ALGEBRAIC APPROACH TO TRANSFORMATIONS ON HYPERCUBE SYSTEM

Ş.Ş.KAHRAMANLI and N.M.ALLAHVERDİ

Section of Computer Programming, Teknik Bilimler Meslek Yuksekokulu, Selçuk University, 42031, Konya, Turkey

and

Department of Computers and Systems, Azerbaijan Technical University, 370622,

Baku, Azerbaijan

Abstract: In this study some algebraic transformations on hypercube system are proposed. The presented transformations of forbidden subcubes provide defining set of all maximal and nonfaulty subcubes, and subset of minimal and nonfaulty subcubes included certain given vertices. It provides formal and simple findings of all the current sources and targets of information, and also the pathways between them. The correctness of all these transformations is proved by the aid of three theorems. In the proposed transformations there were three special operations of cube algebra applied, as subtraction, product and intersection on coordinates.

Index: Hypercube, fault tolerance, cube algebra, nonfaulty and maximal subcubes.

1. INTRODUCTION

Hypercube is a distributed parallel system consisting of 2^n identical processors, each provided with its own sizable memory and interconnected with n neighbors. It has a homogeneous symmetric structure and has necessarily rich connectivity. It also has useful topology in which many other topologies, such as meshes, rings, trees and etc. can be embedded [13, 2, 3, 6, 11, 15, 1]. Hypercube multiprocessor systems usually have a large number of processors (nodes), so the probability that some processor fails can be high. In order to prevent and determine the faulty nodes and links in the data communication there are many different kind of methods to reach the shortest paths between the source and the target nodes. For solving this problem it is applied traditional mathematical theory or heuristics, which can not completely take into account the peculiarities of the hypercube structure. These methods depend on the number of faulty components of hypercube and can not achieve necessary reliability when the number of faulty elements exceeds of cube dimension [2, 3, 11, 15]. The last years for this purpose it has become used the operations of cube algebra, which completely take into account the properties of hypercube structure and make possible to operate in hypercube in presence of a lot of faulty elements, exceeded the cube dimension.

Earlier, the cube algebra was applied in connection with presentation the Boolean functions in form of cubes and its minimization [9, 12, 14, 4]. These and others studies were considered with more detail and generalized in [10, 5]. Latter the operations of cube algebra was used for processor allocation in hypercube [6]. We also used the cube algebra for compact minimization of multiple and single output combination circuits [7, 8].

In this study we attempted for generalization and development of the methods and algorithms, based on cube algebra, with the aim of exactly and unambiguously solving the problem of reliable sequential and parallel routing for data communication in presence of faulty elements in a system of hypercube type with arbitrary dimension. Some of proposed transformations in this study are also useful for finding minimal forms of Boolean (switching) functions. Since these transformations allow to solve the problem of minimization with minimal possible and pre-known number of intermediate results, that is very important in the automatization of minimization process of Boolean functions with many variables.

2. ON CUBE ALGEBRA AND ITS OPERATIONS

Each vertex of *n*-dimensional cube is marked by *n* bit unique binary code. This cube has the coordinates $k_1, k_2, ..., k_n$. Naturally, the value of coordinate k_i is determined in the set $\{0,1\}$ for all i=1,2,...,n-1,n. Thus, the code of a certain vertex at the same time also became the algebraic expression of this vertex. In such coding of the vertex the neighbor vertices of *n*-dimensional cube are marked by *n* bit neighbor codes. Remind that, two *n* bit codes are called the *neighbors* if and only if they differ by the value of one bit. For example, the codes 0110 and 0100 are neighbors.

3-dimensional cube as presented in Fig.1, has coordinate axes k_1 , k_2 , k_3 and $\tilde{k}_1, \tilde{k}_2, \tilde{k}_3 \in \{0,1\}$, where \tilde{k}_i is the value on the coordinate axis k_i . As can be seen from the Fig.1 for the exact determination of any vertex of cube, it is necessary, that all the bits of this vertex's code accept the values from the set $\{0,1\}$. In other words, to set any vertex,

it is necessary, that presenting its code has exact value of all its coordinates without exception. But for the exact definition of a certain edge of cube it is sufficient to set the exact values of two respective coordinates. Since for indication of a line (edge), that unites the vertices 000 and 001, it is sufficiently to give the values of coordinates $\tilde{k}_1 = 0$, $\tilde{k}_2 = 0$, i.e. $\tilde{k}_1 \tilde{k}_2 = 00$; for line, that unites the vertices 000 and 010, it is sufficient to know the values of coordinates $\tilde{k}_1 = 0$, $\tilde{k}_3 = 0$, i.e. $\tilde{k}_1 \tilde{k}_3 = 00$; for edge, that unites the vertices 000 and 100, it is sufficient to set values of coordinates $\tilde{k}_2=0$, $\tilde{k}_3=0$, i.e. $\tilde{k}_2 \tilde{k}_3=00$. In the first of these examples the coordinate k_3 is not obviously indicated, in the second - k_2 , and in the third k_1 , since they are nonessential for giving the respective line. But, as a matter a fact, a nonessential coordinate is very essential in line presentation, since the respective line lies just along this coordinate axis. That is also valid for planes and all figures with more dimensions without exception. This circumstance sharply differs the interpretation of the algebraic expressions of figures (vertex, line, plane, 3-dimensional cube, etc.) applying to routing in hypercube from the interpretation of these expressions applying to minimization of the Boolean functions. Here nonessentiality of a certain coordinate is grasped as nonessentiality of respective variable.

To execute the algebraic operations on the elements of cube, it is necessary that they were presented in the form of algebraic expressions. Then, to avoid ambiguities, it is necessary to use the means, which provide a precise detachment of all coordinates and their values. For this purpose a certain symbol is placed to the position of nonessential coordinate, that does not belong to the set $\{0,1\}$. In many cases this symbol is marked by * and is called "don't care symbol". For example, the lines $\tilde{k_1}\tilde{k_2}=00$, $\tilde{k_1}\tilde{k_3}=00$ and $\tilde{k_2}\tilde{k_3}=00$ will have the expressions 00*, 0*0 and *00, respectively. In such notation, the value of the coordinate k_i stands on *i*th position of the expressions the properties of vector. In connection with this, for generality, the algebraic expression of whatever figure will call the vector of values of the coordinates (VVC). Then $k_i \in \{0,1,*\}$ for all i=1,2,..., n.

VVC, including the values of m don't care coordinate (the coordinates having the values *), expresses the figure that is called *m*-cube [10]. We will call the value m as size. In accordance to this definition a vertex is called 0-cube, since its VVC includes only zeros and ones. A line (the edge of cube) is called 1-cube, since its VVC includes the values of one don't care and n-1 care coordinates. A plane (quadrangle) is called 2-cube, since its

VVC includes the values of two don't care and n-2 care coordinates. 3-dimensional cube is called 3-*cube*, since its VVC includes the values of three don't care and n-3 care coordinates. Further it follows for the figures that have not geometrical analogues and, therefore, are called 4-*cube*, 5-*cube*, 6-*cube* and etc.

It is necessary to note, that the concepts of *m*-cube and *m*-dimensional cube are not always equal. If all the theoretical possible vertices and edges of *m*-dimensional cube enter to the same class of equivalency (according to a certain sign), then *m*-dimensional cube at the same time is *m*-cube. Otherwise, *m*-dimensional cube splits up into series of *l*-cubes (l < m), which have dimension *m* and can not form *m*-cube. On the other hand, *m*-cube can be a subcube of a certain *n*-cube, where n > m, and can have dimension *n*. For example, in Fig.1 all the vertices and edges of the 3-dimensional cube enter to one class of equivalency according to sign of nonfaulty vertices (exactly, placed on these vertices active objects) and communications among them. Therefore, this 3-dimensional cube is a 3-cube. If, for example, the vertices 111 and 110 and edges 10* and *00 are faulty, then this cube splits up into 2-cube 0** (plane - quadrangle), 1-cube *01 (line-edge) and 0-cube 100 (point vertex) (Fig. 2). But all these cubes have dimension 3 since they are not independent figures and are the parts of bigger figure, i.e.3-dimensional cube (Fig.1).

From above mentioned and the Fig.2 it is distinctly clear that the second difference of the application of cube algebra from routing point of view in hypercube and from Boolean (switching) functions point of view. In the minimization theory of SFs the relation of equivalency is spread only to vertices, since the lines between the vertices have no significance and are used only for visualization of neighborhoods between the vertices. Therefore, from point of minimization view, Fig.2 has aspect as shown on Fig.3, i.e. instead of the cubes 0**, *01 and 100 there are obtained 2-cubes 0** and *0*.



Fig.1

Fig.2

Fig.3

Hence it follows that the use of the cube algebra for solving the problem of routing (also broadcasting) in hypercube has its distinct properties by comparison with application of this algebra for minimization of SFs.

The cube algebra includes almost all operations of the set theory and three own operations, that are called *coordinate product (star product, consensus or \$\pi-operation)*, coordinate subtraction (sharp product or #-operation) and coordinate intersection (\frown operation). Notice, that from cube algebra point of view these own operations are fundamentals, but they are not fundamentals from logical algebra point of view, since each of these operations represents a certain operator, realizing a certain standard logical expression, which is transformed with the purpose of obtaining results of definitive type. For example, the results of multiplication, adhere, intersection and so on. Thus, the own operations of cube algebra are convenient tools for transformations on hypercube and, therefore, we they consider briefly.

Following [10, 5, 4], the coordinate subtraction of two cubes C_1 and C_2 , denoted $C_3 = C_1 \# C_2$, is the set of subcubes including the nodes from C_1 , but not from C_2 . For example, **0 # *11 =**0 or {**0, 1**}# 110 = {0*0, *00, 10*, 1*1}. The coordinate product of two cubes C_1 and C_2 , denoted $C_3 = C_1 \not\approx C_2$, is the subcube founded in C_1 and C_2 simultaneously or the subcube one part of which is in C_1 and the other is in C_2 . For example, *00 $\not\approx$ 1*1 =10*. The coordinate intersection of two cubes C_1 and C_2 , denoted $C_3 = C_1 \cap C_2$, is the subcube presented in C_1 and in C_2 at the same time. For example, 10* $\cap 1*1 = \emptyset$ or $10* \cap 1*1 = 101$.

The coordinate subtraction operation was applied to finding of the local prime implicants of the SFs. It was later known that, this operation is a good tool for definition of complete nonfaulty subcubes of hypercube with faulty or prohibition vertices, also for definition of the paths, including the shortest path from any vertex to other vertex in hypercube [1]. By means of the coordinate product the following operations may be realized: 1) Definition of the Hamming distance between the cubes, which is the number of links between them along the shortest path; 2) Intersection of two or more cubes with the aim of definition of their common parts (subcubes); 3) Unification of two *m*-cubes A and B in the (m+1)-cube C. The intersection operation is very useful for determination of the common parts of cubes. At first it was applied to definition of the parts of some prime implicant of SFs which are common with other prime implicants. This allowed to answer

the question: is the given prime implicant extreme? From the point of routing in hypercube the coordinate intersection operation can be used for revelation of the common parts of cubes with the aim of defining the parts of passage from one subcube to another.

3. TRANSFORMATIONS OF CUBE EXPRESSIONS

The coordinate product operation is applied to receive complete set of the prime implicants of SFs [10]. Applying to hypercube on vertices of which there are the processors, computers or other similar active objects, the prime implicant term can be interpreted as maximal subcube including only nonfaulty or nonforbidden vertices. In accordance to definition of prime implicant, subcube A of hypercube Q is maximal, if it is not a subcube of any more subcube of hypercube Q. The definition of complete set of nonfaulty and maximal subcubes of hypercube with faulty vertices is necessary for effective solution of such questions as allocation of the tasks over subcubes of hypercube [6], as routing of the information over vertices of hypercube [1] and etc.

The theorems given below allow to determine the method of finding the set of nonfaulty and maximal subcubes (NMS) in faulty hypercube [1].

Theorem 1. If F is a set of faulty cubes of complete *n*-cube and if $z_i \in \{0,1\}$ is the value of *i*th coordinate of any nonfaulty and maximal subcube Z of this *n*-cube, then at least one faulty subcube $f \in F$ exists in which the value of *i*th coordinate is $\overline{z_i}$.

Theorem 2. If F is a set of complete faulty cubes, then the set of NMS of *n*-cube including F is determined by the #-subtracting the set F from the complete *n*-cube and avoiding nonmaximal cubes from the result.

We apply the Theorem 2, as an example, for a set faulty nodes $F=\{0100,0111, 1010, 1011, 1100, 1101, 1110\}$. It is necessary to find a set of NMS of 4-cube. For simplicity of further transformations we interpret the nodes of the set F as the nodes for which the Boolean functions F are true. Thus, we can find a minimal form of the function F. Hence, $F_{min} = \{*100, 110*, 0111, 101*, 1*10\}$. Now we subtract the set F_{min} from the complete 4-cube.

 $K = **** \# F_{min} = **** \# \{*100, 110*, 0111, 101*, 1*10\} = \{00**, *00*, 1111, 0*10, 0*01\}$. The set K, that is obtained as a result of the #-subtraction from complete 4-cube all elements of a set of F_{min} is a set of NMS of faulty 4-cube which has the set of faulty nodes F.

If the cube $c_i \in C$ is formed in the result of subtraction operation of the cube *B* from the cube *A* on coordinate *i* ($i \in \{1,2,...,n\}$, then the cube c_i is called *difference* cube on coordinate *i*. For example, if $**0#110=\{0*0,*00\}$, then 0*0 and *00 are difference cubes on coordinate 1 and coordinate 2, respectively. In this example there is not the difference cube on coordinate 3.

After determination of a set of NMS we can realize different operations over the cubes which are contained in the set of NMS. Determination together of the cubes, which are contained in the set of NMS, can lead to difficulties from parallel routing of view in hypercube. Consequently, it is important to search the method for determination of NMS, each of which includes a current source in every stage of routing process. This method is based on the Theorem 3.

Theorem 3. If $A = a_1 a_2 \dots a_j \dots a_n$ is any nonfaulty node and $B_i = b_1^i b_2^i \dots b_j^i \dots b_n^i$, $B_i \in F$ is any faulty cube, then for definition of the NMS including node A by procedure $K_i = K_{i-1} \neq B_i$, $i = 1, 2, \dots, m$, $K_0 = ** \dots *$, it is necessary and sufficiently preserved for only all such

 b_i^i , which have opposite value with respect to corresponding a_i .

Proof: Assume that $a_j = b_j^i$ for any coordinate *j*. Then $* \# b_j^i = \overline{a}_j$. Consequently, for $a_j = b_j^i$ the difference cube formed on the base of pair (a_j, b_j^i) , contains the coordinate \overline{a}_j which shows that this cube does not contain the vertex *A*. In this case for averting the formation of nondesirable cube, it is necessary to substitute of the value b_j^i by the symbol *. For $a_j \neq *$ and $b_j^i = *$ the difference cube on coordinate *j* is not formed. The version $a_j = *$ and $b_j^i \neq *$ is not possible, since *A* is a vertex, i.e. *A* is 0-cube. For $a_j = b_j^i \neq *, * \#$ $b_j^i = \overline{b}_j^i = a_j$. Therefore, in the case $a_j = b_j^i$ the difference cube, formed on coordinate *j*, contains the value \overline{a}_j which witnesses that this cube can not contain the vertex *A*. Consequently, for averting the formation of this cube it is necessary to substitute the value b_j^i by the symbol *. If $a_j = \overline{b}_j^i$, then $* \# b_j^i = \overline{b}_j^i = a_j$. Therefore, in the case of $a_j = \overline{b}_j^i$ the difference cube, it is necessary to substitute the value b_j^i by the symbol *. If $a_j = \overline{b}_j^i$, then $* \# b_j^i = \overline{b}_j^i = a_j$. Therefore, in the case of $a_j = \overline{b}_j^i$ the difference cube, formed on coordinate *j*, the difference cube, formed on coordinate *j*.

• Using the Theorem 3 in the search of subcube, including the node A, the faulty cube B_i can be transformed to cube $Q = q_1^i q_2^i \dots q_n^i$ by the means of rule:

If
$$b_j^i = *$$
 or $a_j = b_j^i$, then $q_j^i = *$
If $b_j^i = \overline{a}_j$, then $q_j^i = b_j^i$

$$(1)$$

For example, assume that A=1101, $F=\{B_1, B_2, B_3\}=\{*100, 101*, 0111\}$. According to the transformation (1) $Q_1=***0$, $Q_2=*01*$ and $Q_3=0*1*$. According to the Theorem 3 the set I(A) of subcubes including the node A is defined as I(A)=**** # $\{Q_1, Q_2, Q_3\} = \{11*1, **01\}$, that means the node A=1101 is contained in the subcube 11*1 and **01.

Splitting the cubes I(A) into the set of 1-cubes one may obtain the set of paths P(A) which are incidence to node A: $P(A) = \{11*1, 1*01, *101\}$. If some paths which are incidence to node A are faulty, then firstly it is necessary to remove faulty paths from the set P(A) and secondly to define the set of neighboring nodes V(A). Assume that the path 1*01 is faulty. Then: $P'(A) = P(A) \setminus 1*01 = \{11*1, *101\}$; $V(A) = P'(A) \# A = \{1111, 0101\}$.

As can be seen from the abovmentioned transformations, each node can independently define its achieving neighbors. The procedure allows to find all the paths existing between the nodes and is independence from number of faulty elements and their placed configuration in hypercube. If many paths exist, it makes possible to find the shortest path among them.

4. CONCLUSION

In the study some transformations are presented, allowing to realize the procedures, that is very important in hypercube multiprocessors (routing, multicasting, broadcasting, subcube and task allocation and deallocation, etc.). By aid this approach the data communication in hypercube is carried out between nonfaulty subcubes and it causes no cases to determine by-passing paths in presence of faulty components. The formal approach allows to operate with fault-free components and so, is independent from the number and the scattering of faulty elements in hypercube. This is the advantage of the proposed approach and with this the approach differs from the other respective methods. By the aid of Theorems 1 and 2 we can obtain a set of nonfaulty and nonmaximal subcubes. The Theorem 3 allows to define a path between the source and destination nodes, using only the nonfaulty cubes taking from the source node to destination node. Thus, these algebraic transformations allow to increase a reliability of hypercube multiprocessors and by the aid of these transformations one can obtain a set of fault-free subcubes, that is a beginning set for further manipulation in such systems.

REFERENCES

- [1] Allahverdi, N.M., and Kahramanlı, Ş.Ş. Fault Tolerant Routing Algorithm in Hypercube with Application Cube Algebra, Submitted to J. Fac. Engineering and Architecture, Selçuk University, 1996, 21p.
- [2] Chang, Y, and Bhuyan, L. Subcube Fault Tolerance in Hypercube Multiprocessors, IEEE Trans. Comput., IEEE Trans. Comput. 44, (9), (Sept., 1995), 1108-1120.
- [3] Chiu, G.-M., and Wu, S.-P. A Fault-Tolerant Routing Strategy in Hypercube Multicomputers, *IEEE Trans. Comput.* 45,(2) (Febr. 1996), 143-155.
- [4] Dagenais, M.R., Agrawal, V.K., and Rumin, N.C. McBOOLE: A New Procedure for Exact Minimization, IEEE Trans. Comput.- Aided Design, Vol.CAD-5 (Jan. 1986), 229-237.
- [5] Dietmeyer, D.L. Logic Design of Digital Systems, Boston, MA: Allyn and Bacon, 1979.
- [6] Dutt, S., and Hayes, J.P. Subcube Allocation in Hypercube Computers, IEEE Trans. Comput. 40 (3) (March 1991), 341-352.
- [7] Kahramanh, Ş.Ş., and Allahverdi, N.M. Compact Method of Minimization of Boolean Functions with Multiple Variables, *Proc.Intern.Symp.of Comput. Applications*, Selçuk University, Konya, Turkey (June 1993), 433-440.
- [8] Nadjafov, E.M., and Kahramanov, Ş.Ş. On the Synthesis of Multiple Output Switching Schemes, Scientific Notes of Azerbaijan Institute of Petroleum and Chemistry, Baku, Azerbaijan, Vol. IX, (3) (1973), 65-69.

- [9] Lee, C.Y. Switching Functions on the N-dimensional Cube, Trans. of the AIEE, Vol.73, Part 1 (1954), 289-291.
- [10] Miller, R.E. Switching Theory, Vol.1, Combination Circuits, New York; John Wiley and Sons, 1965.
- [11] Raghavendra, C.S., Yang,, P.-J., and Tien S.-B. Free Dimensions An Effective Approach to Achieving Fault Tolerance in Hypercube, *IEEE Trans. Comput.* 44, (9), (Sept. 1995), 1152-1156.
- [12] Roth, J.P. Algebraic Topological Methods for the Synthesis of Switching Systems in *n*-variables, *The Institute for Advanced Study*, Princeton, New Jersey, ESP56-02 (April 1956).
- [13] Saad, Y., and Shultz, M.H. Topological Properties of Hypercubes, IEEE Trans. Comput., 37 (July, 1988), 867-872.
- [14] Urbano, R.H., and Mueller, R.K. A Topological Method for the Determination of the Minimal Forms of the Boolean Function. IRE Trans. Electronic Comput., EC-6, 4 (Sept. 1956),126-132.
- [15] Wu, J. An Optimal Fault-Tolerant Nonreduntant Broadcasting Scheme in Injured Hypercubes, J.Paral. and Distrib. Computers, 22, (2), (1995).