AD-A040 518 UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES ELECTR--ETC F/G 9/3
SYSTEM TESTING AND DESIGN FOR DIAGNOSABILITY.(U)
JUL 76 M A BREUER, A D FRIEDMAN, J P HAYES N00014-67-A-0269-0019

UNCLASSIFIED NL

| OF |
AD A040518

END

DATE FILMED
7-77

0484 299

(9) FINAL REPORT. (1)

on

(15) ONR CONTRACT N00014-67-A-0269-0019, AND N00014-76-C-0168

(6) SYSTEM TESTING AND DESIGN FOR DIAGNOSABILITY.

(11) July 1976

(12) 17p.

Co-Principal Investigators:

(10) M. A. Breuer
A. D. Friedman
J. P. Hayes

Students:

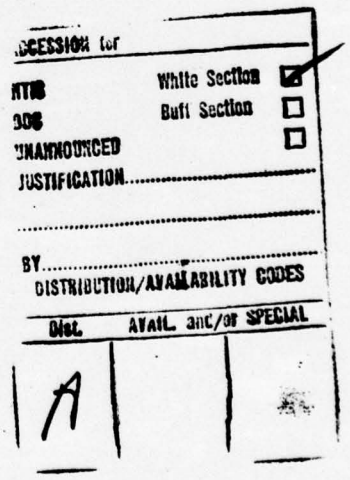
Y. Levendel
M. Marouf
J. Shen

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

Distribution:

J. Trimble, ONR-Washington
S. W. Swart, ONR-Pasadena
H. Carrier, Government Contracts-USC
M. A. Breuer
A. D. Friedman
J. P. Hayes

The work described in this final report is divided into two sections. Section A describes work initiated in the last year and Section B describes continuation of work previously begun.



SECTION A

I. Algebraic Methods for Synchronizing Sequence and Test Sequence Generation for Sequential Circuits.

It is known that many practical sequential circuits cannot be treated by single multivalued systems. The reason for this failure is the loss of information, which is inherent to the use of x as a unique unknown.

It is natural to overcome this difficulty by algebraic methods. One technique has been proposed by S. S. Yau and Y. S. Tang (1) and is well defined for synchronous circuits. A modification has been done by M. J. Flomenhoft (2) by computing the "synchronous equivalent" of an asynchronous circuit. These techniques are similar for synchronizing sequence generation and for test sequence generation and use the equations of the circuit.

In our work, a simpler way of computing the synchronous equivalent has been proposed and some suggestions for speeding up the procedure of sequence generation have been brought.

Since, the equational methods use resubstitution techniques, it was interesting to investigate the possibility of replacing substitution by matrix operations. This was done by using the state transfer matrix of the circuit as its description. The state transfer matrix is a matrix, the entries of which represent Boolean condition on the

state transitions: m_{ij} is the Boolean condition on the transition from state j to state i.

It was possible to compute synchronizing and test sequences for synchronous circuits. A method of computing the synchronous equivalent of the matrix representation of an asynchronous circuit has been developed.

Another matrix representation using Reed Muller form was also found and a linear representation of the sequential circuit was produced. This matrix representation made possible to solve the synchronizing and test sequences generation problem, and it also provided a way of analyzing cycles and paths in the state transfer diagram of an asynchronous circuit by using a Jordan decomposition of the linear matrix.

Finally, a test generation procedure using conditional fault lists and their logical propagation was proposed. This method produces all of the tests for a set of faults in any combinational or sequential circuits. This method made possible the computation of a simultaneous synchronizing sequence for the good machine and a set of bad machines (if the sequence exists).

II. Design of Self Checking and Easily Diagnosable Systems

With the increasing use of digital systems, it is desired that the systems perform more reliably. Classically the solutions to design problems have been subject to the minimization of hardware complexity.

The testing problem is considered only after the design of logic circuit have been completed. A software diagnostic program is usually generated for a given design. It is applied, frequently, to diagnose the design.

Software diagnostic programs represent time redundancy in the system. It is no surprise that such programs are in general lengthy, since the testing criteria was not included during the design phase.

By incorporating testing into design algorithms, we can reduce the size of diagnostic programs or even eliminate them. Two related problems have been examined:

1. Design of Self Checking Circuits:

A system is said to be self checking if all the faults in its circuits are tested during its normal operation and these circuits also have automatic error detecting capabilities for every false output.

The design of self checking circuits can be achieved by using coding techniques. Binary information is coded in a specific code, for which a self checking checker design is possible.

Codes that have exactly m ones and n-m zeros are referred to as m-out-of-n (m|n). They are useful because of their ability to detect a type of errors most likely to occure in computers. In previous work, design of totally self checking checkers for specific codes, namely m|2m and m|2m+1 [3, 4], were given.

We have investigated the general problem of designing a totally self checking checker for any arbitary m|n code. The following results

were obtained:

1. A general theory for designing a totally self checking checkers for m|n codes is established.

2. All the previous results are special case of our results.

3. An efficient algorithm to design a totally self checking checker for m|(2m+2) code.

4. The design problem of a totally self checking checker for an arbitary m|n code is equivalent to a combinatorial problem called the Ramsey Numbers.

The Berger code is a separable code (the information bits are separated from the check bits). A very efficient procedure for designing a self checking checker for an optimal Berger code have been founded.

The general problem of designing a self checking checker for other types of separable codes (e.g., arithmetic codes) are under investigation. These codes are promising for designing self checking sequential circuits.

2. Easily Testable Combinational Realizations:

The problem of functional decomposition, to simplify testing, has been investigated. A Boolean function is to be decomposed into sub-functions which have nice properties from testing viewpoint. The classes of unate functions and linear functions have such interesting properties.

An efficient procedure to decompose a combinational function f in the form:

$f = f_1 + f_2$ where

f_1 : is a linear function, and

f_2 : is a unate function has been developed using the Walsh transform of Boolean functions.

Another procedure to decompose a function f in the form:

$f = f_1 + f_3$ where

f_1 : is a linear function, and

f_2 : requires a minimal number of tests has been developed and is currently being improved.

SECTION B

The work described here is divided into two sections. Section 1 deals with the theory of testing digital circuits using automatic test equipment. The major problems here are test generation, fault simulation and test hardware. Section 2 deals with the problems of designing systems so that they are easier to test, or are automatically tested by the system itself.

1. TEST GENERATION AND SIMULATION

1.1 Automatic Test Generation

A great amount of work has gone into the theory and development of software systems capable of automatically generating tests for digital systems. This area of fault tolerant computing has had a significant impact on digital systems development and maintenance. However, even with this success, there are still numerous problems associated with automatic test generation. Some of these problems are listed below:

a) modeling of complex circuits

b) race and hazard analysis

c) circuit initialization

d) functional level components

e) buried flip flops.

Because of these problems, modern test generation systems cannot cope with the complexity of today's technology, such as LSI chips containing 2000 or more equivalent gates.

During the last few years we have been developing ways for overcoming some of these deficiencies. Specifically we have worked on the problems of model construction for complex digital circuits as well as how to eliminate races and hazards from tests generated via automatic means. Our results have appeared in three recent papers (Numbers 6, 7, 9 on list of publication).

In addition we are engaged in the analysis of a new test generation problem where one is able to inject signals internal to the circuit. This type of testing corresponds to an in-circuit-component tester, sometimes referred to as a "bed-of-nails" tester. Due to the signals injected on the output of devices, abnormal circuit operation occurs. We plan to pursue this work because it offers the greatest potential for deriving tests for complex circuits.

1.2 Testing Logic Circuits by Transition Counting and Related Methods

Several digital testers used quite extensively in industry employ a technique called transition counting for fault diagnosis. In these systems, the number of times a test response sequence changes state (the transition count) is observed rather than the entire response sequence. Transition count (TC) testing has two major advantages.

(1) Fault dictionary size can be drastically reduced by recording only transition counts.

(2) Very simple testers can be used, since only a means of counting and comparing transitions counts is required.

We have studied the basic properties of TC tests, and have developed techniques for generating optimal or near-optimal TC tests for certain classes of circuits (Papers 10, 12 on list of publications).

TC testing is based on the detection of 0/1 transitions in binary sequences. A question that naturally arises is: are there other properties of binary sequences whose detection can provide equally good, or even better, testing methods? Furthermore, can we provide some a priori measures of the cost and performance of such testing methods? Two suitable measures suggested by the characteristics of TC testing are: the complexity of the test equipment used, and the degree of fault detection or location that can be obtained. We have attempted to answer the foregoing questions by introducing a general class of testing methods called check sum (CS) tests which include both conventional testing and transition count testing as special cases. CS tests can be arranged in a hierarchy based on a "memory" parameter m which is a simple measure of tester complexity. A concept of fault resolution is defined for CS tests which measures their ability to detect and locate faults. We have also identified and analyzed some special classes of CS tests. One class of tests called zero-memory CS tests, have some unique and potentially useful properties (this work is reported in Papers 13 and 15 in the list of publications).

2. DESIGN FOR TESTABILITY

2.1 Design of Diagnosable Systems Using Test and Control Logic

Work was continued on the problem of test and control logic insertion at the gate level. In particular, the extension of our previous test point placement algorithms to cover both fault detection and fault location was considered. Some heuristic algorithms were developed to permit near-optimal test point selection in situations where the circuits involved are complex, e.g., contain reconvergent fanout.

The extension of this work to the system level of design was initiated. We have studied the relationship between our models for control logic and the "blocking gate" approach of Ramamoorthy, Mayeda, et al. It appears that we can extend our definition of control logic to include the blocking gate approach. This extension promises to provide useful insights into design for diagnosability at the system level.

2.2 Partially Self Checking Circuits

Another important area within fault tolerant computing is the design of self checking circuits. In such circuits faults are automatically detected on line by the use of codes on both inputs and outputs, and by hardware checkers (instead of diagnostic programs). Thus a totally self checking system could devote all of its time to data processing instead of requiring periodic application of diagnostic programs for maintenance.

Most of the research in this subject has attempted to design self checking circuits for checking specific codes (parity code, m/2m code, residue code). In practice circuits cannot be made totally self checking if we assume the existence of a single error signal. Thus it might be advantageous to consider the design of partially self checking circuits which are fault secure but only partially self testing. A measure of the effectiveness of the design would be the degree of self testing vs the amount of redundancy. We have shown that for a specific type of code the use of three outputs instead of two leads to a dramatic decrease in circuit complexity while causing only a very small increase in hardcore.

3. SYSTEM LEVEL DIAGNOSIS

3.1 Measures of Diagnosability

In recent years several attempts have been made to formulate a model of digital system diagnosis. For the most part the results obtained from these models fail to accurately measure the actual diagnosability of systems. One reason for this is that the measures of diagnosability utilized in these models fail to consider the probabilities associated with system failure and imply a very conservative philosophy of system repair in that only definitely faulty modules are diagnosed and replaced. In addition, artificial and unrealistic assumptions are made on the system which severely limit the usefulness of the model. Our recent work in this area has attempted to overcome these limitations and we have made considerable progress as follows:

(1) We have developed a new measure of diagnosis which enables a more realistic appraisal of a systems' diagnosability and corresponds more closely to actual system repair. Each system module is identified as definitely faulty, possibly faulty, or definitely not faulty. If for any set of at most t faults there are at most s modules identified as definitely or possibly faulty, the system is t/s (t out of s) diagnosable and s-t is a measure of the efficiency of repair. We have analyzed this measure for a canonical class of systems and have shown how this measure greatly affects the appraised diagnosability of these systems. Results on such systems can be used to determine a limit on the

diagnosability of more general types of networks. Designs which are optimal with respect to this model have been developed.

(2) We have developed a new algebraic model of system diagnosis which does not require the artificial restrictive assumptions of other models, and which can be used to analyze easily a test outcome to determine those modules which are definitely or possibly faulty.

(3) Procedures for sequential diagnosability have been developed. The use of sequential diagnosability for analysis of intermittent faults is under study.

Papers published or submitted for publication during 1974-76, which were funded wholly or in part by ONR Contract.

[1] J.P. Hayes, "On modifying logic networks to improve their diagnosablility," IEEE Transactions on Computers, vol. C-23, pp. 56-62, January 1974.

[2] J.P. Hayes and A.D. Friedman, "Test point placement to simplify fault detection," IEEE Transactions on Computers, vol. C-23, pp. 727-735, July 1974.

[3] J.P. Hayes, "Modeling faults in digital logic circuits," (invited paper) Symposium on Rational Fault Analysis, Texas Tech University, Lubbock, Texas, August 1974.

[4] A.D. Friedman, "Diagnosis of short-circuit faults in combinational circuits," IEEE Transactions on Computers, vol. C-23, pp. 746-752, July 1974.

[5] A.D. Friedman, "A new measure of digital system diagnosis," Proceedings Int'l Fault Tolerant Computing Conference, Paris, France, June 1975.

[6] M.A. Breuer and L. Harrison, "Procedures for eliminating static and dynamic hazards in test generation," IEEE Transactions on Computers, vol. C-23, pp. 1069-1077, October 1974.

[7] M.A. Breuer, "The effects of races, delays, and delay faults on test generation," IEEE Transactions on Computers, vol. C-23, pp. 1078-1092, October 1974.

[8] M.A. Breuer, S.J. Chang and S.Y.H. Su, "Identification of multiple stuck-type faults in combinational networks," IEEE Transactions on Computers, vol. C-25, January 1976.

[9] M.A. Breuer, "Modeling circuits for test generation," Proc. 1974 Int'l Symposium on Fault Tolerant Computing, June 1974, pp. 1/13-3/18.

[10] J.P. Hayes, "Testing logic circuits by transition counting," Proceedings International Fault Tolerant Computing Symposium, Paris, France, June 1975, pp. 215-219.

[11] M.A. Adham and A.D. Friedman, "Digital system fault diagnosis," submitted to Journal of Design Automation and Fault-Tolerant Computing.

[12] J. P. Hayes, "Transition count testing of combinational logic circuits," IEEE Transactions on Computers, vol. C-25, pp. 613-620, June 1976.

[13] J. P. Hayes, "Check sum test methods," Proceedings International Symposium on Fault Tolerant Computing, Pittsburgh, June 1976, pp. 114-120.

[14] J. P. Hayes, "On the properties of irredundant logic networks," to appear in IEEE Transactions on Computers.

[15] J. P. Hayes, "Check sum methods for test data compression," to appear in Journal of Design Automation and Fault Tolerant Computing.

REFERENCES

1. S.S. Yau and Y.S. Tang, "Generation of shortest test sequences for individual faults in sequential circuits," Submitted to IEEE, November 1971.

2. M.J. Flomenhoft, "Algebraic techniques for finding tests for logical faults in digital circuits," Ph.D. Dissertation 1973, Lehigh University.

3. D.A. Anderson and G. Metze, "Design of totally self-checking circuits for m-out-of n codes," IEEE Trans. on Computers, Vol. C-22, pp. 263-269, March 1973.

4. S.M. Reddy, "A note on self-checking checkers," IEEE Trans. on Computers, Vol. C-23, pp. 1100-1102, October 1974.