11011000 \wedge 10110001)]$.It gives a tautology; similarly, the method for (28) will be the same.
$(11011000 \vee 11100100) \wedge (11011000 \vee 10110001) \Longleftarrow (11011000 \vee [11100100 \wedge (11011000 \vee 10110001)]$. It is a tautology; Hence, we get the tautology for the required modular inequality that validates the theorem.

Theorem 5: *The Distributive inequalities hold in the structural representation of an Interconnection Network.*

Proof: Let's define the Distributive inequalities by using the vertex-to-vertex relation from Table 1.

$$v_1 \vee (v_2 \wedge v_4) \leq (v_1 \vee v_2) \wedge (v_1 \vee v_4) \tag{29}$$

$$v_1 \wedge (v_2 \vee v_4) \geq (v_1 \wedge v_2) \vee (v_1 \wedge v_4) \tag{30}$$

Consider the vertex-to-vertex vector's binary value from Table 1 for (29) and (30).
$(11011000 \vee (11100100 \wedge 10110001)) \Rightarrow (11011000 \vee 11100100) \wedge (11011000 \vee 10110001)$ is 11111111, it holds tautology.
Similar to the way for (30).
$(11011000 \wedge (11100100 \vee 10110001)) \Leftarrow (11011000 \wedge 11100100) \vee (11011000 \wedge 10110001)$ is 11111111, it holds tautology.
Hence, both (29) and (30) hold the distributive inequalities and are verified by tautology.

Theorem 6: *In the hypercube interconnection network, the lattice modules hold the isotonicity.*

Proof: Let's define the theorem same way as Theorems 4 and 5.

$$v_4 \leq v_2 \Longrightarrow \begin{cases} v_1 \wedge v_4 \leq v_1 \wedge v_2 \\ v_1 \vee v_4 \leq v_1 \vee v_2 \end{cases} \tag{31}$$

We can write to the (31) in the following manner:

$$(v_4 \Rightarrow v_2) \Longrightarrow \begin{cases} (v_1 \wedge v_4) \Rightarrow (v_1 \wedge v_2) \\ (v_1 \vee v_4) \Rightarrow (v_1 \vee v_2) \end{cases} \tag{32}$$

Consider the vertex-to-vertex vector's value from Table 1 for (32).

$$(10110001 \Rightarrow 11100100) \Longrightarrow \begin{cases} (11011000 \wedge 10110001) \Rightarrow (11011000 \wedge 11100100) \\ (11011000 \vee 10110001) \Rightarrow (11011000 \vee 11100100) \end{cases} \tag{33}$$

In the (33) we will get 11111111, which is a tautology that holds the lattice property of isotonicity.

Theorem 7: *A tripod in the hypercube interconnection network is a chain.*

Proof: Let's consider the following case using a vertex-to-vertex relation and vector values from Table 1 of a hypercube interconnection network:

$$(v_i \leq v_{i+1}) \Leftrightarrow ((v_1 \wedge v_2) \Leftrightarrow v_1) \Leftrightarrow ((v_1 \vee v_2) \Leftrightarrow v_2) \tag{34}$$

$$v_i = 11011000 \tag{35}$$

$$v_{i+1} = 11100100 \tag{36}$$

$(11011000 \Rightarrow 11100100) \Leftrightarrow ((11011000 \wedge 11100100) \Leftrightarrow 11011000) \Leftrightarrow ((11011000 \vee 11100100) \Leftrightarrow 11100100)$ is 11111111 which is a tautology.
Hence, we get a tautology which satisfies the distributive law, so it is a chain. Inequality holds in the hypercube interconnection network because of the three-dimensional representation of the hypercube. The position of the tripod is different in all directions, with the same properties.

b) Duality Principle: Statement about lattices involving the operations $*$ and $\oplus$ and the relations $\leq$ and $\geq$ remains true if $*$ replaced by $\oplus$, $\oplus$ by $*$, $\leq$ by $\geq$ and $\geq$ by $\leq$. The operators $*$ and $\oplus$ are called the dual of each other, as are the relations $\leq$ and $\geq$. Same way, the lattices (L, $\leq$) and (L, $\geq$) are called duals of each other [19]. In the hypercube interconnection network set of vertices (points) and edges (lines) follows the duality and can be defined as a dual structure. The duality principle exists; therefore, meet and join operations are interchangeable because the greatest lower bound and least upper bound have the same value and have one set. This property of greatest lower bound and least upper bound is called the duality principle because of this meet and join are interchangeable [23].
c) Incidence Relationships: Lattices ensure structural relationships [15] among subspaces, enabling efficient computation where $\wedge$ (AND) and $\vee$ (OR) be the lattice operation [25]. In mathematical logic, the incidence relation is replaced by the mathematical logical relation as AND ($\wedge$).

In a hypercube interconnection network, every node represents a vertex using a lattice-based subspace relation. The lattices are very important and play a crucial role in structuring the projective space.

5. Results and discussion

In the hypercube interconnection network, each node represents a vertex using a lattice-based subspace relation. Lattices are very important and play a key role in the structure of projective space. The mathematical logical analysis of connectivity between different vertices of an interconnection network by using different combinations of edges and vertices in a connectivity matrix has been shown. How different vertices are connected to different edges by exploring the connectivity matrix has been represented. The basic idea is to analyze the connectivity of edges and vertices, which are shown as tripod connectivity due to 2^n connectivity matrix of the interconnection network, where n is the degree. If degree is n, then we can define it a pod. This n-pod is the basic building block of a hypercube of degree n and is also the centre of consideration for the analysis. In our study, we have used a hypercube of degree 3; therefore, our n-pod is a tripod, which is a building block of our study. Any connectivity matrix shows the structural relationship between edges and nodes of an interconnection network. The structural representation of the hypercube can be represented in the form of a set, which is derived from the vectors of the vertex-to-vertex connectivity matrix. These vectors of a connectivity matrix are a string of binary numbers. The set-theoretic notations are known as intersection and union, which can be replaced by 'AND' and 'OR' operations in the study of the relation between edges and vertices. We have also used the different combinations of values of rows or columns of the connectivity matrix, which is also known as vectors of the connectivity matrix, for exploring the relationships. Now we have an opportunity to use set-theoretic notations on sets and the concept of mathematical logical operations on the binary vectors of the different combinations of the connectivity matrix. This basic presentation of edges and vertices is used to find out the relationship between edges and vertices.

The vertex-to-vertex connectivity matrix shows a total number of edges between the vertices, which also shows the degree of a vertex. Here we can assume some time self-loop and sometimes no self-loop of a vertex; it depends on the application. In this paper, we have proved through operations on vertices that we can find edges and also vice versa.

In our theorems, we have proved that a tripod is a lattice and it also holds the property of an equivalence relation, besides having the property of connex. We have also used binary relations of reflexivity, symmetric, anti-symmetric, and transitivity for different combinations of vectors of different connectivity matrices, and we found that all the properties of binary relations hold properly in our study of connectivity and complexity of vectors of the hypercube. In Theorem 1, we have studied the behavior of set-theoretic notation and mathematical logical functions using the assumptions of set theory and statement calculus. In Theorems 2 and 3, we have shown that every tripod is also a module and is a modular lattice that holds the properties of modular lattices, and also proved it through logical operations on tripods. In addition, Theorem 4, Theorem 5, Theorem 6, and Theorem 7 proved the modular inequalities, distribution inequalities, isotonicity, and tripod of the hypercube interconnection network as a chain in the structural representation of the interconnection network. These properties are used to analyze binary relations and help to provide bounds and relationships among elements within a relation based on their “modular inequalities”. Also, provide the background for binary relations, explaining how logical operations behave.

6. Conclusion

We have interpreted an efficient algorithmic system of switching by using vectors of different combinations of connectivity matrices. We used properties of binary relations of sets of vertices and edges. The lattice and Boolean algebra validate the properties by finding a tautology. This system of algebra will help to develop efficient communication algorithms.

References

[1] Pattanayak, D., Tripathy, Devashreel, 2014. Star-Mobius Cube: A New Interconnection Topology for Large Scale Parallel Processing, International Journal of Emerging Technologies in Computational and Applied Sciences, 7(1). https://api.semanticscholar.org/CorpusID:18807747.

[2] Muhammad Imran, Sakander Hayat, Muhammad Yasir Hayat Mailk, 2014. On topological indices of certain interconnection networks, Applied Mathematics and Computation, Volume 244, Pages 936-951. https://doi.org/10.1016/j.amc.2014.07.064.

[3] Bhuyan and Agrawal, 1984. Generalized Hypercube and Hyperbus Structures for a Computer Network, *IEEE Transactions on Computers*, vol. C-33, no. 4, pp. 323-333. https://doi.org/10.1109/TC.1984.1676437.

[4] Katare, R.K., Singh, N., Pandey, P., Katare, C., 2021, Study of Communication Structural Relationship between Processors of Parallel and Distributed System, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, Volume 7, Issue 5. https://doi.org/10.32628/CSEIT21759.

[5] Tiwari, S., Bhardwaj, M., Katare, R.K., 2021. Study of Communication Pattern for Perfect Difference Network. In: Bajpai, M.K., Kumar Singh, K., Giakos, G. (eds) *Machine Vision and Augmented Intelligence—Theory and Applications. Lecture Notes in Electrical Engineering*, vol 796. Springer, Singapore. https://doi.org/10.1007/978-981-16-5078-9_11.

[6] Y-Chuang Chen, Jimmy J.M. Tan, 2007. Restricted connectivity for three families of interconnection networks, *Applied Mathematics and Computation*, Volume 188, Issue 2, Pages 1848-1855. https://doi.org/10.1016/j.amc.2006.11.085.

[7] Katare, R.K., Premikar, S., Singh, N., Tiwari, S., Katare, C. 2019. Vector Operation on Nodes of Perfect Difference Network Using Logical Operators. International Journal of Advanced Research in Computer Science, 10(6), 29–39. https://doi.org/10.26483/ijarcs.v10i6.6485.

[8] Singh, Sarini, Singh, Prof. Neelam, 2024. Investigating Binary Relationships Among Vectors in Interconnection Networks for Parallel and Distributed Systems, *International Research Journal of Modernization in Engineering Technology and Science*, Vol.06, Issue09. https://www.irjmets.com/uploadedfiles/paper//issue_9_september_2024/61300/final/fin_irjmets1725206675.pdf.

[9] Pandey, Surya Prakash, Katare, Rakesh Kumar, 2018. Application of Fixed-Point Algorithm in Parallel Systems, *International Journal of Computer Sciences and Engineering*, Vol.6, Issue 6. https://doi.org/10.26438/ijcse/v6i6.714719.

[10] Wen-Sz Chiue, Bih-Sheue Shieh, 1999. On connectivity of the Cartesian product of two graphs, *Applied Mathematics and Computation*, Volume 102, Issues 2–3, Pages 129-137. https://doi.org/10.1016/S0096-3003(98)10041-3.

[11] Y-Chuang Chen et al., 2003. Super-connectivity and super-edge-connectivity for some interconnection networks, *Applied Mathematics and Computation*, Volume 140, Issues 2–3, Pages 245-254. https://doi.org/10.1016/S0096-3003(02)00223-0

[12] Zhi Kong et al., 2009. Comment on “A fuzzy soft set theoretic approach to decision making problems”, *Journal of Computational and Applied Mathematics*, Volume 223, Issue 2, Pages 540-542 https://doi.org/10.1016/j.cam.2008.01.011.

[13] Da-Wei Yang et al., 2023. Symmetric property and edge-disjoint Hamiltonian cycles of the spined cube, *Applied Mathematics and Computation*, Volume 452, 128075. https://doi.org/10.1016/j.amc.2023.128075

[14] Liu, M., 2024. Cycle Embedding in Enhanced Hypercubes with Faulty Vertices. *Symmetry*, *16*(1), 44. https://doi.org/10.3390/sym16010044.

[15] Yang, R., & Yuan, M., 2023. The Structural Properties of (2, 6)-Fullerenes. *Symmetry*, *15*(11), 2078. https://doi.org/10.3390/sym15112078.

[16] Nam Sung Woo and Agrawala, 1985. A Symmetric Tree Structure Interconnection Network and its Message Traffic," *IEEE Transactions on Computers*, vol. C-34, no. 8, pp. 765-769. https://doi.org/10.1109/TC.1985.1676626

[17] Tiwari, S., Saket, P, Katare, R.K., Katare, C., Tiwari, A., Gautam, K., 2020. Study of Data Structure and Symmetry Properties for Interconnection Network. *Research Journal of Science Engineering and Technology*, Vol.10. https://doi.org/10.32804/RJSET.

[18] Song, J., Qi, X., & Cao, Z., 2023. An Independent Cascade Model of Graph Burning. *Symmetry*, *15*(8), 1527. https://doi.org/10.3390/sym15081527.

[19] J.P. Tremblay, R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, *Tata McGraw Hill Edition 1997.*

[20] Choudhary, S., Sharma, H. and Patkar, S., 2011. Optimal folding of data flow graphs based on finite projective geometry using lattice embedding, *Discrete Mathematics, Algorithms and Applications*.

[21] Choudhary, S., Sharma, H., and Patkar, S., 2013. Optimal folding of data flow graphs based on finite projective geometry using vector space partitioning. *Discrete Mathematics, Algorithms and Applications*, 05(04), 1350036. https://doi.org/10.1142/S1793830913500365.

[22] Rakesh Kumar Katare and N. S. Chaudhari, 2009. Study of Topological Property of Interconnection Networks and its Mapping to Sparse Matrix Model, *International Journal of Computer Science and Applications*, 6(1):26–39. https://api.semanticscholar.org/CorpusID:12150831.

[23] Kantorovich, L.V., Akilov, G.P., in Functional Analysis (Second Edition), 1982. https://www.sciencedirect.com/topics/mathematics/vector-lattice.

[24] Merriam-Webster, Antisymmetric relation, Merriam-Webster.com, https://www.merriam-webster.com/dictionary/antisymmetric , 2025. (Accessed 14 Mar. 2025).
[25] V. D. Shmatkov, Incidence algebras over lattices, 1992. *Russian Mathematical Surveys*, Volume 47, Issue 4, 230–231, https://doi.org/10.1070/RM1992v047n04ABEH000933
[26] Mengjie Lv, Jianxi Fan, Jingya Zhou, Baolei Cheng, Xiaohua Jia, 2020. The extra connectivity and extra diagnosability of regular interconnection networks, Theoretical Computer Science, Volume 809, Pages 88-102, ISSN 0304-3975. https://doi.org/10.1016/j.tcs.2019.12.001.
[27] Ting Lan, Huazhong Lü, 2025. Hamiltonian cycles of balanced hypercube with disjoint faulty edges, Information Processing Letters, Volume 187, 106518, ISSN 0020-0190 https://doi.org/10.1016/j.ipl.2024.106518.
[28] Nengjin Zhuo, Shumin Zhang, Jou-Ming Chang, Chengfu Ye, 2025. Non-inclusive diagnosability of folded hypercube-like networks, Discrete Applied Mathematics, Volume 364, Pages 237-246, ISSN 0166-218X, https://doi.org/10.1016/j.dam.2025.01.006
[29] Noguez, C.A., et al., 2024. The Hypercube Model: Digital Inclusion. EDULEARN24. https://library.iated.org/view/ASTENGONOGUEZ2024HYP.
[30] StalinMary, R., Rajasingh, I. (2025). Minimum Vertex Cover in Extended Fibonacci Cubes. In: Rajasingh, I., Mary Arul, S., Grace Thomas, D. (eds) Applied Computational Mathematics. ICCE 2023. Springer Proceedings in Mathematics & Statistics, vol 480. Springer, Cham. https://doi.org/10.1007/978-3-031-77764-6_10.