Study of Connectivity Using AND and XOR Logical Operation on JOSEPHUS CUBE for Parallel and Distributed Systems

1Manisha Singh, 2Neha Singh, 3Prof. Rakesh Katare, 4Charvi Katare
1,2,3,4Department of Computer Science, A.P.S. University, Rewa, M.P., India
Email- manishasinghgaharwar@gmail.com

Abstract
Graph is a conceptual mathematical data structure. In this paper, Josephus cube (JC) architecture is studied as a graph. We consider Josephus architecture as a self-loop free graph. We study Josephus cube interconnection connectivity properties where vertices present the processor (nodes) and communication arcs / links as edges of the graph. Josephus cube architecture connectivity is presented through mathematical tool: Matrices to perform logical operations. Incidence Matrix, Circuit Matrix and Path Matrix were considered to map the graph properties to the matrix properties of Josephus Cube interconnection network. Here we take matrix and in each matrix all columns are arranged using the same order of edges. Binary AND (\(\cdot\)) logical operation is performed in MATLAB on the incidence matrix & transpose of circuit matrix and vice-versa for Josephus graph in this paper. The result shows that, for any \(p\)\(^{th}\) processor lies in the \(n\)\(^{th}\) circuit, then the nonzero values in a matrix at corresponding position exist, if the particular communication link is incident on the \(p\)\(^{th}\) processor (vertex) and is also in the \(n\)\(^{th}\) circuit of the Josephus cube graph structure. Path matrix of Josephus cube was also presented to define all the connectivity’s between two processor nodes \(P(P_x,P_y)\). Here we observed that AND (\(\cdot\)) logical operation was executed on the incidence matrix and transposed of path matrix, the resultant matrix has 1’s in exactly two rows \(P_x\) and \(P_y\), the processor nodes for which we had considered all the possible paths and the rest of the \((R_n-2)\) rows are filled with 0 bits, that confirms \(P_x\) and \(P_y\) nodes does not lies in any other connectivity paths. The XOR of any two rows of path matrix of Josephus cube corresponds to the circuit that contains the traversed paths.

Keywords: - Josephus Cube, Interconnection Network, Incidence Matrix, Circuit Matrix, Path Matrix, Transposed.

1- INTRODUCTION
Josephus problem was discussed in 1\(^{st}\) century according to the josephus history, he and his soldier were trapped in a cave this told by Jewish in flavius josephus’s books of the history of the jewish war against in roman. The formulation was in terms of points and lines in a finite projective plane \([3,4,7]\). The josephus cube considered for being develop into a interconnect network mainly through the help of perfect difference network works of Parhami, Behrooz and Rakov, M.A \([8,16]\). 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, and similarly here in this paper we try to show that the josephus cube network interconnection scheme is best possible in the sense that it can cover nodes and smallest nodes degree with power of 2. Compared JC and some of their derivatives to interconnection networks with similar cost and performance with hypercube and its other variants Josephus cube network are a robust high-performance interconnection network for parallel and distributed systems.
systems. The most exhaustive comparative study of hypercube and perfect difference network was done by Katare et al., 2007[17], based on topological properties. Topological properties of josephus cube compared with the corresponding properties of hypercube. Here we tried to implement sparse linear system with the scheme. It was already proved that access function or routing function to map data on hypercube contains topological properties. The study of circuits based on the architecture of josephus cube is further taken forward by Loh Peter 2000[6] in their research work on study of link utilization of JC and Hypercube. They have shown that the circuits formed in JC are a combination of odd and even length. Adjacency matrix of nxn of JC presented to study the link utilization and topological properties.

The JC which has just about similar properties as Hypercube enjoys a superior speed up and efficiency as compared to Hypercube. It is a super graph of the N-cube and several of its variation. The topology can systematically interconnect an capricious number of nodes. Josephus cube scalability, diameter, communication performance and total number of edges compare favourably with the N-cube. In josephus cube, there is one advantage is that we can use any number of processors whereas in hypercube the number of processors is a power of 2.

2- INCIDENCE MATRIX OF

JOSEPHUS CUBE

Matrix is a mathematical representation of particular function that to study the arrangement of processors (processing nodes) and communication links/ arcs of interconnection network. For processing the connectivity between the communicating processors, a graph representation is used to convert into matrix form. Binary logical operations can be easily performed on these matrices to have a real elucidation of topological properties of an interconnection network. A Matrix of processing elements (PEs) can be written as PEs= [Pij]ε RiXj. In this paper, we map the graph theoretic properties to matrix properties to study the Josephus cube interconnection network. Incidence Matrix, Circuit Matrix and Path Matrix columns and rows vectors were explored for connectivity analysis of Josephus cube.

We consider Josephus cube as an undirected graph with n vertices as processor nodes and e edges as communications arcs with no self-loops. Matrix JC= [Pij], each processor element of interconnection network is represented as Pij =1, if processor Pi is having an incident communication arc j as e, Pij =0 if no incident communication arc is present [3]. Fig. 1 represents the Incidence graph structure of Interconnection Network JC (G) for node 8.

Fig-1 Incidence graph of interconnection network Josephus cube node =8
Incidence Matrix of Interconnection Network JC for node 8 is $P_{8 \times 20}$

Table 1: Josephus cube incidence matrix $P_{8 \times 20}$

The following observations on incidence matrix JC (G) are made:

1) Each column vector of the matrix has exactly two 1’s, this signify that every communication arc of Josephus cube is incident on two processor nodes.

2) Each row vector of the matrix has exactly four 1’s, therefore the node degree ($d_n$) of every processor node in the incidence graph is four, $d_n = 5$, $0 \leq n \leq 7$.

3) The sum of the degrees of processor node of JC (G) is $d_0 + d_1 + d_2 + d_3 + d_4 + d_5 + d_6 + d_7 = 20$.

Set of communication links of PDN S = \{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t\}, Cardinality of Set $|S| = 8$

4) The incidence matrix of JC (G) contains no two identical columns, hence no parallel arcs are observed in the graph.

5) No isolated processor node is observed in the JC (G) because no row vector with all zeros exists in the incidence matrix.

6) JC is a connected graph as the incidence matrix cannot be written in a block diagonal form.

3- CIRCUIT MATRIX OF JOSEPHUS CUBE

In this section, the circuit matrix of graph JC (G) is a c by e matrix where c is the number of distinct circuits in and e is the number of communication links in the Josephus cube architecture. The circuit matrix is written as $CMs = [c_{ij}]_{c \times e}$, $c_{ij} = 1$, if $j^{th}$ communication arc is part of $i^{th}$ Josephus cube circuit graph else $c_{ij} = 0$, if no communication arc is part of the circuit under consideration [9]. Table 1 lists the number of independent circuits of figure 1; JC (G) for node 8 is 24.
Table 2- JC (G) for node 8 is 24.

Table 3- Josephus Path Matrix $P_{17 \times 20}$

| Circuit | a | b | c | d | e | f | g | h | i | j | k | l | M | n | o | p | q | r | s | t |
|---------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_1$   | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_2$   | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_3$   | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_4$   | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_5$   | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_6$   | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_7$   | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_8$   | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_9$   | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{10}$ | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{11}$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{12}$ | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| $C_{13}$ | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| $C_{14}$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{15}$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{16}$ | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{17}$ | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| $C_{18}$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| $C_{19}$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| $C_{20}$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{21}$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $C_{22}$ | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $C_{23}$ | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| $C_{24}$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
The following observations on circuit matrix of JC (G) are made:-

1) No non-circuit edge exists in the JC (G) because no column vectors with all zeros exist in the matrix.

2) In Each row vector of circuit matrix is a circuit.

3) There is no self-loop in JC (G), as no column with single 1 bit exists in any of the column vectors.

4) The number of communication bow in a single circuit is equal to the number of 1 bit in each row of the matrix. 0 bits represents the non-participating arch in the circuit.

5) Circuit matrix cannot be represented in block-diagonal form. JC (G) is non separable in any form as no two independent sub-graphs can be derived from the parent graph.

Here We have verified a theorem, i.e for a self-loop free graph (presenting interconnection network) every row vector of circuit matrix is orthogonal to every row of incidence matrix with the columns are arranged in the same order of communication arcs [9].

\[ I \cdot CM^T = CM \cdot I^T = 0 \]

where T denotes the transposed matrix.

Binary AND (.) logical operation is performed using MATLAB between incidence matrix and transpose of the circuit matrix and vice-versa. Intermediate resultant matrix is passed as parameter to function. Screen Shot1 shows the resultant matrix below.

**Screen shot 1**

In This result, we try to shows that, for any \( p \)th processor lies in the \( n \)th circuit, then the nonzero values in a matrix at corresponding position exist, if the particular communication link is incident on the \( p \)th processor (vertex) and is also in the \( n \)th circuit of the JC graph structure. If \( p \)th (processor) vertex is not in the \( n \)th circuit, then there is zero entry and the AND operation of the two rows is zero else \( p \)th vertex is in the \( n \)th circuit, there will be exactly two 1 bits in the sum of the products of individual row entries. As we know \( 1+1=0 \), the AND logical operation of any two row vectors of incidence matrix & transpose of circuit matrix and vice-versa, result is 0.
4- PATH MATRIX OF JC
A path matrix of JC in fig.1 P(P₀, P₄) is defined between two processor nodes from P₀→P₄. The column vector of the path matrix represent the labeled communication arcs and row vector correspond to the independent paths between the two processors (node) on the path [9]. Table 2 lists the intercommunication processor node paths defined from P₀ to P₄.

| Path | Independent Paths from P(P₀ → P₄) |
|------|----------------------------------|
| P₀   | PT₀{j}, {i,d}, {k,e}, {h,c,d}  |
| P₁   | PT₁{l,d}, {m,e}, {f,e,d}       |
| P₂   | PT₂{c}, {p,f,e}, {o,e}         |
| P₄   | PT₃{d}, {q,t,e}                |
| P₅   | PT₄{o,c,d}, {e}                |
| P₆   | PT₅{r}                         |
| P₇   | PT₆{s}, {q,d}                  |

Table 4 - independent path of jc

| Path P(P₀ → P₄) | A | B | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t |
|-----------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PT₀{j}          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| {i,d}           | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| {k,e}           | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| {h,c,d}         | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PT₁{l,d}        | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| {m,e}           | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| {f,e}           | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PT₂{c}          | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| {p,f,e}         | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| {o,e}           | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| PT₃{d}          | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| {q,t,e}         | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| PT₅{o,c,d}      | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| {e}             | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PT₆{r}          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Volume-9 Issue-12 October 2020, DOI: 10.26671/IJIRG.2020.12.9.101  Page 112
Table 5 path metric of jc

| PT/s | 0 0 0 0 0 0 0 0 0 0 0 0 1 0 |
| {q,d} | 0 0 0 1 0 0 0 0 0 0 0 0 0 0 |

The following observations on path matrix of JC (G) are made:

1) It is observed that no common communication arc exists in all the paths between two processors (P0 → P4) because a column vector with all 1’s does not exist in the matrix.
2) All the communication arcs are traversed in a path from P0 → P4, as no column with all 0’s exist in the above matrix.
3) All the possible current paths are covered, as no row vector with all 0’s exist in the path matrix.
4) The XOR logical operation is performed on any two rows of path matrix P(P0 → P4) the resultant bit value corresponds the row of the circuit matrix of JC [9].

5- ADJACENCY MATRIX OF JC

Adjacency of josephus cube is (P, E), which is P={P0, P1, P2…Pn} and E={a, b, c……t} and without parallel edges. The adjacency matrix of josephus cube is an n x n symmetric binary matrix X={Xij} defined over the integers such that Xij = 1 if and only if pi and pj ∈ E otherwise 0.

| Processors | P0 | P1 | P2 | P3 | P4 | P5 | P6 | P7 |
|------------|----|----|----|----|----|----|----|----|
| P0         | 0  | 1  | 1  | 1  | 1  | 0  | 0  | 1  |
| P1         | 1  | 0  | 1  | 1  | 0  | 1  | 1  | 0  |
| P2         | 1  | 1  | 0  | 1  | 0  | 1  | 1  | 0  |
| P3         | 1  | 1  | 1  | 0  | 1  | 0  | 0  | 1  |
| P4         | 1  | 0  | 0  | 1  | 0  | 1  | 1  | 1  |
| P5         | 0  | 1  | 1  | 0  | 1  | 0  | 1  | 1  |
| P6         | 0  | 1  | 1  | 0  | 1  | 0  | 0  | 1  |
| P7         | 1  | 0  | 0  | 1  | 1  | 1  | 1  | 0  |

The following observation on adjacency matrix of JC(G) are made:

1) The entries along the processors of X are all 0’s if and only if the processors have no self loops.
2) The degree of each processor equals the number of 1’s in the corresponding row or column in matrix.
3) According to the definition of adjacency matrix no provision for parallel edges.
4) If there is no self loop, then degree of a vertex equals the number of 1’s in the corresponding row or column of matrix.

6- RESULTS AND DISCUSSIONS

Using any type of matrices, represent graph, for their structure storage and connectivity processing in computer system. We have reveal the usefulness of three eloquent and efficient incidence, circuit and path matrix on josephus cube network architecture; JC (G).

It is observed that incidence matrix of JC gives the node degree four which is equivalent to the number of 1’s bit in each row. Summation of node degree (dn) of n processors for the JC where |S| is the cardinality of set of communication links. An interconnection network of josephs cube with no self loop connectivity and high node degree can presented preferably with incidence matrix, as all the processor connectivity information is preserved.

Circuit matrix confirms that Josephus Cube is a self-loop free interconnection network with no row containing a single 1 bit. MATLAB is used to demonstrate that the each row of circuit matrix is orthogonal to every row of incidence matrix using the AND and mod (2) operation. Path Matrix of JC(G) from processor node P0 to P4, shows that all the distinct paths are traversed only once such that no communications arcs are missed. All 1’s Column vector is missing shows that no
edge is common in all the paths. AND operation on the incidence matrix with a transpose of path matrix (\(P_0 \rightarrow P_3\)) gives a resultant matrix that has 1’s in two rows. All other rows data are filled with 0’s.

7- CONCLUSION
In this paper, we have derived the incidence matrix, circuit matrix and path matrix of Josephus cube Network (JC). Abstract graph data structure of JC is studied to establish the connectivity between the processor nodes of JC through incidence matrix while the circuit matrix confirms that JC is a self loop free graph with maximum length of the circuit is the total count of processor nodes that formed the closed circuit. \(\delta\) times the cardinality of the set of communication link is equivalent to the summation of the node degree of the JC. Here Path matrix presents all the distinct paths are traversed only once. The AND (\(\wedge\)) logical operation performed on the incidence matrix and transpose of circuit matrix and vice-versa it confirms that the \(n^{th}\) processor is in the \(i^{th}\) circuit when the result of dot product of any two rows is zero. The XOR (\(\oplus\)) of any two rows of path matrix of JC confirms that the corresponding circuit contains the traversed paths.

8- REFERENCES
i. Katare, R.K., Chaudhari, N.S. (2009). Study of Topological Properties of Interconnection Networks and its mapping to Sparse Matrix Model. International Journal of Computer Science and Application, Technomathematics Research Foundation, 6(1), 26-39.
ii. Parhami, B., Rakaov, M. (2005). Perfect Difference Networks and Related Interconnection structures for Parallel and Distributed Systems. IEEE Journal on Parallel and Distributed Systems, 16(8), 725-736.
iii. Katare, R.K., Chaudhari, N.S. (2007). A Comparative study of hypercube and perfect difference network for parallel and distributed systems and its application to sparse linear system. Vladimir Journal of Computer Information Sciences, Sandipani Academy, Ujjain(M.P.), India, 2, 13-30.
iv. Siegel, H. J. (1979). A model of SIMD machines and a comparison of various interconnection networks. IEEE trans. Comput, C-28, 907-917.
v. Tiwari, S., Katare, R. K. (2017). A Study of Interconnection Network for Parallel & Distributed System, BEST: IJMITE, 5, 69-74.
vii. Jakobczyk, F. (1973). On the generalized Josephus problem. Journal-Glasgow Math., 14, 168-173.
viii. Deo, N. (2000). Graph Theory: with applications to Engineering and computer science. Prentic Hall of India, New-Delhi.
ix. Squire, J.S., Palais, S. M. (1963). Programming and design considerations of a highly parallel computer. proc. AFIPS spring-joint computer Conf. Jour, 23, 395-400.
x. Singer, J. (1938). A theorem in Finite Projective Geometry and Some Applications to Number Theory. Trans American Mathematical Society, 43, 377-385.
x. Khan, M. Zubair, Singh, R., Alam, J. (2012). A Routing Algorithm for Triple Loop (TL): A Fixed Degree Two Dimensional Geometric Interconnection Network for Parallel System. Journal of Information System and Communication, 3(1), 220-224.
x. Kumar, A., Peh, L. S., Kundu, P., Jha, N. K. (2007). Express virtual channels: Towards the ideal interconnection fabric. In ISCA’07: 34th Annual International Symposium on Computer Architecture, Conference Proceedings, 35(2), 150-161. https://doi.org/10.1145/1250662.1250681.
xii. Saad, Y., Schultz, M.H. (1988). Topological Properties of Hypercubes, IEEE transactions on Computers, 37(7), 867-872.