PROCEDURE FOR DEFINITION OF NEIGHBORHOOD IN FAULTY HYPERCUBE MULTIPROCESSOR

Novruz ALLAHVERDI

Selçuk University, Technical Education Faculty, Electronical and Computer Education
 Department, Campus, 42031, Konya/TURKEY

Abstract: A procedure for definition of neighborhood and fault free subcubes in a faulty hypercube is developed in this study. The cube algebra operations are used for proposed procedure. With this, we can increase the number of faulty elements in the hypercube, which is obstacle for effective data processing and communication processes in such a system.

1. INTRODUCTION

A hypercube multiprocessor consists of 2^n processors, each provided with its own memory and interconnected with n neighbors [1, 2]. In hypercube each node contains two units, namely, a computational part and a communication part that can operate independently. As the size of a hypercube grows, the probability of some processors or links in the system failing increases. One of the important issues in these systems is how to communicate messages in the presence of component failures. Reliable data communication is essential, especially when hypercubes are used for safety-critical and time-critical applications required high reliability [3]. It is very important to detect and isolate faulty processors to ensure correct completion of computation in such applications. In order to determine and avoid the faulty components and to find the shortest path between the source and the target nodes there are many different kind of methods [3-11]. But most of these methods depend on the number and the scattering of the faulty components in hypercube.

The procedure developed in this study allows us to calculate firstly a configuration that has only non-faulty components. Secondly we search the subcubes, covered the source and target vertices (nodes). Thirdly we look for intersection or bridge between the fault free subcubes, with the aim of definition of their common parts. Thus, we have a possibility to define a path between the source and destination nodes, using only the non-faulty cubes (subcubes). It is clear that the source node and the target node must not be placed in isolated subcubes. The procedure does not depend on the number of faulty components and is not affected by the scattering of these components in the hypercube.

To develop this procedure we use the operations of cube algebra, considered briefly in Section 2. In Section 3 we develop the procedure to define a neighborhood between fault free subcubes. Conclusion notes are described in Section 4.

2. OPERATIONS OF CUBE ALGEBRA

The cube algebra was developed to execute some operations on the binary cubes (hypercubes). It includes almost all operations of the set theory and three own operations, that are called *coordinate product* (*star product*, *consensus* or $\star$ -operation),

coordinate subtraction (sharp product or #-operation) and coordinate intersection (∩-operation).

Earlier, the cube algebra was applied in connection with presentation of the Boolean functions in form of cubes [12, 13]. Later the operations of cube algebra were used for minimization of switching functions. Now these operations are also used for subcube allocation problem [6] and for routing of the information over fault-free nodes and links in hypercube [1]. In these studies it is shown that these operations are a good tool for definition of complete non-faulty subcubes of hypercube with faulty, prohibition or bused components, also for definition of the paths, including the shortest path from any vertex to other vertex in hypercube. The operations of cube algebra were enough explained in [12, 13] and [2]. So far as, in this paper we also use these operations, we consider them briefly.

Following [13, 12, 1], the *coordinate subtraction* of two cubes C_1 and C_2 , denoted by $C_1 \# C_2$, is the set of subcubes including the nodes from C_1 , but not from C_2 . For example, $0*0\#11=0*$ or $\{0*0, 1**\} \# 110 = \{0*0, *00, 10*, 1*1\}$. The *coordinate product* of two cubes C_1 and C_2 , denoted by $C_1 \star C_2$, 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, $0*0 \star 1*1 = 10*$. The *coordinate intersection* of two cubes C_1 and C_2 , denoted by $C_1 \cap C_2$, is the subcube presented in C_1 and in C_2 at the same time. For example, $0*0 \cap 10* = \emptyset$ or $10* \cap 1*1 = 101$.

After executing these operations one can obtain repeated cubes and cubes with smaller dimension that can enter in the cubes with bigger dimension. Everywhere in this study we will assume that the repeated cubes and the cubes with smaller dimension including to the cubes with bigger dimension, are avoided.

The coordinate subtraction operation was applied to finding the local prime implicants of the switching functions. 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]. To discard the faulty components from the hypercube the subtraction operation may be used. By means of the coordinate product operation 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 for switching functions which are common with other prime implicants. This allowed answering 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.

Coordinate (star) product operation also allows finding a bridge subcube between the cubes, if it is existing. This feature is very useful for determination a path between two subcubes. For example, for the subcubes $A=0*00$ and $B=01*1$ we obtain a bridge (link) $010*$, i.e. $A \star B = 0*00 \star 01*1 = 010*$. This also means we can find a bridge subcube if the subcubes A and B have a Hamming distance equal to one. On the other hand, the intersection operation can only discover the common part (i.e. intersection) of these subcubes and therefore we use the coordinate product operation here. For the same example: $A \cap B = 0*00 \cap 01*1 = \emptyset$, actually.

3. PROCEDURE FOR DEFINITION OF NEIGHBORHOOD OF FAULT FREE SUBCUBES

In a hypercube there may be faulty nodes and/or faulty links. In this case the hypercube is called *faulty hypercube*. In case when the nonfaulty nodes and links exist in n -cube it is called *complete cube*. A cube is called a *subcube of complete n -cube*, when at least one dependent (0 or 1) coordinate exists in the cube with dimension n . A cube S is the *subcube* of cube C if $C \star S = S$. A subcube S is called a *maximal subcube*, if subcube S is not subcube of any other subcube. Otherwise the subcube S is called *non-maximal*. A subcube F is called *complete faulty subcube (cube)* if F is a subcube (cube) included only faulty nodes and/or links. A set of the maximal subcubes nonevidently includes non-maximal subcubes. Thus, in order to define the desired non-faulty subcubes, it is necessary and sufficiently to have a set of maximal subcubes.

In order to determine the set of non-faulty and maximal subcubes (NMS) in faulty hypercube we use the following Theorem [1, 2].

Theorem 1. If F is a set of complete faulty cubes, then the set NMS of n -cube including F is determined by the $\#$ -subtracting the set F from the complete n -cube and avoiding non-maximal cubes from the result.

After determination of a set NMS we can realize different operations over the cubes, which are contained in the set of NMS. It is important to search the method of determination of NMS, each of which includes a current source in every stage of routing process. This method is based on the Theorem 2 [2].

Theorem 2. If $A = a_1a_2...a_j...a_n$ is any using vertex and $B_i = b_1^ib_2^i...b_j^i...b_n^i$, $B_i \in F$ is any forbidden cube, then for calculation of the prime implicant covered vertex A by procedure $K_i \setminus K_{i-1} \# B_i$, $i=1,2,...,m$, $K_0 = **...*$, it is necessary and sufficiently preserved for only all such b_j^i , which have opposite value with respect to corresponding a_i .

Using the Theorem 2 in the search of subcube (prime implicant), covered the vertex A , the faulty cube $B_i = b_1^ib_2^i...b_j^i...b_n^i$ can be transformed to cube $Q_i = q_1^iq_2^i...q_j^i...q_n^i$ by the means of rule [2]:

$$q_j^i = \begin{cases} *, & \text{if } b_j^i = * \text{ or } a_j = b_j^i \\ b_j^i, & \text{if } b_j^i = \bar{a}_j \end{cases} \quad (1)$$

Thus, one can define the fault free subcubes including the source node, also the target node. Now we will look for the existing connection from the subcube (s) including the source node (let source subcube or subcubes) to the subcubes, including the target node (let target subcube or subcubes).

Now we have at least two subcubes including the source and target nodes. Next, we will look for any bridge subcube between the source subcube and the target subcube. We can do it by executing the star product operation. If the result of this operation is not empty, this will mean that we have a path from the source node to the target node. Else, we must look for bridge between the source subcube and the other fault free subcubes, obtained after determination of the set NMS. But here first we have to subtract the source and target subcubes from this set. After star product operation the source subcube(s) with the elements of the set NMS we define the set, elements of which are not empty. Now these elements become as source subcubes. Then we repeat the star product operation between these subcubes and the target subcube. Thus we repeat these operations until the result of star product operation between the source subcubes and the target subcube is not empty. Non emptiness will mean that we have at least one path from the source node to the target node.

If the problem only consists of definition of any path, then the problem is solved. But if the number of the paths is more than one, which of them will have a minimum distance? This problem may have the following solution. In the any stage when we define the source subcubes, first we must take a subcube for future investigation that has a maximum dimension, than the others. In this way, we a priori define a minimal path between the source and target nodes.

Now we describe the developed procedure in a fermal way. Let $S=s_1s_2\dots s_n$ is a source node and $T=t_1t_2\dots t_n$ is a target node in an n -dimensional hypercube. Let F is a set of complete faulty cubes. Then:

1. Define the set NMS (Sharp subtract the faulty set from the given cube set and avoid non-maximal cubes).
2. Define $S \nrightarrow T$. If $S \cap T \neq \emptyset$ then goto 10, else goto 3.
3. Define the source subcube(s) and target subcube(s) by (1).
4. Define $S \nrightarrow T$. If $S \cap \nrightarrow T \neq \emptyset$ then goto 10, else goto 5.
5. Subtract the source subcube(s) and target subcube(s) from the set NMS. The obtained set be NMSS.
6. Put in order the set NMSS on dimension. The ordered set is NMSO.
7. Execute the star product operation the source subcube(s) with NMSO. Obtained nonempty set is NMSN.
8. Control if some elements of NMSN include the other elements. If such elements exist, remove them from the set NMSN. Put in order the set NMSN.
9. Goto 4.
10. End.

Example. Let $n=5$, $S=00000$, $T=11111$ and $F=\{100*0,01*11,010*0,001*0,10*11\}$. This means that hypercube has five faulty lines or $f=10$ faulty nodes. In Fig. 1 the source and target nodes are shown by gray color and cut lines show the faulty subcubes. According to the above-proposed procedure we obtain the following results:

1. $NMS=*****/F=\{***01,*11*0,*110*,00**1,000**,1*1*0,1*10*,11****\}$.
The obtained set of NMS is shown in Fig. 2.
2. $S \nrightarrow T = \emptyset$.
3. $S \in \{000**\}$ and $T \in \{11****\}$. $S'=\{000**\}$ and $T'=\{11****\}$.
4. $S' \nrightarrow T' = \emptyset$.
5. $NMSS=\{***01,*11*0,*110*,00**1,1*1*0,1*10*\}$.
6. $NMSO=\{***01,*11*0,*110*,00**1,1*1*0,1*10*\}$.
7. $NMSN=NMSO \nrightarrow S=\{***01,00**1\}$.
8. $00**1 \in ***01$. $NMSN=\{***01\}=S'$.
9. Goto 4. $S' \nrightarrow T'=\{11*01\}$. The obtained neighbor subcubes are shown in Fig. 3 by thick lines.
10. End. One of the paths between S and T will:

00000 $\xrightarrow{00000}$ 00000 $\xrightarrow{00001}$ 00001 $\xrightarrow{***01}$ ***01 $\xrightarrow{11*01}$ 11*01 $\xrightarrow{11111}$ 11111 (Fig. 4).

This means that one of the possible paths is: 00000 $\rightarrow$ 00001 $\rightarrow$ 10001 $\rightarrow$ 11001 $\rightarrow$ 11101 $\rightarrow$ 11111.

As seen we have found the neighbor subcubes between the source node 00000 and the target node 11111 (Fig. 2). By means of these fault free subcubes one can define the number of the paths searched. Now the problem is which of these paths has a minimal length? The definition of the length of these paths is not so difficult. Thus, the developed procedure allows us to find a short path (thick lines) between the source and target nodes in a faulty hypercube system (Fig. 4).

For another example we take $n=5$, $S=00000$, $T=11111$ and $F=\{100*0, 01*11, 010*0, 001*0, 10*11, 10*01, 1110*, 01001\}$. This means that $f=15$ nodes are faulty in the hypercube. In this case we have four neighbor subcubes ($000**$, $0*101$, $011*0$, $11*1*$) among which we look for a minimal path. The length of this path is seven ($00000 \rightarrow 00001 \rightarrow 00101 \rightarrow 01101 \rightarrow 01100 \rightarrow 01110 \rightarrow 11110 \rightarrow 11111$). As a matter of fact, this length is the possible minimal distance between the given source and target nodes for this example.

Fig.1. Hypercube with five faulty links ($n=5$)

Fig.2. Non-faulty and maximal subcubes obtained from the hypercube on Fig.1 by coordinate subtracting operation the faulty subcubes from the complete hypercube.

Fig.3. Neighbor subcubes between the source and target nodes.

Fig.4. The minimal path

4. CONCLUSION AND DISCUSSION

The finding of minimal path in a faulty hypercube multiprocessor in data communication processes has some aspects that is an object of studies. In many studies the number of the faulty elements is limited by n or near n . Despite of these methods there are no methods for the finding a minimal (or non-minimal) path when the number of the faulty elements in the hypercube is many greater than n . This case is interesting as theoretical and practical point of view. In this study we attempt to exploit a procedure

to look for a neighborhood of fault free subcubes in a faulty hypercube with a view to define a path between two subcubes in a faulty hypercube. Presented procedure does not depend on the number of faulty elements and on scattering of these elements in the hypercube. As the program, which carried out this procedure shows it, by increasing the dimension n the time of the finding of any path is raised. But for $n \leq 10$ this time is acceptable.

The future studies in this area should be the development of methods for decreasing the time of the calculation for the finding of the minimal path between the nodes (processors) in a hypercube multiprocessor when n will have great a value then 10. On the other hand the using of cube algebra operations allows working with non-faulty subcubes and thus it can help to develop different hypercube algorithms, which may be implemented in a faulty hypercube.

References

- [1] Allahverdi N., Kahramanlı Ş. and Erciyeş K. A Fault Tolerant Routing Algorithm Based on Cube Algebra for Hypercube Systems, to be published in *Journal of System Architecture (JSA)*, 1999.
- [2] Kahramanlı Ş. and Allahverdi N. Algebraic Approach to Transformations on Hypercube System, *Mathematical and Computational Applications*, Vol.1, No:1, pp.50-59, 1996.
- [3] Chang Y. and Bhuyan L., Subcube Fault Tolerance in Hypercube Multiprocessors, *IEEE Trans. Comput.* Vol.44, No.9, pp.1108-1120, 1995.
- [4] Chen M.-S. and Shin K.G. Depth-first Search Approach for Fault-Tolerant Routing in Hypercube Multicomputers, *IEEE Trans.on Paral.Distrib.Syst.* 1, 2 (April), pp.152-159, 1990.
- [5] Chiu, G.-M. and Wu, S.-P., A Fault-Tolerant Routing Strategy in Hypercube Multicomputers, *IEEE Trans.Comput.* Vol.45, No.2, pp.143-155, 1996.
- [6] Dutt, S. and Hayes, J. P., Subcube Allocation in Hypercube Computers, *IEEE Trans.Comput.* Vol.40, No.3, pp.341-352, 1991.
- [7] Wu J. Adaptive Fault-Tolerant Routing in Cube-Based Multicomputers Using Safety Vectors, *IEEE Trans. Paral. and Distrib.Systems*, Vol.9, No:4, pp.321-334, 1998.
- [8] Lan Y. An Adaptive Fault-Tolerant Routing Algorithm for Hypercube Multicomputers, *IEEE Trans.on Paral.and Distrib.Systems*, Vol.6, No.11, pp.1147-1152, 1995.
- [9] Lee, T. C. and Hayes, J. P. A Fault-Tolerant Communication Scheme for Hypercube Computers, *IEEE Trans. on Comput.* 41, 10 (Oct.), 1242-1256, 1992.
- [10] Bruck J., Cypher R., and Soroker D. Tolerating Faults in Hypercubes Using Subcube Partitioning, *IEEE Trans.Comput.* Vol.41, No.5, pp.599-605, 1992.
- [11] Raghavendra C.S., Yang P.-J., and Tien S.-B. Free Dimensions - An Effective Approach to Achieving Fault Tolerance in Hypercube, *IEEE Trans.on Comput.*, Vol.44, No.9, pp.1152-1156, 1995.
- [12] Miller R. *Switching Theory*, Vol. 1, Combinational Circuits. New York, J. Wiley and Sons, 1965.
- [13] Dietmeyer, D.L. *Logic Design of Digital Systems*, Boston, MA: Allyn and Bacon, 1979.