

Failure Diagnosis on the LT1280

The high-density circuitry and I/O pin population of the thermal conduction module (TCM), the VLSI package used in the IBM 3081 processor models, dictates that there be a precise and cost-effective method of detecting and diagnosing TCM defects. This paper describes the challenge faced in testing the logic and random access memories of the TCM and the diagnostic approach used in the LT1280 test system for testing the TCM through its I/O pins. The generation and application of tests are discussed, and the automated multiple-defect diagnostic (AMDD) algorithm is presented in detail.

Introduction

The distinguishing characteristics of the thermal conduction module (TCM) used in the IBM 3081 processor—high circuit density, high chip population, and high input/output (I/O) count—require that the smallest possible number of repairable items be replaced. These same characteristics make testing and diagnosis more difficult because of the volume of data and complexity of analysis.

To facilitate repair, the TCM has been designed with a high degree of repair capability. More than 98% of the defects detected on an assembled TCM are repairable. The features of the TCM [1, 2] that are pertinent to diagnostics include the following. There are a maximum of 133 chips, each capable of being independently removed and rejoined to the module substrate via the controlled collapse chip connection (C4) bonding [3]. There are 1800 input/output (I/O) pins, 1200 of which are logic signal pins. There are a maximum of 96 engineering change (EC) pads surrounding each chip site. These pads allow wiring repairs, wiring updates, and probing to be done. Table 1 shows the attributes of a typical TCM.

A test system was needed for the TCMs. It evolved from two test approaches. In the first of these, chip-in-place (CIP) testing, the tester applies the tests via EC pads. In the second approach, through-the-pins (TTP) testing, the tests are applied via the TCM I/O pins. Diagnosis of test results using CIP is comparatively straightforward since the scope of CIP

testing is limited to one chip at a time. In contrast, the TTP approach tests all chips at once, and failure results are observed at the TCM output pins. Thus, the diagnostic challenge using the TTP approach involved the ability to transform the test results into a cost-effective repair action.

The LT1280 test system was developed for TTP testing of capped and uncapped TCMs [2]. The LT1280 tester also has two single-point diagnostic probes which are independently controllable through software. Their major function within the LT1280 diagnostic system is described later.

Figure 1 shows a schematic of the test system, which consists of three parts: test data generation, test data preparation, and test application and diagnosis. Each of the parts is discussed further; emphasis is placed on the LT1280 diagnostic algorithm, automatic multiple-defect diagnosis (AMDD).

Table 1 Typical thermal conduction module (TCM) attributes.

Logic circuits	25 000
RAM cells	65 000
Chips	96
Logic	60
RAM	28
Terminator	8

© Copyright 1983 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

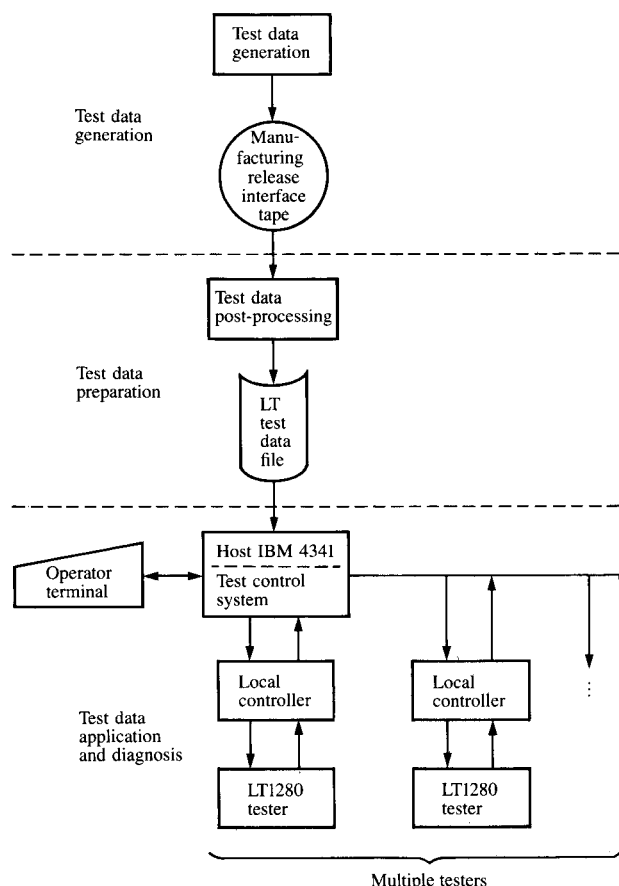


Figure 1 LT1280 test system overview.

Test data generation

The generation of test data for combinational logic tests is discussed separately from that for array or random access memory (RAM) tests. This is because of the inherent differences in the problems encountered and in the approaches to solving those problems.

The IBM engineering design system generates the traditional logic single stuck-fault tests. The logic design of each TCM must conform to the constraints imposed by the level-sensitive scan design (LSSD) [4, 5] if the objectives established for testability and diagnosis are to be achieved. The LSSD allows shift-register latches (SRLs), which have been serialized to form a shift-register string, to be used both as internal test points and as test data input and output (Fig. 2). This effectively reduces the problem of test generation for an entire structure to one involving only subsets of the logic. The design constraints guarantee the operability of the shift-register strings for test use.

Partitioning [6] of the TCM logic is a further requirement because the number of circuits and interconnections on a full

TCM exceed the limitations of the test generators and simulators. Thus, the logic is subdivided into smaller networks which are manageable by test programs; these are referred to as *test generation partitions* (TGPs). Test patterns for each subset are generated and accumulated in a test file. The concept of TGP, as exploited by the AMDD algorithm as part of the LT1280 diagnostic strategy, is discussed later.

The actual generation of the stuck-fault tests is performed by programs based on the D Algorithm [7]. Much has been written about the topic of stuck-fault test generation; thus it is not repeated here. The test generators used for the TCM have proven very effective. In practice, a calculated average stuck-fault test coverage of 94% has been achieved on some fourteen unique TCM logic designs investigated. Calculations show that the actual test coverage for all module types is greater than 98%; the remaining 2% are redundant. These are subject to testing at later stages.

While the principal test generation effort focused on the testing of logic data paths, there was also a need to test automatically RAMs through the TCM I/O pins. However, no automatic means existed for generating test patterns for imbedded RAMs, i.e., RAMs whose inputs and outputs are accessible only through combinational logic surrounding the RAMs. (In the case of the TCM, the RAMs are chips which are imbedded from the perspective of the TCM I/O pins.) Since RAM pattern generators existed for testing RAMs in which the control, address, and data lines were directly accessible (such as for RAM chips tested on the CIP tester), there was a need to provide this same type of feature for the TTP tester.

Test data generation and failure diagnosis for RAM chips imbedded in combinational logic turned out to be a difficult undertaking. (This type of problem is discussed by Eichelberger et al. [4(a)].) Typically, RAM inputs and outputs are connected to SRLs and module I/Os through complex combinational logic, making it difficult to control and isolate each RAM for test purposes. Also, the RAMs are commonly interconnected in bused I/O configurations, adding further complexity to testing and defect diagnosis. In order to successfully test imbedded RAMs, RAM preconditioning data had to be defined. In this manner, each read/write operation could be controlled through SRLs and module I/Os.

An experiment was conducted in which RAM preconditioning data were determined manually, on the basis of analysis of the TCM logic diagrams and the IBM 3081 system-level design data. Through the TCM I/O pins and the SRLs, the conditions for controlling the read/write sequences and accessing the data lines of the RAMs were found. The preconditioning data were then merged with the

actual RAM test patterns derived by tester algorithms. Testing of the TCM RAMs on the LT1280 was thus achieved. This approach to RAM testing has become the normal mode of operation. Some relief for the manual effort was provided first by development of special LT1280 program aids. But even with these aids, generation of RAM test data required extensive manual work. This manual approach was used, however, until imbedded RAM preconditioning data generation was incorporated into the engineering design system tools.

In the generation of RAM data, consideration was given to the volume of test data and to the testing time required. In an attempt to take advantage of the regular and repetitive structure of the RAMs, the preconditioning data and test patterns were generated for groups of RAM chips, each group being called a matrix. Thus, multiple RAM chips on the same TCM could be tested in parallel in the same time it took to test one chip. By using this matrix concept, we have been able to realize significant savings in the required testing times and data volumes.

The test data currently generated by the engineering design system contain the logic test and RAM preconditioning data, as well as diagnostic data used by AMDD to guide its probing. All of these data are merged with physical data for the TCM and packaged into a standard formatted manufacturing release interface tape (RIT).

Test data preparation

Before the test data can be applied at the tester, the RIT must be tailored to the LT1280 format and must have non-logic (power, etc.) parametric data and RAM test patterns added. The first step (Fig. 3) is a corporate processing system which builds the common data base (CDB). Its purposes are to format the data for efficient processing by later steps and to perform checking functions.

The corporate processing system copies the logical and physical records from the RIT onto an internal data base. This data base is structured by record, with each record type containing different information, e.g., logic structure, test data, and so on. During this and later operations, the data are checked for consistency, validity, and adherence to format and product ground rules. To complete the logical and physical description of the TCM, a correlation is established between the logical names used by the test generator and the physical names associated with them. This process makes use of component descriptor rules and the physical model on the RIT. As a last step, all records and the test data are stored in the CDB.

The test data supply system then personalizes the test data into final LT1280-executable form by updating some CDB

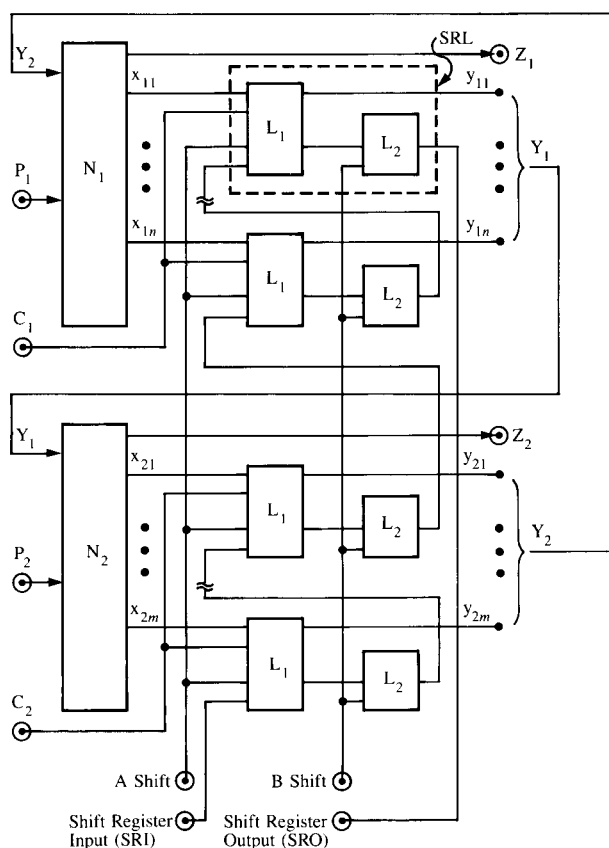


Figure 2 General structure of level-sensitive scan design (LSSD). All latches (L) are segregated from the combinational logic and interconnected in shift-register latch (SRL) strings for testing purposes, thereby requiring only a single data input and a single data output, plus A- and B-shift control signals. N_1 and N_2 are combinational logic; C_1 and C_2 are clocks; P , Y , Z , x , and y are logic inputs, outputs, interconnections, etc.

records and by using others to create new records in the logic test verification file (LTVF). This is the file actually used for testing. It accesses rules for component descriptions and specific tester personalization. These rules provide technology-dependent information for creating non-logic tests and for correlating data between pin and tester I/Os to match the product to the tester.

The test data supply system performs many functions. First, it provides additional technology and tester data (analog voltage limits and tester loading requirements). It generates non-logic parametric data (series/parallel resistances, voltages) and diagnostic data used by AMDD in the pre-power-on tests. It also determines logic path information (come-from, go-to) used by AMDD in defect isolation. It generates the RAM tests and merges them with the preconditioning data sent from the engineering design system. Finally, it converts the test and diagnostic data into a format executable by the LT1280.

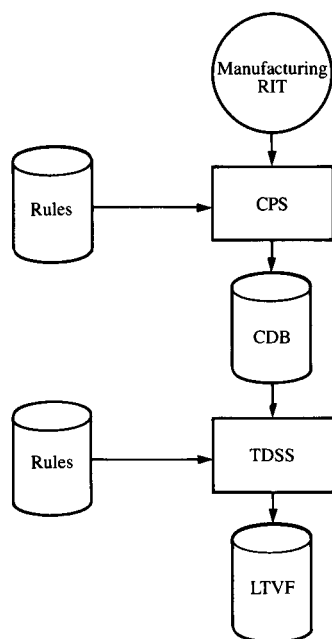


Figure 3 LT1280 test data preparation: RIT = release interface tape, CPS = corporate processing system, CDB = common data base, TDSS = test data supply system, LTVF = logic test verification file.

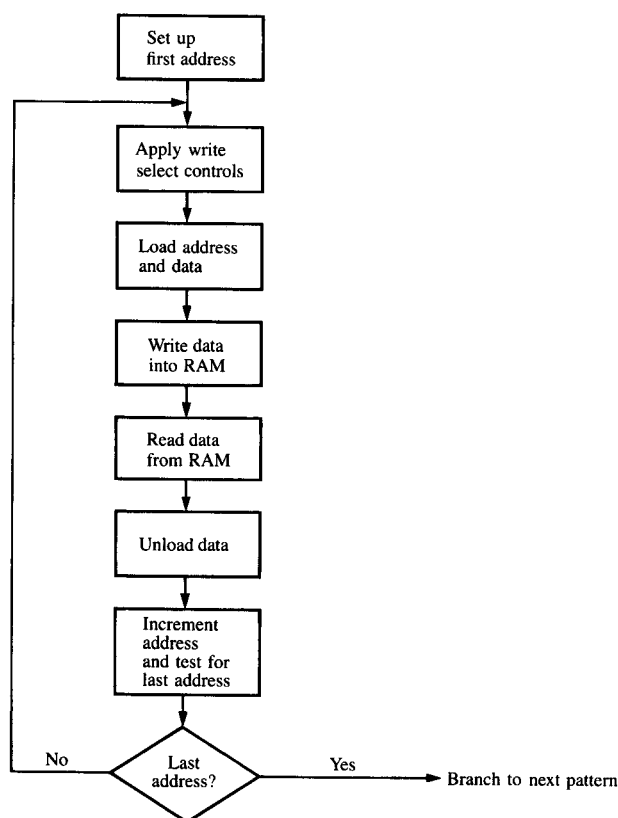


Figure 4 RAM test flowchart.

Through all steps of the test data preparation, a RIT that averages 80 megabytes of data is converted to a highly structured LTVF which averages 20 megabytes. The data volumes occasionally run as high as 225 megabytes for a RIT and 150 megabytes for an LTVF.

Test data application

The test data are applied by using three components: the host system (an IBM 4341), a local controller, and the LT1280 tester (Fig. 1). The host system operates the test control system (TCS), which communicates with a tester via the local controller. (Note that the host can support multiple local controller/tester pairs.) The AMDD programs and LTVF data base reside on the host system; TCS is accessed via a terminal connected to the host system.

Execution of tests is initiated from the host-attached terminal. Preliminary keyed-in commands are required to identify the TCM under test so that the proper test data can be accessed by the test control system. The TCM must be mounted in the tester and the local controller must have been initialized. AMDD then applies the test patterns and performs the diagnosis.

The logic stuck-fault tests for each test generation partition (TGP) are applied to one TGP at a time. The appropriate LTVF records containing operations, such as LOAD input, APPLY clocks, and UNLOAD shift-register string, are sent to the tester and are executed sequentially by utilizing the recursive hardware in the local controller [2]. The recursive hardware increases the rate of tester operations by a factor of eight compared to normal bus commands.

The RAM tests are applied in a manner very different from the combinational logic tests. One matrix (a group of RAM chips) is tested at a time. To reduce testing time and the volume of test data, the RAM test patterns consist of high-level address, branch, and loop commands. A RAM test is represented in the LTVF as an algorithm which loops; for each execution of the loop, the RAM address is incremented (see Fig. 4). The volume of test data is reduced significantly compared to a sequential, non-looping representation. In the latter case, the volume of data required for the same operations would be prohibitive. Also, because the volume of test data is reduced, there is less interaction between the host and the local controller during RAM testing. Thus, the test application time is also substantially reduced.

The tester sends back failing responses to the test control system via the local controller. Diagnosis of these test results is performed after each test. Because of the dependency of AMDD on physical access to the TCM via the probes, the module remains in the tester for the duration of test application and diagnosis.

LT1280 diagnostic approach

The objectives of the LT1280 diagnostic approach are two-fold. First, the diagnostic algorithm must pinpoint a fault to the smallest repairable area. Second, the diagnostic algorithm must minimize the number of repair cycles needed to produce a good product.

As mentioned earlier, the TCM product has been designed with a high degree of repair capability. Since 98% of the defects found in a TCM are repairable with manufacturing tools, a TCM manufactured for the first time can be shipped as new after the repair. However, too many repair cycles can degrade the performance to an unacceptable quality level. This need to minimize the number of rework cycles, as well as the economic advantages of detecting and correcting a defect as early in the manufacturing stage as possible [8], dictates that the diagnostic routine be accurate. It must specify replacement of the defective component and leave the rest intact so as not to introduce new assembly defects on tested good parts of the TCM. It must also be capable of diagnosing as many independent defects as possible in one test application.

The automatic multiple-defect diagnosis (AMDD) algorithm has been developed for diagnosis of TCMs on the LT1280; its objectives are fault isolation, diagnostic effectiveness, and efficiency. Fundamental to the AMDD strategy is that diagnosis of the results of each test is performed immediately after the test is applied. The tests, described later, are applied in the following order: pre-power-on tests, shift-register tests, RAM tests, and combinational logic tests.

The pre-power-on tests, which detect hazardous and similar conditions, abort the test pass if a hazard is detected, thereby avoiding product damage. Otherwise, testing continues, using test generation partition (TGP) elimination and fault mark-off techniques to prevent unnecessary diagnosis of the same fault by different routines. TGP elimination is triggered by the results of the pre-power-on and shift-register test results. For example, if a module primary-input (PI) pin fails a pre-power-on test, the TGP(s) containing that PI are eliminated from further testing. If a shift-register string is defective, TGPs containing that shift-register string are eliminated from testing. Other TGPs not affected by these earlier-detected defects are eligible for further testing.

Fault mark-off is a technique used in RAM and combinational logic diagnosis. If the diagnostics isolate a defect, the observation points [i.e., the module primary-output (PO) pins or the SRLs logically fed by that defective logic] are flagged. This indicates that the defect causing the failures observed at the POs and SRLs has already been determined. Thus, RAM and logic diagnostic routines do not attempt to rediagnose the same faults.

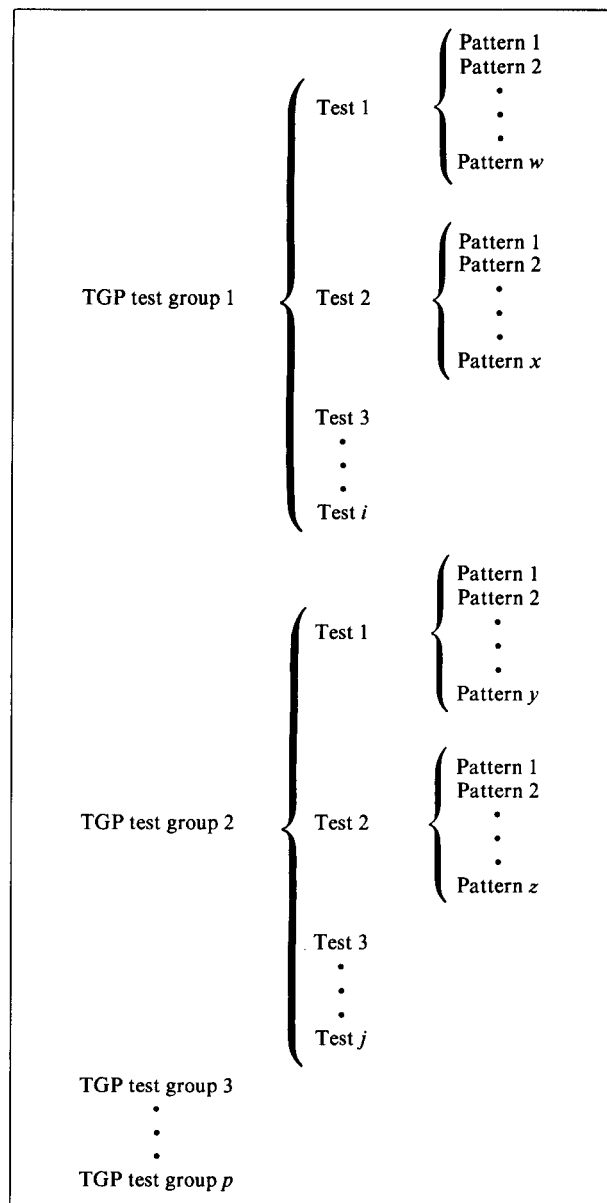


Figure 5 Logic stuck-fault test structure. A set of patterns comprise a test. A set of tests comprise a test generation partition (TGP) test group.

The logic TGP concept for multiple-defect diagnosis is further exploited by AMDD. The logic test structure is shown in Fig. 5. Each TGP has a series of tests composed of input patterns and the corresponding expected output values. AMDD sequentially applies the tests for each eligible TGP until all tests for a TGP are successfully completed or until the program stops at the first failing test. For a failing test, failure isolation is attempted for each failing PO or SRL not marked off by RAM or logic diagnosis. Thus, multiple

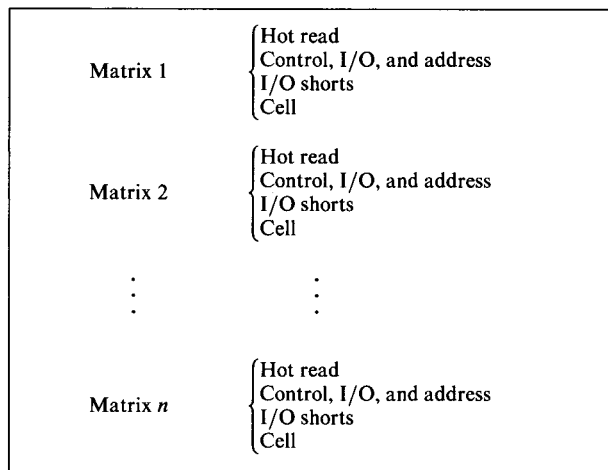


Figure 6 RAM test structure.

(a)

TEST (CONTROL I/O AND ADDRESS TEST)

ADDR	CHIP 1	CHIP 2
0000	XXXXXXXXXX XXXXXXXXXXXX	
0001	XXXXXXXXXX XXXXXXXXXXXX	
0002	XXXXXXXXXX XXXXXXXXXXXX	
0004	XXXXXXXXXX XXXXXXXXXXXX	
0008	XXXXXXXXXX XXXXXXXXXXXX	

(b)

TEST (CONTROL I/O AND ADDRESS TEST)

ADDR	CHIP 1	CHIP 2	CHIP 3	CHIP 4
0000				
0001				
0002				
0004				
0008				

(c)

TEST (CELL TEST)

ADDR	CHIP 1	CHIP 2	CHIP 3	CHIP 4
0008				

Figure 7 Sample RAM failure manifestations. (a) Control line failure: RAM 1 (chip 1) fails all address and all data positions. (b) I/O failure: data bit 2 on chip 2 fails all addresses. (c) Cell failure: data bit 1 on chip 4 fails at one address.

defects can potentially be detected within a TGP, as well as across TGPs. Today, AMDD is not capable of detecting multiple defects within one TGP when the defects cause failures in different failing tests, because AMDD stops test application at the first failing test in each TGP. Also, AMDD assumes that a single defect is responsible for a failure observed at a PO or an SRL. In either case, when AMDD cannot detect multiple defects in one test pass, an additional test and repair cycle is required after the first set of defects have been repaired (because the repaired defects have masked other defects). The ability of the RAM tests to isolate and exercise each RAM chip permits all the RAM tests to be executed (Fig. 6). The PO/SRL fault mark-off technique is used by RAM diagnostics to aid in its own diagnosis and to assist logic failure isolation.

To meet the objective of diagnostic effectiveness, AMDD uses two approaches. First, the concept of a RAM defect analysis is used primarily in RAM diagnosis. Some defects manifest themselves in known failing patterns, especially in RAMs which have highly ordered control and data structures. Figure 7 shows examples of RAM failure manifestations. Whenever possible, AMDD performs RAM defect analyses to determine the defect. When such analyses are not sufficient, AMDD uses the second approach, failure isolation. This is accomplished by utilizing the two single-point probes on the LT1280. The probes contact EC pads surrounding each chip and act as internal module test points, thus providing AMDD with the diagnostic data needed for further defect isolation.

AMDD diagnostic algorithm

The AMDD sequence is diagrammed in Fig. 8. A description of each test and test diagnosis follows.

• Pre-power-on tests

The pre-power-on tests are performed to ensure that the product has made good contact with the tester and that the TCM I/O pins are operational. Two types of tests are performed: terminators, opens, and shorts tests (TOST) and low-voltage tests (LVT). These tests check for shorts between different I/O nets, for opens on the same I/O net, for poor, missing, or misconnected terminators on I/O nets, for I/O nets shorted to ground, and for I/O nets shorted to some voltage level.

Analysis of some failures is done directly from the test results without probing. Opens and terminator failures require probing by the *repair* program, the final step of AMDD. The fifth failure type, I/O nets shorted to some voltage level, is hazardous to the TCM; thus, testing ends if this situation is detected. Each other error type is flagged as a terminal net containing a defect; TGPs containing these terminal nets are flagged for elimination from further testing.

• Shift-register tests

The LSSD structure requires that all latches be clocked and contained in one or more shift-register strings (Fig. 2) [4(b)]. Thus, the entire logic structure is controllable and observable via shift-register strings and, subsequently, via the TCM input and output pins. Therefore, the shift-register circuitry must be verified as being functional prior to testing of the RAMs and combinational logic on the TCM. This is accomplished with flush and scan tests.

The flush tests verify the ability of the shift register to function as a data bus. The two clocks controlling the scan operation are activated at the same time. A logic value of one is applied at the shift-register input (SRI) and must be

observed at the shift-register output (SRO) after a time equal to the delay through the latches in the string. Similarly, a logic value of zero is applied at the SRI and must be observed at the SRO after the appropriate delay.

The scan test is a serial operation which tests the ability to scan values into and out of the string, one bit at a time. The scan clocks are alternately turned on and off in a non-overlapping manner as a logic 10101010 . . . pattern is scanned through the shift-register string, followed by a 01010101 . . . pattern. This ensures that both a logic 1 and a logic 0 can be scanned into and out of each SRL.

Typically, a shift-register string on a TCM spans many chips. Only the SRI and SRO are accessible at the module pins. The objective of AMDD is to isolate the defect (or defects) to a chip or a wire in the string. AMDD uses a bisection routine, first placing one of the movable LT1280 probes on a chip scan input midway in the string. The failing shift-register test is reapplied on a shortened string (i.e., to all chips in the first half of the string), and the probe is treated as the SRO. This procedure is repeated until the shift-register test passes, and the defective area is isolated. For each defect one more probe move is needed to determine if the defect is in the wire feeding the problem chip. If not, the chip is called out for replacement. Test generation partitions containing the defective shift-register string(s) are eliminated from further testing.

• RAM tests

The RAM test data order is shown in Fig. 6. A description of the tests is given here. *Hot read tests* detect interference on a data bus across chips and matrices by applying a unique pattern for each chip and by verifying that only the unique pattern belonging to a chip is read at its own outputs. The *control*, *I/O*, and *address tests* are used to detect faulty control, address, or "stuck" data inputs (outputs) by writing and reading a 1 (0) at all primary addresses (0, 1, 2, 4, 8, . . .). The *I/O shorts test* detects short circuits between data-in and data-out lines by applying opposite values to two data lines and observing whether a short circuit exists. This is repeated for all physically adjacent data-in and data-out pairs. Finally, the *cell tests* detect the ability of each cell to retain both one and zero values by writing and reading a one and then a zero to each cell in the RAM. The implementation of the tests is based on the RAM test algorithm described by Knaizuk and Hartmann [9].

The interpretation of the RAM test results is done by array defect analysis, as shown in Fig. 7. When probing is required, it is limited to nets (wires) going directly into or out of the RAM. Probing involves locating the probe on an EC pad and reapplying a portion of the RAM tests. Precalculated expected values for the net being probed are compared

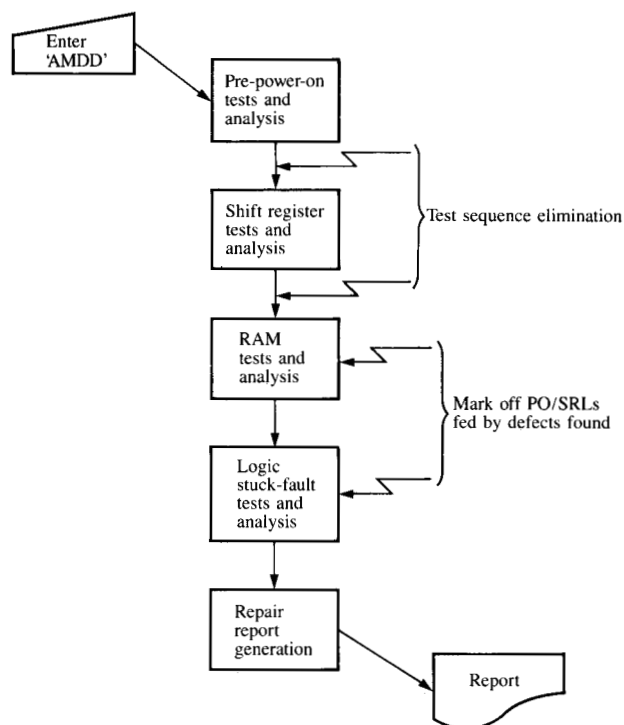


Figure 8 Automatic multiple-defect diagnosis (AMDD) flow.

to observed values to determine where the fault exists. If diagnosis cannot isolate the defect to the RAM chip or to its inputs or outputs, the observed values are recorded. These are used to help find the defect(s) by logic failure isolation. The diagnostic results fall into four categories. If the RAM diagnosis has isolated a failure within a chip or within its connection to the module substrate, the chip is classified as a *failing RAM chip*. The chip is called out for replacement. PO/SRLs fed by the chip data outputs are marked off to eliminate further diagnosis of failures observed at those PO/SRLs. If the RAM diagnosis has isolated a failure to a net feeding or fed by a RAM chip, it is classified as a *RAM terminal net*. RAM diagnosis detects three types of terminal nets: an open net, a net failing at a wrong analog value, and a net possibly shorted. Further diagnosis of this failing net is performed by the *repair* program, which makes the repair call. The PO/SRLs fed by this net are marked off to eliminate further diagnosis of failures observed at the PO/SRLs. When classified as a *defect in logic feeding RAM*, the diagnosis has determined that the input(s) to the RAM chip have incorrect logical values, thus indicating that a defect exists in the logic feeding the RAM. This net is flagged as a starting point for logic failure isolation. Finally, when the RAM diagnosis has probed and found no defects, it

TCM SERIAL # 12345678 PART # 2345678 EC # 345678
DATE - 083182 TIME - 09:18 TESTER # 02

SRL FAILURE ANALYSIS
LT2RR001-THE FAILING CHIP OUTPUT IS CHIPSITE(J03) C4(A01) EC(D01).
LT2RR007-SRL(0320) FAILED DURING FAILURE ISOLATION
LT2RR01A-REPLACE CHIP AT LOC(J03) IT CONTAINS TERMINAL SRL #(0320).

RAM FAILURE ANALYSIS
LT2RR003-REPLACE THE FOLLOWING RAM CHIPS:
CHIPSITE(K07) CHIP PN(4567890)

Figure 9 AMDD repair report.

'AMDD' STARTED AT:	09:21:16	
COMPLETED AT:	09:30:31	
TOTAL TEST TIME:	09:15	MIN:SEC
TOTAL NON-DIAGNOSTIC TIME:	06:39	MIN:SEC
LOGIC:	05:00	MIN:SEC
RAM:	01:39	MIN:SEC
TOTAL DIAGNOSTIC TIME:	02:36	MIN:SEC
RAM ANALYSIS:	01:42	MIN:SEC
FAILURE ISOLATION:	00:20	MIN:SEC
REPAIR ANALYSIS:	00:34	MIN:SEC
TOTAL PROBE MOVES:	9	
TOTAL DEFECTS:	2	
TOTAL TIME PER DEFECT:	01:18	MIN:SEC

Figure 10 AMDD performance report.

is classified as *no defects at RAM*. This indicates that a defect exists in the logic between the RAM and the PO/SRL where the failing result was observed.

◆ Combinational logic tests

Eligible TGP's are tested using the stuck-fault test patterns generated by the engineering design system. The tests for a TGP are applied until the first failing test, or until all the tests for a TGP have been applied and have passed. After the TGP's have been tested, a failure isolation routine is invoked. In a manner similar to RAM probing, the probe is used as an internal module test point while the failing test is reapplied. Failure isolation attempts to find defects for all PO's and SRL's which fail in the first failing test within a TGP.

Using logic interconnection data (come-from, go-to) received on the release interface tape and loaded into the LTVF, AMDD builds a "backtrace" list containing all the logic feeding a failing PO or SRL. If this list contains a failing net flagged by RAM diagnosis, failure isolation begins probing from that failing net. Otherwise, the probing sequence is guided by use of the precalculated diagnostics, which were generated by the engineering design system test generation programs. These diagnostics contain faults, weighted by their probabilities of occurrence, which may cause a test to fail. The precalculated diagnostics are accumulated for chips in the list and thus the probe is directed toward the component most likely to be defective. When a

defect is found, the PO or SRL being processed is marked off, and the next eligible PO or SRL is analyzed.

◆ Repair report generation

The final step of AMDD is to generate a list of suggested repair actions for the TCM under test. In some cases, the repair action has been determined by an earlier diagnostic routine (such as RAM diagnosis which concludes that a RAM chip should be replaced). In most cases, the defect has been found (e.g., a shorted net) but the action to be taken to repair that net has not yet been determined. The repair program must make the final determination of the action required by analyzing information passed from the earlier diagnostic routines and by performing a TOST on nodes in failing nets. The repair routine can call for a direct repair (e.g., replace chip) or a conditional repair (e.g., make a visual inspection, consider the repair history of the TCM before selecting the repair action from AMDD alternatives), or may request additional analysis (if AMDD was unable to isolate the defect). In the last case, the probing capabilities of AMDD are available in a manual mode to aid an analyzer in finding the defect. A sample repair action is shown in Fig. 9.

In addition, the repair routine produces a summary containing statistics for this AMDD test pass (the time spent in each step of AMDD, the number of probe moves, etc.). These data are collected and used to monitor the performance of AMDD. A sample performance report is shown in Fig. 10.

AMDD experience

Since the early 1970s, AMDD has evolved into a large and complex system, comprised of greater than one million lines of code and some sixty programs. It was originally designed only with the pre-power-on, shift-register, and logic test and analysis routines; the repair routine was added later. AMDD was first introduced into a manufacturing environment in 1980; the RAM diagnosis algorithm was added the following year.

The effectiveness and efficiency of AMDD are evaluated by determining the average time needed for application of all tests (*test application time*), the average time used by the diagnostic programs in AMDD to determine the required repair action for a defect (*time per defect*), and the percentage of AMDD test passes on defective TCMs for which AMDD is ineffective in fully determining the repair action (*retest percentage*). The latter situation requires retesting after the AMDD-diagnosed problem(s) are fixed.

AMDD has been steadily improved and its performance is acceptable for manufacturing operations, although new problems always occur. For example, the generation of RAM test data and the integration of RAM diagnostics into an already established AMDD proved to be a major accom-

plishment. The high density and complexity of the logic in the TCM challenged all phases of the test system, both in terms of data volumes and processing times. Unanticipated defect types, found only after AMDD was introduced into the manufacturing environment, have required many enhancements.

Figure 11 shows how AMDD performed over a recent four-month period. During that interval, the test application time averaged 2.35 min and the time per defect averaged 2.2 min. Both times showed continuing improvements. The AMDD retest percentages have averaged 10%, indicating correct diagnostic repair calls for 90% of the defective TCMs. However, the retest percentage has been increasing. This is attributed mainly to the increased occurrences of unanticipated defect types encountered during large-volume production.

Additional AMDD improvements are being made. The area of multiple-defect diagnosis within a TGP is being addressed to help achieve reduced test times. The LT1280 data plan is to have fewer and larger TGPs. Thus, the ability to detect additional faults with tests beyond the first failing test within a TGP is required. Also required is the ability for AMDD to use the TCM repair history to make more definitive repair calls and to automate the tracking of AMDD effectiveness and efficiency.

Conclusions

Automatic multiple-defect diagnosis has demonstrated that an automated, through-the-pins diagnostic approach is achievable for a dense VLSI package. Although some enhancements must still be made, AMDD has integrated the TCM reparability, the test-generation capabilities of the IBM engineering design system, and the LT1280 hardware and manpower resources into an effective diagnostic system.

Acknowledgments

The AMDD concept and algorithm were originally designed at IBM's General Technology Division facility in Endicott by M. S. Narasimha. G. Hack, W. McAnney, J. Manning, and A. Moschner were responsible for the original RAM diagnostic algorithm; A. Kopec was responsible for the repair algorithm. The author thanks D. Calvin and E. Barna for their assistance with this paper. The effort of these individuals and many others who contributed to the success of the LT1280 test system is gratefully acknowledged.

References

1. A. J. Blodgett and D. R. Barbour, "Thermal Conduction Module: A High-Performance Multilayer Ceramic Package," *IBM J. Res. Develop.* **26**, 30-36 (1982).
2. R. L. Pierson and T. B. Williams, "The LT1280 for Through-the-Pins Testing of the Thermal Conduction Module," *IBM J. Res. Develop.* **27**, 35-40 (1983, this issue).

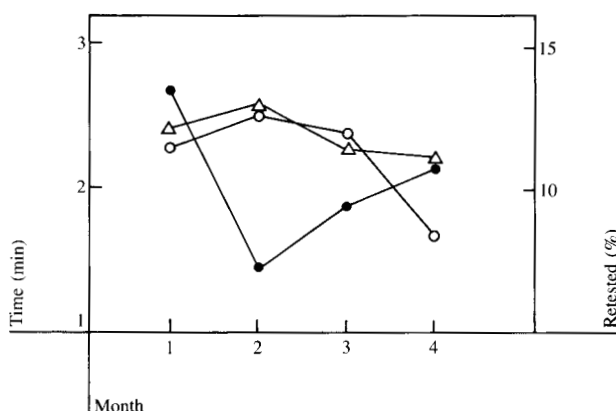


Figure 11 AMDD performance plots showing (Δ) test application time, (○) diagnostic time per defect, and (●) retest percentage.

3. L. F. Miller, "Controlled Collapse Reflow Chip Joining," *IBM J. Res. Develop.* **13**, 239-250 (1969).
4. (a) E. B. Eichelberger, E. I. Muehldorf, T. W. Williams, and R. G. Walther, "A Logic Design Structure for Testing Internal Arrays," *Proceedings of the 3rd USA-Japan Computer Conference*, 1978, pp. 266-272. (b) E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testability," *Proceedings of the 14th Design Automation Conference*, New Orleans, LA, June 1977, pp. 462-468.
5. H. C. Godoy, G. B. Franklin, and P. S. Bottorff, "Automatic Checking of Logic Design Structures for Compliance with Testability Ground Rules," *Proceedings of the 14th Design Automation Conference*, New Orleans, LA, June 1977, pp. 469-478.
6. P. S. Bottorff, R. E. France, N. H. Garges, and E. J. Orosz, "Test Generation for Large Logic Networks," *Proceedings of the 14th Design Automation Conference*, New Orleans, LA, June 1977, pp. 479-485.
7. J. P. Roth, "Diagnosis of Automata Failures: A Calculus and a Method," *IBM J. Res. Develop.* **10**, 278-291 (1966).
8. D. S. Cleverley, "Product Quality Level Monitoring and Control for Logic Chips and Modules," *IBM J. Res. Develop.* **27**, 4-10 (1983, this issue).
9. J. Knaizuk, Jr. and C. R. P. Hartmann, "An Algorithm for Testing Random Access Memories," *IEEE Trans. Computers* **C-26**, 414-416 (1977).

Received October 26, 1981; revised September 7, 1982

Patricia Latus Barry IBM Data Systems Division, P.O. Box 950, Poughkeepsie, New York 12602. Ms. Barry is a development engineer and manager of the Advanced Test Engineering Department, working on self-test. She joined IBM at East Fishkill, New York, in 1974, and worked in the design automation field in East Fishkill and Kingston, New York, until joining the TCM testing project in 1980. She was responsible for the testing and introduction into manufacturing of the TCM array diagnostic programs and later became manager of the TCM Test Data and Diagnostics Department. She received an IBM Division Award in 1980 for the development of software tools for programmable logic arrays (PLA). Ms. Barry received her B.A. in computer science from Cornell University, Ithaca, New York, in 1974, and also graduated from the IBM Systems Research Institute, New York, in 1979.