International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET22919

A Collision Free Network with Minimal Routing Path for 2^n Nodes

Anup Kumar Biswas

Assistant Professor, Department of Computer Science and Engineering, Kalyani Govt. Engineering College, Kalyani, Nadia, West Bengal, India

Article Info

Volume 9, Issue 1

Page Number : 55-63

Publication Issue :

January-February-2022

Article History

Accepted : 05 Jan 2022

Published: 20 Jan 2022

ABSTRACT

In the present work, a simple but complex network which has collision free communications is created. With the help of the binary n-cube the network is formed. This n-cube network connection is formed by connecting distinct N number of nodes which are expressed as a power of 2. If each node has an address of m bits then there are N=2^m number of nodes in the n-cube network. To find out the predefined routing path for the case of this n-cube network, we can apply deterministic algorithm providing us a collision /deadlock free concept. In calculating the predefined routing path(s), node addresses involved in the path are calculated by using the exclusive OR operation, firstly, upon the two node addresses of source and destination, and then, on the derived node-addresses according to the algorithm stated in section 2. Here, the Exclusive-OR operation is performed simply by using 2-input XOR gate which may be made up of either (i) classical CMOS based material, or (ii) *Multiple input threshold logic gate* , or (iii) Single electron tunneling Transistor (SET). We are concentrated in the case of CMOS based XOR circuit. In this work, an E-cube Routing on a 6-dimensional hypercube has been designed and constructed. We have tried to find out the node addresses for predefining the deadlock free routing path for any set of "*source and destination*". To develop a "Collision free network with minimal Routing", a number of Exclusive-OR gates are arranged in a pattern discussed in section 3 and the desired circuit is implemented. The implemented circuit can supply the predefined path nodes through which the signal/packets/data can pass. Regarding the experimental input –output values, some truth tables are depicted in substantial places. At the time of applying the algorithm, we have investigated that for 6-dimentional hypercube, the maximum path length are confined to 6, i.e. whenever the source and destination points are provided, then 6-node points are produced. Among these 6-pionts some of them may have identical values. As a result path length should be less than 6. For the network, the time required to travel a path is measured and shown. The net designed and implemented must a be a collision free one, since when a truth table is written, there would not be any pair of nodes identical in a column.

Keywords : XOR, Routing-Path, N-Cube, Collision Free, Deadlock, Tree

I. INTRODUCTION

Packet routing may be conducted either by deterministically or adaptively, in a network. Relying on the network conditions we can prefer the adaptive routing. In the adaptive routing process an alternate path is possible. But for the case of deterministic routing, the routing path from source point to the goal/destination is strictly determined by the starting/source point node address as well as goal/destination node point address. In such case, the routing path must be measured irrespective of network condition in advance. Dimension order routing is the concept emerging from the algorithm of deterministic routing.

Dimension order routing requires to choose the successive channels or sequential node addresses in order to follow a specific order depending on the dimensions of multidimensional network. If we choose the networking in two dimensional mesh, the scheme is called X-Y routing. When we are thinking of a $n-cube$ network, then the scheme is defined as the E-cube routing [1-5, 12].

Our intention is to make some logical expressions by which we must be able to measure the routing path related nodes so as to travel the routing path through the nodes calculated.

With the help of the tree, the travelling path can be represented. As every nodes contains 6 arms in the network, the tree will be of branch factor = 6. In a full-fledged tree of branch factor 6, there will be $\sum_{i=1}^{6} 6^i$=55986 nodes where some of nodes are repeated. But for the welfare of the algorithm, only 2-6 nodes are sufficient to go from one node to another. This is the beauty of the algorithm used here. The less the nodes used, the less the time complexity and space complexity will be there.

Routing over a n-dimensional hypercube for 2^n nodes

We think of an n-cube having 2^n nodes (peaks). Every node (p) in n-cube is coded in binary form like $p = p_{n-1}p_{n-2}p_{n-3}\dots p_1p_0$. The source node from which we are interested in starting routing is $s = s_{n-1}s_{n-2}s_{n-3}\dots s_1s_0$. And the destination node to which the routing ends is $d = d_{n-1}d_{n-2}d_{n-3}\dots d_1d_0$. Now we need to find out the destination length from source(s) to destination (d) with the help of a small number of nodes or steps required.

We indicate the n-dimensions in the present situation as $i = 0, 1, 2, 3, \dots, n-1$; where the i th dimension marks to the $(i-1)^{th}$ bit in a node to be represented by $q = q_{n-1}q_{n-2}q_{n-3}\dots q_1q_0$. Suppose q is any node along the route to be gone through. The route from source(s) node to the destination (d) node can be determined by the following algorithm.

Algorithm:

(i) Measure the direction bit $k_i = s_{i-1} \oplus d_{i-1}$ for all i in n-dimensions i.e. $i = 1, 2, 3, \dots, n$; Commence with dimension $i = 1$ and $q = s$ and follow the following.

(ii) Start routing from the node q to the next node as $q \oplus 2^{i-1}$ if $k_i = 1$. Skip the step whenever $k_i = 0$.

(iii) Move to the next dimension $i+1$, that is $i \leftarrow i+1$. $If\ i \leq n$ go to the step (ii), else done.

By taking an example, the algorithm written above can be explained. Consider an E-cube routing over a 6-dimensional hypercube given in Fig. 1.

Here, we have chosen the dimension $n = 6$, so the number of nodes is equal to 2^6=64, the nodes are represented by 000000, 000001, 000010,…,111111. For the case of routing, we assume that the source and destination nodes are S=000100 and D=011100 respectively. So, $k = k_6k_5k_4k_3k_2k_1 = (S \oplus D)$=011000. We skip the dimension $i = 1, 2, 3$ *and* 6 as k_1=0⨁0=0, k_2=0⨁0=0 , $k_3 = 1 \oplus 1 = 0$ *and* $k_6 = 0 \oplus 0 = 0$ respectively. Now we are to route from source(s) node 000100 to $S \oplus (00k_4000) =$ 0 00100⨁01000=001100, since $k_4 = 0$⨁1=1. Next route from q = 001100 to the to q ⨁$(0k_50000)$=011100, since $k_5 = 0 \oplus 1 = 1$. As a result, the routing path should be: 000100→001100→011100.

Similarly, If we take another set of inputs, i.e. source and destination like S=100110 and D=001001 we obtain-

Source = (s5s4s3s2s1s0) =100110

Destination = (d5d4d3d2d1d0) = 001001

k6 k5 k4 k3 k2 k1=101111, in this case k5=0

Routing path :

100110→100111→100101→100001→101001→ 001001 and this path is shown in the Fig. 1 by thick green lines with arrows.

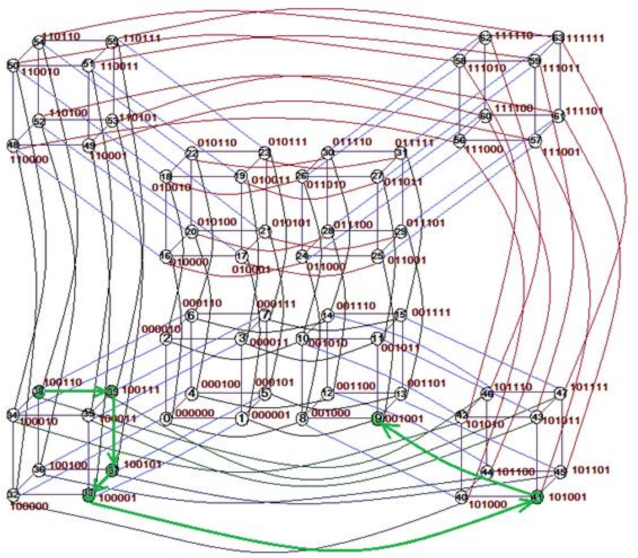

Fig.1 A routing Path with green color from S=100110 and D=001001

Circuit diagram for creating Path points

The circuit shown in Fig.1 consists of 64 node points and every node has 6 arms (which are bidirectional: incoming or out-going). If a packet is being tried to send from a point to another point, which path it follows is to be determined earlier otherwise the packet will fall in a labyrinth which is unexpected and undesirable. To overcome this problem, a deterministic path of 6 points will be predicted previously. Such a circuit drawn in Fig. 2 is capable of producing a set of six points, among them some may be identical, by using some identical XOR gates.

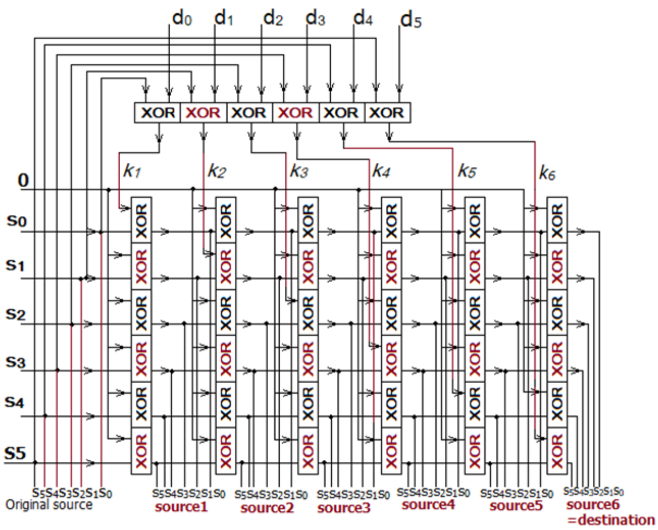

Fig. 2 Block diagram of finding out intermediate nodes of any routing path

According to the diagram drawn in the Fig. 2 as well as the algorithm, we find it easy to get the routing path from source to destination. Some paths consisting of the nodes or vertices of the cube are listed in the Table-1 below. In the diagram above, if we give the source and destination values to input terminatals, the output results we got from the output terminals which are given in the Table-1. Depending upon the number of nodes involved in a path, we have calculated the path length. Equation of the path length is given by the following relation:

Path length = number of nodes involved-1.

For instance, consider the serial number 25 in Table-1. There are 6 non-identical nodes (00000, 00001, 00011, 00111, 01111, 11111) in a routing path, so the $path\ length = 6 - 1 = 5$. Similarly, for serial number 9, there are 4 different nodes (101011, 101010, 101110, 001110) involved in the path, therefore the

$path\ length = 4 - 1 = 3.$

Path number can also be measured as total number of '1' in $k = k_6k_5k_4k_3k_2k_1 = (S \oplus D)$. Consider the 1st row in Table-1, we get $k = k_6k_5k_4k_3k_2k_1 = (110010 \oplus 101101) = 011111$. As there are five 1s in $k = k_6k_5k_4k_3k_2k_1 = 011111$, so the path length is 5.

Note that if somebody look into the Table-1, (s)he will get rarely the same 6-bit words in a column. It is an indication that there will be no confliction in the network in Fig.1 when applied above cited algorithm. In short, there should be no dead lock in the 6-dimensional routing process.

Table-1. Minimum routing path(s)

Sl. No.	Source(s) node number	Destination node number	*Source 1 original*	*Source 2*	*Source 3*	*Source 4*	*Source 5*	*Source 6*	goal	Routing Path Length
1	50	45	110010	110011	110001	110101	111101	101101	101101	5
2	10	18	001010	001010	001010	001010	000010	010010	010010	3
3	0	18	000000	000000	000010	000010	000010	010010	010010	3
4	46	12	101110	101110	101100	101100	101100	101100	001100	2
5	20	30	010100	010100	010110	010110	011110	011110	011110	2
6	39	54	100111	100110	100110	100110	100110	110110	110110	2
7	3	59	000011	000011	000011	000011	001011	011011	111011	3
8	14	61	001110	001111	001101	001101	001101	011101	111101	4
9	43	14	101011	101010	101010	101110	101110	101110	001110	3
10	1	33	000001	000001	000001	000001	000001	000001	100001	1
11	24	36	010110	010110	010100	010100	010100	000100	100100	3
12	20	7	011000	011001	011011	011111	010111	000111	000111	5
13	25	12	011001	011000	011000	011100	011100	001100	001100	3
14	62	5	111110	111111	111101	111101	110101	100101	000101	5
15	7	15	000111	000111	000111	000111	001111	001111	001111	1
16	63	9	111111	111111	111101	111001	111001	101001	001001	4
17	0	22	000000	000000	000010	000110	000110	010110	010110	3
18	55	19	110111	110111	110111	110011	110011	110011	010011	2
19	11	42	001011	001010	001010	001010	001010	001010	101010	3
20	50	60	110010	110010	110000	110100	111100	111100	111100	3
21	2	10	000010	000010	000010	000010	001010	001010	001010	1
22	41	11	101001	101001	101011	101011	101011	101011	001011	2
23	23	7	010111	010111	010111	010111	010111	000111	000111	1
24	6	48	000110	000110	000100	000000	000000	010000	110000	4
25	0	63	000000	000001	000011	000111	001111	011111	111111	6

From the Table -1, it is realizable that the maximal routing path from any set of "(source node, destination node)" is 6. For any set of two non-identical nodes, the routing path length varies from 1 to 6. If we wish to draw a tree for a source node as root node regarding the Table-1 or Fig. 1, we will have a tree of height 6 and branch factor is 6. Below, in Fig. 3(a), Fig. 3(b) and Fig. 3(c) are the three examples which indicate routing path(s) from source/root node(s) to the destination node(s).

Assuming that all 25 sets of (source, destination) are applied at a particular time in Fig.1, then from the Table-1, we will observe that in every row, there are seven words of length 6. But you will not find any two words identical in any column. So there is no probability/chance of conflict/ collision in the network in the Fig.1.

In general, for an n-bit number of a network system will have total number of nodes $=2^n$. When we take a node as the source point, there would have 2^n-1 destination(s) and travelling the path towards the destination is a more complex job. For searching the goal point from the source point, different searching process can be applied and the respective time and space complexity can be measured. But the algorithm used in this work can determine the different node numbers present in the routing path very easily and it tells us that path length will be no more than n i.e. $1\leq$ path length$\leq$n. As we have chosen n=6, so $1\leq$ path length$\leq$6. And the time complexity will be limited to n=6.

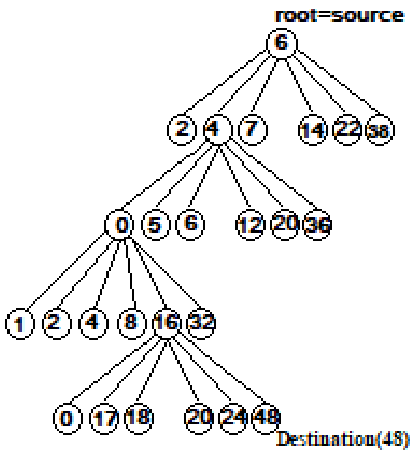

Fig. 3(a) Source to destination length=4

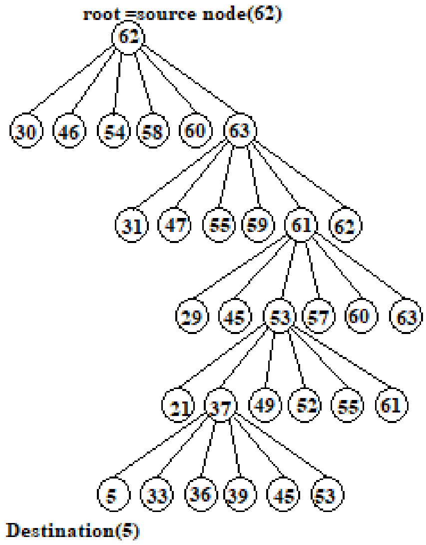

Fig.3(b) Source to destination length=5

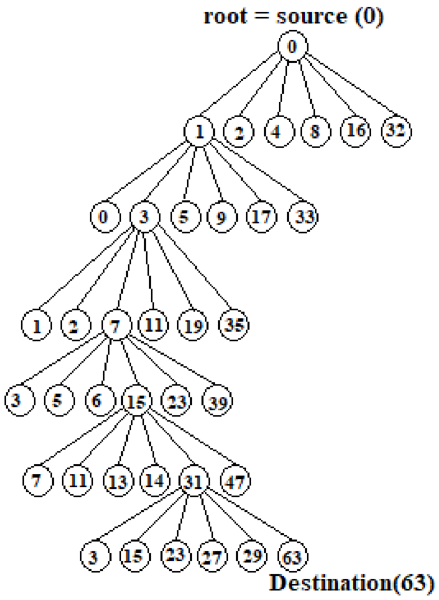

Fig.3(c) Source to destination length=6

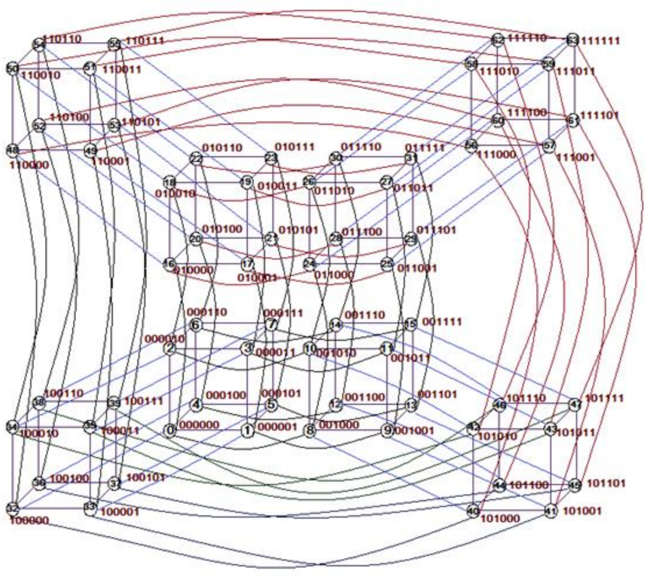

Fig. 4 a network connection of 2^6=64 nodes starting from 0 to 63

Fig.5 Pictorial view of 64 nodes based on gray code and different source to destination nodes with routing paths

For clear understanding, the communication path between two nodes of any set of two numbers (0, 1, 2, ..., 63), a pictorial view of nodes are arranged in 8×8 small squares in Fig. 5. If we want to communicate between any pair of two nodes, we can put the two numbers for the source and destination. The circuit shown in Fig. 2 will provide the communication path showing the nodes connecting the path. A 8×8 node presentation view is depicted in Fig. 5 and some routing paths for some sources and destinations are marked by arrow lines. For example, two numbers 50 and 27 we have chosen, where 50 is the source and 27 is the goal. As soon as we put these two numbers in the circuit in Fig.2, the outputs from the circuit provide us the numbers 50→51→59→27 with path length 3 and this path is shown in Fig. 5. Similarly, some pairs of nodes are taken arbitrarily and are set in the circuit in Fig.2 as inputs, we get their respective outputs, which are tabulated in the Table-2. According to node arrangement in Fig.3, the travelling direction from one point to the next will be either left or right or above or below, but never diagonally.

Table-2

Sl. No.	Input node	Output node	Routing path	Path length
1	9	33	9→1→33	2
2	16	20	16→20	1
3	15	51	15→11→3→19→51	4
4	48	34	48→34	1
5	16	46	16→20→22→30→14 →46	5
6	42	21	42→43→41→45→37→5 3→21	6
7	50	27	50→51→59→27	3
8	4	63	4→5→7→15→31→63	5
9	34	39	34→35→39	2
10	47	61	47→45→61	2

Collision free/Deadlock free arrangement

According to our algorithm maximum path length =6 when n=6 and the nodes concerned here is 6+1=7. For the worst case, a full-fledged tree having branch factor 6, will be $1+6+6^2+6^3+6^4+6^5+6^6$=55986 nodes where some of nodes are reused in our present work. But for the curtesy of the algorithm, only 2-6 nodes are sufficient to go from one node to another. This is the beauty that the algorithm ensures us a small number of nodes are necessary.

In Table-2 and Table-3, some input–output nodes and the corresponding paths are shown. Unlike Table-2 and Table-3, in Table-1 we have listed all the sequential node points involved in a routing path. If you go through the table, you must feel that no two or more node values are the same in a column. So, at a particular time there will be no chance of collision, therefore it would be a deadlock free arrangement in the network involved here.

Memory size

As the network we are considering has 64 nodes and every node address is of 6 bits. So storing this 6-bits in a memory system at least 1byte memory space is essential. For entire network the memory space required =64byte. Since, in our network, path length for a routing path varies from 1 to 6, the corresponding nodes required would be from 2 to 7. For storing these node addresses bit, the memory size needed will be 2byte to 7byte. In average it is 4.5byte≅5byte. At a time, for travelling a routing path, the memory size manipulated is 7 byte (in worst case). So the memory utilization is $\frac{7}{64} \times 100\%$ =10.94%, at most.

Delay and speed Calculation

The circuit drawn in Fig. 2 is made up of XOR gates only. Total number of XOR gates is 42. We know the delay for CMOS based XOR gate is 12ns [9, 10], for SET (single electron tunneling transistor) based circuit it is 4ns[8] and for linear threshold logic gate (TLG) based circuit is 2.21ns [5-7]. The communicating delay between two direct connected node points varies and depends upon the communicated materials used to connect the nodes in general. For this present work, we are assuming the direct connected path distance between any two points are identical and of length 100m for simplification.

Propagation Delay of copper wire: The understanding of signal propagation delay in copper media is essential. Electrical signals through copper wires travel at approximately 2/3 times the speed of light in vacuum. This is the propagation speed of the signal through the copper. As we are aware of that an Ethernet operates at a speed of 10,000,000 bits per second or 10Mbps, we can find out that the length of wire which is occupied by a single bit is approximately equal to 20 meters.

Speed of light in a vacuum =3$\times 10^8$=300,000,000 meters/ second

Speed of electricity in a copper cable = (2/3) × 300,000,000 meters/ second=200,000,000 meters/ second

Length required for a single bit ⟹ (200,000,000ms) / (10,000,000 bits/s) = 20 meter/bit

The delay for a single bit in copper wire is $=\frac{20 \text{ meter/bit}}{200{,}000{,}000 \text{ meters/ second}} = 10^{-7}$second/bit =100ns/bit

So, from the above calculation, to travel 20m for a single bit requires 100ns.

Relying upon these delays, a table for different path lengths including processing delay of node is given in Table-3. Assuming a node delay is 24ns for classical CMOS. While calculating the path delay we are to involve all the nodes in a path. Hence a path having length 1 contains 2 node, having length 2 contains 3nodes and so forth. So the delay of a path of two nodes is $24 \times 2 + 100 = 148$ns. In the same way, all the possible delays for different path lengths of 1, 2, 3, 4, 5 and 6 for a 2^6 node-network drawn in Fig. 6 are presented in Table-3 below. As the maximum delay is 768ns, so in the worst case, the speed of the network will be $f = \frac{1}{768\text{ns/bit}}$ =0.001302 $\times 10^9$ bit/sec =1.302Mb/sec.

Table-3

Sl. No.	Input node	Output node	Routing path	Path length	Delays (ns)
1	16	20	16→20	1	148
2	9	33	9→1→33	2	272
3	24	36	24→ 20 → 4 →36	3	396
4	15	51	15→11→3→19→51	4	520
5	16	46	16→20→22→30→14 →46	5	644
6	42	21	42→43→41→45→37→53 →21	6	768

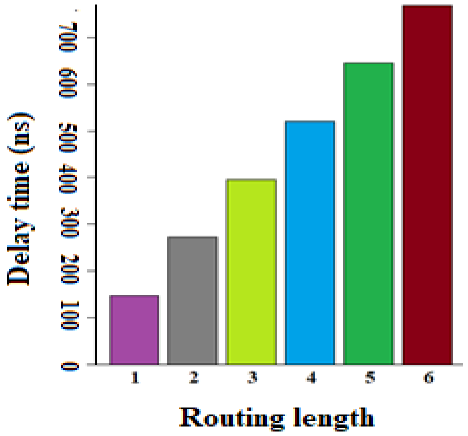

Fig.6 Delay time vs. routing path length

II. Conclusion

In the present work, we are concentrated in making a network having 2^n nodes in general. For the space limitation, the value of n=6 is chosen. And accordingly, we are concentrated in 64 nodes. Among these 64 nodes, any pair of separate two nodes at any time can be selected for data/packet transferring. These data/packet would reach the destination point without any trouble as we have investigated that in any column for the table-1, no two node points (of six bits) are identical. If the two such identical node points would have stayed in any column, there would have a collision or deadlock. For creating a proper sequence of nodes, a circuit is implemented that can provide the required nodes whenever a pair of input points (source, destination) are set to. A 8×8 grid is created to show the distinct communication paths using arrow lines. The memory size utilized for manipulating the nodes in a routing path is at most 10.94%. Time delays for different possible path lengths are calculated. Time complexity is proportional to the depth of the tree i.e. $O(l)$. Speed through the network will be 1.302Mb/sec.

III. REFERENCES

[1]. H. Sullivan and T.R. Bashkow,'A Large Scale Homogeneous, Fully Distributed Parallel Machine," Proc. 4th Symp. Computer Arch., vol.5 pp105-125. Mar, 1977

[2]. Xiaola Lin; P.K. McKinley; L.M. Ni, " Deadlock-free multicast wormhole routing in 2-D mesh multicomputers" IEEE Transactions on Parallel and Distributed Systems , Volume: 5, Issue: 8, Aug 1994 Page(s): 793 - 804

[3]. Chin-Heng Liu, Chia-Chun Lin et al, "Threshold Function Identification by Redundancy Removal and Comprehensive Weight Assignments" IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 38, NO. 12, DECEMBER 2019

[4]. Anup Kumar Biswas, "Routing path calculation of a 5-dimensional hypercube using linear threshold logic of an emerging Technology" International Journal of Research in Engineering and Science (IJRES) ISSN (Online): 2320-9364, ISSN (Print): 2320-9356 www.ijres.org Volume 10 Issue 1 ǁ 2022 ǁ PP. 15-33

[5]. Anup Kumar Biswas, "Threshold Logic Technology based E-cube Routing on a 4-dimensional hypercube", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 8 Issue 6, pp. 07-24, November-December 2021.

[6]. Anup Kumar Biswas, "A High-Speed Bidirectional Register with Parallel Loading using single electron Threshold Logic Technology" ,International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Vol. 8 (4) : pp394-408; July-August-2021, Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)

[7]. Anup Kumar Biswas, "Integrated-Circuit Random Access Memory based on an emerging Technology—electron tunneling through Tunnel Junction" , International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Vol. 8 (4) pp.409-424 July-August-2021, Print ISSN: 2395-1990 | Online ISSN : 2394-4099

[8]. A. K. Biswas and S. K. Sarkar: "An arithmetic logic unit of a computer based on single electron transport system" (SPQEO) Semiconductor

Physics, Quantum Electronics & Opt-Electronics. 2003. Vol 6. pp 91-96 No.1,

[9]. Jacob. Millman and C. C. Halkias; “Integrated Electronics- Analog and Digital Circuits and Systems -second edition” McGraw Hill Education;

[10]. Millman's Electronic Devices & Circuits 4th Edition (English, Paperback, Jacob Millman)

[11]. Kai Hwang & Naresh Jotwani, “Advanced Computer Architecture Parallelism, Scalability, Programmability” Second Edition, Mc Graw Hill.

BIOGRAPHY

ANUP KUMAR BISWAS, is an Assistant Professor in Computer Science and Engineering, Kalyani Govt. Engineering College,West Bengal, India. He defended his Ph.D.[Engg.] thesis in 2005 in the stream of Electronics and Telecommunication Endineering at Jadavpur University. He was a Senior Research Fellow(SRF) in Faculty of Engineering and Technology (*FET*) from 2002 to 2005 at the Department of Electronics &TC. Dr. Biswas has published 27 papers in national, international journals and conferences. The area of his scientific interests are computer arithmetic, parallel architectures, nanotechnology, and computer-aided design. He is engaged in research activities and teaching last 18 years.

Cite this article as :

Anup Kumar Biswas, "A Collision Free Network with Minimal Routing Path for 2n Nodes", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 9 Issue 1, pp. 55-63, January-February 2022. Available at doi : https://doi.org/10.32628/IJSRSET22919
Journal URL : https://ijsrset.com/IJSRSET22919