UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

(2)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

(14) 1. REPORT NUMBER: AFOSR-TR- 78-1355

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

(6) 4. TITLE (and Subtitle): ANALYSIS AND DESIGN OF FAULT-TOLERANT COMPUTER SYSTEMS,

(9) 5. TYPE OF REPORT & PERIOD COVERED: Interim rept.

6. ~~PERFORMING ORG. REPORT NUMBER~~

(10) 7. AUTHOR(s): John P. Hayes

(15) 8. CONTRACT OR GRANT NUMBER(s): AFOSR-77-3352

9. PERFORMING ORGANIZATION NAME AND ADDRESS
University of Southern California
Electronic Sciences Laboratory
Los Angeles, California 90007

~~10. PROGRAM ELEMENT, PROJECT~~, TASK AREA & WORK UNIT NUMBERS
61102F (16) 2304/A6 (17) A6

11. CONTROLLING OFFICE NAME AND ADDRESS
Air Force Office of Scientific Research/NM
Bolling AFB, Washington, DC 20332

12. REPORT DATE: August 1, 1978

13. NUMBER OF PAGES: 26

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

LEVEL

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

(11) 1 Aug 78

Approved for public release; distribution unlimited.

(12) 39 p.

DDC RECEIVED OCT 4 1978

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Bit-sliced microporcessors	Graph models
Communication networks	Microprocessors
Connecting networks	Multiprocessors
Fault diagnosis	Recovery
Fault-tolerant computing	Test generation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the first-year results of an investigation of fault-tolerant computer systems. A new method for measuring recovery time in fault-tolerant multiprocessors was developed. A complete characterization of optimally t-step recoverable systems was obtained, and certain graph transformations that simplify recovery analysis were studied. Some diagnosability properties of n-cube interconnection networks were derived. A study of fault tolerance in large connecting networks was initiated using a new concept of dynamic full access. A design theory based on recursive

DD FORM 1 JAN 73 1473

364 620

UNCLASSIFIED

20. Abstract continued.

component expansion capabiilities was developed for MSi/LSI systems. The use of similar recursive methods for test pattern generation was aiso initiated. Promising resuits were obtained for testing bit-siiced microprocessors and related components.

AFOSR-TR- 78 - 1355

1977-78 Annual Technical Report

Air Force Office of Scientific Research

Grant No. AFOSR-77-3352

ANALYSIS AND DESIGN OF FAULT-TOLERANT COMPUTER SYSTEMS

Prepared
by

John P. Hayes

Electronic Sciences Laboratory
University of Southern California
Los Angeles, California 90007

August 1, 1978

TABLE OF CONTENTS

ABSTRACT

This report describes the first-year results of an investigation of fault-tolerant computer systems. A new method for measuring recovery time in fault-tolerant multiprocessors was developed. A complete characterization of optimally t-step recoverable systems was obtained, and certain graph transformations that simplify recovery analysis were studied. Some diagnosability properties of n-cube interconnection networks were derived. A study of fault tolerance in large connecting networks was initiated using a new concept of dynamic full access. A design theory based on recursive component expansion capabilities was developed for MSI/LSI systems. The use of similar recursive methods for test pattern generation was also initiated. Promising results were obtained for testing bit-sliced microprocessors and related components.

1. RESEARCH OBJECTIVES

The purpose of this research project is to develop methods for the analysis and synthesis of complex fault-tolerant computer systems. It is motivated by recent rapid developments in large-scale integration (LSI) technology, especially the introduction of microprocessors, which are expected to increase greatly the use of multiple computer systems that are required to be highly reliable. The research is particularly concerned with dynamic reconfiguration and recovery in the event of failures, topics which have received relatively little research attention in the past. It is intended to develop specific measure of the cost and complexity of reconfiguration and recovery, and to derive efficient fault tolerance algorithms based on these measures. Various graph theoretical and algebraic tools are used in this research, with the facility graph model [1], developed by the Principal Investigator, serving as a starting point. The special problems associated with the design of systems containing many microprocessors, particularly the problem of interprocessor communication, are also being investigated.

2. RESEARCH ACCOMPLISHMENTS

During 1977-78 results were obtained in three main areas:

(1) Recovery modeling in multiprocessor systems

(2) Communication networks for multi-microprocessors

(3) Design and testing of MSI and LSI systems

These results are described in detail in the following subsections.

2.1 Recovery modeling in multiprocessor systems [2, 3][1]

A new method for characterizing the recovery time of fault-tolerant multiprocessor systems was developed. The system is represented by a facility graph G_r in which nodes correspond to processors and edges correspond to communication links [1]. The fault-free nodes include nodes actively engaged in data processing and nodes acting as standby spares. A fault is represented by the removal of a node and its associated edges from G_r. Faults are tolerated by reconfiguring the pattern of active and spare nodes in G_r so that there always exists an active subnetwork that is isomorphic, that is, has the same (logical) interconnection structure, as a certain minimum configuration G_b called the basic system. G_b can be taken as the minimum fault-free system needed to perform a particular set of tasks.

A system G_r is called k-fault-tolerant (k-FT) t-step recoverable (t-SR) if it can recover from up to k faults by changing the states of at most t fault-free nodes. k is clearly a measure of the amount of damage the

[1] Reference [3] forms an appendix to this report.

system can tolerate. A state change e.g., from spare to active, typically involves the establishment of new logical paths in the system, and the transfer of programs and data between the affected nodes. If n state changes of average duration c are required to recover from a particular fault, then nc is the total recovery time. Thus the parameter t defined above is proportional to the maximum recovery time required by G_r.

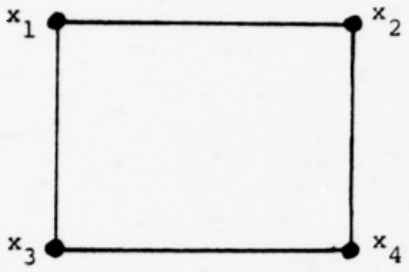
Clearly $t \geq k$. A case of particular interest, corresponding to a class of systems with minimum recovery time, is where $t = k$. In such systems recovery from t faults is achieved by immediate replacement of each failed node by a fault-free spare. G_r is defined to be optimally t-SR with respect to an n-node basic system G_b if

(1) G_r is t-FT/t-SR with respect to G_b

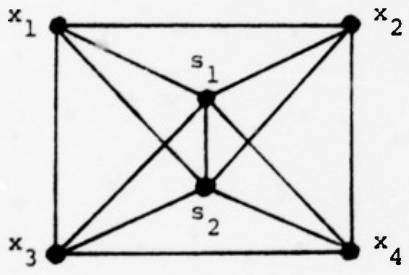
(2) G_r contains the minimum number of nodes, viz. $n + t$

(3) G_r contains the fewest edges among all systems satisfying conditions (1) and (2)

In [3] we prove that the optimal t-SR realization of every G_b is unique, and that it has a surprisingly simple structure. Figure 1a shows an example of a basic graph I_b consisting of four processors arranged in a ring. Figure 1b shows the corresponding optimal 2-SR graph I_2^{OPT}. It consists of I_b with two additional spare nodes, labeled s_1 and s_2, and additional edges connecting s_1 and s_2 to all nodes, including each other. Every fault graph formed by removing one or two nodes from I_2^{OPT} contains a subgraph isomorphic to I_2 (the 2-FT property). Furthermore, each such subgraph can be chosen so that it differs from the original active subgraph in at most two nodes (the 2-SR property).



(a)



(b)

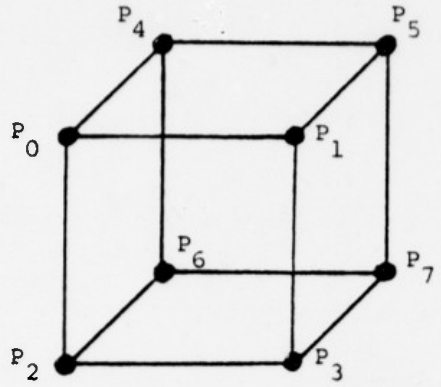
Figure 1. (a) A 4-node basic system I_b. (b) The corresponding optimal 2-SR system I_b^{OPT}.

Optimal t-SR systems have the disadvantage that the number of edges connected to some nodes (the node degree) may be very large. Since this represents the number of parallel data paths to a processor, it is often severely restricted by physical considerations, for example, microprocessor pin limitations. Thus nonoptimal fault-tolerant systems with limited node fanout are of interest. We have investigated a class of graph transformations, called line graph transformations, which lead to t-SR designs with nodes of lower degree than the corresponding optimal t-SR systems [3]. We have also shown that line graph transformations greatly simplify the computation of the parameters k and t.

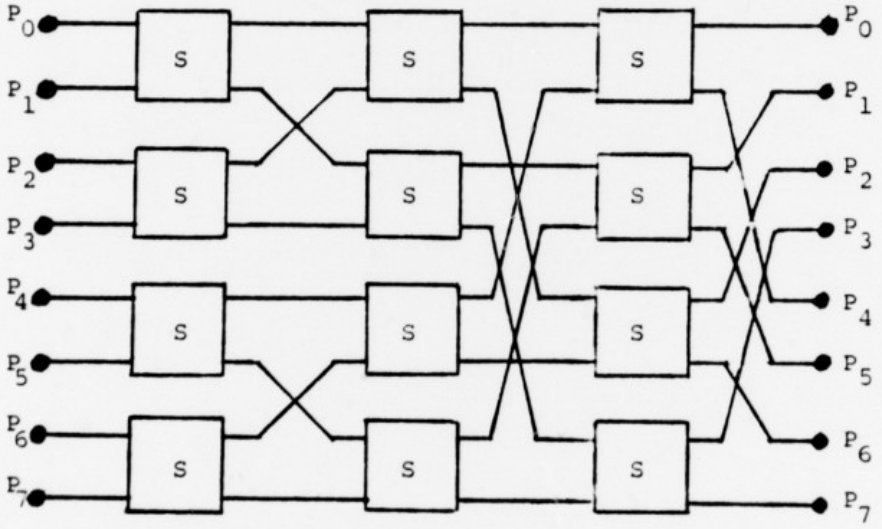
2.2 Communication networks for multi-microprocessors [4]

An extensive survey of systems containing many microprocessors was completed. Two major communications structures for such systems were identified; the hierarchical bus organization represented by Cm* [5], and the n-cube organization proposed by several researchers [6, 7]. Most of the published work in this area deals with unimplemented paper designs with little analytical basis. System reliability and fault tolerance have also been largely ignored.

A network organization with a relatively sound analytical basis is the binary n-cube structure [7]. This contains 2^n processors whose logical interconnection structure can be represented by an n-dimensional cube. Figure 2a shows the structure of the 3-cube. We have investigated several aspects of the fault tolerance of n-cube networks. Using the approach of



(a)



(b)

"Through" state of S

"Cross" state of S

(c)

Figure 2. (a) 3-cube network. (b) Implementation of a 3-cube network (c) States of the switch S.

Preparata et al. [8] we have shown that the diagnosability of an n-cube system is n for $n \geq 3$, where the diagnosability of a system is defined as the largest number k such that the system is one-step k-fault diagnosable [4].

N-cube arrays can be implemented using connecting networks of the type long used in telephone exchanges [9]. Figure 2b shows one such implementation of the 3-cube using twelve switches denoted S. Each S may be considered to have two states, the "through" and "cross" states depicted in Figure 2c. We have begun investigating the fault tolerance properties of connecting networks of this kind. A study of actual circuits used for S [10] indicates that most faults in the network can be modeled by switches that are stuck at the through state (s-a-T) or stuck at the cross state (s-a-X).

We have defined a connecting network N to have the dynamic full access property if each processor P_i can be connected to any other processor P_j via a finite (but unspecified) number of passes through the connecting network. This is a generalization of the usual full access property [9]. N is said to be k-fault tolerant (k-FT) with respect to the foregoing s-a-T/X fault model if the failure of k or fewer switches in N does not destroy the dynamic full access property. We have begun investigating the conditions for N to be k-FT. It is hoped that this work will lead to methods for designing efficient and fault-tolerant communication networks for large multi-microprocessor systems.

2.3 Design and testing of MSI and LSI systems [11, 12]

Most existing analytical tools are inadequate for dealing with digital components above the gate and flip-flop levels, which correspond

to small-scale integration (SSI) in current technology. There is at present no adequate theory for the design or testing of MSI and LSI devices, although the need for such a theory has long been recognized. Perhaps the only LSI device for which a promising theory of testing is emerging is the semiconductor random access memory (RAM) [13].

We have observed that a significant property of components at all complexity levels is expansibility, which is the ability of components of a given type to be interconnected in a systematic way to form larger components of the same type [12]. The larger component performs the same operation as its constituent elements, but processes more and/or bigger operands. Many MSI and LSI design rules are merely recipes for component expansion, e.g., how to build a 1-out-of-N decoder using 1-out-of-n decoders where N > n, or how to build an N x M RAM using n x m RAM IC's where N > n or M > m [14]. Expansibility plays a particularly important role in the architecture of microcomputers. The major design problems revolve around the number, size and interconnections of the ROM's, PAM's and IO interface circuits used, problems which are intimately associated with the expansibility of these components. With bit-slice architecture the CPU (microprocessor) becomes an expandable design component. Two main expansion techniques have been identified, expansion by composition and by replication [12]. Expansion methods, which correspond to design rules, can be concisely defined by recursive equations. For example, a typical MSI component, a ripple-carry adder can be defined as follows:

$$\text{Basis: } ADD^{1}_{0:1}(x_0, y_0, c_{in}) = x_0y_0 + x_0c_{in} + y_0c_{in},\ x_0 \oplus y_0 \oplus c_{in}$$

$$ADD^{n+1}_{0:n+1}(x_{0:n}, y_{0:n}, c_{in}) = ADD^{1}_{0:1}(x_0, y_0, ADD^{n}_{0}(x_{1:n}, y_{1:n}, c_{in})),$$

$$ADD^{n}_{1:n}(x_{1:n}, y_{1:n}, c_{in}).$$

Here x_i and y_i denote input data lines, and c_i denotes a carry line. We have proposed a classification scheme for expansion algorithms based on three parameters: the presence of feedback, the use of constant inputs or outputs, and the logical depth of the interconnections used. We have shown that most standard components can be expanded using FS2 algorithms which allow neither feedback nor constant input/output values, and which require two (the minimum number) logic levels. Some other useful expansion methods have also been identified [12].

We have also demonstrated that recursive techniques can be used for test pattern generation. As a simple illustration consider the n-input AND function AND^n. Let $T^n(x_0, x_1, \ldots, x_{n-1})$ be a Boolean function denoting the (unique) set of test patterns for stuck-type faults in AND^n; $T^n(X) = 1$ if and only if X is a test pattern. We can define the tests for AND^n recursively as follows.

Basis: $T^2(x_0, x_1) = \bar{x}_0 x_1 + x_0 \bar{x}_1 + x_0 x_1$

$$T^{n+1}(x_0, x_1, \ldots, x_n) = T^n(x_0, x_1, \ldots, x_{n-1}) x_n + x_0 x_1 \ldots x_{n-1} \bar{x}_n$$

We have started to extend this test generation philosophy to obtain efficient and systematic test procedures for MSI/LSI systems. Besides leading to analytic testing methods, this approach has the added advantage of being relatively independent of such factors as word size, making it possible to analyze all members of a family of components simultaneously.

We have carried out a study (unpublished) of the feasibility of this general approach for testing bit-sliced microprocessors. We use as the basic component the 1-bit processor cell M shown in Figure 3. M has most of the major features of a commerical bit-sliced microprocessor, such as the Intel 3002 2-bit processor [14] or the Am2901 4-bit processor [16]

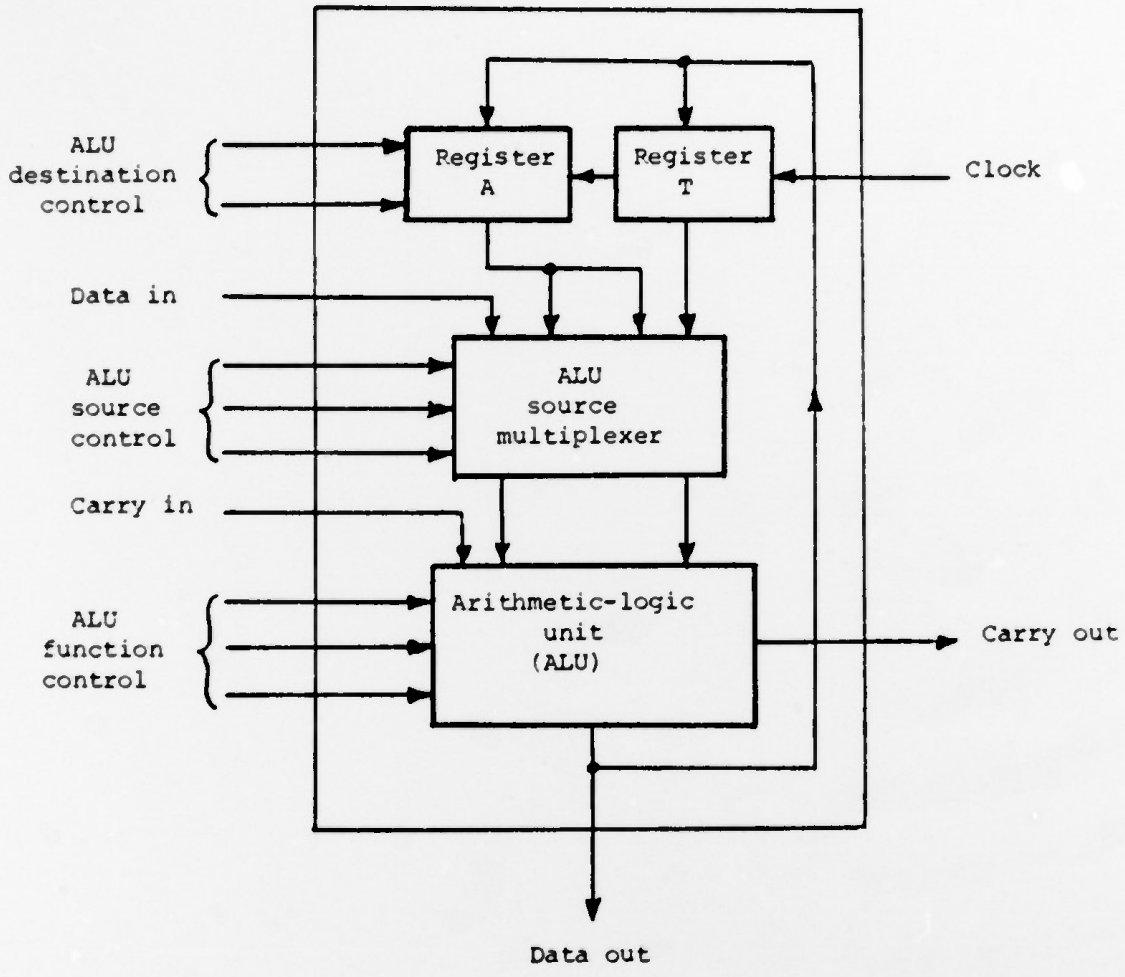


Figure 3. Processor cell M used for analyzing bit-sliced microprocessors.

(only the shift function and the status flags have been omitted). It contains two registers A and T and two complex combinational circuits, a multiplexer and an arithmetic-logic unit ALU. Using the most general functional fault model, which allows aribtrary functional changes in the individual registers and combinational circuits, we have shown that M can be tested with $t \approx 100$ test patterns. Furthermore, a k-bit processor array constructed from k copies of M can also be tested with t tests, independent of k, and the array tests can be easily derived from those of the individual cell.

2.4 References

[1] J. P. Hayes: "A graph model for fault-tolerant computing systems," IEEE Trans. Computers, Vol. C-25, pp. 875-884, September 1976.

[2] R. Yanney: "Design and analysis of fault tolerant multiprocessor systems," Ph.D. Thesis Proposal, University of Southern California, January 1978.

[3] J. P. Hayes and R. Yanney: "Fault recovery in multiprocessor networks," Digest of Intl. Symp. on Fault-Tolerant Computing (FTCS-8), pp. 123-128, Toulouse, June 1978.

[4] J. Shen: "Fault-tolerant multi-microcomputer systems," Ph.D. Thesis Proposal, University of Southern California, January 1978.

[5] R. J. Swan et al.: "Cm* -- a modular multi-microprocessor," AFIPS Conf. Proc., Vol. 46, pp. 637-644, 1977

[6] H. Sullivan and T. R. Bashkow: "A large scale homogeneous fully distributed parallel machine, I," Proc. 4th Ann. Symp. Comp. Architecture, pp. 118-124, March 1977.

[7] M. C. Pease: "The indirect binary n-cube microprocessor array," IEEE Trans. Computers, Vol. C-26, pp. 458-473, May 1977.

[8] F. P. Preparata et al., "On the connection assignment problem of diagnosable systems," IEEE Trans. Electronic Computers, Vol. EC-16, pp. 848-854, December 1968.

[9] V. E. Benes: Mathematical theory of connecting networks and telephone traffic, New York, Academic, 1965.

[10] K. N. Levitt et al.: "A study of the data commutation problems in a self-repairable multiprocessor," Proc. 1968 Spring Joint Computer Conf., pp. 515-527.

[11] T. Sridhar: "Functional testing of complex combinational circuits," EE 556 Project Report, University of Southern California, January 1978.

[12] J. P. Hayes: "Component expansion techniques in computer design," March 1978, Submitted for publication to Digital Processes.

[13] J. P. Hayes: "Detection of pattern sensitive faults in random-access memories," IEEE Trans. Computers, Vol. C-24, pp. 150-157, February 1974.

[14] Fairchild Inc.,: The TTL application handbook, Mountain View, California, 1973.

[15] Intel Corp.: 3000 Series Reference Manual, Santa Clara, California 1976.

[16] Advanced Micro Devices: Am2900 Family Data Book, Sunnyvale, California 1976.

3. PUBLICATIONS

The following documents were sponsored wholly or in part by Grant No. AFOSR-77-3342.

[1] R. Yanney: "Design and analysis of fault tolerant multiprocessor systems," Ph.D. Thesis Proposal, University of Southern California, January 1978.

[2] J. P. Hayes and R. Yanney: "Fault recovery in multiprocessor networks," Digest of Intl. Symp. on Fault-Tolerant Computing (FTCS-8), pp. 123-128, Toulouse, June 1978.

[3] J. Shen: "Fault-tolerant multi-microcomputer systems," Ph.D. Thesis Proposal, University of Southern California, January 1978.

[4] T. Sridhar: "Functional testing of complex combinational circuits," EE 556 Project Report, University of Southern California, January 1978.

[5] J. P. Hayes: "Component expansion techniques in computer design," March 1978, submitted for publication to Digital Processes.

4. PERSONNEL

The following people were associated with the research effort reported here.

Principal Investigator

John P. Hayes

Research Assistants

John P. Shen

Thirumalai Sridhar

Raif Yanney

Note: R. Yanney received no fianacial support from Grant No. AFOSR-77-3352.

5. INTERACTIONS

Meetings with Air Force Personnel

J. P. Hayes met with Dr. Joseph Bram, AFOSR Directorate of Mathematical and Information Sciences, in Los Angeles, on January 30, 1978. Current progress and future plans for the project being reported here were reviewed.

J. P. Hayes met with Mr. Armand Vito of RADC (ISCA) in Marina Del Rey, California on April 6, 1978 to discuss research topics of mutual interest.

J. P. Hayes visited RADC, Rome, New York, May 12-13, 1978. He met with Mr. Murray Kesselman (ISCA) who provided him with a detailed overview of Air Force research interests in the areas of computer architecture and fault-tolerant computing. He also met with Lt. Michael Troutman (ISCA) and discussed the Air Force sponsored Total System Design (TSD) and Multi-Microprocessor System (MMS) projects. Dr. Hayes had an opportunity to see some of RADC's research facilities, including its QM-1 and STARAN computers.

Attendance at FTCS-8

J. P. Hayes and R. Yanney attended the 1978 International Symposium on Fault-Tolerant Computing (FTCS-8) in Toulouse, France, June 21-23, 1978. This is the major annual conference on research in fault tolerance. Approximately 350 researchers from 25 countries attended FTCS-8. The paper "Fault recovery in multiprocessor networks" (see Apprendix) was presented at this conference.

6. SUMMARY AND FUTURE PLANS

We have developed a new model for measuring the recovery time of a fault-tolerant system based on the facility graph concept. Necessary and sufficient conditions for an arbitrary system to be k-step recoverable were obtained. A survey of communication networks for multi-microprocessors was carried out. The diagnosability of the n-cube interconnection network was characterized. An analysis of the fault tolerance properties of connecting networks was initiated using the concept of dynamic full access. A design theory for MSI/LSI systems based on a formal definition of recursive expansibility was developed. It was shown that this approach can be used for test pattern generation for a variety of complex systems including bit-sliced microprocessors.

In the area of reconfiguration and recovery we propose to investigate strategies for achieving fault tolerance in distributed systems when the individual processors have limited information about the system as a whole. We also intend to study graceful degradation in such systems. We propose to continue our analysis of communication networks for multi-microprocessors, with the aim of completely characterizing their fault tolerance properties. We plan to extend our analysis of bit-sliced microprocessors to include all the features of real systems. We further aim to extend it to other bit-sliced components such as microprogram sequencers and RAM's so that ultimately we can automatically generate a near-optimal test set for complete microcomputers that use bit-slicing technology. Finally, we hope to use our knowledge of the test requirements of bit-sliced microcomputers to analyze non-bit-sliced systems.

APPENDIX

FAULT RECOVERY IN MULTIPROCESSOR NETWORKS *

John P. HAYES
Department of Electrical Engineering
University of Southern California
Los Angeles, California 90007 USA

Raif YANNEY
Hughes Aircraft Company
Culver City, California 90230 USA

ABSTRACT

A method for characterizing dynamic reconfiguration and recovery in fault-tolerant networks of processors is proposed. A network is represented by a graph G_r whose nodes correspond to processors and whose edges correspond to communication links. Each node or edge has three major states: active, inactive (spare) and failed. G_r tolerates a fault F by activating spare nodes and edges to reconfigure around the failed components so that an active subnetwork isomorphic to a basic system G_b is maintained. G_r is called k-fault-tolerant (k-FT) t-step recoverable (t-SR) if it can recover from k or fewer node failures by changing the states of at most t fault-free nodes, e.g., by activating t spare nodes. Thus t is a measure of system recovery time. A t-FT system is called optimally t-SR if it contains t spare nodes and the minimum number of edges that permit t-step recovery from all tolerated faults. Necessary and sufficient conditions for G_r to be optimally t-SR with respect to an arbitrary network G_b are obtained. Techniques for achieving t-step recovery where $t > k$ are discussed, with particular reference to networks with restricted node fanout, a constraint imposed by most microprocessors. A graph transformation technique based on line graphs is described that simplifies the calculation of k and t.

I. INTRODUCTION

Most previous research in fault-tolerant computer design has been concerned either with system reliability or fault diagnosis. Other important aspects of system behavior, notably recovery, have received little attention, even though they play a central role in fault tolerance. In this paper a graph theoretical model for fault recovery in complex systems is presented. The model is particularly applicable to large multiprocessors. Systems containing thousands of microprocessors have been proposed recently and are likely to proliferate in the future [1, 2]. It can be expected that many of these multimicroprocessor systems will have fault tolerance as a major design goal.

A system is modeled here by a graph whose nodes represent hardware components, e.g., processors or computers, and whose edges represent communication links, e.g., switching networks or buses. Similar models have been used previously in the analysis of computer network reliability [3], self-diagnosability [4], and fault tolerance [5]. These are all primarily structural rather than behavioral models, since the graphs used represent the physical or logical interconnection structure of the system under consideration. As such they are to be contrasted with models such as Petri nets or state graphs that are primarily behavioral [6].

II. RECOVERY MODEL

Following [5], a computer system is described by a (facility) graph whose nodes represent (micro-) processors and whose edges represent communication paths. All nodes are assumed to be of the same type and to have the same processing abilities. Edges are assumed to be undirected. A fault is represented by the removal of nodes and edges from the graph.

Definition 1: A basic graph G_b is a graph that represents the minimum system configuration needed to perform a certain set of tasks. Thus a basic system cannot tolerate any faults.

Definition 2: A redundant graph G_r with respect to a basic graph G_b is one that contains G_b as a proper subgraph. In other words, a proper subgraph G_b' of G_r is isomorphic to G_b, denoted $G_b' \equiv G_b$. G_r is viewed as a fault-tolerant realization of G_b.

At any time, some subgraph $G_b' \equiv G_b$ of G_r represents an active system engaged in data processing. The remaining part of G_r, denoted $G_r - G_b'$, represents either unused (spare) or unusable (faulty) components. Thus every node x of G_r can be viewed as having three possible states:

(1) active, that is $x \in G_b'$

(2) spare

(3) faulty.

Definition 3 [5]: G_r is k-fault tolerant (k-FT) with respect to G_b if the removal of any k nodes (and the edges connected to those nodes) from G_r results in a graph that contains G_b.

It is assumed that the systems of interest contain a mechanism for continuous self-diagnosis. For example, each node may be regularly tested by one or more of its neighboring nodes. The precise manner in which diagnosis is achieved is not of direct interest here. Once a faulty active node is detected, a process of recovery is initiated which involves replacing the active subsystem G_b' by another subsystem $G_b'' \equiv G_b$ which contains no faulty nodes. This means that if G_b' contains k faulty nodes, at least k previously spare nodes must be changed to the active state

* This research was supported by the Air Force Office of Scientific Research under Grant No. AFOSR-77-3352, by the Joint Services Electronics Program under Contract F44620-76-C-0061, and by a Fellowship from Hughes Aircraft Company.

and must be included in G_s''. The manner in which the new active subsystem G_b'' is determined constitutes the recovery strategy. In this paper aspects of recovery are considered that are largely independent of the particular recovery strategy employed. Note that recovery is being viewed primarily as a process of reconfiguration around the faulty nodes. The possible changes of state that a node can experience during system operation are illustrated in Fig. 1.

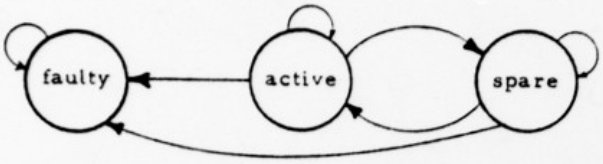


Fig. 1. State diagram for a system node.

The recovery process often involves a considerable amount of information transfer among the system nodes. For example, a spare node s that is being activated to replace a defective node x must be provided with all information defining the functions of x, as well as the status of x at the last known (error-free) check-point. This information is transferred to s from x or from some other processor that stores the status of x, e.g., a system supervisor. The number of fault-free nodes whose state or identity is changed when forming G_b'' from G_b' is taken as a measure of system recovery time, and leads to the following definition.

Definition 4: G_r is t-step recoverable (t-SR) with respect to G_b if G_r is t-FT with respect to G_b and G_r can recover from any fault affecting $k \leq t$ nodes by changing the state or identity of at most t fault-free nodes.

In many cases recovery can be accomplished by replacing the k faulty nodes of G_b' by k spare nodes. Spare nodes are assumed to be fault-free when they are first activated; they may subsequently become faulty and require replacement. It may also be necessary to replace active nodes as well, either by changing active nodes to spares, or requiring an active node to assume the identity of another active node. The parameter t defined above is independent of the recovery strategy R used and the choice of the initial active configuration G_b'. It states that some R and G_b' exist making t-step recovery possible for all sequences of up to t faults.

Example 1: Consider the graphs shown in Fig. 2. H_r is clearly 1-FT with respect to H_b since if G_b' comprises nodes B and C, the system can recover in one step by replacing the faulty node B (C) by the spare node D (A). Note that if the subgraph consisting of A and B is chosen as G_b', recovery requires two steps in the event of the failure of node B. In this case, the active node A must also be replaced by one of the spare nodes C or D. □

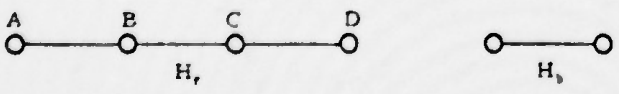


Fig. 2. Example of a system H_r that is 1-step recoverable with respect to H_b.

The calculation of the fault tolerance, i.e., recovery measures k and t for arbitrary graphs G_r and G_b is very difficult. In order to find out if G_r tolerates a given fault F, it is necessary to determine if the graph G_r' representing the faulty system contains a subgraph isomorphic to G_b. This is the well-known subgraph isomorphism problem. It may be necessary to examine all subgraphs of G_r that are isomorphic to G_b in order to determine if G_r is t-SR with respect to G_b. While the general subgraph isomorphism problem is computationally very complex, efficient (polynomial time) algorithms are known for many special classes of graphs, while efficient heuristic procedures are known for the general case [7].

III. OPTIMAL t-STEP RECOVERY

It is clearly desirable that G_b' and G_s'' should share as many unaltered nodes as possible in order to minimize the recovery time. The fastest recovery will be achieved when none of the fault-free active nodes of G_b' are affected in forming G_s'', i.e., exactly t spare nodes are used to replace the t faulty nodes.

Definition 5: G_r is optimally t-SR with respect to the n-node system G_b if

(1) G_r is t-step recoverable with respect to G_b;

(2) G_r contains the minimum possible number of nodes, namely, n+t;

(3) G_r contains the fewest edges among all redundant systems satisfying (1) and (2).

We now show that every nontrivial connected basic system G_b has a unique and easily-characterized optimal t-SR realization G_t^{opt}.

Theorem 1: Let G_t^{opt} be formed from G_b as follows. Introduce t spare nodes $s_1, s_2, \ldots, s_t$ and introduce edges connecting each s_i to every node in G_b and the t-1 nodes s_j where $i \neq j$. G_r is optimally t-SR with respect to G_b if and only if $G_r \equiv G_t^{opt}$.

Proof: First we show that G_t^{opt} is t-SR if G_b is the original active subsystem. Let x be any faulty node. x can be replaced in one step by any spare node s_i, since s_i is adjacent to all the nodes that are adjacent to x. Any sequence of t node failures can be tolerated similarly, since every node in G_t^{opt}, including the t original spares, can be replaced by a spare in one step. Thus the t spares allow t faulty nodes in G_b to be replaced in t steps, implying that G_t^{opt} is t-SR with respect to G_b.

Let G_t^* be any optimal t-SR system. We now show that G_t^* contains a subgraph isomorphic to G_t^{opt}, hence $G_t^* \equiv G_t^{opt}$. Let G_b^* be the initial active subsystem of G_t^*, so that $G_b^* \equiv G_b$. Let the t nodes of $G_t^* - G_b^*$ be designated $s_1^*, s_2^*, \ldots, s_t^*$. It remains to show that each s_i^* is adjacent to every node of G_b^*. Suppose by way of contradiction that s_i^* is not adjacent to y_j^*. There are two possible cases:

Case 1: $y_j^* \in G_b^*$. (Since G_b is nontrivial, G_b^* contains at least two nodes.) Let $y_k^* \in G_b^*$ and let y_j^* and y_k^* be adjacent. Suppose that a sequence of t nodes failures occurs affecting y_k^* and each of the spare nodes activated to replace y_k^*. At some point s_i^* must be used to replace y_k^* since G_t^* is t-SR and only t spare nodes are available, including s_i^*. However s_i^* is not adjacent to y_j^* and $y_j^* y_k^*$ is an edge of G_b^*, hence s_i^* cannot replace y_k^*. Consequently G_t^* is not t-SR, a contradiction. Thus s_i^* must be adjacent to every node of G_b^*.

Case 2: $y_j^* \in G_t^* - G_b^*$, i.e., $y_j^* = s_j^*$. Again consider a sequence of t node failures. After fewer than t-1 failures either s_i^* or s_j^* must be activated, say s_i^*. s_j^* has at least one neighbor z_k^* which is part of the currently active system. Suppose that all subsequent faults involve z_k^* and its replacements. Eventually s_i^* will be the only nonfaulty spare node available to

replace z_i^*. Since s_i^* is not adjacent to s_j^* (which is now part of the active subsystem), s_j^* cannot take the role of z_i^*, hence G_t^* cannot tolerate the t-fault in question, a contradiction. Thus s_i^* is adjacent to every node $s_j^* \neq s_i^*$.

We have shown therefore that the spare nodes of G_t^* are connected to every node of G_b^* so G_t^* and G_t^{opt} are isomorphic. Hence every optimal t-SR system is isomorphic to G_t^{opt}. □

Example 2: Fig. 3a shows a basic graph I_b, and Fig. 3b shows the corresponding optimal 2-SR system I_2^{opt} obtained by the procedure described in Theorem 1. □

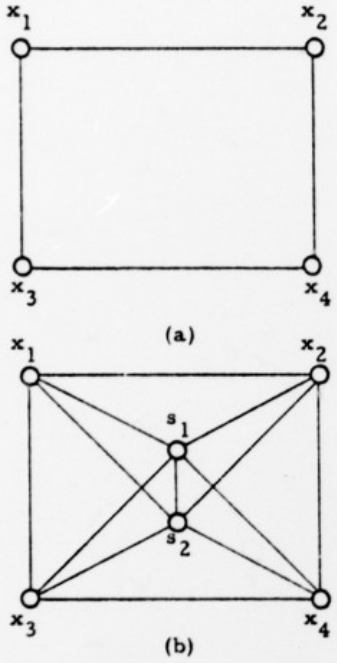


Fig. 3. (a) A basic graph I_b. (b) The corresponding optimal 2-SR graph I_2^{opt}

Optimal t-SR systems can also be characterized in terms of their clique graphs. Let K_n denote a complete graph of n nodes, i.e., an n-node graph containing all possible edges.

Definition 6 [8]: A clique of a graph G is a maximal complete subgraph of G. The clique graph K(G) of G is the intersection graph formed by the cliques of G, i.e., there is a one-to-one correspondence between the cliques of G and the nodes of K(G), and two nodes in K(G) are adjacent if and only if the intersection of the corresponding cliques in G is nonempty.

Theorem 2: If G_r is an optimal t-SR realization of some G_b, then $K(G_r)$ is complete.

Proof: Suppose $K(G_r)$ is not complete. Then G_r has two cliques C_1 and C_2 which have no node in common. There exists a spare node in the initial configuration of G_r which is not adjacent to any nodes in C_1 or C_2. Hence G_r cannot be isomorphic to G_t^{opt} and so, by Th. 1, it is not optimally t-SR, a contradiction. Hence $K(G_r)$ must be complete. □

Fig. 4 shows the clique graphs for H_r and I_2^{opt} from Figs. 2 and 3, respectively. H_r has three cliques isomorphic to K_3. Since two of these cliques are disjoint, $K(H_r)$ is not complete. I_2^{opt} has four cliques isomorphic to K_4, hence $K(I_2^{opt}) \equiv K_4$. Note that the optimal 1-SR graph for H_b in Fig. 2 is K_3, and $K(K_3) \equiv K_1$.

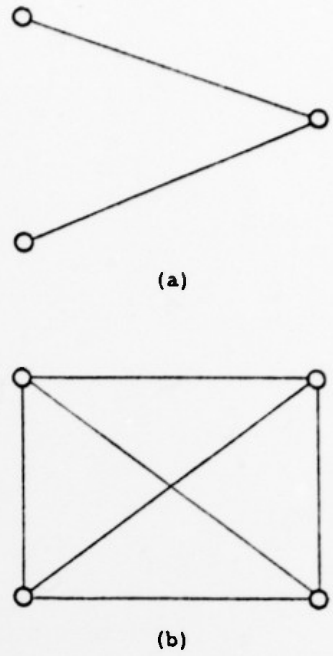


Fig. 4. Clique graphs (a) for H_r of Fig. 2. (b) for I_2^{opt} of Fig. 3.

IV. GENERALIZED t-STEP RECOVERY

The optimal t-SR design considered in the preceding section have the disadvantage that the maximum node degree in G_t^{opt} can be very large. If G_b contains n nodes then the spare nodes s_i in G_t^{opt} have degree n+t-1, which is the maximum possible degree in an (n+t)-node graph. Node degree corresponds to the number of input/output ports of a processor, or its fanout, and this is usually limited by physical considerations. In the case of microprocessors, the number of parallel data paths that can be connected to the microprocessor is severely restricted by integrated circuit pin limitations. Thus it is of interest to consider nonoptimal redundant systems in which node degree is limited.

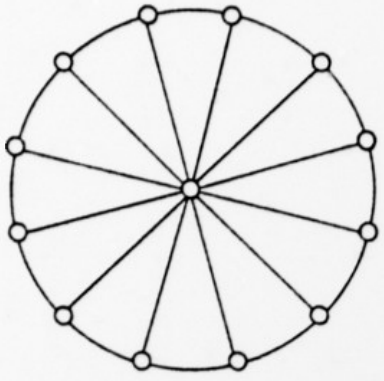
In the definition of t-SR given earlier it was assumed that the system was required to tolerate up to t faults. We now give a more general definition in which the number of faults tolerated and the number of recovery steps are distinguished.

Definition 7: G_r is k-fault tolerant t-step recoverable (k-FT/t-SR) with respect to G_b if G_r can recover from up to k faults in G_r in at most t recovery steps, that is, by changing node states or identities at most t times.

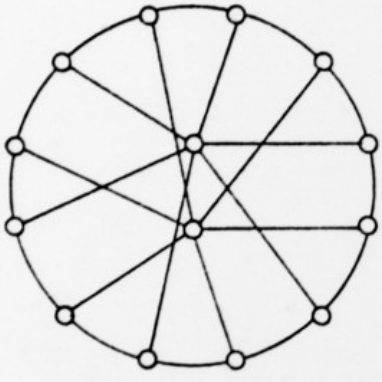
In general, $k \leq t$. When $k = t$ the system will also be called simply t-SR conforming with the earlier definition.

Example 3: Fig. 5 shows three different 1-FT realizations of the basic graph C_{12}, which is the cycle with 12 nodes. Fig. 5a shows the optimal 1-FT/1-SR graph as defined by Th. 1. Note that the central "spare" node has degree 12. Fig. 5b shows another 1-FT/1-SR version of C_{12} which contains two spare nodes and so is nonoptimal; however, its maximum node degree is only 6. The graph in Fig. 5c is the 1-FT realization of C_{12} which, as proven in [5], contains the minimum number of edges. It also has the smallest possible node degrees, however, it is 8-SR.

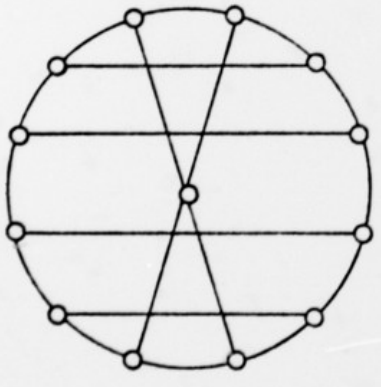
Thus there are fundamental tradeoffs involving the number of spares, the maximum node degree, and the maximum number of recovery steps t.



(a)



(b)



(c)

Fig. 5. Three 1-FT/t-SR realizations of C_{12}

As noted in §2, the computation of k and t for arbitrary k-FT/t-SR systems is very difficult. There are two possible ways in which this computational complexity problem can be avoided.

(1) We can restrict our attention to graphs with properties such as structural regularity which simplify fault analysis.

(2) We can attempt to transform the given graphs into graphs that are easy to analyze, and are such that the fault tolerance properties of the original graphs can be obtained from the transformed graphs.

In the remainder of this paper, we examine a special class of graphs called line graphs for which fault analysis is relatively easy. Moreover, an arbitrary graph can readily be converted into a line graph by the addition of nodes and edges [8]. First we define and characterize line graphs.

Definition 8 [8]: The line graph of a graph G, denoted L(G), is a graph whose nodes are in one-to-one correspondence with the edges of G. Two nodes in L(G) are adjacent if and only if the corresponding edges of G are adjacent. If H is a line graph, then there exists a graph G such that L(G) is isomorphic to H. G is called the root graph of H and will be denoted by $L^{-1}(H)$.

It is obvious that every graph has a line graph, however it is not necessary for every graph to be a line graph of another graph. Very efficient algorithms are known for determining if G is a line graph and, if it is, for generating its root graph [9]. Line graphs have been studied extensively; the following theorem summarizes their major characteristics. Let $K_{1,n}$ denote the star graph [8] which contains n+1 nodes, and n edges, with n of the nodes joined to the remaining node.

Theorem 3 [8]: Properties of line graphs.

(a) If G_1 and G_2 are any two nontrivial connected graphs except K_3 and $K_{1,3}$, then $L(G_1)$ is isomorphic to $L(G_2)$ if and only if G_1 is isomorphic to G_2.

(b) G is isomorphic to L(G) if and only if G is a cycle.

(c) If G is a line graph then the edges of G can be partitioned into complete subgraphs $\{C_i\}$ in such a way that no node lies in more than two of the subgraphs, and there is a one-to-one correspondence between $\{C_i\}$ and the nodes of $L^{-1}(G)$.

(d) Line graphs of regular graphs with degree d are regular with degree 2(d-1).

Example 4: Fig. 6 illustrates Th. 3c. The complete subgraphs $\{C_i\}$ in the line graph L(J) correspond to the nodes $\{x_i\}$ in its root graph J. □

Def. 8 implies that we can define a function L that transforms a graph into its line graph, and a function L^{-1} that transforms a line graph into its root graph. The following notation is also useful

$$L^{i+1}(G) = L(L^{i}(G))$$

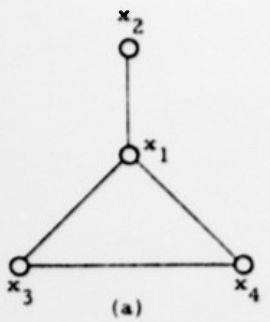
$$L^{-(i+1)}(G) = L^{-1}(L^{-i}(G))$$

where $i \geq 1$. Menon [10] has shown that $L^{-1}(G)$ has fewer nodes than G if G is not a cycle or a path, hence $L^{-1}(G)$ is usually simpler than G. We will now show that if a redundant system G_r is a line graph, many of its properties pertaining to fault tolerance can be determined with less computation from $L^{-1}(G_r)$.

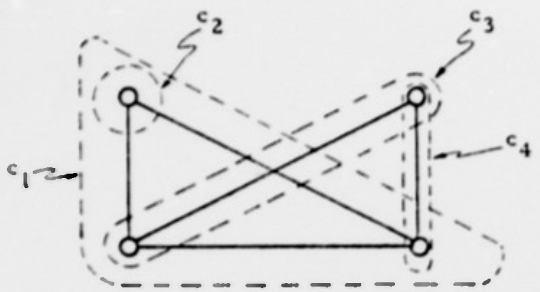
Thenrem 4: If G_r is k-FT with respect to G_b, then $L(G_r)$ is k-FT with respect to $L(G_b)$.

Proof: Th. 3c implies that a one-to-one correspondence exists between the nodes $\{x_i\}$ of G_r and a subset $\{C_i\}$ of the complete subgraphs of $L(G_r)$ where the $\{C_i\}$ include all nodes of G_r. Suppose a k-fault in $L(G_r)$ effectively eliminates a set S of k nodes to form a new graph H. Let $C_1, C_2, \ldots, C_j$ be any set of $j \leq k$ members of $\{C_i\}$ that contain S, and let H^* be the result of removing $C_1, C_2, \ldots, C_j$ from $L(G_r)$. There are j nodes $x_1, x_2, \ldots, x_j$ in G_r such that x_i corresponds to C_i in $L(G_r)$ for $i = 1, 2, \ldots, j$. If G_r^* is the result of removing these j nodes from G_r, then

$L(G_r^*) = H^* \subset H$.



(a)



(b)

Fig. 6. (a) A graph J. (b) Its line graph L(J) showing the complete subgraphs $\{C_i\}$ of L(J) that correspond to the nodes $\{x_i\}$ of J.

Since G_r is k-FT with respect to G_b and $j \leq k$, we conclude that $G_b \subset G_r^*$. Hence

$$L(G_b) \subset L(G_r^*) \subset H$$

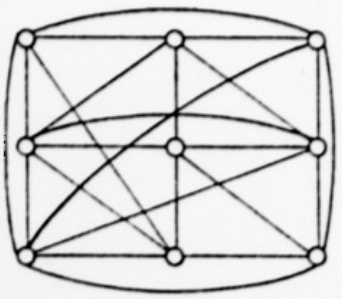
implying that H, which represents $L(G_r)$ with a k-fault present, contains a subgraph isomorphic to $L(G_b)$. It follows that $L(G_r)$ is k-FT with respect to $L(G_b)$. □

Note that the converse of Th. 4 is false.

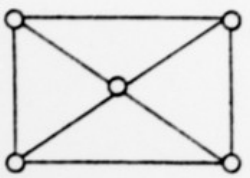
Theorem 5: If G_r is k-SR with respect to G_b, then $L(G_r)$ is k-FT/(2kd-k)-SR with respect to $L(G_b)$ where d is the largest degree of any node in G_b.

Proof: As in the proof of Th. 4, every set of k nodes in $L(G_r)$ is contained in $j \leq k$ complete subgraphs $C = \{C_1, C_2, \ldots, C_j\}$ which correspond to nodes $X = \{x_1, x_2, \ldots, x_j\}$ in G_r. Since G_r is k-SR, G_r can recover from the removal of X in at most k steps, i.e., by changing the state or identity of at most k fault-free nodes. Every clique in $L(G_r)$ contains at most d nodes. Hence $L(G_r)$ can recover from a k-fault in C by deactivating at most kd-k fault-free nodes in C, i.e., by removing C, and by changing the states or identities of an additional kd nodes to replace C. Hence $L(G_r)$ can recover from a k-fault in at most 2kd-k steps. □

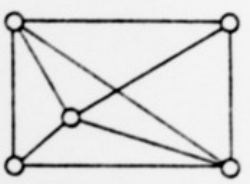
Example 5: Figs. 7a and 7b show two line graphs P_r and P_b. Consider the problem of determining values of k and t such that P_r is k-FT/t-SR with respect to P_b. The problem is greatly simplified if we replace P_r and P_b by their root graphs $L^{-1}(P_r)$ and $L^{-1}(P_b)$ which appear in Figs. 7c and 7d, respectively. By Th. 1, $L^{-1}(P_r)$ is optimally 1-SR with respect to $L^{-1}(P_b)$. The maximum node degree of $L^{-1}(P_b)$ is three, hence by Th. 5, P_r is 1-FT/5-SR with respect to P_b.



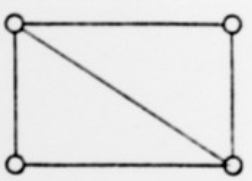
(a)



(b)



(c)

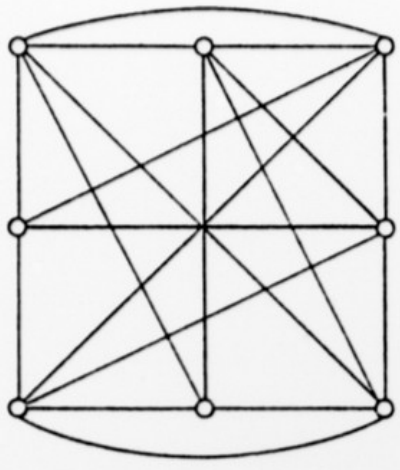


(d)

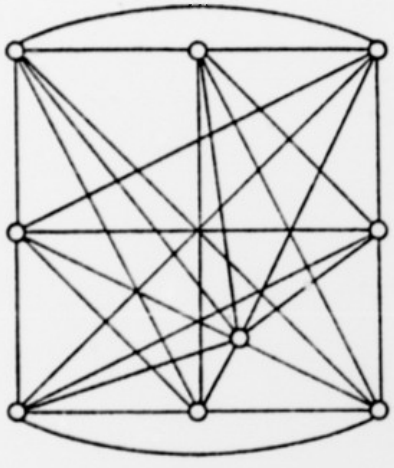
Fig. 7. (a) The redundant graph P_r. (b) The basic graph P_b. (c) The root graph $L^{-1}(P_r)$. (d) The root graph $L^{-1}(P_b)$.

Theorems 4 and 5 can also be used to construct k-FT/t-SR systems with nodes of lower degree than the corresponding optimal t-SR systems, the case of regular basic graphs. (A graph is regular if all its nodes have the same degree d.) The reduction in the node degree of G_r becomes more apparent as d increases. The following example illustrates this.

Example 6: Suppose a 1-FT realization of a certain regular graph Q_b is required where Q_b has 20 nodes of degree 8. Fig. 8a shows $L^{-1}(Q_b)$. (We omit the diagram for Q_b because of its complexity.) Using Th. 1 the graph $L^{-1}(Q_r)$ shown in Fig. 8b can be constructed. $L^{-1}(Q_r)$ is an (optimal) 1-SR realization of $L^{-1}(Q_b)$. Now construct the line graph Q_r of $L^{-1}(Q_r)$, which by Th. 5, is a 1-FT/9-SR realization of the original system Q_b. Q_r has 28 nodes, and 12 is its maximum node degree. While Q_r has far more spare nodes than the optimal 1-SR realization of Q_b, the latter contains nodes with degree 20. □



(a)



(b)

Fig. 8. (a) The graph $L^{-1}(Q_3)$. (b) An optimal 1-SR realization of $L^{-4}(Q_3)$.

REFERENCES

1. H. Sullivan and T. R. Bashkow: "A Larg-Scale Homogeneous, Fully Distributed Parallel Machine, I," Proc. Fourth Ann. Symp. Computer Architecture, pp. 105-117, March 1977.
2. M. C. Pease: "The Indirect Binary n-Cube Microprocessor Array," IEEE Trans. Computers, vol. C-26, pp. 458-478, May 1977.
3. H. Frank and I. T. Frisch: Communication, Transmission and Transportation Networks, Addison-Wesley, Reading, Mass., 1971.
4. F. P. Preparata, G. Metze and R. T. Chien: "On the Connection Assignmant Problem of Diagnosable Systems," IEEE Trans. Electronic Computers, vol. EC-16, pp. 848-854, Dec. 1967.
5. J. P. Hayes: "A Graph Model for Fault-Tolerant Computing Systems," IEEE Trans. Computers, vol. C-25, pp. 875-884, Sept. 1976.
6. R. Troy: "Dynamic Reconfiguration: An Algorithm and its Efficiency Evaluation," Proc. 7th Int'l. Conf. on Fault-Tolerant Computing, pp. 44-49, June 1977.
7. D. C. Schmidt and L. E. Druffel: "A Fast Backtracking Algorithm to Test Directed Graphs for Isomorphism using Distance Matrices," Journ. ACM, vol. 23, pp. 433-445, July 1976.
8. F. Harary: Graph Theory, Addison-Wesley, Reading, Mass., 1969.
9. P. G. H. Lehot: "An Optimal Algorithm to Detect a Line Graph and Output its Root Graph," Journ. ACM, vol. 21, pp. 569-575, Oct. 1974.
10. V. Menon: "On Repeated Interchange Graphs," Amer. Math. Monthly, vol. 73, pp. 986-989, 1966.

John P. Hayes was born in Newbridge, Ireland on March 3, 1944. He received the B. E. degree from the National University of Ireland, Dublin, in 1965 and the M. S. and Ph. D. degree from the University of Illinois, Urbana, in 1967 and 1970, respectively, all in electrical engineering.

From 1965 to 1967 he was with the Digital Computer Laboratory at the University of Illinios, where he participated in the design of the ILLIAC 3 computer. In 1967 he joined the Switching Systems Group at the Coordinated Science Laboratory of the University of Illinois, where he worked in the area of fault diagnosis of digital systems. From 1970 to 1972 he was a member of the Operations Research Group at the Shell Benelux Computing Centre of the Royal Dutch/Shell Company in the Hague, the Netherlands, where he was involved in operations research and software development. Since 1972 he has been with the Departments of Electrical Engineering and Computer Science at the University of Southern California, Los Angeles, where he is currently an Associate Professor. His research interests include fault-tolerant computing, computer architecture, switching theory and microprocessor/microcomputer-based systems. He was Technical Program Chairman of the 1977 International Symposium on Fault-Tolerant Computing (FTCS-7). He is the author of the book Computer Architecture and Organization (New York, McGraw-Hill, 1978).

Dr. Hayes is a member of IEEE, ACM and Sigma Xi.

Raif Yanney was born in Cairo, Egypt on February 29, 1944. He received the B. S. degree in electrical engineering from Cairo University in 1965, the M. S. degree in electrical engineering from Pratt Institute, Brooklyn, New York in 1970, and the Engineer degree in electrical engineering from the University of Southern California, Los Angeles, California in 1975. He is currently studying for his Ph. D. in electrical engineering at the University of Southern California.

Mr. Yanney joined the Department of Neurology, Cornell University Medical College, New York as a research associate in 1967, a position he held until 1973. He joined Hughes Aircraft Company, Culver City, California in 1973, first as a member of the technical staff and then as a staff engineer. His responsibilities with Hughes Aircraft Company include the analysis and design of fault tolerant digital systems and self-diagnosable digital systems. His main interests are computer architecture, fault diagnosis of digital systems and fault tolerant systems. He is a member of IEEE and Sigma Xi.