Finding Faulty Components in a Live Fleet Environment

Technical Paper

August 2024

Revision 1.1

Intel Corporation Authors:

Benson Inkley

Michael Mishaeli

Document Number: 822279

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.

Intel technologies may require enabled hardware, software, or service activation.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Performance varies by use, configuration, and other factors. Learn more on the Performance Index site.

Your costs and results may vary.

"Conflict-free" refers to products, suppliers, supply chains, smelters, and refiners that, based on our due diligence, do not contain or source tantalum, tin, tungsten or gold (referred to as "conflict minerals" by the U.S. Securities and Exchange Commission) that directly or indirectly finance or benefit armed groups in the Democratic Republic of the Congo or adjoining countries.

All product plans and roadmaps are subject to change without notice.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visiting the Intel Resource and Document Center.

Contents

Figures

Revision History

Revision Number	Description	Date
1.0	• Initial release of the document.	May 2024
1.1	• Revised to include In-Field Scan web site link.	August 2024

1 Abstract

Over time, computing systems experience component failures due to natural wear-out phenomena, typically called reliability failures. Events such as these cannot be predicted or prevented; they can only be detected after the failure has occurred. Depending on the specifics of the failure, it may cause an application to halt, or it may manifest as a silent data error.

System administrators schedule regular maintenance sessions during which diagnostics are run to find failures that occurred since the prior session. How often and how extensive of a test to run is a complex question involving many variables.

Recognizing the importance of minimizing downtime, Intel developed Intel® In-Field Scan, a processor feature that enables very fast testing of processing cores without taking the node off-line. This paper discusses Intel In-Field Scan and key considerations regarding how often to test the processors.

Keywords: Silent data errors, fleet maintenance, Intel In-Field Scan.

2 Introduction

2.1 Intel® In-Field Scan Overview

5th Gen Intel® Xeon® Scalable processors, formerly codenamed Emerald Rapids, introduce a new Reliability, Availability, and Serviceability (RAS) family of tools called Intel In-Field Scan. These tools are designed to help system administrators quickly and easily find processors that have failed over time. In-Field Scan has a well-defined roadmap of capabilities that are included on 5th Gen Intel Xeon processors and future processors. Scan-at-Field (SAF) and ArrayBIST are the first two tools within the In-Field Scan family; both are available on 5th Gen Intel Xeon processors.

In-Field Scan tools have minimal impact on fleet operation because testing can take place on one core while all other cores in the node continue to run application workloads.

Scan[1] is an industry standard method for detecting defects in semiconductor devices. Until now, scan has been used only by specialized test equipment in the chip manufacturing factories. Intel uses scan to test processors during High-Volume Manufacturing (HVM).

Scan-At-Field enables customers to run a subset of Intel's manufacturing scan tests to check individual processing cores for defects. Using Intel supplied test patterns (called scan test images), each core within the processor package can be independently tested.

ArrayBIST checks the L1 and L2 caches and many other arrays in each core. Being a Built In Self-Test (BIST), there are no test images to load, all the testing is coordinated by a dedicated test module in each core.

Considerations as to how often to test a system for failures is provided in this document as well as an overview of SAF and ArrayBIST.

More information, including links to an enabling guide and test images, is available at the In-Field Scan web site.

2.2 Purpose of Intel In-Field Scan

To assist system administrators in finding the aforementioned reliability failures, Intel developed DCDiag – the Intel® Data Center Diagnostic Tool[2] – a comprehensive processor test suite. It runs a thorough test of the entire processor and can be used by system administrators for periodic maintenance on the fleet, or to test a node that is suspected of having a defect. Typical run time is about 45 minutes, and it is most effective when the entire processing node is dedicated to testing.

Recognizing that taking a node offline for testing can be disruptive, Intel developed Intel In-Field Scan to test one processing core quickly while all other cores in the node continue to run user workloads. It is designed and optimized for live fleet testing.

2.3 Capability for Rapidly Screening Reliability Defects

In-Field Scan provides system administrators with a very fast (~200 milliseconds or less) core test that can be run on live nodes (meaning a node that is on-line and running user applications) without interrupting the entire node's operation. By running Intel In-Field Scan periodically on each core, a system administrator can gain confidence that the processor is functioning correctly.

Scan-at-Field tests the functionality of the combinatorial logic within the processing cores. 5th Gen Intel Xeon processors utilize multiple scan test images; each image has different test vectors that exercise the circuits in different ways. 5th Gen Intel Xeon processors plan to have 6 scan test images, future processors may have a different number of test images. Running all six test images on one core takes approximately 6 x 200 milliseconds (ms), or ~1.2 seconds (not including overhead for loading the test images from disk). SAF is capable of testing over 80% of the logic contained within each 5th Gen Intel Xeon processor core; it does not test the un-core, shared caches, caches within a core, or other arrays within the processor. As part of the test process, the architectural state is saved prior to running SAF, and then restored after SAF is finished. Data in the core caches is flushed, with modified data being written back to a higher-level cache or to memory.

ArrayBIST uses integrated hardware within the core to check the cores' L1 and L2 data and instruction caches, and many other arrays. It does not test the shared last level cache (L3 cache). Total test time per core is approximately 5 milliseconds. Prior to running, data in the caches is flushed to memory. Upon completion, the cache can be considered a "cold" cache.

2.4 What Can ArrayBIST and SAF Find?

SAF and ArrayBIST run tests that detect defects which cause the input or output of circuit to be stuck-at 1 or 0. These defects do not always generate a Machine Check Error (MCE), or a parity error during normal operation. For example, if the data bit is stuck-at 1 and the result of an operation should have a 1 in that bit location, then the answer is correct. On the other hand, if the stuck bit causes the result of a calculation to be 1000h rather than 0000h, the impact could be anything from a machine check to a silent data error. As an example, if the error occurs in a memory address, a machine check might be generated due to the application attempting to access an invalid memory location. In other cases, such as the incorrect result being used to determine a financial transaction, a silent data error may occur.

SAF and ArrayBIST are not capable of detecting all defects in the processor core logic or caches. Marginal defects that are sensitive to precise frequency or voltage conditions, or defects that cause a circuit to be marginal, may not be detected. Future enhancements to In-Field Scan may add capabilities for the detection of these types of defects.

If the defective circuit is in the un-core or a shared last level cache, In-Field Scan will not detect it since those areas of the processor are not tested.

While the number of cores in a processor varies across SKUs, a generalization can be made that the core logic area is approximately one half of the total 5th Gen Intel Xeon processor die area. Since In-Field Scan tests only the cores, that leaves the rest of the processor untested. Therefore, In-Field Scan should be just one of the ways in which processors are periodically tested.

In-Field Scan and DCDiag are complementary tools and together make for a comprehensive and powerful test suite.

3 How Often to Run In-Field Scan

As mentioned above, periodic testing of the fleet is recommended to find components that have failed over time. How often to run is influenced by several factors including: How long the processor has been running; the predicted Failure in Time (FIT)[3] rate of the processor; the customer's tolerance for SDE; and how much time the system administrator is willing to devote to proactive system maintenance.

In-Field Scan can be used in a variety of ways to maximize system up-time and minimize end-user disruptions. For example, In-Field Scan can be run periodically to test the entire fleet for failures that occur over time. Another option, which is common in High-Performance Computing (HPC), applies to when a specific number of processors are to be dedicated to one job for a defined length of time. In-Field Scan can be run on all the nodes before the job starts and again after the job is completed. A third option is to run In-Field Scan on a node that is behaving erratically. A quick In-Field Scan check can identify a defective processor and reduce the debug and repair time.

For the example in this paper, the goal is to detect a failure as soon as possible after it occurs. If failures are assumed to have a uniformly random distribution over time, the predicted interval between failures will be a function of the time it takes to reach 1 billion run-time hours divided by the FIT rate.

Since failures cannot be detected before they occur, the test interval should be the same interval as failures that are anticipated to occur.

4 Intel In-Field Scan Example

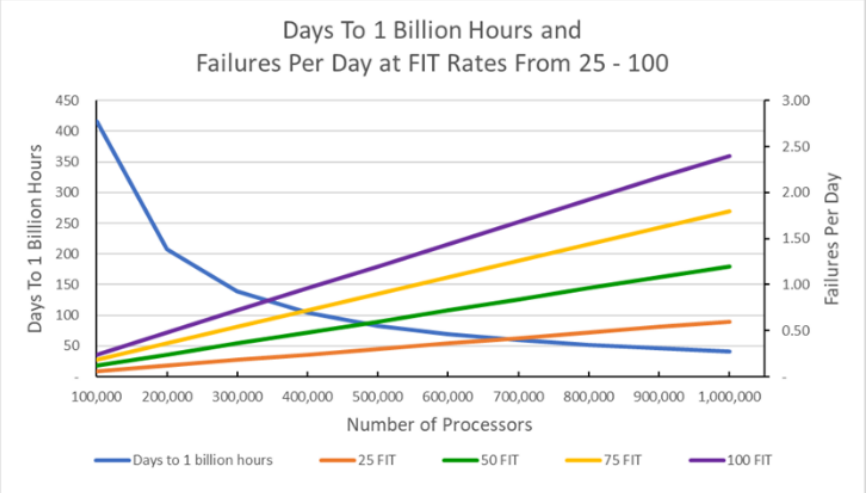
It takes approximately 11.2 seconds to run one SAF test image on each core in a 56-core device (200 ms/core x 56 cores). In addition, the overhead associated with loading the SAF test content into memory must also be considered. For this example, it is estimated to take 15 seconds to load one SAF image file and run the test on all cores. If there are six test image files, the total test time to run all images on all cores would be about 90 seconds. Adding in the overhead and recalculating the test time per test image = 90 seconds/56 cores/6 test images = 0.268 seconds/test image.

ArrayBIST takes approximately 280 ms to run on a 56-core device (5 ms/core x 56 cores). This test time is much less than the 90 seconds for SAF. For ease of use, the overall test time for 6 test images of SAF and one run of ArrayBIST will be rounded to 90 seconds.

For the example in this paper, a FIT[3] rate of 50 FIT will be used. In a fleet of 500,000 processors, it takes 2,000 hours to reach 1 billion run-time hours, which is 83.3 days, or just under 3 months. At a FIT rate of 50, there would be, on average, 50 processor failures in that 3-month timeframe or about 1 failure every 2 days (40 hours will be rounded up to 2 days).

At this example failure rate, testing all the cores on each processor once every two days will allow most failing parts that are detectable by In-Field Scan to be identified quickly, and will limit the opportunity of defects to impact the operation of the fleet. Figure 4-1 shows the number of days to reach 1 billion run-time hours for various fleet sizes and includes an estimate of the number of failures per day that might be seen at different FIT rates.

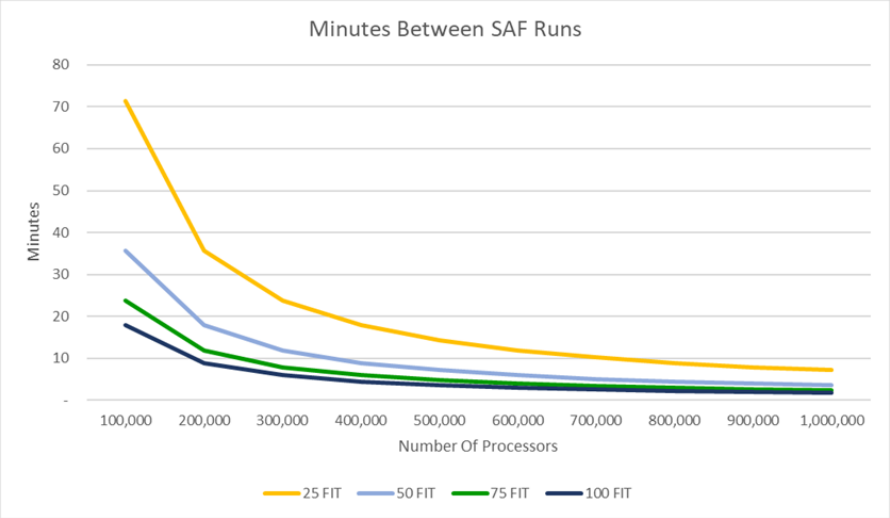
Figure 4-1. Days to 1 Billion Hours and Failures Per Day at FIT Rates from 25-100. Left axis plots the number of days to 1 billion run-time hours for fleets ranging in size from 100,000 to 1,000,000 processors. Right axis plots the estimated number of failures per day at various FIT rates for fleets ranging in size from 100,000 to 1,000,000 processors.



If the work of testing all the cores on a processor is to be distributed equally over 2 days (48 hours), it would require testing one core on each processor every 0.86 hours (48 hours/56 cores), which is every 52 minutes. With 6 test images for SAF, that would mean running one test image about every 8 minutes (52 minutes/6 images). ArrayBIST, which does not require test images, would only be run once every 52 minutes.

Figure 4-2 graphs the number of minutes between running one SAF test image on a core for various size fleets and at different FIT rates.

Figure 4-2. Minutes Between SAF Runs. Graphs the number of minutes between running SAF on one core in a 56-core processor. Data is shown for various fleet sizes and processor FIT rates.



At 268 ms/test image (including overhead), running SAF on one core every 8 minutes, or ~7 runs per hour, would take about 2 seconds every hour (268 ms X 7 runs/hour). In one hour, there are 3600 seconds, which means 2 seconds is 0.05% of one hour. Since only one core is being tested at a time, the other 55 cores are running to their full potential. Taking this into account, the hourly performance impact to the processor would be 1/56th of 0.05% = 0.001%. Detecting this very low level of performance impact would be exceptionally difficult.

The example given here applies only to a 500,000-node system with processors having 56-cores and a FIT rate of 50. A higher FIT rate will increase the frequency of failures and how often In-Field Scan should be run, while a lower FIT rate would reduce the frequency of how often to test.

Running In-Field Scan every 8 minutes is one way to implement the periodic testing, but there are multiple ways to achieve the same goal. SAF can opportunistically run during idle periods on the node, or multiple cores could be tested in bursts of SAF activity. Each system administrator can choose the implementation that best fits their requirements[4].

5 Conclusion

In a fleet with hundreds of thousands, or millions, of processors, failures typically occur on a regular basis. Finding these defects as quickly as possible is key to minimizing the interruptions to customer operations.

Intel is leading the industry by providing multiple tools to test processors for correct operation. In-Field Scan is a new family of tools that make this job easier for the system administrator. Starting with 5th Gen Intel Xeon processors, In-Field Scan tools have the following characteristics:

- Very fast, less than 200 microseconds per core.
- Runs on a single core without impacting other cores.
- Has high test coverage of the cores and core caches.
- Can be run in a live fleet environment to identify defects that have occurred over time.

The example given above should be considered a starting point for understanding how often to run In-Field Scan and its ability to detect various failures. Used in conjunction with Intel DCDiag, a system administrator has a variety of ways to measure and ensure the proper operation of the fleet.

6 References

[1] Scan is a design for test capability that is a well-known industry technique for testing silicon. Details of how scan works can be found here:

http://ece-research.unm.edu/jimp/vlsi_test/slides/html/scan1.html

[2] Intel® Data Center Diagnostic Tool for Intel® Xeon® Processors: https://www.intel.com/content/www/us/en/support/articles/000058107/processors/intel-xeon-processors.html

[3] Failures in Time (FIT) is the predicted average number of failures that will occur in 1 billion hours of run-time. For example: If 1 million processors are running 24 hours/day, and the predicted FIT rate is 10, then it is estimated that 10 devices, on average, will fail every 1000 hours (1000 hours X 1,000,000 devices = 1 billion run-time hours). Or 1 device could fail every 100 hours on average, which is once every 4 to 5 days. FIT rates apply to the statistics of a large population of parts, and do not indicate that a specific part will fail in a particular timeframe.

[4] The most efficient way of running SAF is to load scan test image #1 and then run it on all available cores. Once all 56 cores have been tested with scan test image #1, scan test image #2 should be loaded into memory and used to test all the cores. That would then be followed by scan test image #3, #4, etc.