

S/390 G5 CMOS microprocessor diagnostics

by P. Song
F. Motika
D. R. Knebel
R. F. Rizzolo
M. P. Kusko

This paper describes the strategies and techniques used to diagnose failures in the IBM 600-MHz S/390® G5 (Generation 5) CMOS microprocessor and the associated cache chips. The complexity, density, cycle time, and technology issues related to the hardware, coupled with time-to-market requirements, have necessitated a quick diagnostic turnaround time. Beginning with the first prototype of the G5 microprocessor chip, intense chip diagnostics and physical failure analysis (PFA) have successfully identified the root causes of many failures, including process, design, and random manufacturing defects. In this paper, three different diagnostic techniques are described that have enabled the G5 to achieve its objective. An example is presented for each technique to demonstrate its effectiveness.

Introduction

Fault diagnosis is a very important concern in modern VLSI design and testing. Initial silicon problems have many sources in the testing environment. These sources include (but are not limited to) inaccurate fault models, incorrect test-pattern generation, erroneous pattern translation from the design program to the manufacturing tester, improper tester setup, and hardware failure due to process problems, random manufacturing defects, or marginal design problems.

Time-to-market demands have required quick and accurate diagnostics for the initial chip implementation. At the same time, high-performance microprocessor design is driving performance cycle times to several hundred MHz while increasing chip density and complexity. The custom-designed S/390* G5 microprocessor chip (CP) contains more than 25 million transistors and runs at 600 MHz [1]. Diagnosing this dense, high-frequency chip is an enormous challenge. Depending on only one diagnostic technique (e.g., software) to localize defects is not always an efficient approach. During the G5 system bring-up, several diagnostic methods were used, including software-based, tester-based, and light-emission-based techniques. These three techniques are described in detail in this paper.

Currently, the most common diagnostic approach in both industry and academia uses several post-test software algorithms. The purpose of a software diagnostic tool is to identify a list of potential fault candidates, or fault "callouts," given a faulty response to a particular chip stimulus. These fault candidates should be precise enough so that physical failure analysis can be done on the chip to identify the physical defect and, most significantly, the root cause. Many diagnostic algorithms have been published and implemented in commercial tools over the past several decades. Generally, there are two types of diagnostic techniques: *cause-effect* analysis techniques [2–6] and *effect-cause* techniques [7–11]. Cause-effect techniques depend on the stored symptoms caused by possible faults, and use the observed responses to locate

©Copyright 1999 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.

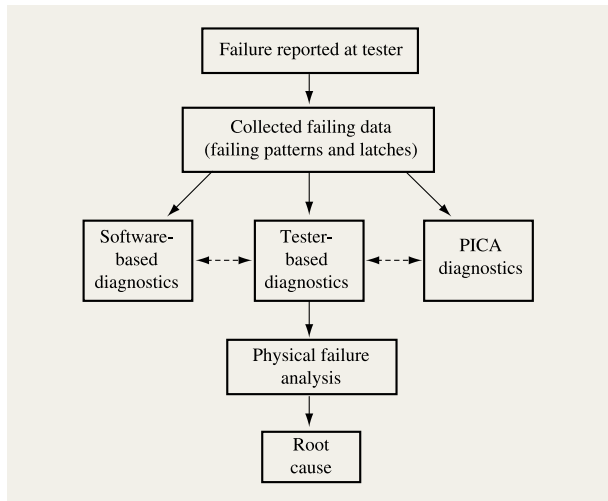


Figure 1

Overview of S/390 G5 chip diagnostics.

the fault. The *fault dictionary* approach [5, 6] is one of these techniques. Problems with this approach, especially for large chips, include 1) excessively long simulation time with prohibitively large memory requirements, and 2) ineffective physical and electrical failure analysis due to low diagnostic resolution. In contrast, effect-cause techniques do not depend on pre-stored data, but instead process the response obtained from the device under test (DUT) to determine the possible faults that generate the response. Effect-cause algorithms are less CPU-intensive and are well suited to fault diagnostics. In [6, 8], a deduction algorithm is used to determine the faults on the basis of faulty responses, while in [9, 11] fault simulation is adapted to predict the faults. The IBM tool *TestBench* [12] uses effect-cause techniques to diagnose faults for which the effects are the failing patterns and failing latches or primary outputs (POs) obtained from the tester. Fault simulation is used to diagnose failures discovered with various pattern types, such as deterministic “stuck-at” patterns, weighted random patterns (WRP), and logic built-in self-test (LBIST) in both dc and ac test modes.¹

A major advantage of the software diagnostic technique is that it is usually faster than other methods. There are, however, a few disadvantages. Fault candidates can be wrong, or there can be many candidates with low scores. A fault with a low score is one with a low probability of explaining the failure. Several scenarios can explain these situations: 1) incomplete failure data; 2) unmodeled fault types, such as path-delay faults and bridging faults; and 3)

faults in those areas of logic that are not fully represented in the fault model (such as clock logic). All of these have led to the development of several tester-based diagnostic techniques.

Tester-based diagnostic techniques can be classified in the hardware diagnostic category. These techniques are simple and fast, and they take advantage of the scannable latches in the design. They are particularly useful in diagnosing ac defects in the scan chain, and can often be done in minutes. The diagnostic resolution is usually sufficient for physical failure analysis. Tester-based diagnostics become more complicated when the fault is embedded in the logic.

Diagnostic techniques based on front-side photon-emission microscopy (PEM), liquid crystal hot-spot analysis, fluorescent microthermal imaging (FMI), electron beam, and so on belong to the category of hardware diagnostic tools [13], but are more specialized and difficult to use. There is a substantial hardware investment, while as device sizes shrink and the number of metal layers increases, resolution becomes worse. As a result, IBM researchers have developed static and dynamic *back-side* light-emission techniques [14–17]. Two important advantages of this approach are that 1) chips may be optically analyzed while all I/O contacts are accessed by a tester, and 2) the metal layers (including the I/O solder balls) do not block the emission. This technique has been used to diagnose ring oscillator recirculating loops [14, 15], I_{ddq} problems [16], and ac-defective circuits [17]. The time-resolved light-emission tool is called picosecond imaging circuit analysis (PICA). Extremely high diagnostic resolution results in the identification of a single defective transistor, easing the task of physical failure analysis. However, the diagnostic time is longer than that of both software and tester-based diagnostic techniques.

Overview of diagnostics on the G5 microprocessor

Figure 1 shows the three methods being used in S/390 G5 diagnostics: software-based diagnostics (TestBench), tester-based diagnostics, and diagnostics using dynamic light imaging (PICA).

After the failing chip is identified on the tester, these three methods are used synergistically to speed up the diagnostic turnaround time. The methods complement and overlap one another. Depending on the type of failure, one or another technique will have greater success. For instance, PICA successfully handles ac failures, TestBench has greater success with dc failures, and tester-based diagnostics work best when the failure is in the scan chain or involves a small number of logic elements. Since TestBench diagnostics are fast to run and do not require additional tester resources, every failure is first diagnosed using this approach. The candidates determined in this

¹ In this paper, the term *dc test* refers to a stuck-at-fault test, and *ac test* refers to a transition-fault test.

step provide an initial logical location for further diagnostics. This first step greatly reduces the overall diagnostic time.

Tradeoffs among the three diagnostic methods in terms of diagnostic resolution, analysis time, and cost are shown in **Figure 2**. As previously stated, software diagnosis is always applied first because it is easy to use, fast, and less expensive in resources than the other two methods. Analyzing one set of failure data usually takes from several minutes to several hours. In situations for which the failure data is sufficiently detailed, software diagnosis can identify individual logic gates for physical failure analysis. However, for a variety of reasons (as described in the Introduction above), the actual diagnostic resolution is sometimes not as good as required.

Tester-based diagnosis has the advantage of flexibility. Logic values on the input latches of the cone can be modified on-the-fly to pinpoint the failure. Other conditions, such as voltage, temperature, and pin timings, can be varied to aid in isolating the failure. Another benefit of the tester-based technique is that the device under test can be used as an on-line good-machine simulator (GMS). In a typical case, it can take several days to identify with confidence the physical location of a failure. Generally, the resolution is to the logically equivalent gates. Easily diagnosable with these tester-based techniques is an ac scan-chain failure. Within a few minutes, a physical location for failure analysis is typically found, and this is highly effective in rapid yield learning. It is generally assumed that failures in the scan chains represent similar failures in the remaining logic. Many examples have demonstrated the success of this method.

In contrast to the other two methods, the PICA technique does not require monitoring of the POs or measure latches; the faulty behavior needs only to be set up or activated. Only the timed light emission from the back side of the chip is measured during circuit switching. The chip must be thinned, and special tester fixtures are also required before the light emission can be measured. This is the most expensive and time-consuming of the three techniques, with a typical PICA diagnosis taking a week or more. However, the resolution is extremely high, and examples show that failures can be isolated down to an individual transistor. With such a precisely identified failing physical location, less time is needed in physical failure analysis, resulting in a very high success rate.

Design for test and diagnosis

Diagnostics must be planned for from the beginning of the design program. The correct software tools, along with the necessary design interfaces, are critical for fast debugging and diagnostics. A fundamental requirement is scan design, with level-sensitive scan design (LSSD) being the ideal situation. Scan design not only provides internal

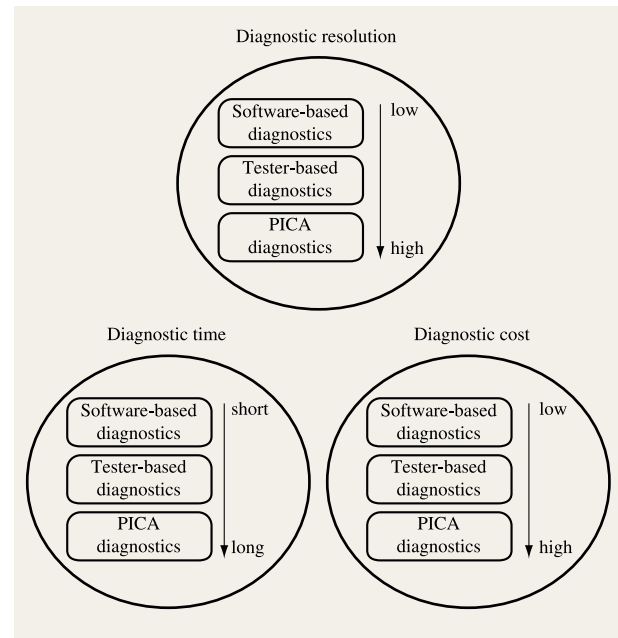


Figure 2

Relationship among three diagnostic methods with respect to diagnostic resolution, time, and cost.

control points for test stimulation but also provides access for measuring or observing internal circuit behavior. A scan-based design facilitates the use of automated software diagnostic systems. Using scan-based patterns, diagnostic algorithms can be applied which analyze the failing patterns, determining the most likely fault to explain the failure. Different fault models, along with additional failure data, can be used in conditions for which the candidate is not clear, i.e., the score is low. These automated techniques are relatively fast and inexpensive, and leave the failing chip intact. Other tester- or hardware-based techniques complement these software techniques. In these cases, the software techniques supply the starting point from which the tester techniques can work. These additional tester techniques also require the design for diagnosis (DFD) interfaces, the most important being full-scan design.

The S/390 G5 processor chip is a full-scan design with more than 70 000 latches [18], meaning that all latches are accessible by means of a serial scan chain. Nearly all latches are LSSD [19, 20], with standard master/slave L1/L2 shift-register latches (SRLs) as shown in **Figure 3**. The L1 or master latch has two data ports and may be updated by either a scan clock or a functional clock. The L2 or slave latch has only one clock input, and that clock

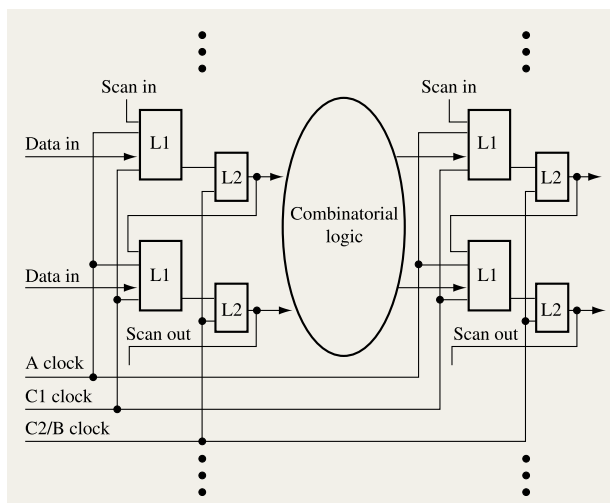


Figure 3

LSSD structure.

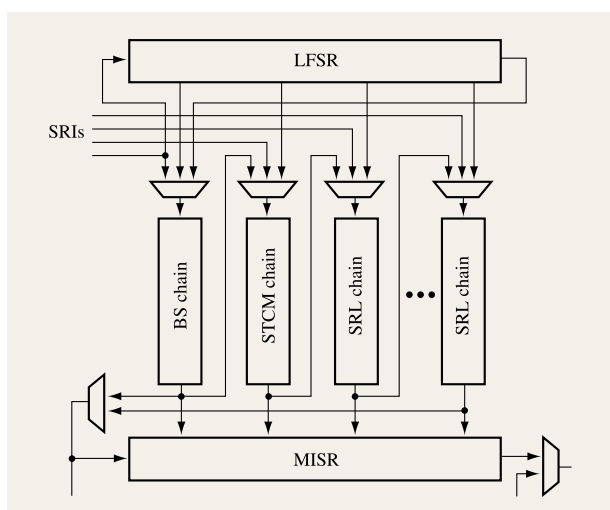


Figure 4

STUMP structure.

is out of phase with both L1 clocks. Scanning is done using separate A and B clocks.

Scan has the benefit of being key in both the design-for-test (DFT) and design-for-diagnostics (DFD) arenas. Sometimes design is implemented just for debugging or diagnostics, but the goal is to maximize the usefulness of any technique employed [21, 22]. The built-in self-test (BIST) implementation in the G5 processor chip is

optimized to increase its usefulness in both test and diagnostics. The BIST for the embedded memories is programmable [23]. Inherent in this programmability is support for modifying tests and varying conditions on-the-fly in support of debugging. This flexibility enables precise excitation and isolation of the failure.

The BIST for the random logic is based on a self-test using MISR and parallel SRSG (STUMPS) configuration [24], as shown in **Figure 4**. The scan chains are split into 122 evenly balanced chains, with the input stimulus fed from a pseudorandom pattern generator (PRPG) and the output responses collected into a multiple-input signature register (MISR). Both the PRPG and MISR reside within the chip. This LBIST is designed to run at 600 MHz. Speeds for scan-in and scan-out are variable and are independent of the speeds at which the functional clocks are applied. Once the LBIST is started, on each scan clock cycle the PRPG is updated and new data is loaded into each stump channel. After all of the stump latches are loaded, or, equivalently, all latches are initialized, one or more of the functional clocks are applied. The data is then scanned out with the data from each stump channel latch and compressed into the signature register. Both DFT and DFD benefit from this technique. Reliance on the tester is minimized, the tester throughput time is relatively fast, there is greater diagnostic isolation, and the designed-in flexibility in the clocking enhances detection, excitation, and resolution.

In addition to supporting the STUMPS configuration, the G5 processor chip also supports a weighted random pattern (WRP) [25, 26] scan-chain configuration. From a configuration perspective, WRP is similar to LBIST, except that the pseudorandom stimulus is provided externally by the tester and the output responses are also collected at the tester. The scan chains are split into 16 balanced chains. Both WRP testing and straight deterministic testing use this configuration. Since the scan-chain inputs and outputs are PIs and POs, scan-chain isolation for debugging is easier.

The use of the scan chain can be further optimized by controlling scan loading and unloading. The latches may be scanned in two ways, nonskewed or skewed. In a nonskewed load, after the scan, both the L1 and L2 have the same data. That is, the scan-in operation ends with a "B" scan clock pulse. Since the scan operation is not critically timed, the data from the L2 has a generous amount of time to propagate to the L1 before the functional L1 clock is pulsed. Thus, this nonskewed load is used primarily for dc testing. In a skewed load, the scan-in ends on an "A" scan clock, allowing the L1 and L2 latches to have different values. The subsequent functional clock sequence of a "B" or "C" clock can be critically timed to support the detection of dynamic faults or delay defects [27]. Other clock sequences or combinations of scan and

functional clocks can be used to target specific faults in clock logic or logic not bounded by master-slave latches. Generally, G5 testing methodology can use many diverse clock sequences.

The G5 chip design also uses scan-only latches that are internal control points and are initialized before the test technique (LBIST, WRP, etc.) is executed. These scan-only latches are used to reconfigure the scan chains, enable and disable chains, and serve in a variety of other roles in “flip-flopping” the test technique between supporting DFT and supporting DFD.

Throughout this discussion, scan is the fundamental building block. As the chips become denser and lithographies shrink, the importance of scan will only increase.

TestBench diagnostics

TestBench supports chip diagnostics in different test modes. As shown in **Figure 5**, the chip logic models, test patterns, and failure data are all used as diagnostic inputs. Fault simulation is done using a general-purpose significant events simulator. Diagnostics on WRP and LBIST patterns first require the patterns to be converted into stored patterns, and this conversion is done automatically by TestBench. TestBench diagnostics use fault-grading algorithms to compute scores on the simulated faults. Both dc and ac faults can be included in the diagnostic simulation. The fault that best explains the failure receives the highest score. In many cases, more than one candidate fault can be given the highest score.

Figure 6 shows an example of a TestBench diagnostic callout. The highest possible score is 100. In the case in which many faults receive a high score, additional work is needed to determine which fault really explains the failure. Two methods are used to distinguish the candidate faults. The first method simulates the suspect fault by injecting it into the simulator and checking whether any other tests detect this fault. If they do, these tests are reapplied on the tester to verify the targeted fault.

Another way to distinguish the faults is to generate a new test pattern which detects the suspect fault. This special test can be generated using the interactive back-trace justification capability of TestBench. This test pattern is then applied on the tester, and the suspect fault is either verified or found not to be true.

The TestBench visual test-pattern analysis tools are also used in diagnosing the faults by displaying both good-machine and faulty-machine waveforms of a circuit at the desired pins. Also shown are the values of all of the internal nodes for a specific test pattern. In the case of a dynamic test pair, transitions are clearly visible on the internal nodes, helping to trace the ac fault.

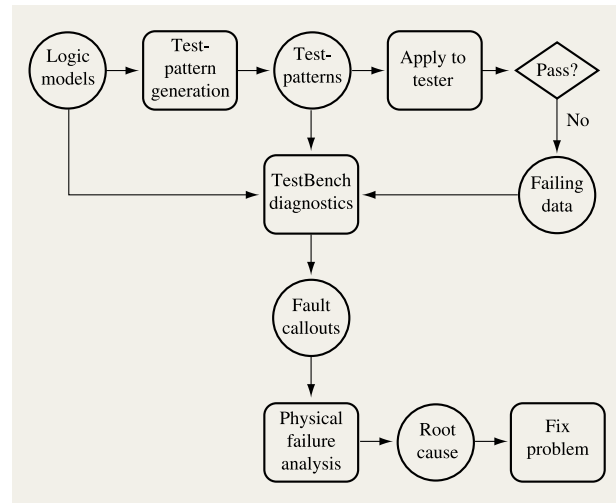


Figure 5

Flow diagram of TestBench diagnostic procedure.

• TestBench diagnostic example

This sample is a G5 CPU chip that failed after several hours of stress. Figure 6 shows the TestBench diagnostic results. Six faults have the highest score of 100, and two of them are redundant (Fault Index 3541421 and 3541430). To find which one really causes the failure, the techniques described in the preceding section are used. Once the fault is isolated to a single AND gate (Fault Index 3541421), the chip is ready for physical failure analysis.

Figure 7 shows the transistor-level schematic of the defective gate and the corresponding scanning electron microscopy (SEM) image of the defect. A metal-and-polysilicon (“poly”) line short was found to be the source of the failure, as shown in Figure 7(a). In Figure 7(b), bright lines are metals while dark lines are poly. It is noted that the leftmost metal line is shorted to its neighboring poly line.

Tester-based diagnostics

The tester-based diagnostic method consists of a highly interactive set of engineering and diagnostic techniques and tools developed to support quick and accurate pinpointing of both systematic and random circuit faults. These diagnostic techniques are intended primarily to be used during early bring-up of complex high-performance designs coupled with the introduction of new high-density technologies.

The philosophy at the root of this diagnostic technique is to provide the test and diagnostic engineer with an open-ended set of tools and simple interfaces that encompass the test-generation tools, test system,

FaultIndex	ScoreA	ScoreB	TFSP	TFSP	TPSF	Consistent	Conflicting	NoContribution	Dropped								
1012793	97	0	7	0	2	1	1	0		1012793	^	OSA1	5020410	NBUF	2	Pin.f.l.cp.nl	.
3540545	97	0	7	0	2	1	1	0		3540545	r	OSA0	5020406	NBUF	2	Pin.f.l.cp.nl	.
1011648	98	0	7	0	1	1	1	0		1011648	^	OSA0	5021720	NBUF	1	Pin.f.l.cp.nl	.
1013389	98	0	7	0	1	1	1	0		1013389	^	OSA1	5019636	NAND	1	Pin.f.l.cp.nl	.
1013487	98	0	7	0	1	1	1	0		1013487	^	OSA0	5019528	NOR	1	Pin.f.l.cp.nl	.
1011908	100	1	7	0	0	1	0	0		1011908	^	ISA0	5021474	NOR	2	Pin.f.l.cp.nl	.
3541421	100	1	7	0	0	1	0	0		3541421	r	OSA0	5021470	AND	2	Pin.f.l.cp.nl	.
1013506	100	1	7	0	0	1	0	0		1013506	^	OSA0	5019510	NAND	1	Pin.f.l.cp.nl	.
1013508	100	1	7	0	0	1	0	0		1013508	^	ISA1	5019508	NAND	2	Pin.f.l.cp.nl	.
3541430	100	1	7	0	0	1	0	0		3541430	r	OSA1	5021476	NOR	2	Pin.f.l.cp.nl	.
1199625	100	1	7	0	0	1	0	0		1199625	^	OSA1	4878082	NBUF	1	Pin.f.l.cp.nl	.

FaultIndex	ScoreA	ScoreB	TFSP	TFSP	TPSF	Consistent	Conflicting	NoContribution	Dropped
TD6226 (1)	$\text{ScoreA} = (\text{TFSP} / (\text{TFSP} + \text{TPSF}/10)) * 100$ $\text{ScoreB} = \text{Consistent} - \text{Conflicting}$ TFSP = How many pins/latches that failed at the tester, passed(didn't fail) in simulation TFSP = How many pins/latches that failed at the tester, failed in simulation TPSF = How many pins/latches that passed(didn't fail) at the tester, failed in simulation Consistent = Count of the faults that cause "at least one" measure point to fail in simulation that failed at the tester (TFSP>0) Conflicting = Count of the faults that cause "at least one" measure point to fail in simulation that passed at the tester (TPSF>0) Nocontribution = Count of the faults that have NO measure points fail in simulation. (TPSF=0 and TFSP=0)								

Figure 6

An example of TestBench diagnostic callouts.

conventional diagnostic tools, and associated techniques. This tightly coupled set of tools should be simple to use, highly interactive, easily adaptable to new test approaches, and able to offer the engineer a high degree of test and diagnostic freedom.

The ultimate goal of tester-based diagnostics is to utilize all available laboratory resources to electrically diagnose the fault to within a couple of logic blocks or a dozen or so devices as rapidly as possible. A further goal is to bridge the diagnostic process between the electrical model and the physical location by providing conventional PFA and PICA tools with special diagnostic test patterns and acceptable physical locations for the potential defect.

These tester-based diagnostic tools, both software and hardware in nature, usually evolve as a result of new test methodologies, technology enhancements, and design concepts, for which the standard diagnostic approaches and tools are incompatible or have not yet been developed. Furthermore, these new approaches are inherently fault-model-independent and are highly effective when diagnosing unmodeled faults, ac defects,

and intermittent failures that do not conform to the classical or conventional stuck-at or transition fault models. As a result, many of these tools are somewhat customized and specifically implemented for the product design and problem encountered, but many of the underlying basic concepts can be generalized and integrated into general-purpose automated test-generation and diagnostic products.

Generally, tester-based diagnostics encompass and support a broad area of testing, including structural, functional, both ac and dc parametrics, several logic and memory built-in self-test methods, and even some quite uncommon test techniques. Furthermore, the hardware equipment, test systems, and test-generation and diagnostics software also span a varied and diverse set of tools and product packages.

• Tester diagnostic techniques

This section of the paper describes some of these tester-based diagnostic methods and tools, as developed and applied toward the initial bring-up and power-on of the

G5 system. As previously indicated, the system is based on LSSD and structural test methodology, integrating logic and memory built-in self-test structures.

Specifically, two areas of diagnostics are elaborated on. First, since scan design and structural test methodology are dependent on one or more scan chains to access the internal logic, several approaches used to diagnose problems associated with these scan chains are discussed. Additionally, the logic built-in self-test concept used in the G5 design is based on on-board random pattern generation and signature analysis [18, 24]. The techniques used for diagnosing logic within this self-test environment are also discussed.

Before the specifics of some of these tester-based approaches are described, it should be noted that several global test and environmental parameters are varied throughout the diagnostic process. These parameters may include the device temperature, power-supply voltages, timings, light, and others. Some of the parameters are typically “shmoosed” [28] throughout the operating range and beyond to determine defect sensitivities and to improve or aggravate the device response.

Another basic technique invoked across several tester-based diagnostic methods is the use of a “good” or “golden” reference operating point for on-the-fly or dynamic comparison to the failing test condition. This reference point can be on the same defective device being diagnosed or on an alternate good device previously tested and found to be fully functional. There are several approaches in using the reference, depending on the hardware and software tools available on the test system. Ideally, one would prefer the reference point to be on the device being diagnosed, but using a multiple test head to test, save, and compare is also a viable approach, although it is usually slower.

- *Scan-chain diagnostics*

The diagnostics associated with scan chains can be separated into two main categories. The first category can be characterized as dc stuck-at or “broken” chains [29–32]. Conversely, the second category is typically encountered when one or more scan chains exhibit an ac defect, but operate properly or “shift” at a slower clock rate.

1. DC stuck-at chains

The diagnosis of these chains is often critical when a low- or zero-yield situation is encountered, usually due to a process problem. Typically, a dc stuck-at chain may consist of a scan-path problem, a latch problem, or a scan-clock problem. This may result in a scan chain that is a “hard” stuck-at, i.e., a chain that is stuck in such a way that data supplied to the scan-in port cannot be measured or observed at the scan-out port when the scan clocks are

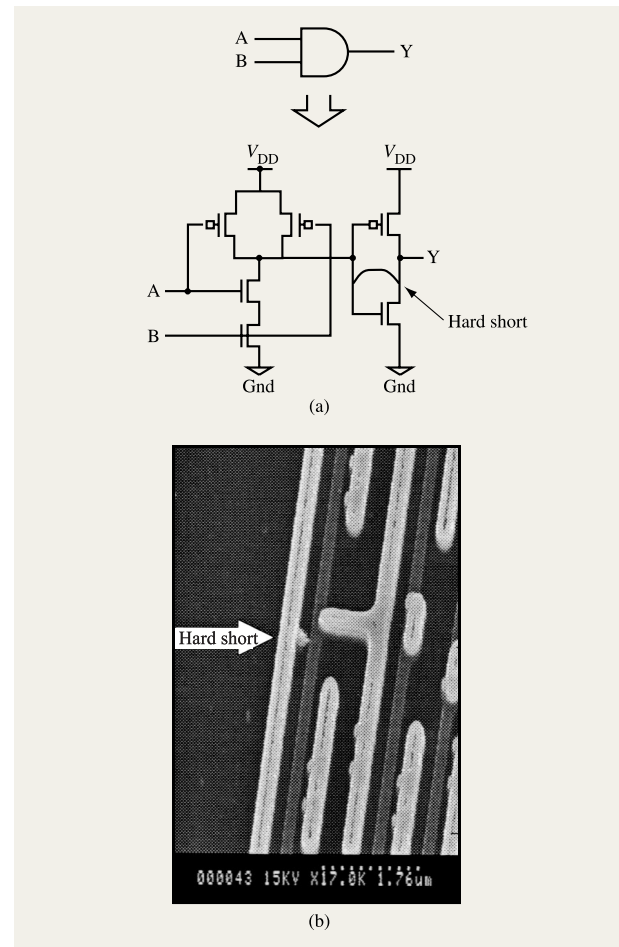


Figure 7

TestBench diagnostic example: (a) Defective gate schematic; (b) SEM image of the defect.

turned on. More complex defects can cause other than simple stuck-at-like fault conditions, making electrical fault isolation nearly impossible at times.

Since a defective chain often inhibits or limits sufficient stimulation and observation of the internal logic, conventional test and diagnostic methods become extremely ineffective. Several tester-based techniques have been developed to address this type of problem when partial loads and unloads of the chain are possible and the defects are relatively simple in nature. Some of these techniques include I_{ddq} testing and device power-supply sequencing variations.

Often several of the techniques are used concurrently to isolate the failing latch, narrow the range of possibly failing latches, or determine the associated faulty scan clock.

2. AC-defective scan chain

The diagnosis of ac defects in a scan chain is somewhat simpler than that for the stuck-at scan-chain problem described above. Usually an ac-defective chain functions or shifts properly at a slower clocking rate, but fails at higher scan rates or tighter timing setup conditions. The methodology applied in this diagnostic approach is based on using a "good" scan reference point and a three-phase test process. The reference point used is the same chip operating at a slower scan rate (i.e., relaxed timings) and possibly with a different power-supply setup. When dealing with marginal, intermittent, or multiple fails, more than one reference point may be characterized and used. The three test phases can be characterized as follows:

Identification phase The goal of this phase is to establish a stable test condition that exposes the ac failure. This is accomplished by varying several environmental variables, power settings, and timing parameters in conjunction with the application of diverse scan-pattern sequences. This set of predefined scan patterns may include propagating alternating sequences, single-transition sequences, single-latch sequences, adjacent-latch sequences, and other customized sequences.

When the design incorporates multiple scan chains or allows for reconfiguration of scan chains, the pattern sequences described above are applied to one or more chains while the remaining chains are held in a quiescent 0 or 1 state. Of course, when there are a large number of scan chains, the number of possible pattern combinations can also become relatively large, and judicious diagnostic pattern selection must be used. The scan-pattern generation and diagnostic process should consider all latch inversions within the scan chains.

Once the failure can be replicated and is stable, the patterns are used in the next phase of the diagnostic process. In cases for which the initial identification phase is unsuccessful, specialized patterns may be generated, but the diagnostic process turnaround time increases quickly and generally cannot be integrated into automated diagnostic tools.

Verification and localization phase The second phase of this diagnostic process is to select a specific failing pattern sequence and verify the passing reference point and the failing test-point conditions. These two test points and previously identified failing pattern or patterns are then used to localize the failure to a specific shift-register chain, latch, or range of latches. This is achieved by on-the-fly modifying the above pattern data and timing in conjunction with the execution of a binary search algorithm.

For simple single ac failures, this approach usually narrows down quickly to a single shift-register latch (SRL).

Alternatively, when one or more ranges of latches are identified, the problem is usually associated with a scan-clock distribution tree. By analyzing the physical layout of the associated clock distribution tree and latches, one can often identify the common tree branch and associated re-drivers.

Characterization phase Once the failure is localized to a latch or clock, the last phase of the diagnostic characterization is executed to determine the size of the ac defect, parameter sensitivity, and further circuit localization. This is typically done by modifying all timing edges and clock pulse widths for a specific set of scan-path data transitions.

Specifically, as in the case of the G5 designs described in this paper, the scan-chain shift-register latch usually consists of a master-slave pair (L1/L2) supported by a set of scan clocks, as shown in Figure 3. The initial goal of the characterization phase is to attempt to further isolate the failure to either the L1 or the L2 latch, the scan data path between the latches, or one of the scan clocks.

The size of the ac defect, once localized, can be evaluated by relaxing all of the timing edges except the launch and capture edge of interest and shmooing these two edges. For latch defects this may also involve shmooing the clock pulse width to determine the feedback or latching properties of the circuit.

Although the types of ac defects diagnosed with this scan-chain technique are relatively large because of noncritical scan-clock distribution or scan design methodology, and are limited by the tester timing accuracy, it has been highly effective in identifying and diagnosing some gross process problems early in the program. As device densities increase in the future and scan-chain latches increase proportionally, and as scan-clock distributions adhere to tighter system timing requirements, this diagnostic technique has the potential of becoming even more effective.

• Logic built-in self-test (LBIST) diagnostics

In this section, the diagnostic methods and techniques used to diagnose defects exposed by the LBIST pattern execution are described. The G5 system design architecture and test methodology are based on the STUMPS configuration [24] shown in Figure 4, and utilize linear-feedback shift registers (LFSRs) as on-board pseudorandom pattern generators (PRPGs) and multiple-input signature registers (MISRs). Furthermore, the system design is also based on level-sensitive scan design (LSSD) or general scan design (GSD) concepts [19, 20] supporting structural test techniques and basic combinational test and diagnostic methods.

In addition to the LBIST configuration, the system design incorporates phase-locked loops (PLLs) and a self-

generated clock [18], enabling bursts of at-speed test-vector clocking. By combining the concepts described above, one can generate effective dc and ac structural tests that can achieve high stuck-at and transition fault coverage while still maintaining the diagnostic concepts and tools developed for basic combinatorial logic.

Most of the basic LBIST diagnostic techniques used for the bring-up of the G5 system apply to both dc and ac failures, but here primarily the approaches used for ac diagnosis are described. One of the significant differences between the ac and dc tests is that the reference test point for ac is usually on the same defective device, while the reference test point for the dc test might be on a different “golden” reference device. Of course, when diagnosing initial design problems or systematic defects, the reference test point might not apply.

LBIST ac diagnostic techniques

Similar to the ac scan diagnostic methodology described above, the LBIST diagnostic is also a three-phased approach that uses many of the same basic techniques. In addition, some LBIST-specific techniques have been developed to support the STUMP structure and self-generated clock environment. The LBIST diagnostic process flow is shown in **Figure 8**; the three phases in this case can be characterized as follows.

- *LBIST extraction phase* The goal of this phase is to identify which pattern in the LBIST test sequence fails and then to extract the failing latches or observation points. Since the nature of LBIST signature analysis is to compress the response data for many patterns into a MISR signature, the first step in the diagnostic process is to identify the individual failing signatures and associated patterns using a binary search technique. When the result data for multiple failing test patterns is required for enhanced diagnostic resolution, a signature superpositioning algorithm is invoked to determine the expected signatures to be used during the follow-on binary searches [33]. A further side benefit of this approach is that the actual stimuli can be extracted for the desired test pattern by unloading the scan chain prior to applying the system launch and capture clocks. This feature becomes very convenient when the random patterns applied far exceed the basic pattern set. In [21], an analytical BIST diagnostic technique was proposed, and multiple failing latches per test can be found. However, some extra hardware and a post-test processing procedure are required. Another scheme for locating the failing latches and tests was introduced in [22] by using the pseudorandom elimination technique. Again, some hardware overhead is imposed for implementing the scan-cell/chain-selection block.

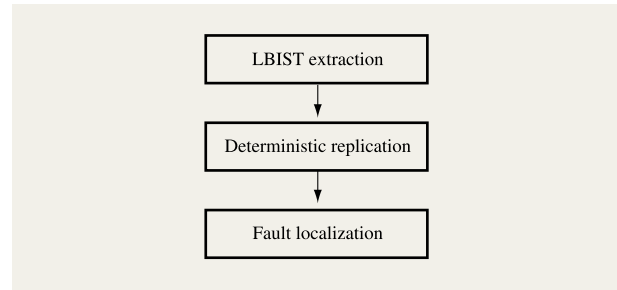


Figure 8

LBIST diagnostic process.

The end result of the first phase is the specific stimuli and associated failing responses for one or more test patterns, similar to those for equivalent stuck-at or transition test patterns. At this point one can use all of the above failing and passing test-pattern information to run standard fault-simulation and diagnostic tools to isolate or localize the defect. However, these off-line diagnostic tools often do not provide sufficient diagnostic resolution for quick and accurate PFA, and one must also invoke the next two phases of the interactive tester-based diagnostic process.

- *Deterministic replication phase* The task of the second phase is to use the above information to generate a set of quasi-deterministic patterns that replicate the failing conditions in the LBIST environment. Along with the stimuli for these patterns, one must also provide the corresponding clock sequences and expected good-machine values. When the system incorporates memories or sequential logic, the proper initialization patterns must be executed prior to the desired diagnostic test-pattern application. In designs that use extensive BIST support, one may need to invoke internally generated initialization sequences and clock-generation macros in conjunction with these patterns.
- *Fault localization phase* This highly interactive phase uses a broad range of tester resources, on-product DFT support, and off-line software diagnostic tools to pinpoint the logical location of the fault. Many of these tools have been developed specifically to address LBIST diagnostic needs and are to some extent customized to a particular test methodology and design. Of course, as with many logic diagnostic techniques, the design tools, test-generation tools, logic modeling, and fault modeling are integral components of the diagnostic process.

The fundamental concepts used by this interactive diagnostic technique consist of determining and minimizing the stimulus and observation cones of influence, selective path sensitization and transition

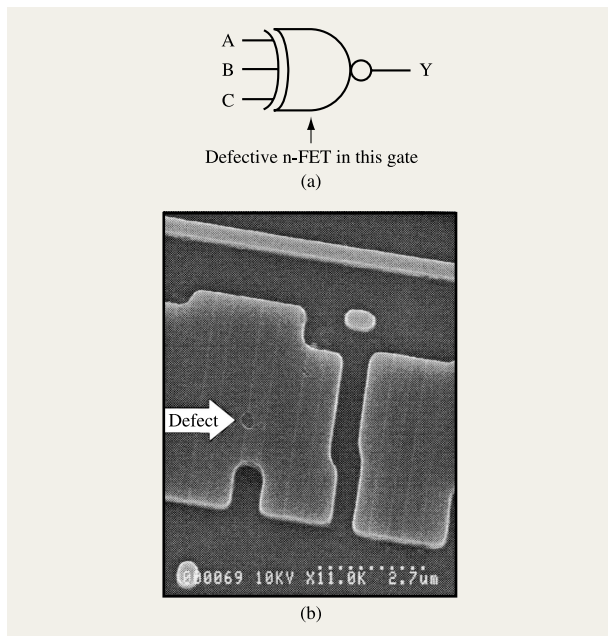


Figure 9

Tester-based diagnostic example: (a) Defective three-input exclusive-NOR gate; (b) SEM image of the defect.

propagation, and critical path timing control. These concepts, often used concurrently and complemented by external diagnostic support, allow for rapid convergence to both modeled and unmodeled logic faults. This is achieved by dynamically generating test patterns that iteratively reduce the forward and backward logic cones associated with the fault. These on-the-fly patterns are conditionally generated by algorithms that extend from simple single-transition propagation to extensive, exhaustive sequences.

An additional useful feature of this diagnostic technique in the LBIST environment is that the pattern set can easily be extended far beyond the simulated-signatures set and clocking sequences. The device operation at the good-reference test point can serve as GMS executing at system clocking rates. This enhances the detection of faults embedded in random-pattern-resistant logic designs. Furthermore, since the detection can be set up in a continuous or infinite test loop, intermittent defects can also be exposed and eventually diagnosed.

Of course, many of these defects can be complex in nature, resulting in equivalent multiple faults, nonmodeled faults, feedback shorts resulting in sequential circuits, intermittent defects, and many others. Often these defects

are difficult to model accurately, or they may belong to large fault-equivalence classes, causing accurate physical localization to fall short of desired goals. Pattern generation can also encounter restrictions such as latch adjacency, as well as orthogonality considerations.

In this section, a few of the tester basic diagnostic techniques used successfully in the initial power-on support of the G5 system are described. Although no single method or technique satisfies all diagnostic needs, the combination of many diverse approaches complementing one another can meet most diagnostic challenges. This tester-based diagnostic methodology and the associated techniques address one fault in the diagnostic methods spectrum.

• *Tester-based diagnostic example*

The second example is a logic ac failure on a G5-associated cache chip. It failed after burn-in stress. Since it is an ac failure, the golden reference condition is running at a slow cycle time. The failing latches and patterns are identified using a binary search technique (extraction phase). Then, TestBench is used to trace back the whole failing cone from the failing latches to the input scan latches and PIs on which the deterministic patterns can be applied (replication phase). Depending on the number of input scan latches and PIs involved, different algorithms are chosen to generate the deterministic patterns. In this example, there are only seven input scan latches (no PIs) in the failing cone, and it is possible to apply an exhaustive set of patterns to these latches. Thereafter, the test results can be analyzed, and the possible defective gates can be determined. To achieve a high diagnostic resolution suitable for physical failure analysis, the fault must be localized to a specific gate (fault-localization phase). This is often possible, especially when there is fan-out from the suspicious gates to multiple observation points. In this example, a three-input exclusive-NOR gate was confirmed to be the defective gate.

The three-input exclusive-NOR gate is a complex gate which contains 18 transistors; only the gate-level representation is shown in **Figure 9(a)**. On the basis of the test results obtained during diagnostics, circuit fault simulation was performed with the injected defects [34]. From the results, a defective n-FET was predicted to be the source of the failure. **Figure 9(b)** shows the SEM image of the defect. It is noted that a silicon attack during process is observed (the poly has been removed).

PICA diagnostics

The use of PICA as a diagnostic tool is an emerging technology that can provide precise identification of defect location. It is important to locate defects precisely to improve both the speed and the likelihood that a defect

can be analyzed to determine its root cause. PICA is positioned in the diagnostic process with other tools, such as e-beam (electron-beam probing), emission microscopy, and FIB (focused-ion-beam milling and repair). E-beam and emission microscopy provide information about the operation of circuitry that is not directly measurable by electrical testing or other forms of contact probing. FIB may be used to expose otherwise hidden circuit components for contact or contactless probing or may be used to modify internal circuit connectivity as an aid to indirectly deducing a failure mechanism.

PICA improves upon these techniques by providing a probeless means for detecting precisely when transistors switch. Probeless measuring techniques are important as integrated circuit feature sizes shrink and single-die designs become more complex. The G5 microprocessor is a high-speed, complex design that challenges current tester speeds and diagnostic instrument resolution.

In the following discussion, the PICA measurement technique and methodology for use as a chip diagnostic tool are briefly described. A specific example is outlined to show the detailed steps taken to pinpoint a defect location. The example is taken from a particular defect that could be dependably isolated to a single latch pair using scan-chain diagnostics, but more precise location of the defect was needed to improve the turnaround time of root-cause analysis.

- *PICA operation and methodology*

Normally biased CMOS logic circuits emit photons for only a short period during the switching transient, allowing precise timing of individual transistors. The emitted light can be detected from the front or back side of the die, but the large number of metal levels and the packaging of the G5 processor chip prohibit measuring through the front side. Lightly doped silicon substrates absorb a portion of the bandwidth of the emitted light, so samples to be analyzed were first thinned to improve emission intensity.

The samples require no further preparation, and the chip package and socketing used throughout the measurement are the same as those used in electrical testing. **Figure 10** shows a sample and the components of the measuring system.

An automated tester is used to stimulate the packaged device so that the transistors to be studied are switched repetitively. A standard infrared microscope is used to magnify and focus these devices onto the detection apparatus. A thermoelectrically cooled microchannel-plate (MCP) photomultiplier with a position-sensitive resistive anode is used to determine both the location and the time of a photon emission.

Two steps are employed to reduce the overall measurement time. One is the use of software or tester diagnostics to minimize the number of devices to be

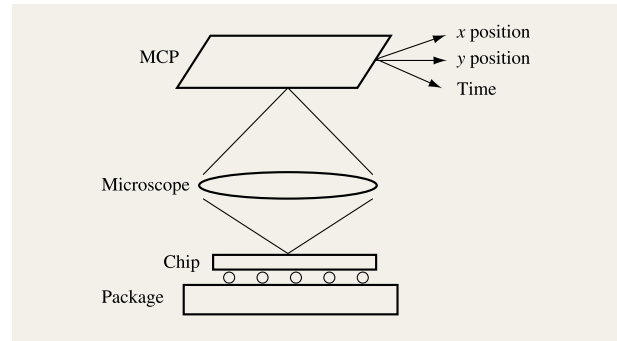


Figure 10

Schematic diagram of PICA measuring apparatus for back-side analysis.

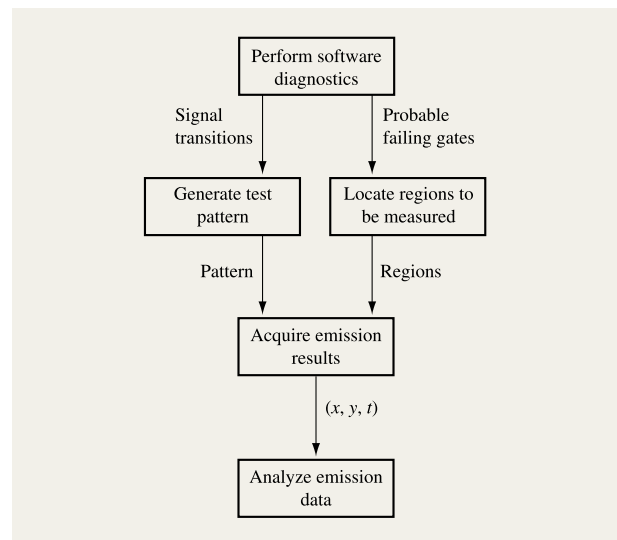


Figure 11

Flow diagram of PICA diagnostic procedure.

investigated. This information is used to select the magnification needed to spatially resolve the nearest transistors and to determine the number of measurements needed and their locations, given the field of view determined by the minimum usable magnification. The other step is selection of a test pattern that will rapidly cycle the circuits of interest through a desired switching state. **Figure 11** is a flow diagram of the diagnostic procedure.

The data collected from the measurements is postprocessed to provide insights into the device operation. Integration of the measured data over time

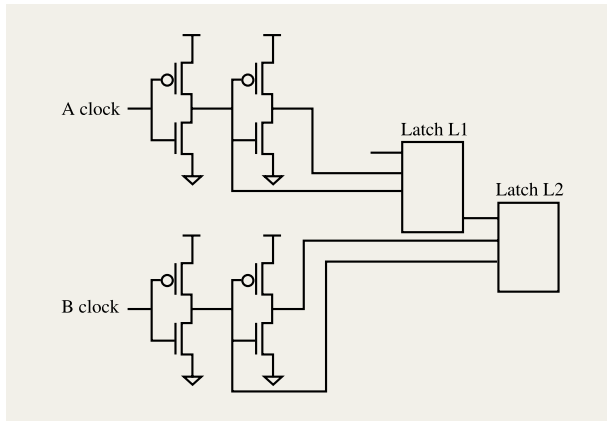


Figure 12

Schematic diagram of scan latch.

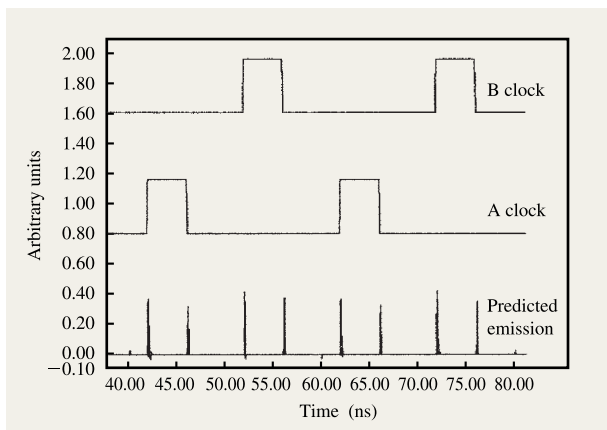


Figure 13

Test pattern and predicted emission amplitude vs. time.

creates an “emission photograph,” which shows the locations of all devices that switched throughout the test sequence. Selecting a single emission “spot” in the (x, y) plane of the collected image and plotting the time dependence of the emission intensity of the spot yield an optical waveform of the emission of the transistor or transistors within the spot.

Layout-to-schematic mapping is used to relate optical waveforms to circuit schematic elements, and provides a means for comparison to circuit simulation. Circuit delays and logic evolution can be deduced from the waveform and circuit schematic information. A circuit stuck at a high or low value is detectable by comparing the predicted

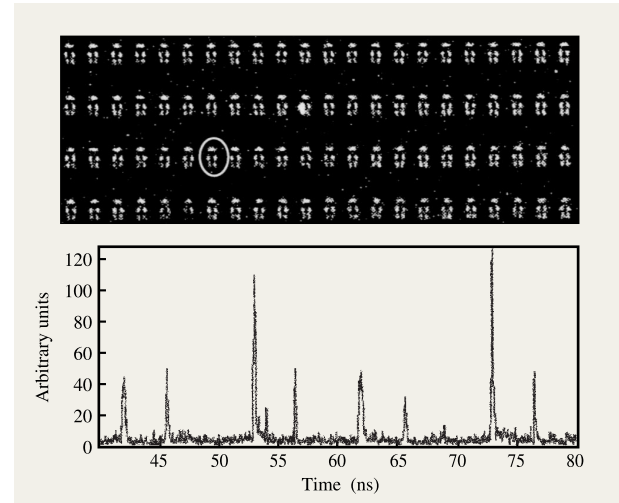


Figure 14

Emission measurement for a good latch.

switching activity of a good device for a tester pattern to the measured switching activity for that pattern. A timing failure is located by comparing the simulated time of such switching to the measured time of the switching.

• *PICA diagnostic example*

This example of defect diagnostics for G5 is taken from the yield learning cycle of the program. A high dc (stuck-fault) yield had already been calculated from wafer-level test statistics, but module-level testing at high speed revealed timing-related failures. As was previously mentioned, certain failures were detectable in the scan chain, making rapid diagnostic calls available through electrical testing.

In this case, however, locating the defect by contact probing would be very time-consuming, even after tester diagnostics, because of the large number of transistors and connections in the latch circuit. The scan-chain test identifies the timing defect by detecting a wrong value stored in the latch when the chain is running at a frequency that is within the specification of a good scan chain, but it alone cannot identify whether the failure is due to clock, gating, or memory circuitry. **Figure 12** shows a schematic diagram of the latch circuit.

Simulation of a good latch-circuit model was done to predict both the static emission pattern of the register file and a time-resolved emission waveform for a latch pair. **Figure 13** shows the voltage waveforms of the tester-generated clock signals and the predicted emission (current) waveform of the latch pair when loading a 1 and a 0 into latch B. Locating the peaks in the current

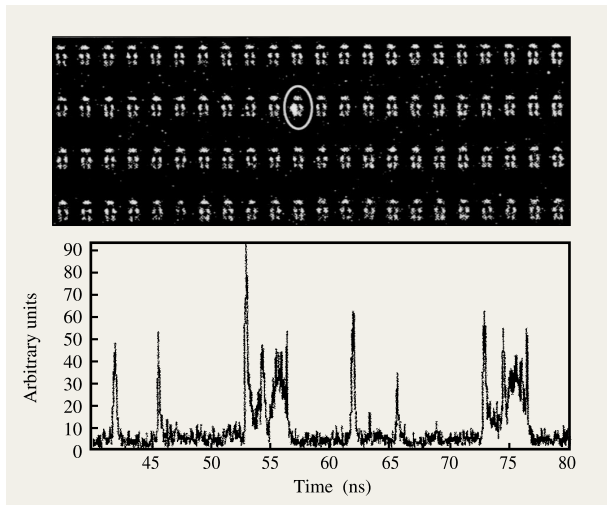


Figure 15

Emission measurement for the defective latch.

waveform predicts the timing of the light emission relative to the clock signals.

Measurements gathered over the field of view include a number of latches in the scan chain. **Figure 14** shows a time-integrated image of the emitted light and a time-resolved waveform for a single latch pair, as circled in the top portion of the figure. The locations of the emission spikes compare favorably with the predicted emission pattern and coincide with the edges of the clock signals.

The time-resolved emission waveform of the faulty circuit is extracted in a similar fashion. **Figure 15** shows the faulty circuit circled in the time-integrated image and the extracted emission waveform for that circuit. The time-integrated image shows an anomaly in the circuit, and the time-resolved waveform shows a relationship between the excess light emission and the period of the cycle during which the B clock is high.

The relationship between the measured switching activity and the B clock narrows down the range of possible defective devices or nets to a few possibilities. These possible defect mechanisms were examined in detail, and only one was found that could cause the emission pattern when gating either a 1 or a 0 with the B clock. The defect is modeled by a series resistor in the B clock signal path, as shown in **Figure 16**.

Simulation of the defective circuit compares qualitatively with the measured emission pattern, as shown in **Figure 17**. At this point there is enough evidence to commit the sample to deprocessing and physical analysis. **Figure 18** shows a small crack in the line that connects the B clock signal to the p-FET device layout.

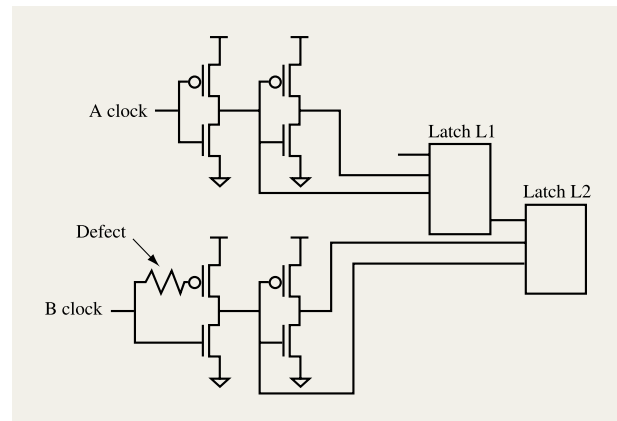


Figure 16

Circuit schematic showing simulated defect location.

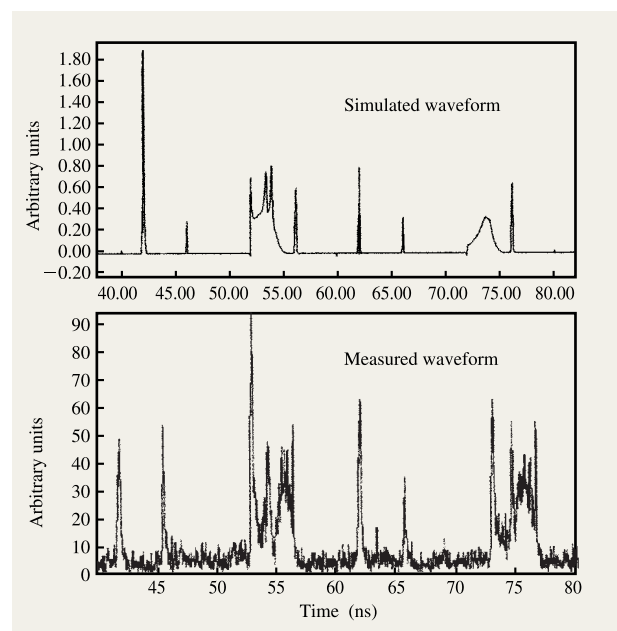


Figure 17

Comparison of simulated and measured emission waveforms.

The measurements taken in this example were made with a relatively low magnification objective, demonstrating that a large number of transistor measurements can be made in parallel. It is a goal of the PICA method to be able to take advantage of this to locate defects that are more difficult to isolate with software and tester diagnostics. Other measurements have

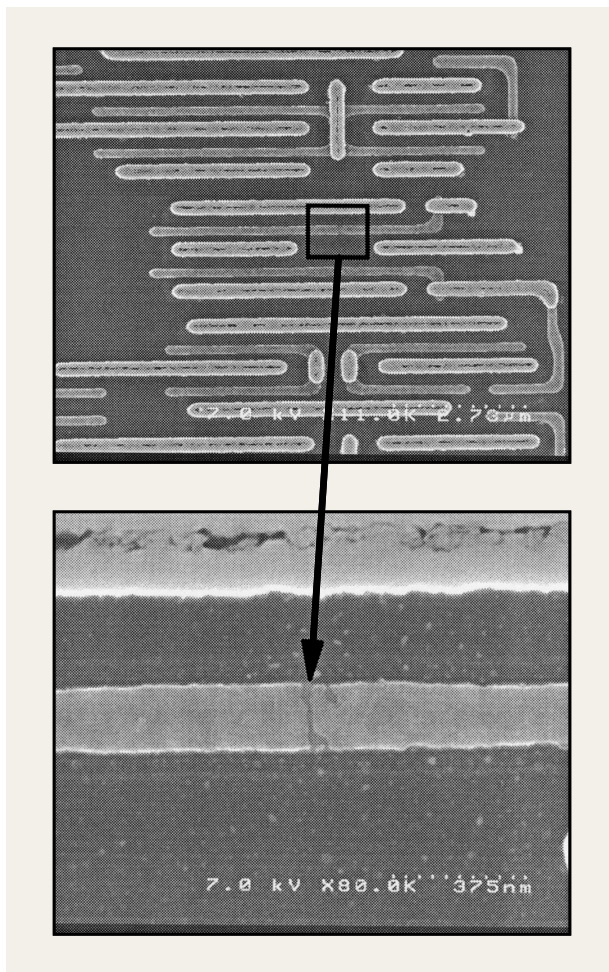


Figure 18

PICA diagnostic example: SEM image of the defect.

been made with higher magnification to spatially resolve a single transistor within a smaller field of view. This has been useful for determining the specific transistor or net containing the defect when it could not be determined by less direct means.

Fault localization is a critical step in the process of root-cause determination because of the need to turn each problem around as quickly as possible. PICA provides a means for both rough and fine defect localization, as well as defect-related timing characterization. While localization of stuck faults is possible, PICA is most particularly suited to the analysis of failures that have fewer, if any, software or tester-based diagnostic tools available for their analysis. The technique has been effective in the G5 yield learning exercise.

Conclusions

In this paper three diagnostic techniques have been described that complement one another. Software techniques, as exemplified by the effect-cause algorithms in TestBench, are well suited for dc stored-pattern test failures and are always used to narrow the scope of the problem if more detailed diagnostic work is needed. Tester-based diagnostics work well for certain types of failures, particularly ac scan failures, and are valuable for diagnosing other ac failures. PICA, an advanced diagnostic technique that detects device switching activity through the back side of a chip, has been used to successfully diagnose difficult-to-locate ac failures. It offers time resolution in the tens of picoseconds and spatial resolution down to an individual transistor. A powerful advantage is that the failing location need not be exactly known.

It has been noted [35] that the incidence of “soft” defects, those that fail only under certain conditions of voltage, timing, and temperature, is expected to increase. Soft defects, resistive opens being an important class, are a concern not only for manufacturing test, but for long-term reliability as well. Since many of these defects are not modeled well by single stuck-at faults, diagnosis is expected to be more difficult. A repertoire of diagnostic techniques such as we have described will be needed to attack these problems successfully. Software methods will have to integrate tester-based and PICA techniques to diagnose failures to the root cause so that process corrections can be most quickly implemented.

Acknowledgments

The authors gratefully acknowledge all of the G5 test team members, including Bryan Robbins, Larry Lange, Tim Koprowski, Bill Huott, Tom Foote, and Joe Eckelman. Other individuals who contributed to the success of this work are Donato Forlenza, Bill Hurley, Ray Kurtulik, Orazio Forlenza, Steve Wilson, Julie Lee, Bob Clairmont, Bill Bentley, Randy Wells, Rick Wasielewski, Wilt McClain, Dawn Tudryn, Glen Froese, Greg O’Malley, Debbie Hamm, Moe Hamel, Tom Higgins, Pia Sanda, Moyra McManus, Dennis Manzer, Steven Steen, Jeff Kash, and J. C. Tsang. Many thanks are due to Dale Hoffman, Phil Wu, and Rocco Robortaccio for their management support.

*Trademark or registered trademark of International Business Machines Corporation.

References

1. D. E. Hoffman, R. M. Averill, B. Curran, Y. H. Chan, A. Dansky, R. Hatch, T. McNamara, T. McPherson, G. Northrop, L. Sigal, A. Pelella, and P. M. Williams, “Deep Submicron Design Techniques for the 500MHz IBM S/390 G5 Custom Microprocessor,” *Proceedings of the International Conference on Computer Design*, 1998, pp. 258–263.

2. W. H. Kautz, "Fault Testing and Diagnosis in Combinational Digital Circuits," *IEEE Trans. Computers* **C-18**, 352–366 (April 1968).
3. S. Y. H. Su and Y. C. Cho, "A New Approach to the Fault Location of Combinational Circuits," *IEEE Trans. Computers* **C-21**, 21–30 (January 1972).
4. Nandakumar N. Tendolkar and Robert L. Swann, "Automated Diagnostic Methodology for the IBM 3081 Processor Complex," *IBM J. Res. Develop.* **26**, No. 1, 78–88 (January 1982).
5. T. Yano and H. Okamoto, "Fast Fault Diagnostic Method Using Fault Dictionary for Electron Beam Tester," *Proceedings of the IEEE International Test Conference*, September 1987, pp. 561–565.
6. M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*, AT&T Bell Laboratories and W. H. Freeman and Company, 1990.
7. F. Hsu, P. Solecky, and R. Beaudoin, "Structure Trace Diagnosis for LSSD Board Testing—An Alternative to Full Fault Simulated Diagnosis," *Proceedings of the 18th Design Automation Conference*, June 1981, pp. 891–897.
8. M. Abramovici and M. A. Breuer, "Multiple Fault Diagnosis in Combinational Circuits Based on an Effect-Cause Analysis," *IEEE Trans. Computers* **C-29**, No. 6, 451–460 (June 1980).
9. J. A. Waicukauski, V. P. Gupta, and S. T. Patel, "Diagnosis of BIST Failures by PPSFP Simulation," *Proceedings of the IEEE International Test Conference*, September 1987, pp. 480–484.
10. H. Cox and J. Rajski, "A Method of Fault Analysis for Test Generation and Fault Diagnosis," *IEEE Trans. Computer-Aided Design* **7**, No. 7, 813–833 (1988).
11. J. A. Waicukauski and E. Lindbloom, "Failure Diagnosis of Structured VLSI," *IEEE Design & Test of Computers* **5**, 49–60 (August 1989).
12. *TestBench User's Guide*, IBM Microelectronics Division, Endicott, NY, November 1996.
13. D. P. Vallet, "IC Failure Analysis: The Importance of Test and Diagnostics," *IEEE Design & Test of Computers* **13**, 76–82 (July/September 1997).
14. J. C. Tsang and J. A. Kash, "Picosecond Hot Electron Light Emission from Submicron Complementary Metal-Oxide-Semiconductor Circuits," *Appl. Phys. Lett.* **70**, No. 7, 889–891 (February 1997).
15. J. A. Kash and J. C. Tsang, "Dynamic Internal Testing of CMOS Circuits Using Hot Luminescence," *IEEE Electron Device Lett.* **18**, 330 (1997).
16. J. A. Kash, J. C. Tsang, R. F. Rizzolo, A. K. Patel, and A. D. Shore, "Backside Optical Emission Diagnostics for Excess I_{DDQ} ," *IEEE J. Solid-State Circuits* **33**, No. 3, 508–511 (March 1998).
17. D. Knebel, P. Sanda, D. Vallet, L. Huisman, P. Nigh, R. Rizzolo, P. Song, and F. Motika, "Diagnosis and Characterization of Timing-Related Defects by Time-Dependent Light Emission," *Proceedings of the IEEE International Test Conference*, October 1998, pp. 729–733.
18. M. Kusko, B. Robbins, T. Snethen, P. Song, T. Foote, and W. Huott, "Microprocessor Test and Test Tool Methodology for the 500 MHz IBM S/390 G5 Chip," *Proceedings of the IEEE International Test Conference*, October 1998, pp. 717–726.
19. E. B. Eichelberger, "Method of Level Sensitive Testing a Functional Logic System," U.S. Patent 3,761,695, September 25, 1973.
20. E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testability," *Proceedings of the 14th Design Automation Conference*, New Orleans, 1977, pp. 462–468.
21. Y. Wu and S. Adam, "BIST Fault Diagnosis in Scan-Based VLSI Environments," *Proceedings of the IEEE International Test Conference*, October 1996, pp. 48–57.
22. J. Rajski and J. Tyszer, "Fault Diagnosis in Scan-Based BIST," *Proceedings of the IEEE International Test Conference*, November 1997, pp. 894–902.
23. P. G. Shephard III, W. Huott, P. R. Turgeon, R. W. Berry, Jr., P. Patel, G. Yasar, J. Hanley, and F. J. Cox, "Programmable Built-In Self Test Method and Controller for Arrays," U.S. Patent 5,633,877, May 1996.
24. P. H. Bardell and W. H. McAnney, "Self-Testing of Multichip Modules," *Proceedings of the IEEE International Test Conference*, 1982, pp. 200–204.
25. F. Motika and J. Waicukauski, "Weighted Random Pattern Testing Apparatus and Method," U.S. Patent 4,688,233, August 18, 1987.
26. J. A. Waicukauski, E. Lindbloom, E. B. Eichelberger, and O. P. Forlenza, "A Method for Generating Weighted Random Test Patterns," *IBM J. Res. Develop.* **33**, No. 2, 149–161 (March 1989).
27. B. Konemann, J. Barlow, P. Chang, R. Gabrielson, C. Goertz, B. Keler, K. McCauley, J. Tischler, V. Iyenger, B. Rosen, and T. Williams, "Delay Test: The Next Frontier for LSSD Test Systems," *Proceedings of the IEEE International Test Conference*, 1992, pp. 578–587.
28. K. Baker and J. V. Beers, "Shmoo Plotting; The Black Art of IC Testing," *IEEE Design & Test of Computers* **13**, 90–97 (July/September 1997).
29. J. L. Schafer, F. A. Policastri, and R. J. McNulty, "Partner SRLs for Improved Shift Register Diagnostics," *Proceedings of the IEEE VLSI Test Symposium*, 1992, pp. 198–201.
30. S. Kundu, "Diagnosing Scan Chain Faults," *IEEE Trans. VLSI Syst.* **2**, No. 4, 512–516 (December 1994).
31. S. Edirisooriya and G. Edirisooriya, "Diagnosis of Scan Path Failures," *Proceedings of the IEEE VLSI Test Symposium*, April 1995, pp. 250–255.
32. S. Narayanan and A. Das, "An Efficient Scheme to Diagnose Scan Chains," *Proceedings of the IEEE International Test Conference*, 1997, pp. 704–713.
33. F. Motika, J. J. Shushereba, D. Forlenza, and P. Nigh, "Logic BIST Diagnostic Method," IBM Docket BU9-97-022, June 1997.
34. P. Song, F. Motika, M. Kusko, R. Rizzolo, J. Lee, and R. Clairmont, "High Resolution Diagnostic Techniques for the IBM S/390 Microprocessor," *Proceedings of the 8th IEEE North Atlantic Test Workshop*, May 1999, pp. 67–75.
35. Wayne Needham, Cheryl Prunty, and Eng Hong Yeoh, "High Volume Microprocessor Test Escapes, An Analysis of the Defects Our Tests Are Missing," *Proceedings of the IEEE International Test Conference*, November 1998, pp. 25–34.

Received November 30, 1998; accepted for publication July 15, 1999

Peilin Song *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (psong@us.ibm.com).* Dr. Song received the M.S. and Ph.D. degrees in electrical engineering from the University of Rhode Island in 1994 and 1997, respectively. He joined the IBM S/390 Division in 1997 as a member of the Custom Microprocessor Design Department. His current interests are in the area of design for test, diagnostics, modeling, and circuit simulation. Dr. Song has more than ten publications in the area of test generation, fault modeling, and diagnostics. He is a member of the IEEE and serves on the program committee of the IEEE North Atlantic Test Workshop.

Franco Motika *Micrus/IBM Corporation, East Fishkill facility, Route 52, Hopewell Junction, New York 12533 (motika@us.ibm.com).* Mr. Motika is currently a Senior Technical Staff Member in the area of VLSI test and diagnostics. He joined the IBM General Technology Division in 1968 in an advanced semiconductor memory characterization group and has since been involved with custom VLSI design and logic test. During this time Mr. Motika's assignments have included the architecture, software and hardware development, design, and implementation of various VLSI test systems, and the development of diverse test and diagnostic methodologies. He has received an IBM Corporate Outstanding Technical Innovation Award (1988), an Outstanding Innovation Award (1986) for the weighted random-pattern test concept, a GTD Achievement Award and a GTD Division Award (1983) for contributions to logic final test, four Outstanding Technical Achievement Awards for test throughput and product SPQL enhancements, and the Micrus CEO Recognition Award. He has also received several Invention Achievement Awards for various patents and innovations. Mr. Motika received a B.S. degree in 1968 and an M.S. degree in 1969, both in electrical engineering, from the Polytechnic Institute of Brooklyn, New York. He is a Senior Member of the Computer Society of the Institute of Electrical and Electronics Engineers.

Daniel R. Knebel *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (knebel@us.ibm.com).* Mr. Knebel is a Senior Engineer in the Advanced Systems Technology Department at the IBM Thomas J. Watson Research Center. He joined the IBM Data Systems Division in 1982, working on circuit families for high-performance processors. He has since worked in a variety of related areas, including VLSI verification, reliability, and diagnostics, electronic design automation, and microprocessor design. Mr. Knebel is currently part of a research team developing diagnostic analysis techniques for integrated circuit processes and products. He received a B.S. degree in electrical engineering from Purdue University in 1982 and has received four Technical and Invention Achievement Awards since joining IBM.

Richard F. Rizzolo *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (rizzolo@us.ibm.com).* Mr. Rizzolo is a Senior Engineer and served as test team leader for the S/390 G4 and G5 projects. He also had primary responsibility for sorting for S/390 chips designed in Poughkeepsie. He received his B.S. degree in physics from Rensselaer Polytechnic Institute in 1977 and his M.E. degree in electrical engineering from Rensselaer in 1980. Since joining IBM in 1978, Mr. Rizzolo has worked on bipolar and CMOS projects in the areas of design for testability, high-

frequency design and timing analysis, diagnostics, and circuit design. In 1994 Mr. Rizzolo was assigned to Boeblingen, Germany, to develop CMOS timing methodology. He is a member of the IEEE and holds three patents and several published inventions. Mr. Rizzolo is a co-author of several papers in the field of testability and diagnostics; he has received three Outstanding Technical Achievement Awards and two Outstanding Innovation Awards.

Mary P. Kusko *IBM Server Group, 1580 Route 52, Hopewell Junction, New York 12533 (eleve@us.ibm.com).* Ms. Kusko is a Senior Engineer in EDA at IBM East Fishkill, where she has worked as a consultant in test strategy definition and implementation. Earlier, she worked in S/390 processor design in the area of design verification and test. She received her B.S. degree in electrical engineering from the University of Delaware. Ms. Kusko is a member of the IEEE and the SWE, and serves on the ITC program committee. She has several publications and four patents pending.