A COMPARISON CONNECTION ASSIGNMENT FOR DIAGNOSIS OF MULTIPROCESSOR SYSTEMS*

Miroslaw Malek

Department of Electrical Engineering
The University of Texas at Austin
Austin, Texas 78712

ABSTRACT

A comparison method for diagnosis of multiprocessor systems is introduced. Given a system of n units modeled by a linear graph, problems of finding the minimum number of comparison edges required for fault detection and fault location are solved by the use of a covering algorithm. The bounds for the number of comparison edges, the number of necessary comparisons and test cycles in fault detection and fault location in systems with n units are determined and an algorithm for an optimal comparison connection assignment is given. Simplicity and ease of implementation make the described method applicable for fault detection and location in multiprocessor systems.

Index terms: Comparison, covering, 1-diagnosable, diagnosis, fault detection, fault location, multiprocessor system, test cycle

I. INTRODUCTION

With the ever-growing complexity of multiprocessor systems, maintenance costs are increasing rapidly and there is big demand for fast, reliable diagnostic methods for fault detection and fault location in a system. Reliability and fault tolerance have become one of the primary goals in system design.

Substantial progress in LSI and VLSI technology has made single chip processors a reality and multi-microprocessor systems have since attracted many researchers. Among several performance objectives in those systems, diagnostic procedures and fault tolerant system capabilities seem to be of great importance. Several papers have attacked the problem of the diagnosis of a system consisting of n units and the classical approach to this problem required that some units be able to test other units[1 5 6 7]. Our philosophy is based on a comparison method. We assume that in the system of n units, the comparison of the behavior of some or every pair of units is possible so that any discrepancy during the comparison indicates some failure. It seems that this may be a simple and effective diagnosis method for complex systems such as microprocessor chips. A good example of the applied comparison technique is Bell System Electronic Switching Systems (EES) where a pair of proccessors is compared on-line.[8]

In this paper the cases of one and the maximum of n-2 faults (faulty units) are considered. In addition, the parameters of the comparison connection assignment are examined as well. We define lower and upper bounds for comparison connection assignment parameters in terms of the number of comparison edges, the number of comparisons and the number of test cycles. Comparisons may be made between pairs of units, e.g. by running the same program, preferably diagnostic, on those units and comparing the intermediate and final results. Comparisons may or may not require the use of additional hardware. The comparison conneciton in EES has a special unit called a matcher which is used to detect any mismatch between a pair of processors.[8] An ultra-reliable processor or any processor of the system could substitute a matcher provided that the required comparison connections exist. In this paper we will not involve ourselves in the practical implementation of the comparison method. Our objective is to show how to design systems which would require a minimum of additional hardware (comparison edges), a minimum number of comparisons or a minimum number of test cycles in order to detect or locate a faulty unit. We will examine systems with fixed and arbitrary structures. Moreover, it is important to remember that the diagnostic process consists of two activities: 1) fault detection where the objective is only to find out whether a fault has occurred in the system and 2) fault location, where identification of the faulty unit is required. Both cases are investigated and a design algorithm for the comparison connection assignment is also given.

II. THE MODEL

A multiprocessor system is modeled by a graph G(V,E,C) where V is a set of n vertices and E and C are sets of edges. Each element of the set V corresponds to a single computer, processor or unit in the system and each element of the set E represents the communication connection between a pair of units (processors, computers). Set C corresponds to the comparison connections and it may or may not be a subset of E. The comparison connection (comparison edge), represented by an edge, denotes that it is possible to compare the behavior of two units which are the endpoints of the considered edge. It also means that there is some hardware and/or software mechanism making the comparison feasible. Examples of the implementation of those mechanisms may be found in Bell Systems EES's.[8]

It is assumed that there is an executive system which accomodates the information on comparisons and thus derives the state of the whole system: whether it be faulty or nonfaulty, and/or locates the faulty unit(s). A graph G(V,E) is a connected multigraph, i.e. there is a path between any pair of vertices and there may be one or more edges between any given pair of vertices.

The main advantage of this model is its simplicity in detecting or locating a faulty unit because the comparison of pairs of elements seems to be easier than testing one unit by another or others.[1,5,6,7]

We shall consider the cases with one and the maximum allowable number of faulty units in the multiprocessor system. In the analysis, each case will be treated separately. In our study, the term "fault" is equivalent to "faulty unit". A system under t-fault assumption refers to the system in which up to t faults are permitted simultaneously. When two units are compared, their states and the outcome of the comparison are shown in Fig. 1. The outcome "pass" indicates that both units are fault-free, while "fail" indicates that at least one of the units is faulty. In fault detection further comparisons with respect to the compared pairs of units are obviously redundant, but in the case of fault location, the units must be compared with some other units in order to identify faulty unit(s). Generally, we would prefer to compare units that are already connected by communication links, therefore $C \subseteq E$ is desirable.

Unit 1	Unit 2	Comparison Outcome
fault free	fault free	0 (pass)
fault free	faulty	1 (fail)
faulty	fault free	1 (fail)
faulty	faulty	1 (fail)

Fig. 1. States and the outcome of a comparison of a pair of units

Let us define a fault table F as p x q matrix where p is the number of faulty states with up to t faults and q is equal to the number of comparison edges. (A similar table was defined by Kime[4] where p represented the number of faults and q denoted the number of pass-fail tests.)

$f_{ij} = 1$, j-th edge is adjacent to a vertex or vertices in i-th faulty state, and
$f_{ij} = 0$, otherwise.

The comparison table H is also p x q matrix but it is determined during the diagnostic process and $h_{ij} = 1$ iff j-th comparison fails and is adjacent to vertices in i-th faulty state, and $h_{ij} = 0$ otherwise. Upon completion of the comparison process, the row of matrix H that is identical (if any) to a row of matrix F identifies the faulty state uniquely, provided that all rows of matrix F were unique.

We say that the system is k-diagnosable under the given comparison connection assignment if there is at least one set of k identical rows in matrix F. Obviously, in a 1-diagnosable system every row is different, hence every faulty state may be uniquely determined.

The number of faulty states $s = \sum_{i=1}^{t} {}_nC_i$,

where ${}_nC_i = \frac{n!}{(n-i)!i!}$ and t is the maximum number of faulty units that may simultaneously occur in the system.

Since the number grows exponentially, it seems reasonable to consider that at most one faulty unit may be found at a time since the probability of simultaneous failure of more than one processor (unit) between successive diagnosis intervals seems to be minimal. Then k-diagnosability is equivalent to t-diagnosability with t = 1.

An example of a complete graph with four vertices and six edges is given in Fig. 2 and its corresponding fault table in Fig. 3. $s = \sum_{i=1}^{2} {}_4C_i = 10$. In order to make the graph 1-diagnosable, at least five comparison connections are necessary under two fault assumption. With four edges the graph becomes 2-diagnosable (e.g. 1,3 and 2,4 are identical if the edges are a,b,c,d) and with three edges, the graph is 3-diagnosable. If we reduce the number of faults at a time to a single fault only, then three comparison edges suffice to make the system 1-diagnosable. As we will prove later, it is possible to assume some fault of a maximum n-2 units in a system with n elements and have 1-diagnosability. Therefore, in the given example a maximum of two faults has been assumed.

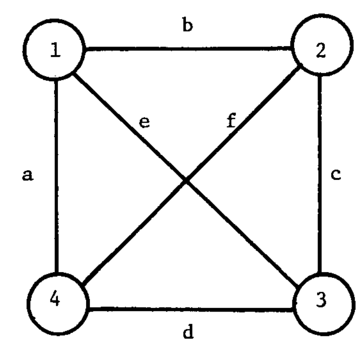

Fig. 2. A complete graph on four vertices

edge / faulty state	a	b	c	d	e	f
1	1	1	0	0	1	0
2	0	1	1	0	0	1
3	0	0	1	1	1	0
4	1	0	0	1	0	1
1,2	1	1	1	0	1	1
1,3	1	1	1	1	1	0
1,4	1	1	0	1	1	1
2,3	0	1	1	1	1	1
2,4	1	1	1	1	0	1
3,4	1	0	1	1	1	1

Fig. 3. A fault table of the complete graph on four vertices

One of the objectives of this paper is to determine 1) lower and upper bounds for the number of comparison connections, 2) the number of comparisons that have to be made in order to detect or locate a faulty unit, and 3) to determine the number of test cycles where test cycle is defined as a time interval necessary to perform comparisons between pairs of units simultaneously. We will also give an algorithm for the comparison connection assignment. Since the algorithm is based on covering problem algorithm, the following definitions are necessary.

In graph G, a set of g edges is said to cover G if every vertex in G is adjacent on at least one edge in g. A set of edges that covers a graph G is said to be an edge covering or simply covering G. The minimal covering is a covering from which no edge can be removed without destroying its ability to cover the graph. A minimal covering with the smallest number of edges is called the minimum covering. The number of edges in the minimum covering is called the covering number of the graph.

The comparison assignment parameters may be denoted as follows:

- q_d - the number of comparison edges in order to detect any fault in the system
- q_ℓ - the number of comparison edges in order to locate any fault in the system
- c_d - the number of comparisons required in order to detect any fault in the system
- c_ℓ - the number of comparisons required in order to locate any fault in the system
- t_d - the number of test cycles required for fault detection
- t_ℓ - the number of test cycles required for fault location

We shall attempt to solve two problems: one is to find a comparison connection assignment in a system with a fixed interconnection structure and another where the comparison connections between elements of the system may be placed arbitrarily.

III. COMPARISON ASSIGNMENT AND ITS PARAMETERS

An efficient comparison diagnostic procedure requires solution of the following problem. Given a graph G(V,E), find the minimum set of comparison connections $C \subseteq E$ which assures 1-diagnosability, i.e. location of every faulty state, and requires the minimum number of comparisons and test cycles. We will examine this problem under the single fault assumption.

Both optimization requirements concern cost minimization directly since a decrease in hardware costs is obtained by minimizing the number of comparison connections. In addition, the computer time used for diagnosis is reduced due to the minimization of the number of comparisons and test cycles.

Since the number of comparison connections is topology dependent, we may only determine lower and upper bounds for this number. The exact number of comparison connections may be obtained by using the minimum covering algorithm.[2 3]

The case of fault detection is straightforward. In order to detect whether there is a faulty unit in the system, every unit should be compared to some other one. In an ideal case, if there is an even number of units and there are connections sufficient to cover every pair separately by a single link, the number of required comparison edges is equal to $\frac{n}{2}$. If the number of units is odd, then the obtainable minimum equals $\lceil \frac{n}{2} \rceil$ and thus determines the lower bound for the number of comparison edges q_d. The upper bound occurs when every link has to be a comparison edge. This happens in some types of tree graphs which have a number of edges equal to n-1. This case occurs, for example, in a star graph where the graph is a tree with one vertex adjacent to all remaining vertices. In this graph, every comparison connection has to be used in order to detect or locate the faulty unit. Summarizing the above, we have

$$\left\lceil \frac{n}{2} \right\rceil \leq q_d \leq n-1.$$

The upper bound for fault location is exactly the same as for fault detection. The difference, however, may be found in the lower bound. Assuming the ideal case for detection, i.e. that all pairs of vertices are matched by a single edge in the first test cycle, we should have an additional connection that would compare a pair of units which have resulted in different outputs due to the fault. These connections should be a separate set of edges independent from the ones already used. We merge vertices adjacent to edges that were used for comparisons in the first step and we search for the minimum covering in the remaining graph. Consequently, the number of edges is equal to $\lceil \frac{n}{2} \rceil$ in the first test cycle plus $\left\lceil \frac{n-\lceil \frac{n}{2} \rceil}{2} \right\rceil$ in the second cycle. This bound may be decreased further if we notice that in order to locate one faulty unit out of three units we need two comparison connections instead of three as the above derived bound might indicate. Based on the fact that two edges are absolutely necessary for fault location in three units, we can now define the lower bound on the number of comparison edges. This bound is equal to $2\left\lfloor\frac{n}{3}\right\rfloor + n-3\left\lfloor\frac{n}{3}\right\rfloor = n - \left\lfloor\frac{n}{3}\right\rfloor$. In effect $n - \left\lfloor\frac{n}{3}\right\rfloor \leq q_\ell \leq n-1$.

In a practical approach it seems that the method of minimum covering would be used. Although it does not guarantee the minimum number of comparison connections for fault location, nevertheless, it assures the minimum for fault detection and also minimizes the number of comparisons and test cycles for fault location which in the long run may be more cost effective than minimization of the number of comparison edges.

The number of comparison edges concerns the total number of pairs of units that may be compared. We will demonstrate that the lower bound of necessary comparisons is smaller or equal to the lower bound for the number of comparison connections.

In the diagnosis process, the fault may be detected after the first test cycle. Subsequently, only one additional comparison is required and sufficient for fault location if the comparison is made between one of the pairs of units that has shown discrepancy and some other unit that proved to be fault-free on the basis of previous comparisons. The bounds for the number of comparisons c_d and c_ℓ may be easily determined.

The bounds for the number of comparisons are 1) in fault detection $\left\lceil\frac{n}{2}\right\rceil \leq c_d \leq n-1$ and 2) in fault location $\left\lceil\frac{n+1}{2}\right\rceil \leq c_\ell \leq n-1$.

The bounds for the number of test cycles for fault detection and location are respectively:

$$1 \leq t_d \leq n-1 \quad \text{and} \quad 2 \leq t_\ell \leq n-1$$

Let us consider an example of graphs (Fig. 4, Fig. 5 and Fig. 6) that attain lower and upper bounds for 1) the number of comparison connections, 2) the number of comparisons and 3) the number of test cycles in fault detection.

In the system represented by the graph in Fig. 4, lower bounds are attained except for the minimum number of comparison edges for the case of fault location. The structure of the system may be considered as optimal in terms of the comparison connection assignment diagnosis for fault detection.

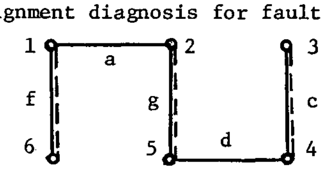

Fig. 4. A system of six vertices with an optimum connection assignment for fault detection (dotted lines) and minimum c_ℓ and t_ℓ

The system in Fig. 5, represented by a star graph, also has six units but all of its diagnostic parameters are equal to upper bounds.

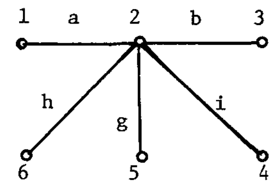

Fig. 5. A system of six vertices (star graph) with comparison parameters equal to the upper bounds

The parameters shown in Table 1 prove that all characteristics of the graph with respect to fault detection in Fig. 4 are optimum and equal to lower bounds as opposed to the graph in Fig. 5 whose comparison assignment parameters are equal to the upper bounds.

connection assignment / parameter	As in Fig. 4	As in Fig. 5	As in Fig. 6
q_d	3	5	4
q_ℓ	5	5	4
c_d	3	5	4
c_ℓ	4	5	4
t_d	1	5	2
t_ℓ	2	5	2

Table 1. The comparison parameters of systems in Figures 4, 5 and 6

The optimum connection assignment for fault location is shown in Fig. 6.

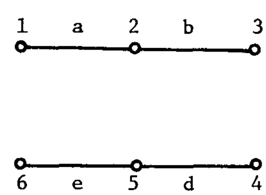

Fig. 6. The comparison connection assignment requiring a minumum number of comparison edges for fault location

In an arbitrary graph, the number of comparison edges for fault detection q_d is equal to the covering number γ. The fault location requires $\gamma + \gamma_r$ edges if the covering method is used where γ_r is the covering number of the graph reduced in the following manner: merge all subsets of vertices that are connected by edges which are in the minimum cover by making them into single vertices and deleting all self-loops. The method of determining the q_d and q_ℓ directly suggests an algorithm for finding a set of comparison connections for fault detection and fault location.

Algorithm for finding the comparison connection assignment:

Step 1: Find the minimum covering g in a given graph G by using the covering algorithm.[2 3]

Step 2: If the connection assignment is for fault detection then stop (the assignment is minimum for fault detection), otherwise go to Step 3.

Step 3: Reduce graph G by merging all connected parts of the covering into single vertices, delete self-loops and leave all remaining connections.

Step 4: Find the minimum covering h in the reduced graph then stop. Coverings g and h form the comparison connections set for fault location which requires a minimum number of comparisons.

The number of comparisons required for fault detection is the same as q_d and equals γ. The number of comparisons for fault location equals $\gamma + 1$ if there is a perfect matching (i.e. when the number of edges in covering g of G equals $\frac{n}{2}$) and $c_\ell \geq \gamma$ otherwise.

The number of testing cycles for fault detection may be determined by $t_d = d_{max}(v)$ where $d_{max}(v)$ denotes the maximum degree among vertices in g and h, i.e. after deleting in G the edges which are not in the covering sets g or h. In the case of fault location, the number of test cycles $t_\ell = 2$ is sufficient if there is a perfect matching, otherwise $t_\ell = d_{max}(v)$.

The comparison connection method may be extended to multiple faults. The comparison parameters in this case could be derived from a fault table. A necessary and sufficient condition for every state to be locatable is to insure that there are enough comparison edges to make every row distinct. This means that every subset of 1,2,...t vertices (if up to t faults are assumed) must have a distinct set of adjacent edges in order to be locatable.

Theorem: If there is a comparison edge between every pair of vertices in the system graph G and G has n vertices, the maximum number of faulty units that may occur and be uniquely located equals n - 2.

Proof: Since the comparison of every pair of vertices is possible, there will be a single comparison without discrepancy locating the only two nonfaulty units. Then all other cases with less than n - 2 faulty units may also be located but the states of the system when there are n - 1 or n faults cannot be distinguished due to the fact that in both cases all comparison edges would indicate discrepancy. Therefore n - 2 is the maximum number of the locatable faulty units.

Another problem that should be considered here is the design of systems with an optimal connection assignment. The optimal connection assignment is one which requires the minimum number of comparison connections, comparisons and test cycles in a system of n units. In fault detection all three parameters may be easily optimized by covering algorithms. The comparison assignment is determined by the minimum covering of n vertices and all three parameters are minimum if the vertices are coupled in $\lceil \frac{n}{2} \rceil$. In fault location the optimization objective must be determined first. If it is the minimum number of comparison connections, the system has to be partitioned into triples of units and each triple should be connected by a pair of edges. If n is not divisible by 3 then one or two units are connected to some triples by single edges. In addition, there is a capability of locating at least $\lfloor \frac{n}{3} \rfloor$ faulty units provided each fault occurs in the separate triple. If the minimization of the number of comparisons and test cycles seems to be more cost-effective than the minimization of the number of comparison connections then the perfect matching, merging and perfect matching again may be applied. Optimization of q_ℓ corresponds to the minimization of the additional hardware costs (if any) in order to make comparisons feasible, optimization of c_ℓ and t_ℓ refers to the minimization of computer time.

IV. CONCLUSION

A comparison method has been introduced for fast and reliable detection and/or location of a faulty unit in multiprocessor systems. This method is primarily applicable for the reconfigurable multimicroprocessor systems where the simultaneous running of the same programs on different processors and diagnostic comparisons are easily feasible. Simplicity and ease of implementation as well as the smaller number of tests than those used in classical methods make comparison connection assignment an attractive alternative in the diagnosis of the multiprocessor systems.

REFERENCES

1. F. Barsi, F. Grandoni and P. Maestrini, "A theory of diagnosability of digital systems," IEEE Trans. on Comput., vol. C-25, No. 6, June 1976, pp. 585-593.
2. J. Edmonds, "Paths, trees and flowers," Canadian Journal of Mathematics, vol. 17, 1965, pp. 449-467.
3. S. Even, "Maximum matching algorithms; computational complexity," Proc. of Int. Symposium on

Foundations of Computer Science, 1975, pp. 100-112.
4. C. R. Kime, "An analysis model for digital system diagnosis," IEEE Trans. on Comput., vol. C-19, No. 11, November 1970, pp. 1063-1073.
5. G. G. L. Meyer and G. M. Masson, "An efficient fault diagnosis algorithm for symmetric multiple processor architectures," IEEE Trans. on Comput., vol. C-27, No. 11, November 1978, pp. 1059-1063.
6. J. A. McPherson and C. R. Kime, "A two-level diagnostic model for digital system," IEEE Trans. on Comput., vol. C-28, No. 1, January 1979, pp. 16-27.
7. F. P. Preparata, G. Metze and R. T. Chien, "On the connection assignment problem of diagnosable systems," IEEE Trans. on E. Comput., vol. EC-16, No. 6, 1967, pp. 848-854.
8. W. N. Toy, "Fault tolerant design of local EES processors," Proc. of the IEEE, vol. 66, No. 10, October 1978, pp. 1126-1145.

*This work was supported in part by the National Science Foundation under Grant MCS77-20698.