# How to Debug Dead Boards in Production

## Based on a Case Study

by Larry Osborn

ScanWorks®
Platform for Embedded Instruments

*By Larry Osborn*

Larry Osborn, Product Manager Processor-Controlled Test, at ASSET InterTech, has over 25 years of experience in product management, hardware/software product design and development, product delivery to the marketplace and user support. Over the years, Larry has a proven track record for identifying user needs and opportunities in the marketplace, providing innovative solutions and exceeding the expectations of users. At ASSET, Larry is responsible for the profit and loss for a product group. Prior to ASSET, he has held positions with Lockheed Martin, OCD Systems, Windriver, Hewlett-Packard, Ford Aerospace and Intel® Corporation. He holds a Bachelor's Degree in Computer Science from the University of Kansas and various technical and marketing training certifications.

## Table of Contents



## Table of Figures/Tables

## Executive Summary

Dead circuit boards or board that won't boot as they come off the production line is a fast growing problem today. Why? First, testing a non-bootable board is nearly impossible or, at the very least, impractical from a cost perspective. Second, the time spent to manually debug dead boards is limited by extremely tight production deadlines that are dictated by a demanding marketplace. Third, perhaps the most critical aspect of this situation is the causes of faults too often remain undetermined instead of providing a feedback loop so that production that could eliminate faults that consistently appear. Without such a production control feedback loop, the causes of problems will continue and the pile of dead boards – the bone pile – will continue to grow.

A recent case study clearly shows that this situation can be corrected and dramatic results achieved when a new methodology consisting of categorizing or 'binning' dead production boards by fault types and testing these boards from the processor outward – inside out – is applied. This is often referred to as employing embedded instrumentation. When this approach is adopted, the weeks previously required to diagnose dead boards can be condensed to days. And, most important of all, information on the categories of faults being found and the causes of faults diagnosed can be fed back as a real time feedback control loop to affect changes in ongoing fabrication processes and improve board yields.

This eBook is based on case study data provided by a high-volume manufacturer of circuit boards. The data demonstrates how characterizing and detecting faults quickly improves production yields and dramatically reduces those bone piles of dead unrecoverable boards.

## Circuit Board Cost Goes Up. Debug Time Goes Down.

Today's Intel-based circuit board designs are highly complex. Depending on the target market, the resulting boards can be very expensive. For example, boards for the high-end telecom market can range from $10,000 to $20,000 each. Operating margins on these boards are being pushed to the limit by increasing operating speeds. With tighter margins, small problems can result in unsolvable dead boards, or boards that won't boot when they come off the production line. Debug these boards from production fall-out is practically impossible, since the functional test systems that are typically used for fault analysis and troubleshooting *require that the boards start up* (boot). When they don't, functional tools are practically useless. To paraphrase a common dilemma: **dead boards can't talk**.

Meanwhile, shorter debug times and faster feedback are imperatives because the production department needs information on the underlying causes of faults, such as issues relating to components like memory or specific vendors. The crux of the problem revolves around this contradiction: the time and labor spent on debugging dead boards is decreasing while, at the same time, manufacturers don't want to scrap costly dead boards. They would much rather recover as much as possible the huge costs they've already sunk into each production board.

Drilling down, here are some of the issues you may have to deal with when you're developing and producing fast and flexible Intel-based circuit boards:

- How to maintain high production yields as test probe access diminishes
- How to deal with dead boards that don't boot
- How to bin or categorize the problems encountered in dead boards and, at the same time, create a fast information feedback loop to correct problems on the fly
- How to debug and recover dead boards quickly and cost effectively

These are just some of the challenges facing today's production manager who is responsible for the manufacturing yields well as the test manager who is responsible for test strategies and procedures.

## Inside Out Testing. Not Outside In.

In the test engineering tools market, an evolutionary move has taken place away from big external hardware testers and toward soft access methods to onboard DFT structures, commonly referred to as embedded instrumentation (Figure 1). In most cases, these embedded instrumentation test processes are coupled with functional test platforms based on the product itself. Embedded instrumentation is embedded in the silicon or placed on the board itself.

**Figure 1: Test structures and functionality have moved from external testers to on-chip and onboard resources that are available to test engineers through soft access embedded instrumentation methods.**

In either case, any test solution today clearly needs to take advantage of an **inside out approach for debug and test, as opposed to antiquated outside in methods**. Employing onboard and on-chip DFT structures to find and eliminate board problems in production is a first step. Taking control of the onboard processor (by means of the so called JTAG emulation process) to test and debug internal board structures *functionally and at-speed* is a second step. Reviving dead boards to a bootable state by using a functional test platform is the third and final step.

Combined, these techniques have the ability to define a test process that is fast, accurate and able to keep pace with production. Of course, all problems on circuit boards will never be solved, but a fast fault feedback loop can remove the bulk of the problems, causing yields to increase. It has

really never been about testing everything (maybe at one time it was), but rather, it's about *testing enough* while building an understanding of the source of the problems. Now, the goal is to bin or categorize problem boards, determine the root causes of the problems and take corrective actions in minutes or hours, not days and weeks.

*What is needed in this process is the ability to quickly sort dead boards into bins of shared problems. That is, quickly determine the type of problem and sort boards into groups of problem types, followed by sharing the information learned with production. Eventually, faults on each board must be isolated so it can be repaired, if possible.*

## Some Background on Device Binning

Semiconductor device test starts with a wafer probe that applies parametric tests to determine whether the device is operational at a basic level while devices are still part of a wafer. Based on this information die are extracted from the wafer for packaging as devices. Once a device is packaged and sent to a device handler, basic parametric tests are applied again to verify it is still operational. If the device is operational, it is tested again with a different and more stringent set of parametrics to test whether it performs as specified.

These parametric tests become increasingly more stringent each time they are applied to evaluate a device and eventually bin or group devices according to their parametric test results. This binning process allows for further testing which will establish the working parameters such as access times on each device. The objective is to spend as little time as possible on devices that will end up as scrap and more time classifying the value of operational devices. For example, the shorter the access time on a device, the more valuable that device will be. This process is sometimes also referred to as grading; components are graded according to performance. Each grade can be conceived of as a bin. So binning or grading are performed to select the cream of the crop – the best of the best – that can be sold at a higher price For example, a 2.7GHz microprocessor (uP) could be sold at a higher price than  a 2.5GHz uP from the same batch of devices. Similarly to devices, boards can also be quickly binned by failure types.

## Binning Boards by Failure Types

A similar approach can be applied to testing boards as they come off a production line or that have ended up on the bone pile or dead boards. Depending upon the value of an assembled board, the goal of the test methodology is usually to *maximize test coverage but minimize test time*. However, this type of test strategy could end up sending boards to the bone pile without any real knowledge of the potential sources of problem on the boards. The objective of binning dead boards is to narrow the focus of the repair technician on one of a few likely causes of the problem so repair can be completed quickly and cost effectively. The binning or classifying methodology can be borrowed from device and applied to boards during production. If the test is too comprehensive in the beginning, you can spend too much time testing a board, only to find that the problem was relatively easy to diagnose, such as a faulty power rail. An early power test could have stopped further testing and caused the board to be placed in a certain bin representing power failures. Knowing this, the repair group will only spend time examining power devices.

The dominant focus in production test development today is often on maximum coverage and test completeness without regard to achieving diagnostic binning results. If we flip it around, and take a binning approach in our test development we would design tests that covered smaller functional areas early in the test process so that boards could be binned as soon as a failure manifests itself. This would provide the repair team a target to focus their efforts upon. By grouping the causes of board failures into larger patterns, we get the coverage we desire *and* the diagnostic granularity we need to speed boards through repair, thus recouping lost revenue sooner.

Again, borrowing from device binning, we can envision a scenario where a board that has been placed in Bin No. 2, for example, has been classified with a relatively simple problem. As a result, it can be recovered quickly, but boards in Bin No. 10 might have more catastrophic problems and be considered for component recovery. How does this affect the return on investment (ROI) for testing? The earlier we find a problem in test and classify or bin the board, the more we save in test time. The quicker we can determine the (general) cause of the problem, the faster we can make changes in production so that additional faulty boards are not produced in the next batch. Moreover, we can quickly determine *if* a particular board is worth repairing. Sometimes, that's a big IF.

The last place we want to find a board problem is at system functional test, where our goal should be to find system integration issues, not board failures. All too often this is not the case.

## How to Test Dead Boards

The causes of circuit boards not completing the startup/boot process typically fall into three categories:

A) Corrupt BIOS
B) A faulty data/address bus from the CPU
C) Memory issues

Let's examine each of these possible faults and provide some insight as to how they might initially present themselves in the traditional functional test approach (that is, an outside in test approach). Then we'll look at how to use the binning technique and embedded instrumentation or Dfx structures to prevent the board from ending up in the bone pile. All of this should be done while reducing test times to a minimum. Of course, test engineers must also determine the optimum test strategy. Central to defining an optimum test strategy will be the number of board-level parametrics tested relative to the eventual value or sales price of the board. For example, the test strategy for a board with a relatively low sales price might not include an extensive number of parametrics while the test strategy for a more expensive board would probably include more parametrics. When devising a test strategy, the cost of test (or the cost of preventing a dead board through extensive parametric testing) must be balanced against the eventual sales price of the board. The cost of additional testing is more difficult to justify financially when the board's sales value is low. A very expensive board can more readily justify extensive parametric testing.

This approach could also classify the boards currently in the bone pile. The reason for addressing this in production is to focus on the ROI of board repair. The earlier a fault is found and classified, the better the rate of return will be on the organization's board repair processes.

### Corrupt BISO

A circuit board will not boot when its BIOS is corrupt. Unfortunately, a functional test or system test environment will give no indication how or why the BIOS became corrupted. The test

operator is left to speculate as to what has happened. Was it the power? The CPU? The chipset, or is the tester itself to blame? The only outcome of this process is to classify the board as dead and place it in the bone pile with a dead board tag reading "Won't Boot". Using a binning methodology and Dfx or embedded instrumentation structures within the CPU, one can access the BIOS or firmware and perform fundamental checking.

1. The first check should be to read the version number of the BIOS at a specific location and compare this against an expected value. If the board passes, additional parametric criteria should be tested.
2. Next, a CRC on the contents of BIOS should be conducted and compared with the expected results.

Both of these tests are simple and can be completed quickly. Based on the results of each test, additional parametric value can be tested to help with the fault classification when a fault is found.

## Data/Address Bus Issues

Various types of faults occur on address and data buses. Common faults are opens and shorts. An open could cause the board to not boot, but this will depend on the address or data line where the open is and how the line is used during the boot process. Depending on the board's design for testability (DFT) features, most manufacturing tests will capture opens. Shorts can also be captured with the same tests. If opens and short testing is omitted, then the board might not boot if the fault is in a certain location. As a result, the board would be classified as a dead board. Faults on data/address buses could prevent a board from booting because in the BIOS initialization process the cache provides RAM storage for critical data structures. After the RAM has been initialized, these data structures and other code are transferred to memory. If a fault interrupts these transfers, the board will not boot and classified as a dead board without any indication as to why.

Another place where address or data bus faults can cause problems and prevent the board from booting is at-speed execution of code. This can be particularly critical with today's high-speed serial buses. These types of faults are discussed in the following section.

In summary, by applying a binning methodology, tests should:

1. Detect shorts and opens
2. Test for at-speed access

## Memory Issues

Some of the issues described above also apply to memory. The interfaces to DDR3/4 memories are high-speed, fault-tolerant serial buses. These buses require initialization that is done by code stored within the BIOS. Memory issues fall into a fault spectrum. The spectrum includes opens/shorts, training or initialization faults, margining or lane faults and a larger category of at-speed faults. As these buses are fault tolerant and because many fault types can be present, no single test methodology can detect all possible faults.

Other white papers, blogs, eBooks and other materials deal with the topic of memory test in detail. Here we will focus on *binning as an outcome of functional tests*. Obviously, if memory is not working properly, a board is usually classified as dead. The aspects of memory that must be tested are address, data, control lines and parity or ECC control. Single-bit parity or ECC faults generally won't cause board to not boot, but these aspects of the board should still be tested because double-bit faults could cause a reset that may appear as a board with a booting problem.

A test procedure for these faults which would follow the binning methodology by adding more parametrics to further classify the problems could consist of the following:

1. Testing memory for lane or rank failures after the initialization code has executed
2. Testing memory for address and data faults
3. Testing for control signal faults
4. Testing for ECC

There are other buses that need to be tested but most of these will not prevent a board to boot. (The exceptions to this are the DMI and QPI buses. Although, how to test these buses is an important discussion, it is beyond the scope of this eBook.)

## Dead Board Case Study

In a recent case study of Sandy Bridge boards that had been classified as dead boards, nearly 50% had memory problem (Figure2). A binning approach classified memory faults into three areas:

1. Data Bit (DQ) or Strobe (DQS)
2. ECC related faults
3. Other

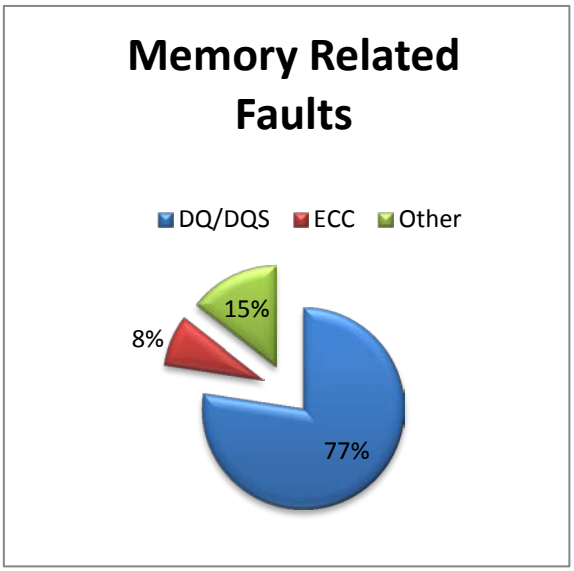



**Figure 2: Memory Related Faults**

DQ and DQS are control signals between the CPU and DDR2/3 memories. If faults are present on these signals, memory access is not possible and the board appears dead. Because these are high speed signals, probing the signals will interfere with the at-speed performance of the bus. In addition, probing is a violation of Intel design guidelines.

Other approaches to testing are necessary to provide fault coverage and diagnostic resolution to aid in the recovery of a board with this type of fault. The ECC errors are single bit and double bit. Double bit errors would cause a dead board because that area of memory could not be reliably accessed. So the testing strategy needs to encompass testing for these types of errors. In the case study a unique and powerful functional test method was applied to classify faults.

By relying on this type of binning or classification, the board recovery team could focus on a smaller number of components and signals. If this type of binning procedure had been in place earlier in the production test stage, the production plant manager would have been alerted to this type of memory problem and corrective action could have recovered the lost product sooner. In addition, a second test setup would not have been necessary.

We also found from the case study a relatively large group of problems that were classified as other faults (Figure 3). By focusing on this data, the test was enhanced so that faults could be binned into fault types that are better understood and could be more easily addressed.

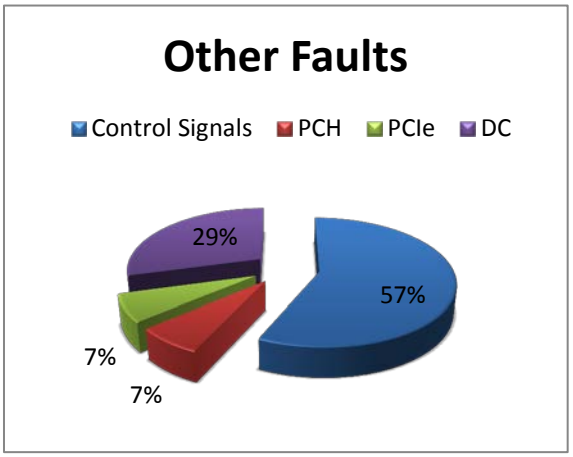



**Figure 3: Other Faults**

## Control Signals

Generally speaking, the control signals that can have an impact on the board's ability to boot are usually found at the CPU. The state of the processor determines what the control signals state should be. These faults can be identified by a pins check and comparing the value found to the expected value. In this case study 57% of the faults classified as other could be binned as control signal faults.

## Processor to Controller Hub (PCH)

Specifically, shorts or opens on the DMI bus can cause a dead board. In the case study, a test was developed to check that the expected devices available to the processor are connected and functioning correctly. Only 7% of these types of faults were found.

## PCIe

Faults on the PCI Express (PCIe) bus could also cause a dead board. PCIe can run at various speeds and use multiple links to increase the bandwidth of the communication to a component. The absence of an operation link can be detected and reported as a fault.

Other faults might appear on PCIe buses. This discussion assumes familiarity with the Unified Extensible Firmware Interface (UEFI) development. In a UEFI firmware implementation during the DXE phase when PCIe devices are being enumerated, control can be passed to a PCIe card that is responsible for completing the boot process. Simply testing the PCIe path may not be sufficient to find a problem on a subtending card or device. Several possible faults that could prevent a board from booting. The problems could be PCIe link down, PCIe link width diminished, RAM problems on the PCIe card or interrupt problems on the card.

Although in the case study only 7% of the faults were PCIe related, the effort to classify these faults is more involved than most buses because of the nature of the bus.

## ROI Case Study

Adoption of this binning test strategy impacts more than just scrap recovery. To determine total return on investment, the entire test spectrum must be analyzed. There are four areas that are impacted by implementing this binning or classification strategy. They are:

1. Reducing scrap costs
2. Reducing debug time
3. Lowering board and system functional test costs
4. Reducing in-circuit test (ICT) costs

These concepts were demonstrated in a different case study where data was gathered from a manufacturer of server boards. Table 1 shows some of the findings of this second case study:

**Table 1: ROI Components**

| Elements of Test | Number, Cost or Yield Data |
|---|---|
| Server Boards | 20,000 per year |
| Board Cost | $750 |
| ICT Yield | 93% or 1,400 failures per year |
| Board functional test (BFT) Yield | 87% or 2,418 failures per year |
| Boards unrecoverable from BFT failure (dead boards added to the bone pile) | 242 (10% of BFT failures) |
| ICT Fixture Cost | $30,000 per design |
| ICT Test Development Cost | $30,000 per design |
| BFT Station Requirements | Four stations at $29,000 per station |
| Debug Cost and Time | $50 per hour and 5 hours average debug time |

Although all areas were examined in this case study, for the purposes of this paper we will only focus on scrap reduction and reducing debug time.

The board functional test (BFT) failed 2,418 boards per year. A minimum of 10% of these boards were classified as dead boards. Thus, 242 boards ended up in the bone pile. With the testing describe throughout this paper 50% of the bone pile was recovered, resulting in the recovery of $90,750 in costs (242 X 50% X$750 = $90,750).

Previous to the case study data being gathered, the average board debug time on the bone pile was five hours per board. By applying the test strategy described in this paper, the debug time could be reduced by an average of 50%. The cost savings from reduced debug time would then be $30,250 (242 X 5 X $50 X 50% = $30,250).To summarize, the annual cost recovery and cost savings as a result of this test strategy could be $121,000. (See below.)

Total Annual Cost Saving of

$121,000 ($90,750 + 30,250)

This is not intended as a complete ROI calculation. It does not include a payback period or projected annual rates of return on capital expenditures and new labor costs. Despite these limitations, it does demonstrate that the annual cost savings and revenue recovery figures can be significant. Since the new test processes described in this paper do not involve large capital expenditures or significant new labor costs, the expected return on investment should be rather high and the payback period short. Of course, these figures are all based upon the assumptions described above. Each individual case will have its own set of costs figures, yield rates and other characteristics.

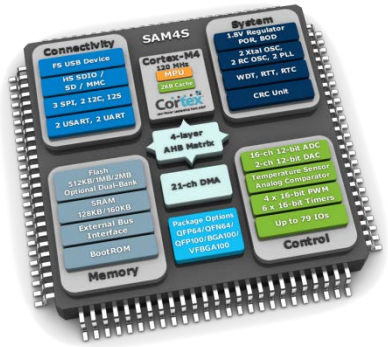
## Dead Boards Really Can Talk…

As we have seen, the growing problem of piles of dead boards can be dealt with by using an inside out test approach based on embedded instrumentation. Combined with effectively binning dead boards by fault categories, big savings in time and money can be achieved. But perhaps the most valuable benefit of this process is faster feedback of statistics on failures to production, enabling an efficient feedback loop that can prevent more failures of the same type from occurring. Shortening the failure feedback loop from weeks to days if not hours is a tremendous step in the right direction toward maintaining higher manufacturing yields. For example, if there is a problem with a certain memory device or type, the production manager wants to know about it ASAP. If the flash is not programmed correctly, manufacturing management wants to know about it ASAP. But just knowing about it is not enough. The responsible engineer also wants good diagnostic information; perhaps even the results of a functional test combined with a halt capability so that he or she can read registers to find out what is going on.

To rely solely on functional test on the expensive and complex boards and high-speed buses of today is no longer an effective option. But mixing functional test with onboard DFx structures and an embedded instrumentation approach so as to quickly bin by failure types and perform inside out testing from the processor perspective is very cost- and time-effective. And, in addition, there's the added benefit of creating a feedback loop of information for production personnel to keep yields and quality high.

## Learn More

*Learn about diagnosing and debugging prototype circuit boards using the processor.*




**Learn More!**

SCANWORKS®
Platform for Embedded Instruments