INTRODUCTION
The automatic diagnostic program has been with us for almost as long a time as the electronic digital computer, however the disparity between its achievements and its potentials has not been as successfully closed as it has for the computer itself. In the opinion of the author this is at least partly due to an only recently acquired understanding that the term “diagnostic program” is a misnomer insofar as it implies that the task is primarily one of programming rather than logic analysis.

For this reason, during the development of a test and diagnostic system (T&D) for the MICRORAC FIELDATA COMPUTER SYSTEM (RAC), a series of studies was undertaken to clarify both the performance attainable by a diagnostic program and the methods for achieving this performance. The results of these studies are summarized in the following sections.

The quality of a diagnostic program
The detection capability
This relates to the ability of the program to determine that a fault exists in the system. This may be integrated with the portion of the system that produces an actual fault diagnosis, however, it is desirable that the detection system be capable of rapid, easy application, since the system must be tested frequently enough to ensure that only a single fault exists at any one time. It is obviously important that the percentage of undetected faults be kept as small as possible, however this system has not been specifically designed to detect intermittents, marginal faults or faults in redundant logic. For example, in some cases, logic elements are gated by a particular timing pulse in order to ensure correct operation under worst case conditions. Since the worst case almost by definition, rarely occurs, it is clear that a failure which causes certain logic elements to be clocked at an improper time will generally not be detected. Similar conditions occur in certain logic areas used in the core memory system to minimize noise disturbances. Marginal testing techniques must be relied upon to deal with such faults.

The diagnostic resolution
In general, most diagnostic procedures will lead the operator to the conclusion that one of a small group of possible faults exist. Although it is frequently possible to refine the procedure by additional automatic semi-automatic or manual tests, to lead to a unique diagnosis, this is not always economically desirable because the RAC maintenance system includes the use of an automatic card tester capable of thoroughly testing every module on a card. The use of the tester, however, requires a fairly long time per card and also requires that cards be removed and reinserted into the RAC frame. This latter procedure, if repeated too frequently puts additional undesirable stresses on the connector and may increase its failure rate. For these reasons the resolution or number of possible faulty cards listed at each diagnosis varies from point to point. The final resolution reflects a compromise between the cost of developing additional diagnostic procedures and the desirability of minimizing unproductive usage of the automatic card tester.

The required operator skill level
The skill level required to maintain a digital computer could conceivably vary all the way from graduate design engineers to laymen completely unfamiliar with electronic equipment, but capable of following detailed explicit directions. Unfortunately many military systems have been built with the former requirement while few have approached the latter.

If a completely automatic system were developed to diagnose 100% of all faults to a single card, a degree of training would still be required because the present nature of computers requires tape handling, program loading, initialization and starting, card replacement, etc. These skills are essentially at the level of the computer operator rather than the maintenance man.

761
The time required for diagnosis

This should be distinguished from the total repair time and from the time required to extract and handle cards and modules because these latter times are more a result of computer design and operator training than diagnostic system design. For this reason diagnostic time is regarded as beginning with a loaded test and diagnostic program and ending when the operator possesses a list of fault cards. In considering a reasonable time requirement it is obviously misleading to call for a time disproportionate with the other time elements in the repair and maintenance cycle, i.e., a diagnostic time on the computer time scale of microseconds or milliseconds is useless because the operator activities required to complete the diagnostic cycle occur on the human scale of seconds or more likely minutes. On the other hand, some elements of the diagnostic test procedure will undoubtedly be used every day and so must take only a modest portion of allowed maintenance time. This suggests that a reasonable time might vary from 1 minute to 1 hour depending upon whether a quick reassurance of system operability is desired, to diagnosis of a subtle fault requiring manual intervention and analysis. In this light a goal of 10 minutes average diagnostic time has been used.

The additional equipment required to reach a diagnosis

Because the state of the art of computer design has not advanced to the point wherein a computer can be produced which is completely and automatically self-diagnosable, it is clear that provisions must be made for operator intervention to perform semi-automatic and manual tests. Such tests may require a wide range of equipment from a simple D.C. voltmeter to an additional test computer. The test philosophy adopted was to provide for a range of techniques and levels of manual intervention, largely at the discretion of the operator. Thus each diagnosis results not only in a list of cards but a set of references which usually indicate to the operator the following additional information:

1. The group of logic elements and gating signals being tested and their normal functions.
2. The normal results of the tests being performed.
3. The tests previously performed and functions and elements found to be correctly operating.
4. Logic drawings references.
5. Pin number and test point number identification data.

This information will frequently enable an operator to refine, verify or extend a diagnosis by simply measuring the D.C. potential or observing the presence or absence of a pulse at a particular test point.

The criteria for automatic diagnosis

To execute any automatic diagnostic program, it appears necessary that a fault not affect the ability of the computer to perform the following functions:

1. Load a program into memory.
2. Automatically sequence through the instructions of the program.
3. Automatically perform a comparison of a result with a known correct result.
4. Automatically execute a conditional transfer of control, as a result of a comparison.
5. Automatically print or display a result.

If a fault makes it impossible to perform any one of these five functions, it is not true that diagnosis is impossible but, rather that such a failure requires manual intervention for its interpretation and so contributes to an excessively long or complex diagnostic cycle.

It is quite difficult to apply these criteria to obtain precise numerical estimates because each of these functions can usually be performed in many ways, even in a computer not particularly designed with diagnostics in mind, much less a military computer such as RAC. For example, if program loading via magnetic tape is impossible due to element failure in one of the I/O converters, a few minutes of cable switching allows the magnetic tape to be loaded via a second I/O converter, or the magnetic tape may be replaced by a paper tape entry using the same converter. If the fault persists, short programs can be loaded, one instruction at a time via the console.

It is thus clear that there are many possible roads, albeit manually oriented, to the initiation of a final automatic diagnostic program. Therefore an estimate of the amount of hardware which must be operative to perform an automatic diagnostic program, in some sense of the word automatic, is largely limited at the top by the ingenuity of the program designer as well as the characteristics of the computer design. However, the problem of economic feasibility limits the number of possibilities which may be explored to provide alternate automatic or semi-automatic paths for automatic diagnostic execution. For this reason the test and diagnostic system for RAC makes no claim to exhaustive use of all possible as well as probable alternatives. However, among the redundancies included in the system are:

1. Program entry by any one of two I/O converters.
2. Program entry by either paper tape or magnetic tape.
3. The use of alternate core memory banks.
4. Results displayed either by a High Speed Printer on IOCNV# or a Flexwriter on IOCNV #2 or from console indicators directly on the B Register.
5. The use of logical instructions to synthesize a redundant comparison mode.

6. The use of a single instruction, console controlled, diagnostic mode for use in diagnosing failures which otherwise cripple the ability of the system to "breathe."

The relationship of card partitioning to diagnostic resolution

It has been observed that the placement of functionally related logic on the same or a small number of circuit cards makes it unnecessary to diagnose or analyze failures down to the logic element level. This occurs because diagnosis or isolation to a function then allows one to replace the card or cards concerned with performance of this function. Two objections exist to this viewpoint:

1. Because of a requirement for the MICRORAC system to minimize the number of card types and resulting inventory costs, RAC uses only 30 different board types. It was therefore not possible, in most cases, to place only functionally related logic on a board. Thus, to determine that a circuit board is faulty, it is generally necessary to determine that a specific logic element on the board is faulty.

2. Even if the circuit boards did contain only functionally related logic, the problem of diagnostic resolution would only be partially ameliorated. This occurs because faults which are not functionally related to a particular test, and therefore, could occur on any card, may still affect a particular test.

A more detailed analysis of logic failures indicates that the ability to use partitioning as an aid to diagnostic analysis is most significantly affected by the presence of two types of faults which may be categorized as:

Type 1 Faults: Those which prevent an action from occurring or:

Type 2 Faults: Those which cause an action to occur, improperly. Localization of Type 1 faults can be made easier by appropriate partitioning, because such faults are those which are ordinarily considered as being related to the function under consideration. However, localization of Type 2 faults is not aided by partitioning because these faults can originate anywhere in the computer.

It thus appears that the problem is not one of diagnosing to 1, 2 or 3 boards, but rather the diagnosis to 1, 2 or 3 elements, each of which may be on different boards. The magnitude of the latter problem may be demonstrated by observing that the Central Processor of MICRORAC contains about 160 cards. Diagnosis directly to 3 cards, would imply a resolution of approximately 2%. However, the number of elements in the Central Processor is about 5000 and diagnosis to 3 elements implies a resolution of 0.06%.

The effective number of logical elements in a diagnostic system

It has been demonstrated that the analysis for a diagnostic program cannot be content with a card by card analysis i.e., that essentially every logical element must be considered. However, an even deeper level of analytical detail is required i.e., every failure mode of an element must be analyzed in order to develop a thorough diagnostic program. This can be seen by considering the basic methods of diagnosis. All methods involve an element of prediction. The prediction process consists of analyzing the effects of a presupposed failure upon observable portions of a computer. This is essentially a method for building up a conceptual dictionary relating failures to symptoms. Obviously, every entry in the table results from selecting a logic element, assuming its failure and then deducing its symptoms. Each element can, however, fail in several ways, each of which can produce a characteristically different symptom. For example, consider a two input - non - inverting AND gate, with output Z and inputs A and B. The modes of operation of this gate including normal operation and all types on non-marginal faults are:

\[ Z = AB \]  
\[ Z = A \]  
\[ Z = B \]  
\[ Z = 0 \]  
\[ Z = 1 \]

In addition, two other modes exist in which each of the input diodes is shorted. Such an element failure does not alter the transfer function of the gate, however, it has the disturbing effect of altering the operation of gates sharing the same input leads, thus producing a sidewise sneak path.

In any case, it is clear that the prediction method requires the consideration and analysis of the equivalent failure mode of operation of every gate. In effect, if the average gate of flip-flop has six modes of failed operation, each normal logic element is replaced, sequentially, by six other logic elements. The effects of these new logic elements must be deduced, thus multiplying the required extent of analysis by six.
CONCLUSIONS

As an example of the successful application of these principles, the RAC Diagnostic System, consisting of approximately 22,000 instructions and data words, and over 7500 individual tests leading to specific diagnostic listings was subject to a two level acceptance test. 400 failures were inserted by RCA during test and debugging of the system and an additional 75 were inserted during government conducted acceptance tests. 100% of all faults were detected, 74% by explicit module listings, to a median resolution of 4 cards out of more than 500 cards. The median diagnostic time was 2 minutes. Implicit diagnosis of unit and function was provided for the remaining 26%.

ACKNOWLEDGMENTS

The author wishes to record the outstanding efforts of a few members of the group who contributed so heavily to the success of this program from conception to final test. They are H. Hellman and M. Kornfeld of the Simulation, Automation and Diagnostics Group. The programming efforts were directed by A. Rose, who performed masterfully in bridging the gap between logic analysis and programming. Assistance in significant areas was provided by P. Carides, J. DeSantis, J. Cohen, B. Burghen, M. Sendrow.