AU J.T. 6(4): 179-186 (Apr. 2003)

On the Oblivious Circuit Switching in Multi-Link Binary Hypercubes

Dobri Atanassov Batovski
Faculty of Science and Technology, Assumption University
Bangkok, Thailand

Abstract

This paper considers the necessary conditions for the existence of oblivious circuit switching in multi-link binary hypercubes. It is also shown that 1(2,3,4,7)-link edge-disjoint, many-one routing exists in the 1(2,3,4,5)-cube. Modular destination graphs are decomposed for the said number of links, where the physical paths are obtained using quasi-dimension-order (cyclic) routing. The basic requirements in obtaining many-one modular destination graphs are discussed. As a result of the combination of two many-one graphs of the same dimension, it directly follows that 1(2,3,4,7)-link, edge-disjoint permutation routing exists in the 2(4,6,8,10)-cube. Also, by combining two, many-one graphs of different dimensions, one can obtain (2,3,4,7)-link edge-disjoint permutation routing in the (3,5,7,9)-cube.

Keywords: *Circuit switching, oblivious routing, multi-link binary hypercube, modular destination graph, edge-disjoint paths, quasi-dimension-order routing, interconnection networks.*

Introduction

The binary hypercube, or just hypercube, is a subject of intensive research efforts since the publication of the first technical papers on the topic by Sullivan and Bashkow (1977) and Sullivan *et al.* (1977). Since that time, most of the studies in the hypercube topology have been concentrated mainly on the establishment of a conflict-free packet routing (Grammatikakis *et al*. 1998), due to the obstacles on the way of implementing circuit switching.

The rearrangeability of the hypercube in circuit switching mode of operation, as an establishment of edge-disjoint paths, for arbitrary permutations, has been proven by Szymanski (1989) for dimension three. As it has been shown by Lubiw (1990), that the five-dimensional hypercube is not rearrangeable if all paths are restricted to minimal lengths. The rearrangeability of the four-dimensional hypercube is to be verified in the future. In a more recent study by Gu and Tamaki (1997), it has been proven that the hypercube is 2-rearrangeable, and an arbitrary permutation can be routed by two sets of edge disjoint paths. However, it is important to notice that the two sets are different for different permutations, and are to be determined by a central controller during the initialization stage prior to data transmission.

The so-called *e*-cube algorithm (restricted Boolean *n*-cube, Lang (1982) was the basic routing technique being implemented in a number of hypercube hardware configurations due to its simplicity. This is an oblivious-like approach, because it establishes a deterministic dimension-order routing for arbitrary source-destination pairs. The *e*-cube algorithm can be used in establishing edge-disjoint paths for a limited number of permutations. Circuit switching for some basic permutations, has been studied theoretically by Veselovsky and Kupryanova (1991). Dimension scrambling has been used by Lin (1990) in obtaining edge-disjoint paths for the bit reversal permutation. A computational study on the applicability of the dimension-order routing to most common permutations has been performed by Veselovsky and Batovski (2003).

When dealing with arbitrary permutations, the use of a central controller for determining the routing paths becomes rather ineffective with the increase of the number of nodes for higher dimensions. An alternative approach is the use of an *oblivious routing,* for which deterministic physical paths are established between the source-destination pairs, and are to be followed for arbitrary permutations.

Oblivious routing has been well studied in the literature when applied to packet routing. In a paper by Hwang *et al.* (2002), modular destination graphs for 2(3,4,7,12)-step many-one routing in the 2(3,4,5,6)-cube, have been presented. As a result of the combination of two, many-one modular destination graphs of lower dimensions, it has been shown that there exist modular destination graphs for 7(8,11,14,19,24)-step permutation routing in the 7(8,9,10,11,12)-cube.

Similar results have been obtained by Batovski and Veselovsky (2002) for a number of dimensions less than, or equal to eight by using quasi-dimension-order (cyclic) routing. An 11-step graph for permutation routing in the 9-cube can also be obtained with the said technique.

A novel approach of using decomposition of modular destination graphs into several conflict-free independent sets in the multi-link hypercube has been presented by Batovski (2002a). It has been shown that there exist tight (*n*-step, where *n* is the dimension of the hypercube) many-one packet routing in the 2(2,3)-link 5(6,7)-cube, which directly follows to tight permutation packet routing in the 2(2,2,2,3,3)-link 9(10,11,12,13,14)-cube.

In many-one packet routing the first routing step is always conflict- free. This fact can be used to construct [1+*l*(*n*-1)]-step many-one modular destination graphs in the standard hypercube (*l* is the number of links (duplex connections) between two connected nodes in the multi-link hypercube), thus resulting in the existence of graphs for 11(19)-step many-one routing in the 6(7)-cube. The result in the 6-cube cannot be further improved and is better with a single routing step, in comparison with the work by Hwang *et al*. (2002). This follows to the existence of graphs for 18(22,30,38)-step permutation packet routing in the 11(12,13,14)-cube. The latest computational studies by Batovski (2003a) prove the existence of decomposed modular destination graphs for tight many-one packet routing in the 5-link 8-cube, thus resulting in the existence of tight permutation routing in the 5-link 15(16)-cube. This follows to the existence graphs for [1+5x(8-1)]-step many-one packet routing in the 8-cube, which results in 55(72)-step permutation packet routing in the 15(16)-cube.

The number of links in the multi-link hypercube can be increased for the purpose of conflict-free many-many routing. As it has been shown by Batovski (2002b), there exist graphs for tight 1(1,2,2,4,6,10)-link many-many packet routing in the 1(2,3,4,5,6,7)-cube. If the number of links is multiplied by the dimension of the hypercube, one can obtain the number of routing steps for many-many routing in the standard hypercube.

Table 1. Number of routing steps, *r*, *versus* number of dimensions, *n*, for which modular destination graphs for conflict-free packet routing have been found in the binary hypercube.

n	Number of routing steps, *r*		
	One-one packet routing	Many-one packet routing	Many-many packet routing
1	1	1	1
2	2	2	2
3	3	3	3x2=6
4	4	4	4x2=8
5	5	7	5x4=20
6	6	1+2x(6-1)=11	6x6=36
7	7	1+3x(7-1)=19	7x10=70
8	8	1+5x(8-1)=36	-
9	4+7=11	-	-
10	7+7=14	-	-
11	7+11=18	-	-
12	11+11=22	-	-
13	11+19=30	-	-
14	19+19=38	-	-
15	19+36=55	-	-
16	36+36=72	-	-

Table 2. Number of links, *l*, *versus* number of dimensions, *n*, for which modular destination graphs for conflict-free tight packet routing have been found in the multi-link binary hypercube.

n	Number of links, *l*		
	One-one packet routing	Many-one packet routing	Many-many packet routing
1	1	1	1
2	1	1	1
3	1	1	2
4	1	1	2
5	1	2	4
6	1	2	6
7	1	3	10
8	1	5	-
9	2	-	-
10	2	-	-
11	2	-	-
12	2	-	-
13	3	-	-
14	3	-	-
15	5	-	-
16	5	-	-

Tables 1 and 2 contain a summary of the known results for the existence of modular destination graphs for conflict-free packet routing. Table 1 contains a summary of the known results in the standard binary hypercube. Table 2 deals with the number of links for tight packet routing in the multi-link binary hypercube.

It should be mentioned that the results for both, many-one, and many-many packet routing in Table 2 are optimal and cannot be further improved. The optimality of the graphs is verified by comparizon with the number of links obtained from the necessity conditions that naturally follow from the link budget analysis. Many boxes in Table 2 are left empty for future verification, due to the increased computational complexity. Also, a more precise theoretical consideration of the necessity conditions should follow to an apparent increase of the number of required links for higher dimensions.

It is reasonable to expect that the level of link utilization for the case of circuit switching will increase significantly. The main difference with the case of packet routing, is that the edge conflicts cannot be resolved anymore by increasing the number of routing steps, because the connections between the neighbor nodes are reserved all the time for certain source-destination pairs. Therefore, the only solution in this topology is to increase the number of links, *l*, between the connected nodes.

Basic Notations

The *all-port model* will be used, where every node can be connected simultaneously to all of its neighbor nodes.

Decimal numbers are used in representing the bit positions. The least significant bit (LSB, decimal number 1) is the right-most one and the most significant bit (MSB, decimal number *n*) is the left-most one.

The *minimum-routing* algorithms perform bit replacements only for the non-identical bit positions between source and destination. Three modifications of the minimum-routing approach are shown in Tables 3, 4, and 5. Table 3 illustrates the deterministic dimension-order routing where the bit replacements are initiated from LSB. Table 4 is an example of a quasi-dimension-order (cyclic) routing where the bit replacements of the dimension-order routing are initiated from a bit position different than LSB, also called virtual LSB (VLSB), Batovski and Veselovsky (2002). Table 5 is based on dimension scrambling where the bit replacements correspond to a certain bit permutation.

A path can be uniquely represented by its source and destination addresses and the positions of bit replacements. Convenient notations for the physical paths have been introduced earlier by Batovski and Veselovsky (2002). For example, the path 31→0 in Table 3 can be viewed as:

31 (12345),

where the destination address 0 is omitted. The path in Table 4 is:

31 (34512),

and the path in Table 5 is:

31 (35142).

Table 3. Determination of the routing path for the source-destination pair 31→0 using dimension-order routing 31 (12345) starting from LSB (VLSB=1).

Edge	Node	Replaced Bit	Binary Representation
-	31	-	11111
31→30	30	1	11110
30→28	28	2	11100
28→24	24	3	11000
24→16	16	4	10000
16→0	0	5	00000

Table 4. Determination of the routing path for the source-destination pair 31→0 using quasi-dimension-order routing 31 (34512) starting the bit replacements from the third bit (VLSB = 3).

Edge	Node	Replaced Bit	Binary Representation
-	31	-	11111
31→27	27	3	11011
27→19	19	4	10011
19→3	3	5	00011
3→2	2	1	00010
2→0	0	2	00000

Table 5. Determination of the routing path for the source-destination pair 31→0 using dimension scrambling 31 (35142) starting the bit replacements from the third bit (VLSB = 3).

Edge	Node	Replaced Bit	Binary Representation
-	31	-	11111
31→27	27	3	11011
27→11	11	5	01011
11→10	10	1	01010
10→2	2	4	00010
2→0	0	2	00000

If d_{ij} is the Hamming distance of an arbitrary source and destination pair $i \rightarrow j$, the number of different physical paths for circuit switching equals the factorial of that distance:

$$p_{ij} = (d_{ij})! \quad (1)$$

With the increase of the Hamming distance, the number of possible physical paths increases drastically. When applying quasi-dimension-order routing, the number of physical paths decreases to:

$$p_{ij} = d_{ij} \quad (2)$$

The reduced number of physical paths allows one to perform fast computations in obtaining modular destination graphs.

Many-One Circuit Switching

In many-one routing, a source node can send data to only a single arbitrary destination. A destination node can receive data from many source nodes (from all source nodes in the worst case).

The modular destination graphs being used for many-one routing have some interesting properties. By definition, a modular destination graph contains all the paths having destination address zero. The paths to all other destination addresses are obtained by performing a modulo-2 addition between the node addresses of the modular path with the new destination address. For example, the path 31→0

31 (34512)

can also be viewed as:

31 → 27 → 19 → 3 → 2 → 0

It is obvious that the four secondary paths having lower Hamming distances 1, 2, 3, and 4, should follow the major path 31→0:

31 → 27
31 → 27 → 19
31 → 27 → 19 → 3
31 → 27 → 19 → 3 → 2

Performing a modulo-2 addition with the destination addresses of the secondary paths, one can obtain the paths that are a part of the modular destination graph:

4 → 0
12 → 8 → 0
28 → 24 → 16→ 0
29 → 25 → 17→ 1→ 0

These paths are directly derived from the major path 31→0 (34512) and their compact decimal representation is:

4 (3)
12 (34)
28 (345)
29 (3451)

Therefore, a modular destination graph can split into two sets consisting of major paths and secondary paths, correspondingly.

The oblivious circuit switching requires all the non-directional connections between the neighbor nodes to be reserved all the time, thus increasing the number of edge conflicts. By increasing the number of links l between two connected nodes, it is possible to decompose the graph into l sets to be routed independently through independent links. *Logically, the use of l independent binary hypercubes allows one to route l independent sets of the decomposed graph.*

It is preferable if the secondary paths belong to the same set with the major path. Also, two (or more) major paths can share the same set. However, the number of paths in a set is limited to only n paths because there are only n connections between a node and its neighbors for a given link (set). In general, the oblivious routing is a non-homogeneous process despite the homogeneous topology provided for this since there is only one major path having Hamming distance n. Here this path forms a separate set together with its secondary paths. For example, if n = 5, and one can choose the major path to be 31 (34512), then the set:

4 (3)
12 (34)
28 (345)
29 (3451)
31 (34512)

is complete and no additional paths can be added because all the possible unidirectional connections to node 0 (the last decimal number in the brackets): 1, 2, 3, 4, and 5 are already in use. This is also an illustration of the main difference with packet routing, where different connections can be used during different routing steps (hops) by different source-destination pairs.

Necessity Conditions

The required number of links, l, in the multi-link binary hypercube can be obtained from basic considerations. Keeping in mind that a total number of $2^n - 1$ non-directional connections must enter node 0 of the modular destination graph (excluding the identity path 0→0). And, that the number of connected nodes per link equals the dimension n of the hypercube, the following inequality holds for the number of required links, l,

$$l\, n \geq 2^n - 1 \qquad (3)$$

Table 6 contains the optimal number of links versus number of dimensions obtained from inequality (3) for a number of dimensions up to 20. As it can be observed from Table 6, the number of links increases rapidly with the dimension of the hypercube, due to the increased number of nodes in the interconnection network.

Table 6. Optimal number of links l *versus* number of dimensions n for which decomposed many-one modular destination graphs for edge-disjoint circuit switching are to be found.

N	*l*	*n*	*l*
1	1	11	187
2	2	12	342
3	3	13	631
4	4	14	1,171
5	7	15	2,185
6	11	16	4,096
7	19	17	7,711
8	32	18	14,564
9	57	19	27,595
10	103	20	52,429

Quasi-Dimension-Order Routing

The existence of decomposed modular destination graphs for the number of links shown in Table 6 is to be verified. The number of physical paths used for this verification can be initially reduced to the ones obtained with *quasi-dimension-order* (*cyclic*) routing. Starting from the assumption that such solutions might exist, it is a matter of a computational experiment to obtain the graphs using an advanced source code which has been constantly improved throughout the years, Batovski (2003b). It has been found that a graph decomposition takes place for a number of dimensions less than, or equal to five. Sample modular graphs for the said dimensions are presented in the Appendix.

The existence of many-one circuit switching follows to the existence of permutation circuit switching at higher dimensions. The combination of two, many-one graphs, the first one is used for forward routing (from the past to the future) and the second is used in the reverse direction (from the future to the past), forming a decomposed graph for permutation circuit switching as shown in Table 7.

Table 7. Number of links, *l*, *versus* number of dimensions, *n*, for which modular destination graphs for edge-disjoint circuit swithcing have been found in the multi-link binary hypercube.
.

Dimension, *n*	Number of links, *l*	
	One-one Circuit Switching	Many-one Circuit Switching
1	1	1
2	1	2
3	2	3
4	2	4
5	3	7
6	3	12 (or 11?)
7	4	-
8	4	-
9	7	-
10	7	-
11	12	-
12	12	-

Table 8. Number of unused duplex connections per node, *versus* number of dimensions *n* for the decomposed modular destination graphs presented in the Appendix.

n	Number of unused connections		Unused duplex connections per node
	Theoretical Limit	Graph Results	
1	0	0	
2	1	1	Link 2: 0 ↔ 1
3	2	2	Link 2: 0 ↔ 1 Link 3: 0 ↔ 2
4	1	1	Link 2: 0 ↔ 1
5	3	0	-

For odd dimensions *n* of the obtained permutation graphs, the number of links *l* equals the higher number of links of the two, many-one graphs used for its construction. Therefore, for odd dimensions, the number of unused duplex connections will increase.

Considering the unused connections in many-one graphs, one can observe that expression (3) does not hold with an equality sign, and there is always a certain number of connections that should be left unused. Table 8 compares the theoretical limit from inequality (3), with the actually obtained unused links from the graphs in the Appendix. For dimension five, all the connections are fully utilized. By using some elements of *dimension scrambling*, it is possible to design alternative graphs, for which the number of unused connections increases. Table 9 contains a sample decomposed graph for dimension five, where the duplex connection 0↔4 of link 6 is left unused.

In summary, the use of quasi-dimension-order routing follows the fast decrease of the number of edge conflicts. The computational procedure becomes more effective by initially applying this form of cyclic bit replacements. If in certain cases, the number of edge conflicts cannot be reduced to zero, dimension scrambling can be applied to the initially optimized graph structure. Dimension scrambling can also be applied if it is necessary to satisfy some additional link constraints as mentioned above.

Table 9. A decomposed modular destination graph for many-one circuit switching in the seven-link five-dimensional binary hypercube obtained with dimension scrambling.

Link	Source (positions of bit replacements)
1	1 (1), 3 (12), 7 (123), 15 (1234), 31 (12345)
2	2 (2), 17 (15), 21 (153), 29 (1534)
3	4 (3), 6 (32), 12 (34), 13 (341), 28 (345)
4	8 (4), 10 (42), 14 (423), 26 (425), 27 (4251)
5	16 (5), 18 (52), 19 (521), 23 (5213), 24 (54)
6	9 (14), 11 (142), 25 (145)
7	5 (31), 20 (35), 22 (352), 30 (3524)

Conclusion

The use of oblivious circuit swithing in interconnection networks is quite important for the design of ultra-fast supercomputers. Since there is a well developed and constantly improving crossbar technology for electrical networks, the possible application of decomposed multi-link hypercubes can be found in the future optical parallel systems, where after the relatively slow electro-optical initialization of the physical paths, the data can travel with the speed of light. Detailed information about the the known results for both packet rotuting and circuit switching in both standard (single-link) and multi-link hypercubes is provided in a series of tables.

Acknowledgements

The Monfort Brothers of St. Gabriel, Assumption University, are gratefully appreciated.

Appendix

The appendix contains sample decompositions of modular destination graphs for many-one circuit switching for a number of dimensions $n \leq 5$. Quasi-dimension-order (cyclic) routing has been used in obtaining the edge-disjoint paths.

Table A 1. A decomposed modular destination graph for many-one circuit switching in the one-dimensional binary hypercube.

Link	Source (positions of bit replacements)
1	1 (0)

Table A 2. A decomposed modular destination graph for many-one circuit switching in the two-link two-dimensional binary hypercube.

Link	Source (positions of bit replacements)
1	1 (10), 3 (12)
2	2 (20)

Table A 3. A decomposed modular destination graph for many-one circuit switching in the three-link three-dimensional binary hypercube.

Link	Source (positions of bit replacements)
1	1 (1), 3 (12), 7 (123)
2	2 (2), 6 (23)
3	4 (3), 5 (31)

Table A 4. A decomposed modular destination graph for many-one circuit switching in the four-link four-dimensional binary hypercube.

Link	Source (positions of bit replacements)
1	1 (1), 3 (12), 7 (123), 15 (1234)
2	2 (2), 6 (23), 10 (24)
3	4 (3), 5 (31), 12 (34), 14 (342)
4	8 (4), 9 (41), 11 (412), 13 (413)

Table A.5. A decomposed modular destination graph for many-one circuit switching in the seven-link five-dimensional binary hypercube.

Link	Source (positions of bit replacements)
1	1 (1), 3 (12), 7 (123), 15 (1234), 31 (12345)
2	2 (2), 5 (13), 13 (134), 18 (25)
3	4 (3), 9 (41), 11 (412), 20 (35)
4	8 (4), 24 (45), 25 (451), 26 (452), 30 (4523)
5	16 (5), 17 (51), 19 (512), 21 (513), 27 (5124)
6	12 (34), 14 (342), 28 (345), 29 (3451)
7	6 (23), 10 (24), 22 (235), 23 (2351)

References

Batovski, D.A. 2002a. On the Invariant Many-one Packet Routing in Multi-link Binary Hypercubes. Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'02), Las Vegas, Nevada, USA, 24-27 June 2002, Vol. IV, pp. 1732-8, CSREA Press, Athens, Georgia, USA.

Batovski, D.A. 2002b. Decomposition of Modular Destination Graphs for Many-Many Packet Routing in Multi-Link Binary Hypercubes. Proc. 8th IEEE Int. Conf. on Communication Systems (ICCS 2002), Singapore, 25-28 November 2002, paper 3P-05-01, pp. 1207-11, IEEE, Piscataway, New Jersey, USA.

Batovski, D.A.; and Veselovsky, G. 2002. An Invariant Oblivious Minimum-routing Algorithm for Binary Hypercubes. Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'02), Las Vegas, Nevada, USA, 24-27 June 2002, Vol. IV, pp. 1610-6, CSREA Press, Athens, Georgia, USA.

Batovski, D.A. 2003a. A Modular Destination Graph for Many-One Packet Routing in the Five-link Eight-dimensional Binary Hypercube. Unpublished.

Batovski, D.A. 2003b. Source code Hypercube.cpp. Unpublished.

Grammatikakis, M.D.; Hsu, D.F.; and Sibeyn, J.F. 1998. Packet routing in fixed-connection networks: A survey. J. Par. Dist. Comp. *54:* 77-132.

Gu, Q.P.; and Tamaki, H. 1997. Routing a permutation in the hypercube by two sets of edge disjoint paths. J. Par. Dist. Comp. *44: 147-52.*

Hwang, F.K.; Yao, Y.C.; and Dasgupta, B. 2002. Some permutation routing Algorithms for low-cubes. Theor. Comp. Sci. 270: 111-24.

Lang, C.R., Jr. 1982. The Extension of Object-Oriented Languages to a Homogeneous, Concurrent Architecture. Ph.D. Thesis, Computer Science Department, California Institute of Technology, Pasadena, California, USA, Technical Report 5014:TR:82, 24 May 1982.

Lin, W. 1990. A Dimension-scrambling Approach to Fast Hypercube Data Permutations. Proc. Int. Conf. on Parallel Processing, Pasadena, California, USA, Vol. III: Algorithms and Applications (ICPP90-3), 13-17 August 1990, pp. 119-22, Pennsylvania State Univ. Press, Pasadena, California, USA.

Lubiw, A. 1990. Counter example to a conjecture of Szymanski on hypercube routing. Inf. Proc. Lett. 35(2): 57-61.

Sullivan, H.; and Bashkow, T.R. 1977. A Large Scale, Homogeneous, Fully Distributed Parallel Machine, I. Proc. 4th Ann. Symp. on Computer Architecture, March, Vol. V, pp. 105-17, ACM Press, New York, New York, USA.

Sullivan, H.; Bashkow, T.R.; and Klappholz, D. 1977. A Large Scale, Homogeneous, Fully Distributed Parallel Machine, II. Proc. 4th Ann. Symp. on Computer Architecture, March, Vol. V, pp. 118-24, ACM Press, New York, New York, USA.

Szymanski, T. 1989. *On the Permutation Capability of a Circuit-switched Hypercube. Proc.* Int. Conf. on Parallel Processing, Pasadena, California, USA, Vol. I: Architecture (ICPP89-1), pp. 103-110, Pennsylvania State Univ. Press, Pasadena, California, USA.

Veselovsky, G.G.; and Kupryanova, M.V. 1991. Comparison of some combinatorial properties of direct and indirect binary n-cube interconnection networks. Proc. ITC Specialist Seminar, Telecommunications Services for Developing Economies, Cracow, Poland, 22-27 April 1991, pp. 461-72, Elsevier, Amsterdam, The Netherlands.

Veselovsky, G.; and Batovski, D.A. 2003. A Study of the Permutation Capability of a Binary Hypercube under Deterministic Dimension-Order Routing. Proc. 11th Euromicro Conf. on Parallel Distributed and Network Based Processing, Genova, Italy, 5-7 February 2003, pp. 173-7, IEEE Computer Society, Los Alamitos, California, USA.