Fault diagnosis of digital systems—a review

R. G. Bennetts and D. W. Lewin

Department of Electronics, The University, Southampton SO9 5NH

The increasing complexity of digital systems over the past decade has been accompanied by a growing awareness of the need for efficient fault diagnosis, as proved by the ever increasing literature published on the subject. The paper is based on 86 referenced sources and its main function is to review the published methods of deriving diagnostic test sequences, indicating the advantages and disadvantages of each technique. In so doing, it traces the interaction between the diagnostic techniques that have evolved and their influence on the design philosophy of digital systems at all levels. It is apparent from the review that there exists a requirement for a unified theory of diagnosis compatible with, and complementary to, current design techniques based on switching theory and ways are suggested in which this might be achieved.

(Received August 1970)

The development of diagnostic routines for logical networks has progressed with the development of digital systems and the requirements of the latter has influenced the algorithmic format of the former. The early (pre 1960) computing systems possessed as their diagnostic aid, sets of programs written by programmers and engineers, and these were mainly orientated towards checking the order code of the machine and were therefore not capable of testing the logical functions. Subsequent diagnostic programs however were more concerned with the hardware, but in general did not provide an exhaustive check.

As a result of this, it became apparent that the overall philosophy of the fault diagnosis method would have to be modified and also that this was applicable throughout the entire system complex—from the system level such as a complete computing installation to the basic building module; at this time discrete components in standard configurations, and now LSI. A variety of techniques have been developed, each striving to produce an efficient diagnostic test sequence (DTS) that can be applied to the system in order to determine the correctness or otherwise of its function. In general, all the techniques involve the overall system in extra hardware and/or software and the particular requirements and specification of the system form a constraint on the diagnosis implementation. For further discussion on this, the reader is referred to the suggested papers for each technique; in particular, Manning [18] and Jones and Mays [38] comment on the implementation of certain of the techniques in the areas of computers and LSI respectively.

The variety of techniques that have evolved all seek to solve specific problems associated with a particular level within the digital system, and one of the main functions of this paper is to identify and summarise these techniques and indicate their applicability and shortcomings.

A further area of associated research has been the interaction between diagnosis and computer design. This has stimulated some interesting theoretical proposals and the ultimate aim is to achieve complete automatic self-diagnosing and repairing computers. The proposals are enumerated in a later section.

An extensive bibliography of 86 papers has been included and reference to most of these is given in the text. The papers are listed in chronological order and two are themselves bibliographies—namely Breuer [17] in 1966 and Salisbury and Enslow [23] in 1967. Breuer contains references to 61 papers and Salisbury and Enslow to 88. The overlap between these two is surprisingly small, being 30 in all thus collectively providing a source for 109 papers. Of these, 17 are included in the list attached to this paper and the three papers provide access to a further 178 papers—an indication of the formidable amount of material that has been generated, mostly since 1960.

It is as well to define formally some of the common terms in use and the following definitions are quoted from Seshu and Freeman [2].

'By a "test" we mean the process of applying a set of inputs to the machine and observing the corresponding outputs. The term "check-out" is used as a synonym for "failure detection" or the process of determining whether the machine is failure-free or not. "Diagnosis" is the process of identifying the failure, if one exists. Thus diagnosis includes checkout.'

Also in common usage are the terms malfunction, error and failure. These are loosely synonymous and are used to describe any deviation from the expected operation of the system, such deviations being the result of design errors, fabrication errors, equipment malfunctioning or program error [47]. The finer nuances of difference between some of these terms has been described by Muller [28]. Each technique, however, imposes its own bounds on the definition and these limitations are discussed more fully in a subsequent section.

In the text, the classification of networks into *combinational* and *sequential* follows the following rigid definitions of switching theory:

Combinational—A combinational logic network is one in which the output(s) obtained from the network is solely dependent on the present state of the input.

Sequential—A sequential logic network is one in which the output(s) obtained from the network is not only dependent on the present inputs, but also the past inputs. This implies storage and feedback of previous input conditions.

1. Techniques of fault diagnosis

The various techniques of producing a diagnostic test sequence (DTS) for fault diagnosis can be broadly categorised into seven main types. There is obviously overlap in some cases and some methods have evolved from others. The methods are summarised below in approximately the order in which they emerged and each summary is preceded with a list of the relevant papers.

1.1. *Partitioning* [1], [2], [9], [38]

This method was first proposed by Brulé *et al.* [1] but the real development was done by Seshu [2], [9]. It is normally applicable to existing rather than proposed systems, although Marlett [27] has used a modified partitioning technique to compare theoretical designs of computer central processing units. The concept can be applied to any class of system and indeed

de Atley [79] comments that it is a common method for manufacturing acceptance tests for LSI circuits.

The method basically consists of a digital computer simulation of the 'good' system along with N previously defined faulty versions. The response of each to an input test set is used to effect a partitioning of the $(N + 1)$ systems such that:

1. The good system is quickly isolated from all others—fault detection, or
2. All systems are uniquely identified—fault detection and location.

The resulting partitioning or decision tree defines a DTS such that when it is applied to the system, the response enables identification of one of the $(N + 1)$ categories.

The *sequential* or *combinational testing* procedure can both be applied to this technique. The combinational procedure implies that a fixed set of tests is applied and system identification does not occur until all the responses are received, whereas the sequential procedure uses the response of one test to determine the next. Seshu's paper contains a useful section on the advantages and disadvantages of the two procedures and the sequential procedure is now generally accepted as being more efficient.

Foremost among the attempts to implement this technique is the Sequential Analyser described by Seshu [9]. The set of tests can be pre-specified and the Analyser details the order in which they should be presented depending upon whether detection or full diagnosis is required—the selection process being assisted by the application of certain criteria (these are enumerated in Section 2.3).

Alternatively the Analyser will attempt to derive the tests itself using certain stratagems aimed at producing a non-minimal but optimal set; again detection or full diagnosis can be accommodated.

1.2. *One-dimensional path sensitising* [5], [13], [19], [80], [82]
The basic technique of one-dimensional path sensitisation relies on three processes:

1. The postulation of a known fault at a known location.
2. The propagation of the fault from its location to one of the primary outputs via a *sensitised path*, i.e. one along which any change in the logical value of the fault will result in a corresponding change at the appropriate primary output. This is called the forward-trace phase.
3. Implicit in the forward-trace phase is the setting-up of other elemental inputs and outputs and these can only be established by their predecessors—in the limit this being the primary inputs. This process is termed the backward-trace phase and the final set of primary inputs constitute the necessary test configuration for the postulated fault.

The major disadvantage with this technique is due to *re-converging fanout*, defined by Armstrong [13] in the following manner:

Fanout paths that reconverge again are referred to as 'reconverging fanout' paths.

The presence of such paths can obscure the final test configuration and indeed Schneider [19] has illustrated a network containing a fault that cannot be detected using one-dimensional techniques. This inadequacy has now been overcome and the extension is described under n-dimensional path sensitising in Section 1.5.

A further disadvantage, certainly in Armstrong's paper, lies in the fact that the network has to be re-configured into its *equivalent normal form*, i.e. the original sum-of-product Boolean form. This is a somewhat retrograde step and Crook and Blythin [82] have overcome this. They have also extended the technique to include bistable elements along the sensitised path. This is achieved by progressive setting-up in successive clock cycle stages.

The true sequential network still presents many problems due to the inherent feedback and the technique of breaking the feedback loops proposed by Galey, Norby, and Roth [5] does not succeed when the network becomes sophisticated. As one-dimensional sensitisation has now been superceded by the n-dimensional process, the extension into sequential networks is now of academic interest only.

1.3. *State-table analysis* [6], [7], [14], [25], [31], [50], [52], [69]
The specific intention here is to produce a DTS from the original state-table and Poage and McCluskey [7] attempted to extend a previous method of theirs that only applied to combinational networks. This extension involved forming the state-tables of all the networks—good and faulty. Each faulty table was then compared with the good table and a suitable input sequence capable of differentiating between the two was formed. The resulting set of sequences was then minimised using prime implicant subset techniques. This approach obviously suffered from the need to form all state-tables and is now of historical interest.

The present state-table method is based on a paper by Hennie [6] and his idea was to derive certain input sequences, called distinguishing sequences (DS) which could be applied at any time to the network, and analysis of the response identifies the initial starting state. The combinations of these DS's and certain state transitional sequences form an overall checking sequence that can be applied to the network no matter what its state, and the output will follow a pre-defined pattern if there is no fault.

One of the main academic problems associated with this technique has been the study of the theoretical upper bound on the length of the checking sequence. This is a function of the internal organisation of the sequence and successive authors [14], [25], [50] and [69] have all succeeded in reduction on previous bounds.

Another important aspect of this approach has been the modification to the state-table due to Kohavi and Lavallee [25]. If a state-table M does not have a DS, then they have suggested a manner in which it may be modified to an equivalent state-table M' that does possess a DS. Thus the diagnosis requirement is incorporated as a design parameter and not an afterthought.

Gelenbe [31] has combined the theory of the DS checking sequence with state-table representation using regular expressions, but the upper bound on the sequence is greater than the (then) current upper bounds.

1.4. *Fault matrix* [11], [20], [35], [48], [62], [74], [83]
The fault matrix relates a set of tests to their associated faults. The D-matrix of Chang [11] is a generalised form in which the d_{ij}^{th} entry is 1 if a fault f_i is detected by a test t_j, otherwise 0. Kautz [48] has described a similar matrix for single output combinational networks, called the F-matrix in which the expected output forms the actual entry. The three problems of fault detection, fault location and fault location to module level only are re-stated within the F-matrix confines and depending on which restraint is selected, a further matrix, the G-matrix is formed. The subsequent reduction of the G-matrix is common to all three problems, and row and column dominance techniques as in the prime implicant subset selection problem are used to derive either an optimal or a minimal set of tests. This technique is also used to minimise Chang's D-matrix.

Kautz has extended the process to include q-output combinational networks and the entries in the F-matrix become q-digit binary numbers. The subsequent G-matrix however has only the two entries of 1 or 0 and is reduced as before. Powell [62] has suggested an alternative method of fault location to

module level and this is achieved by assigning weights to each test according to its applicability to specific modules.

The latest work has been done by Kime [74] who has increased the model flexibility and shown its suitability to systems based on LSI technology and Preparata [77] who has formalised the mathematics of the bounds on the length of the DTS. Boyce [83] describes how he has implemented the technique on a digital computer and demonstrates how bistable elements can be accommodated into the set of tests by representing them as an equivalent model using only logic gates.

The major problem associated with this method is the size of the initial input test set—for example, a simple combinational network has an upper bound of 2^n possible test inputs, where n is the number of input variables. Consequently, for large variable problems, the method soon becomes unwieldy. Unfortunately, there is no real solution to this, since if redundant tests are removed by some means before forming the matrix, the resulting matrix has no value in that it cannot be further reduced. If the initial tests are derived by some technique other than simple 2^n enumeration, i.e. by path sensitising say, it is often the case that a test that has been derived for one fault will also be valid for other faults. It is under these circumstances that the fault matrix reduction process is an invaluable aid.

1.5. n-*dimensional path sensitising* (D-*algorithm*) [15], [19], [32], [33], [60], [84]

The problem of reconvergent fanout associated with the one-dimensional path sensitising technique (Section 1.2) was solved when the *D*-algorithm procedure was proposed by Roth in 1966 [15]. This basic procedure, which has subsequently been expanded by Roth *et al.* [32] is based on the *calculus of* D-*cubes*. This is a mathematical model of a combinational network, not unlike a truth table except that an extra symbol *D*, and its inverse $\bar{D}$ is used. *D* has the capability of assuming either of the Boolean values 0 and 1, but whichever one it assumes, it applies throughout the whole *D*-cube. The *propagation* D-*cubes* of each element within the network are derived and each one indicates the ability of one of the elemental input lines to control the elemental output. A D-*cube of failure* is then derived for a postulated fault and an attempt is made to '*D*-drive' the effect of the failure to the primary output of the network—all such paths being recorded. This forward-trace phase operation is assisted by the D-*intersection* technique where elemental or *primitive* propagation *D*-cubes are intersected with the failure *D*-cube following certain combinational rules [15].

The backward-trace phase, called the *consistency operation* by Roth, is then initiated to determine the primary input configuration necessary to realise the sensitive path(s). This operation is a reverse *D*-intersection technique whereby Boolean values are assigned to any 'don't cares' in the sensitised paths. It is at this stage that any inconsistencies in gate input/output requirements will become apparent.

By enumerating all paths, the method ensures that if a test exists, it will be found and in the case of the Schneider network [19], it does determine the test input configuration required to test the postulated fault that was not obtainable using the one-dimensional path sensitising technique [32].

The basic *D*-algorithm is applicable only to combinational networks. Kubo [60], however, has discussed the standard model for a sequential network and shown that it can be redrawn as a cascaded connection of combinational networks, called the *developed sequential network*. He then effects a modification to the *D*-algorithm to produce a DTS for the original sequential network.

It has been suggested [32] that the *D*-algorithm can be combined with the system/360 Fault Location Technology to produce a more comprehensive diagnosis capability of computer failures and Warburton [84] has reported that he has successfully programmed the *D*-algorithm to produce DTS's for combinational and sequential LSI networks.

1.6. *Boolean difference* [36], [49], [75]

Both Amar and Condulmari [36] and Sellers *et al.* [49] have used the logical exclusive OR operator in generating DTS's for combinational networks. The Boolean difference *Bd* is defined as being the exclusive OR operation between two Boolean functions—one representing the good machine, and the other representing the faulty machine. If the *Bd* is 1, then an error is apparent and a suitable test sequence can be derived. The DTS proper is derived by forming the *Bd*'s for each fault; determining the necessary input sequence for its detection and location, and assembling these together to form a composite sequence.

Its main advantage is that most of the work done in deriving the sequence is straightforward Boolean expression minimisation and there are excellent algorithms for this. Marinos [75] has reported extensions to the technique to cover sequential networks. Carroll *et al.* [64] and Kajitani [66] have also made use of the Boolean difference in their respective treatments of DTS generation.

1.7. *Graph theory models* [22], [39], [56], [63], [64], [66], [67], [81]

The use of graph theoretic techniques in analysing digital systems and determining the DTS's is gaining momentum and it does seem to offer acceptable solutions to some of the problems that are not adequately solved by other methods. Familiarity with the fundamentals of graph theory is necessary to appreciate the detail of the relevant papers, but the concepts are understandable without this.

Any digital system is a combination of its behavioural and structural properties and the latter can be represented by a directed graph, in which the nodes represent functional members and the arcs indicate the information flow or connections between the members. Once this model has been derived, it can be analysed using graph theory techniques. For instance, the system can be partitioned either physically or functionally into smaller preferably disjoint systems—this amounts to finding the maximally strongly connected subgraphs. This results in a new system model that can be used to illustrate the positioning of strategic test points (Kautz [48] commented that the fault-matrix could be initially reduced, not in a dominance sense, if strategic test points were placed around the network. He knew of no satisfactory way of doing this however).

The applications of the graph theory model are many. Ramamoorthy [22], [56] has considered the test point problem in some depth and other authors [39], [66] have studied the problem of multiple fault location. Hornbuckle and Spann [63] have derived an efficient algorithm for DTS generation enabling diagnosis to module level only and Carroll *et al.* [64] have studied fault diagnosis of computers based on a modular construction and using LSI modules. Young [81] has described a dependency chart derived from system block diagrams that illustrates the functional and physical structure of the system and enables a clear indication of inter-element dependency and signal interchange. The chart is eminently suitable for presentation to a computer via a graphic input medium.

The main problem at the moment with this approach to fault diagnosis arises from the fact that most of the papers are original and not based on, or extensions of, previous papers. It is felt that a unified approach can be made, combining the various applications and forming a composite 'diagnosis theory', compatible with current switching theory and digital system design methods.

2. Limitations of the diagnosis techniques

There are a number of limitations that are common to all the techniques of producing DTS's. These are discussed in this section and in general only the more important references are cited.

2.1. *Class of faults*

The general restraints on the type of fault that can be either detected and/or located are as follows:

1. Logical faults that cause any line to become 'stuck-at-1' or 'stuck-at-0' (s-a-1, s-a-0 type).
2. The system remains a fully logical system under the fault condition.
3. Normally a single fault at any one time is assumed.
4. Faults are non-transient.

In addition to these, some of the techniques impose their own extra restraints, e.g. state-table analysis requires that the state-table of the fault-free network is strongly connected and that both the fault-free and faulty network state-table have the same number of states.

The s-a-1, s-a-0 type fault is a very common restriction and only rarely do solutions specifically embrace other common faults such as open and short-circuit or wiring errors. Amar and Condulmari [36] have used the Boolean difference to diagnose these sort of errors and in fact have an example in which they successfully diagnose multiple wiring errors.

The logical functioning of the faulty system is complementary to the logical fault restriction and in general single faults only can be accommodated. The multiple faults diagnosis has been studied among others by the following authors: Amar and Condulmari [36], Preparata *et al.* [39], Schertz and Metze [53], Kajitani *et al.* [66] and Gault *et al.* [72]. Of these, two [39] and [66] are 'graph theory' papers and one [36] 'Boolean difference'.

The intermittent or transient fault is a particularly difficult one to diagnose under any circumstances. Such faults may result from marginal operation of the elements caused possibly by ageing. Chang and Thomis [20] have attempted to apply experimental observations on two similar systems to the test dictionary that is used to identify the fault. By these means, a new test dictionary is derived with built-in 'fuzziness' that allows an inconsistent fault to be diagnosed.

One final group of faults that very seldom receives any mention are those due to the physical construction and layout of the system. Such faults occur through cross-talk, ground plane incompatibilities etc. and are not normally considered to be diagnosible using DTS techniques. Certainly it would be very difficult to generalise specific automatic diagnostics, if they exist, but nevertheless the problem is very real and merits further study.

2.2. *The effects of logical redundancy*

The effects of logical redundancy in the network has been stated by Friedman [21] and studied by Jones and Mays [38]. Friedman demonstrates that the presence of a fault on a logically redundant element can mask further faults on non-redundant elements. Furthermore, he points out that in some cases, i.e. to avoid hazards, redundancies are necessary. He does prove however for combinational networks, that if the redundancy is of the 'Eichelberger' type, i.e. contains all the prime implicants,* then this masking does not occur.

In general, if a fault occurs in a redundant element, it cannot be detected. This causes problems in systems using redundancy to achieve high reliability. Hannigan [34] has described a technique for allocating test points within such systems to alleviate the detection problem.

Jones and Mays present an integrated approach to checking LSI circuits and part of this is a redundancy check. This is achieved by simulation and forward and backward propagation of signals—inconsistencies being indicative of redundancy. Breuer [73] has used the sensitised path concept to determine redundancy in combinational networks.

2.3. *Test criteria*

There are many criteria used in selecting tests to form the overall DTS of a system. Some are obscured and even replaced by the minimisation techniques, whereas others form a vital part in the selection process. Nowhere is this more so than in the partitioning approach and various criteria have been proposed to aid the selection process. These are as follows:

Checkout detection criterion—This is assigned on the basis of the test separating the largest number of faulty systems from the good system. This enables rapid fault detection.

Information gain criterion—This is a measure of the information gained from each test and is similar in concept to the entropy function used in information theory. This is more useful when full diagnosis is required.

The first of these was introduced by Seshu and Freeman [2] and Mandelbaum [8] has formally defined the second. Brulé *et al.* [1] and Seshu and Freeman use a restricted version of the information gain criterion. The restriction is that it is only applicable to binary partitioning whereas the general definition applies to *m*-ary partitioning. In a later paper devoted solely to this topic, Chang [46] mathematically defines these two criteria and also introduces a further one:

Distinguishability criterion—This selects the test that distinguishes the largest number of pairs of systems and effectively it ensures that each test will have the maximum *m*-ary partitioning effect.

The mathematical expression is modifiable to enable the criterion to be applied to module level rather than element level tests and this is also defined in Chang's paper.

For all the other techniques available, the problem is not so much one of selecting tests, but of deriving the test in the first place and it becomes difficult to apply straightforward selection criteria.

Finally **Table 1** indicates some of the limitations discussed above and correlates these to the seven methods identified in Section 1. Also included in the table is a recommended paper describing the particular technique.

3. The effect of diagnosis requirements on computer system engineering

The change in the engineering design philosophy of digital computers was mainly due to the inadequacy of the early diagnostic routines. These were derived after the system design had been completed and in some cases, after the actual machine had been built. They were, as a consequence, not fully effective in their coverage and the obvious solution was for an automatic or at least algorithmic way for deriving the tests and this immediately implied that the initial system and engineering designs should be constrained by the diagnosis requirements. Once this concept had become established, there was a spate of design organisations and nearly all of these have aimed to achieve the ultimate in computer system diagnostic engineering —the automatic self-diagnosing and repairing computer.

Maling and Allen [4], in an often referenced paper, proposed a serial machine structure with a special set of programs, based on hardware checkout requirements and controlled by special console controls. The programs contained tests for the combinational and sequential sections derived in a manner similar to the fault matrix method.

*Eichelberger, E. B. Hazard detection in Combinational and Sequential switching circuits, Proc. 5th symposium on switching circuit theory and logical design, 1964.

Table 1 A comparison of the limitations of each DTS technique

	SYSTEM LEVEL		NETWORK			FAULTS COVERED				RECOMMENDED
	TOTAL	SUB-	MODULE	COMB.	SEQ.	SINGLE s-a-1 s-a-0	MULT.	INTER-MIT.	WIRING o/c, s/c	PAPER(S)
Partitioning	1	1	1	1	1	1				[2] [9]
Sensitised path		1	1	1		1				[13]
State-table		1	1		1	1	1			[6] [25]
Fault matrix	1	1	1	1	1	1	1	1		[11] [48]
D-algorithm		1	1	1	1	1				[15] [60]
Boolean diff.		1	1	1	1	1	1		1	[49]
Graph theory	1	1	1	1	1	1	1		1	[22]

The first real attempt to partition the system specifically to accommodate the diagnosis requirements was made by Forbes *et al.* [10]. They proposed a method of partitioning the system such that each section was almost disjoint and capable of providing and acknowledging test patterns to at least one other section. Their method placed a severe restraint on the system design of the computer, but necessitated very little extra hardware. The work has since been extended [29] to include automatic error detection and automatic program recovery.

Manning [18] proposed and simulated a novel system architecture called *cascade machine organisation*. The machine separates the status of the core store and arithmetic unit from the supervisory controls and also incorporates special microinstructions solely for diagnosis purposes. He claims that this enables the diagnosis to be conducted by means of special programs using the test microinstructions and also program revision can be simply effected. To the present authors' knowledge, no further work seems to have been done on the cascade machine.

An interesting application of the partitioning method for DTS has been reported by Marlett [27]. The proposed computer system is simulated along with all anticipated fault modes and the identification of the system is resolved on the basis of instruction word responses. The resulting list of instruction words now form the diagnostic program. Marlett considers that this approach does not obscure the diagnosis design principles and allows effective comparison of different systems with a view to improving the self-diagnosability.

A computer organisation called ARC, Automatically Repaired Computer, has been described by Bouricius *et al.* [30]. The requirement here was for very high reliability and ARC contains functional units, e.g. control units, channel control, arithmetic units, etc., that each consist of an assembly of identical modules. When an error is detected, ARC 'bootstraps' itself into operation until the defective module is located and then it switches in a standby module whilst a replacement is obtained. The actual diagnosis programs are based on the *D*-algorithm. A later paper [45] by the same authors expands on the reliability aspects of automatically repaired computers.

An approach similar to ARC has recently been described by Linsenmayer [71] and he comments that this form of self-repairing computer is compatible with LSI technology and modularity construction.

Aviziensis [58] has gone a stage further in the use of redundancy and his STAR, Self Testing and Repairing, computer incorporates such features as error detection codes (this is also used by Carter and Schneider [57]); partitioning of functional units—each unit having its own sequence generator and a minimum of input/output lines; special purpose hardware for fault detection, recovery and replacement; complex replication of the hard core, etc.

The use of redundancy and other reliability techniques is not within the terms of reference of this paper, but the interested reader will find a list of 155 papers dealing with self-repairing computers and the ramifications thereof in a recent paper by Dorrough [61].

4. The requirements of digital systems in terms of diagnosis

It is difficult to separate the system requirements in terms of diagnosis from the effect of diagnosis on the system, but some authors have isolated the requirements and in particular Dent [46] who has described three types of checking procedure called *Start Small*, *Multiple Clue* and *Start Big*: Start Small is a sequential bootstrapping approach, Multiple Clue is the combinational approach* and Start Big is an overall rapid system check with more sophisticated diagnosis at subsystem level if required. He demonstrates the applicability of these to the different tests that are required by computer users. These are engineering/manufacturing acceptance tests, field maintenance and quick checkouts (early morning checkout etc.). Further comments along these lines have been made by Jones and Mays [38] and Breuer [51]. Also O'Brien [59] has discussed the pre-flight checkout procedure for a special purpose aerospace computer. This is accomplished by a link-up to a general purpose computer. Horovitz [65] has described a similar technique for using the computer to perform manufacturing acceptance checks on computers in production.

Among the current requirements of digital systems is the concept of graceful degradation under fault conditions. This implies that the system is still able to perform its primary functions although at a reduced efficiency. A further requirement is that initial diagnosis routines should be operable by the normal computer operators and not require complex operation and analysis. This implies a hierarchical test structure as in the Start Big approach, and the method of DTS derivation due to Martlett [27] and mentioned in the previous section, goes a long way to achieving this.

In general, Dent's paper is an adequate summary of the requirements, but most of the papers already cited in Section 3 also have some relevant comment. In addition, Brulé *et al.* [1], Quatse [12], Hannigan [34] and Calhoun [76] are of interest.

*Sequential and combinational here apply to the testing procedure and not the classification of networks. The use of these terms for both testing procedures and logical networks can give rise to confusion and alternative expressions have been suggested by Young [81] who uses non-sequential and sequential and by Boyce [83] who uses single-flow and multi-flow for combinational and sequential respectively.

5. Functional testing and diagnosis of LSI

The functional testing of LSI is receiving more and more attention as its complexity grows—not only by the manufacturers but also by the user. The problems are manifold and the paper by de Atley [79] contains an appraisal of these. In essence, they are problems associated with custom-designed LSI; the type of faults that occur in manufacture compared with the normal restrictions of the diagnosis procedure and problems inherent with dynamic and static testing. Lewis [33] has proposed an algorithm based on path sensitising techniques that will produce acceptance tests for chips containing sequential networks. It has the advantage of not requiring a formal state-table but only the connection details, and the disadvantage of s-a-1, s-a-0 type faults only.

Tammaru and Angell [37] consider groups of interconnected elements as the smallest diagnosible element on the chip. This, combined with a temporary test metallisation, enables more efficient testing during manufacture and the final metallisation pattern only utilises the good groups. Tammaru has also considered aspects of regular cellular arrays and the characteristics they require to make them amenable to terminal testing in his subsequent thesis [55].

Perhaps one of the more significant papers in this area is that of Jones and Mays [38]. They have combined the partitioning and elements of the state-table approach to produce an integrated DTS procedure applicable to either combinational or sequential networks. The method is a curious but efficient blend of redundancy checking procedures, modified local-search and random partitioning and subsequent state-table construction and minimisation to determine the shortest sequence. The method has been simulated and the authors, who at the time were with the Fairchild Corporation, consider it eminently suitable for the fast turn-round of the industry.

Recently Hillman [78] has described a programmable test system for LSI and de Atley [79] describes how some manufacturers have used the partitioning method in deriving the acceptance tests.

6. Conclusion

This brief survey indicates the importance that has been attached to diagnosis by all concerned with the design, construction and operation of digital systems. The methods of deriving the diagnostic test procedures are manifold and to the seven listed can be added an eighth—the heuristic *ad hoc* testing and diagnosing based on a thorough working knowledge of the system. This has its obvious advantages and disadvantages and is still in wide use.

The present authors feel that the time is now ripe for a unified theory of diagnosis and that this theory will probably be mathematically modelled on graph theory concepts. This form of mathematical description lends itself to path enumeration and it is felt that the theory will embrace the techniques of *n*-dimensional path sensitising and state-table analysis—both of which rely on path enumeration. The former is more applicable to intuitively designed digital networks whereas the latter is exclusive to networks designed more rigorously using switching theory. In this way, the diagnosis theory can span both design techniques. It is vitally important however, that no matter what design process is employed, the diagnosis technique becomes an integral part of it—indeed, diagnosis itself must become one of the principal design parameters. There is an obvious case here for more formalisation of the design methods, i.e. the increasing use of switching theory, as such a network has a defineable mathematical form and consequently lends itself to analysis for diagnosis purposes. This applies particularly to networks utilising storage elements, i.e. sequential networks. Combinational network diagnosis is now adequately covered by existing techniques, but problems still exist for large variable networks. In some cases, these combinational methods have been extended to include sequential networks. These have their own inherent problems due to the internal feedback, but the use of graph theory is indicated since the state diagram of the network is no more than a directed graph.

The impact and interaction of diagnosis with computer design has been reviewed. The trend in computer construction is toward more modularity, this being a result of the emergence of LSI technology and this dictates the boundaries of the partitioning of the system, both physical and functional. The diagnosis requirements are constrained by the extra hardware/software that can be tolerated and there are alternative methods available for both generating the diagnostic routines and incorporating them within the system. The diversity of techniques make it difficult to enumerate design principles, but the general ones that have emerged as being important are:

1. The diagnosis requirement is an initial restraint on the theoretical system design. Previous diagnosis routines have been grafted on to an existing system and the inadequacies of this has been amply demonstrated.
2. The diagnosis procedure should be automatic both in detection and location of the fault.
3. Some sort of partitioning constrained by the physical and functional hierarchical structure of the system should be effected to enable efficient sub-system checkout. This implies that inter sub-system dependence is small.
4. The hard core should be minimised. (The hard core is that section of a system that is assumed faultless or is pre-checked independently such that it can be used with certainty as a basis for any subsequent checking procedures.)

The testing philosophies for LSI and digital systems are slightly different. For LSI, the prime requirement is for an input/output check only and subsequent fault location is not generally required unless the chip is a general purpose one with the final metallisation pattern only utilising previously determined 'good' components. The prime requirement for a digital system such as a computer is for full diagnosis and eventual physical replacement of the defective element. This contrast in the requirements is reflected by the diagnosis procedure implementation problems. For LSI, the technique is usually one of program driven testing machines having high flexibility of outputs and low cost programs. The computer installation however requires a hierarchy of tests and these may be of a software and/or hardware nature; these additional features must be self-checking and in effect become part of the hard core. Also required is a coherent print-out of the diagnosis and the ability to maintain operation at a reduced efficiency during the diagnosis period. This implies that the diagnosis procedure is complementary to and not replace existing user requirements.

It is indicative of the importance that is now attached to fault diagnosis that two books have recently been published pertaining to fault diagnosis. In the first, Chang, Manning and Metze [85] have attempted to bring together and compare some of the techniques outlined in Section 1 together with their limitations. Kohavi [86] in his book has devoted space to a discussion on the state-table approach.

Finally, the list of papers referenced is in no way complete, but is representative of the more important contributions that have been made to the general theory and extensions of fault diagnosis in digital systems within the last decade.

7. Acknowledgement and affiliation

This work is financially supported by the Science Research Council, London. One of the authors, R. G. Bennetts, wishes to acknowledge the support of ICL and SRC for an industrial studentship.

Both authors are at present with the Department of Electronics, Southampton University, England.

Bibliography

[1] BRULÉ, J. D., JOHNSON, R. A., and KLETSKY, E. J. (1960). Diagnosis of Equipment Failure, *IRE Trans. on Reliability and Control*, Vol. RQC-9, pp. 23-34.

[2] SESHU, S., and FREEMAN, D. N. (1962). The diagnosis of asynchronous sequential switching systems, *IRE Trans. on Electronic Computers*, Vol. EC-11, pp. 459-465.

[3] MENGER, K., Jnr. (1963). Checkups for combinational gates, *IEEE Trans. on Aerospace—Support Conference Procedures*, Vol. AS-1, pp. 954-960.

[4] MALING, K., and ALLEN, E. L. (1963). A computer organization and programming system for automated maintenance, *IEEE Trans. on Electronic Computers*, Vol. EC-12, pp. 887-895.

[5] GALEY, J. M., NORBY, R. E., and ROTH, J. P. (1964). Techniques for the diagnosis of switching circuit failures, *IEEE Trans. Communications and Electronics*, Vol. 83, No. 74, pp. 509-514.

[6] HENNIE, F. C. (1964). Fault detecting experiments for sequential circuits, Proc. of the 5th Annual Switching Theory and Logical Design symposium, S-164, pp. 95-110.

[7] POAGE, J. F., and MCCLUSKEY, E. J., Jnr. (1964). Derivation of optimum test sequences for sequential machines, Proc. of the 5th Annual Switching Theory and Logical Design symposium, S-164, pp. 121-132.

[8] MANDELBAUM, D. (1964). A measure of efficiency of diagonistic tests upon sequential logic, *IEEE Trans. of Electronic Computers*, Vol. EC-13, p. 630.

[9] SESHU, S. (1965). On an improved diagnosis program, *IEEE Trans. on Electronic Computers*, Vol. EC-14, pp. 69-76.

[10] FORBES, R. E., RUTHERFORD, D. H., and STIEGLITZ, C. B. (1965). A self-diagnosable computer, *Proc. AFIPS, FJCC*, pp. 1073-1100.

[11] CHANG, H. Y. (1965). An algorithm for selecting an optimum set of diagnostic tests, *IEEE Trans. on Electronic Computers*, Vol. EC-14, pp. 706-711.

[12] QUATSE, J. T. (1966). Time-shared troubleshooter repairs computers on-line, *Electronics*, Vol. 39, Jan. 24th, pp. 97-101.

[13] ARMSTRONG, D. B. (1966). On finding a nearly minimal set of fault detection tests for combinational logic nets, *IEEE Trans. on Electronic Computers*, Vol. EC-15, pp. 66-73.

[14] KIME, C. R. (1966). An organization for checking experiments on sequential circuits, *IEEE Trans. on Electronic Computers*, Vol. EC-15, pp. 113-115.

[15] ROTH, J. P. (1966). Diagnosis of automata failure: a calculus and a method, *IBM Journal R. and D.*, Vol. 10, pp. 278-291.

[16] ARMY ELECTRONICS COMMAND, FORT MONMOUTH (1966). Microelectronics ADP Maintenance study, July, AD 664 195.

[17] BREUER, M. A. (1966). General survey of design automation of digital computers, *Proc. IEEE*, Vol. 54, pp. 1708-1721.

[18] MANNING, E. (1966). On computer self diagnosis. Part I—Experimental study of a processor. Part II—Generalizations and design principles, *IEEE Trans. on Electronic Computers*, Vol. EC-15, pp. 873-890.

[19] SCHNEIDER, P. R. (1967). On the necessity to examine *D*-chains in diagnostic test generation—an example, *IBM Journal*, Jan., p. 114.

[20] CHANG, H. Y., and THOMIS, W. (1967). Methods of interpreting diagnostic data for locating faults in digital machines, *BSTJ*, Vol. 46, pp. 289-317.

[21] FRIEDMAN, A. D. (1967). Fault detection in redundant circuits, *IEEE Trans. on Electronic Computers*, Vol. EC-16, pp. 99-100.

[22] RAMAMOORTHY, C. V. (1967). A structural theory of machine diagnosis, *Proc. AFIPS, SJCC*, pp. 743-756.

[23] SALISBURY, A. B., and ENSLOW, P. H. (1967). Diagnostic programming for digital computers—a bibliography, West Point Military Academy, April, AD 813 831.

[24] CHU, W. W. (1967). A mathematic model for diagnosing system failure, *IEEE Trans. on Electronic Computers*, Vol. EC-16, pp. 327-331.

[25] KOHAVI, Z., and LAVALLEE, P. (1967). Design of sequential machines with fault detection capability, *IEEE Trans. on Electronic Computers*, Vol. EC-16, pp. 473-484.

[26] MANNING, E. G., and CHANG, H. Y. (1967). A comparison of fault simulation methods for digital systems, Digest of the 1st Annual IEEE Computer Conference, Sept., pp. 10-13.

[27] MARLETT, R. A. (1967). On the design and testing of self-diagnosable computers, Digest of the 1st Annual IEEE Computer Conference, Sept., pp. 14-16.

[28] MULLER, D. E. (1967). Evaluation of logical and organizational methods for improving the reliability and availability of a computer, Digest of the 1st Annual IEEE Computer Conference, Sept., pp. 53-55.

[29] AGNEW, P. W., FORBES, R. E., and STIEGLITZ, C. B. (1967). An approach to self-repairing computers, Digest of the 1st Annual IEEE Computer Conference, Sept., pp. 60-63.

[30] BOURICIUS, W. G., CARTER, W. C., ROTH, J. P., and SCHNEIDER, P. R. (1967). Investigations in the design of an automatically repaired computer, Digest of the 1st Annual IEEE Conference, Sept., pp. 64-67.

[31] GELENBE, S. E. (1967). Regular expressions and checking experiments, Sept., AD 666 696.

[32] ROTH, J. P., BOURICIUS, W. G., and SCHNEIDER, P. R. (1967). Programmed algorithms to compute tests to detect and distinguish between failures in logic circuits, *IEEE Trans. on Electronic Computers*, Vol. EC-16, pp. 567-580.

[33] LEWIS, R. S., Jnr. (1967). An approach to test pattern generation for synchronous sequential networks, Ph.D. Thesis, Southern Methodist University, USA.

[34] HANNIGAN, J. M. (1967). Redundant system test point allocation and mission reliability estimation procedures, *IEEE Trans. on Electronic Computers*, Vol. EC-16, pp. 591-596.

[35] HADLOCK, F. (1967). On finding a minimal set of diagnostic tests, *IEEE Trans. on Electronic Computers*, Vol. EC-16, pp. 674-675.

[36] AMAR, V., and CONDULMARI, V. (1967). Diagnosis of large combinational networks, *IEEE Trans. on Electronic Computers*, Vol. EC-16, pp. 675-680.

[37] TAMMARU, E., and ANGELL, J. B. (1967). Redundancy for LSI yield enhancement, *IEEE Journal of Solid-State Circuits*, Vol. SC-2, pp. 172-182.

[38] JONES, E. R., and MAYS, C. H. (1967). Automatic test generation methods for large scale integrated logic, *IEEE Journal of Solid-State Circuits*, Vol. SC-2, pp. 221-226.

[39] PREPARATA, F. P., GERNOT, M., and CHIEN, R. T. (1967). On the connection assignment problem of diagnosable systems, *IEEE Trans. on Electronic Computers* Vol. EC-16, pp. 848-854.

[40] ULRICH, E. G. (1967). The evaluation of digital diagnostic programs through digital simulation, Computer Technology, IEE Conference Publication No. 32, pp. 9-19.

[41] LAWDER, R. E. (1967). Computer testing by control wave-form simulation, Computer Technology, IEE Conference Publication No. 32, pp. 20-31.

[42] SESHAGIRI, N. (1967). A decision table approach to self diagnostic computers, *Proc. IEEE*, Vol. 55, pp. 2180-2181.

[43] EPLEY, D. L. (1968). Systematic analysis of combinational network hazards and faults from dual matrices, Proc. 2nd Annual Princeton Conference on Information Sciences and Systems, pp. 64-68.

[44] GARCIA, O. N., and RAO, T. R. N. (1968). On the methods of checking logical operations, Proc. 2nd Annual Princeton Conference on Information Sciences and Systems, pp. 89-95.

[45] Bouricius, W. G., Carter, W. C., and Schneider, P. R. (1968). Reliability modelling of automatically repaired computers, IEEE International Convention, p. 193.
[46] Chang, H. Y. (1968). A distinguishability criterion for selecting efficient diagnostic tests, *Proc. AFIPS, SJCC*, pp. 529-534.
[47] Dent, J. J. (1968). Diagnostic Engineering requirements, *Proc. AFIPS, SJCC*, pp. 503-507.
[48] Kautz, W. H. (1968). Fault Testing and diagnosis in combinational digital circuits, *IEEE Trans. on Computers*, Vol. C-17, pp. 352-366.
[49] Sellers, F. F., Jnr., Hsiao, M. Y., and Bearnson, L. W. (1968). Analysing errors with the Boolean Difference, *IEEE Trans. on Computers*, Vol. C-17, pp. 676-683.
[50] Kohavi, I., and Kohavi, Z. (1968). Variable length distinguishing sequences and their application to the design of fault-detection experiments, *IEEE Trans. on Computers*, Vol. C-17, pp. 792-795.
[51] Breuer, M. A. (1968). Hardware Fault Detection, *Proc. AFIPS, FJCC*, pp. 1502-1503.
[52] Bossen, D. C. (1968). Self-diagnosable decomposition of sequential machines, Thesis, Northwestern Univ., Evanston, USA.
[53] Schertz, D. R., and Metze, G. A. (1968). On the indistinguishability of faults in digital systems, Proc. 6th Allerton Conference on Circuit and System theory, pp. 752-760.
[54] Powell, T. J. (1968). Synthesis requirements for fault detection, Proc. 6th Allerton Conf. on Circuits and System theory, pp. 761-772.
[55] Tammeru, E. (1968). Efficient testing of combinational logic cells in large scale arrays, Thesis, Stanford Univ., USA.
[56] Mayeda, W., and Ramamoorthy, C. V. Distinguishability criteria in oriented graphs and their application to computer diagnosis—I and II, Texas Univ., 1968 AD 685-739 (Part I) and AD 690 127 (Part II). Part I is also published in *IEEE Trans. on Circuit Theory*, Vol. CT-16, 1969, pp. 448-454.
[57] Carter, W. C., and Schneider, P. R. (1968). Design of dynamically checked computers, IFIP Congress, Vol. 2, pp. 878-883.
[58] Aviziensis, A. (1968). An experimental self-repairing computer, IFIP Congress, Vol. 2, pp. 872-877.
[59] O'Brien, J. A. (1968). Computer fault location using tabular functions, IFIP Congress, Vol. 2, pp. 1479-1483.
[60] Kubo, H. (1968). A procedure for generating test sequences to detect sequential circuit failures, NEC R & D, No. 12, pp. 69-78.
[61] Dorrough, D. C. (1969). A methodical approach to analysing and synthesizing a self-repairing computer, *IEEE Trans. on Computers*, Vol. C-18, pp. 22-42.
[62] Powell, T. J. (1969). A procedure for selecting diagnostic tests, *IEEE Trans. on Computers*, Vol. C-18, pp. 168-175.
[63] Hornbuckle, G. D., and Spann, R. M. (1969). Diagnosis of single-gate failures in combinational circuits, *IEEE Trans. on Computers*, Vol. C-18, pp. 216-220.
[64] Carroll, A. B., Kato, M., Koga, Y., and Naemura, K. (1969). A method of diagnostic test generation, *Proc. AFIPS, SJCC*, pp. 221-228.
[65] Horovitz, M. S. (1969). Automatic checkout of small computers, *Proc. AFIPS, SJCC*, pp. 359-365.
[66] Kajitani, K., Tezuka, Y., and Kasahara, Y. (1969). Diagnosis of multiple faults in combinational circuits, *Electronics and Communications in Japan*, Vol. 52-C, pp. 123-131.
[67] Ichikawa, T., and Watanabe, T. (1969). A systematic method of finding diagnostic test functions, *Electronics and Communications in Japan*, Vol. 52-C, pp. 165-172.
[68] Barney, G. C., Hambury, J. M., and Chow, S. P. S. (1969). Diagnostic routines for a hybrid computer, *The Computer Bulletin*, June, pp. 178-183.
[69] Gonenc, G. A method for the design of fault detection experiments, IEEE Computer Group Repository, R-69-134.
[70] Cohen, D. J., and Manning, E. G. A fault simulator for research applications, IEEE Computer Group Repository, R-69-142.
[71] Linsenmayer, G. R. Self-repairing digital systems using few spares, IEEE Computer Group Repository, R-69-155.
[72] Gault, J. W., Robinson, J. P., and Reddy, S. M. Multiple fault detection in combinational networks, IEEE Computer Group Repository, R-69-168.
[73] Breuer, M. A. Generation of fault tests for linear logic networks, IEEE Computer Group Repository, R-70-9.
[74] Kime, C. R. An analysis model for digital system diagnosis, IEEE Computer Group Repository, R-70-19.
[75] Marinos, P. N. A method of deriving minimal complete sets of test-input sequences using Boolean Difference, IEEE Computer Group Repository, R-70-22.
[76] Calhoun, R. C. (1969). Diagnostics at the microprogram level, *Modern Data Systems*, No. 5, p. 58.
[77] Preparata, F. P. (1969). An estimate of the length of diagnostic tests, *IEEE Trans. on Reliability*, Vol. R-18, pp. 131-136.
[78] Hillman, L. (1969). An automated dynamic digital logic circuit test system, *Computer Design*, Vol. 8, Aug., pp. 58-62.
[79] de Atley, E. (1969). LSI testing is a large scale headache!, *Electronic Design*, Vol. 16, Aug. 2nd, pp. 24-34.
[80] Kriz, T. A. (1969). Machine Identification concepts of path sensitizing fault diagnosis, IEEE Conf. 10th Annual Symposium on Switching and Automata Theory, pp. 178-181.
[81] Young, H. W. (1970). Specifying the interface between design and test organizations, Joint conference on automatic test systems, IERE Conference Proc. No. 17, April, pp. 91-110.
[82] Crook, K. J., and Blythin, J. (1970). A computer controlled tester for logic networks and a method for synthesizing test patterns, Joint conference on automatic test systems, IERE Conference Proc. No. 17, April, pp. 187-200.
[83] Boyce, A. H. (1970). Computer generated diagnosing procedures for logic circuits, Joint conference on automatic test systems, IERE Conference Proc. No. 17, April, pp. 333-346.
[84] Warburton, G. C. (1970). Automatic dynamic response system for testing semiconductors, Joint conference on automatic test systems, IERE Conference Proc. No. 17, April, pp. 467-484.
[85] Chang, H. Y., Manning, E. G., and Metze, G. (1970). *Fault diagnosis of digital systems*, Wiley Interscience.
[86] Kohavi, Z. (1970). *Switching and finite automata theory*, McGraw-Hill Computer Science Series.