You wouldn’t buy a car, check the oil – then never check the oil again. Similarly, a mission-critical subsystem with the highest possible MTBF should be complemented by a strategy that verifies reliable operation in deployment. No matter how high the MTBF figure, it is always a finite, statistical quantity. A failure may still occur at any particular moment.

In the embedded market, system verification is typically implemented by test firmware that exercises the hardware during power-up and operation, and reports any anomalies to the application for the appropriate corrective action. In this way, anomalies are prevented from hiding until they occur during a moment of critical system use, when they might threaten the mission. And to continue the analogy: in the same way that not all cars are created equal, not all deployed test firmwares are created equal.

But first, a little history...

**MIL-HDBK-217**

MIL-HDBK-217 (Reliability Prediction of Electronic Equipment) was created by the Department of Defense to establish and maintain consistent and uniform methods for estimating the inherent reliability of electronic equipment and systems, and has been widely used outside of the military environment.

The 217F handbook was published in December 1991, when the very large majority of solutions were proprietary. The developer of a system therefore had complete insight into the detail of all hardware and software elements, including Built-In Test (BIT), and could craft them to coordinate. Beyond that, the hardware would typically be application-specific and not generic.

**COTS challenges**

The advent of Commercial Off-The-Shelf (COTS) solutions has in many ways rendered MIL-HDBK-217 inapplicable. Hardware – such as a typical Single Board Computer (SBC) – is now generic, as is the silicon that populates these boards. Detailed information on the internal logic of such parts is mostly unavailable.

It is also true that a COTS operating system is effectively a black box where the internal device and resource usage is unknown. As such, it cannot be disturbed by the intrusive testing that is typical of BIT. Even if operating system internals were known, they would only be known at a single point in time. While COTS operating system vendors are committed to maintaining the compatibility of the Application Programming Interface (API), they are not committed to maintaining internal structures or resource usage.

An intrusive test of any individual hardware component will almost certainly cause a complete malfunction of the operating system, which relies on the total machine state of the SBC. While this is generally not an issue for a BIT that executes before the operating system runs, it has significant implications for continuous hardware confidence testing after the point at which the operating system and application start their execution.

**Our test firmware approach**

The COTS market requires a different approach to the design, implementation, and operational verification of test firmware. Radstone’s methodology is holistic in that it encompasses not only a hardware confidence test at power-up that is provided via BIT, but also an ongoing hardware confidence test through Background Condition Screening (BCS).

Additionally, Radstone provides comprehensive facilities in these standard products that support system acceptance testing and extra, customer-designed confidence testing.

This article reviews some of the key characteristics of Radstone’s approach to BIT, but it is important to note the vital role that BCS also plays in assuring confidence.
BIT relies on unfettered and exclusive access to system resources, allowing intrusive tests to achieve the highest possible confidence. BCS runs as a background operating system task, and is designed to be non-intrusive and non-destructive. This allows confidence checks to be maintained as the background task does not disturb the operating system.

Alternative approaches to continuous testing (such as allowing callbacks to destructive BIT tests) are questionable, even if an attempt is made to save and restore the key machine state variables around the callback. COTS operating system internals are unknown, and the functioning of the operating system may be disturbed. BCS is designed to work with the operating system, and it utilizes specific hardware partitioning for test purposes. Run in conjunction, BIT and BCS provide the maximum possible ongoing guarantee of hardware integrity.

Meaningful BIT coverage
If deployed test firmware is to preserve the mission from hidden failures, there must be very high confidence in their effectiveness. The magic word in the BIT world is coverage, a term that is meant to indicate the percentage of possible board failures that are detectable (or covered) by the test. Coverage can be estimated in a number of ways, and some of the estimations are inappropriate for a COTS environment. The key is that coverage is only meaningful if it is within the context of a valid, known, and appropriate measurement system.

For example, it was once common practice to equate coverage with the number of gates exercised by the test. This made sense when low-density TTL chips prevailed – but makes little sense today where the vast majority of gates may be used for just one or two functions (such as memory or the processor). Similarly, with today’s high density components, a simplistic exercising of the component will not generally reveal whether all of its gates are functional.

This method of establishing coverage evolved to a circuit analysis method, which recognized the growing importance of the printed circuit board itself, and the fact that the significant memory subsystem had its own built-in hardware strategies to cope with failure – Error Checking and Correction (ECC). Although not without its limitations, circuit analysis was well suited to the designs of yesterday and is still a valuable tool today when determining the efficiency of BIT code.

Another past technique used standard reliability analysis figures to help determine test firmware coverage. This also has its place and drawbacks. For example, the failure rate predictions do not always align well with actual field failures.

Heavier coverage weighting is given to BIT exercised chips with higher estimated failure rates, generally over-emphasising those with a high gate count. This can lead to skewed coverage estimates. Some specific part use is not adequately taken into consideration. For example, edge connectors in the field may have a very significant failure rate, especially as they are exposed to inadvertent abuse, but the failure rate figures in use may not account for this.

There are numerous sources of reliability prediction data in the COTS silicon world, but they are inconsistent and tend to produce results more reliant on the chosen data than on real world performance. A further limitation of standard reliability analysis is that it inadequately addresses the impact of printed circuit board and solder joint reliability.

Our BIT coverage approach
As discussed, COTS hardware and software has challenged the traditional measurement methods. Radstone provides reliability analysis figures (generally to MIL-HDBK-217) for our products, and this is useful data when considering the overall suitability of a particular board for a mission-critical application. However, our fault insertion/stuck nodes approach goes much further to prove BIT coverage.

The advantages of fault insertion are that the evidence it delivers is operationally real and entirely objective: a known fault exists – BIT detects the fault. This method has been used historically to verify BIT operation in the industry, but generally only as an incidental audit, and with only a very small number of inserted faults.

Number of nodes
The challenge is to make the process statistically valid by testing a large and representative sample of faults (circuit board nodes are stuck High or Low to create faults).

A representative sample is achieved through the random choice of a large number of nodes. The question of how many nodes is difficult. The question is resolved with a standard formula which reveals the correct sample size for an appropriate level of confidence (Equation 1).

where:
- \( n \) = necessary sample size
- \( N \) = total node population
- \( d \) = tolerance percentage
- \( P \) = coverage level percentage
- \( t \) = confidence level percentage

The confidence level percentage is directly related via the area under a normal distribution curve, to the confidence that the statistical result can be applied to the whole population.

An example
For this example, the board has a relatively low population of 5,000 nodes. The example values are:

- \( N = 5000 \)
- \( d = 2.75\% \)
- \( P = 95\% \)
- \( t = 90\% \)

The result of this calculation is that 164 random nodes need to be tested (n=164). Even this relatively manageable number of test nodes can pose a significant test challenge. Consider 164 nodes spread randomly across a high-density board with several large Ball Grid Array (BGA) devices.

Failure equivalence
A stuck nodes analysis does not directly address some fault types – for example, a faulty analog component or certain internal silicon failures. However, many failures of this type are covered by equivalence. For example, an SBC failure at a node that is not directly observed causes an observed node to be stuck elsewhere on the SBC. This results in the detection of an equivalent failure, and coverage of the fault in the analysis. The fact that these failures are not always directly detected does not affect the overall confidence factor in the result that describes the efficiency of the BIT.

Where equivalence does not exist, Radstone BIT includes code that is specially targeted to provide additional coverage for such failure modes.
Radstone’s BIT, therefore, has a statistically valid foundation and has been developed to deliver maximum customer confidence in the reliable operation of the board it tests, and Radstone’s testing does not even stop with the board.

**Beyond the board**

Radstone’s BIT philosophy incorporates the optional ability to test edge nodes (SBC nodes that lead off-board). This feature recognizes that BIT’s true value is in real world deployed systems where an SBC may be connected to:

- USB
- serial lines
- SCSI hard drives
- Ethernet LAN
- external VME or PCI bus

This presents some difficulties, in that typical edge nodes do not have the predictable loop-back path of internal nodes. Nevertheless, although the connection path in which edge nodes participate is system dependent, it is normally still possible to verify many of them via the appropriate configuration of a range of standard Radstone tests.

For example, Radstone’s BIT can access pre-configured areas of VME or PCI address space, and test the data capability at each address with various cycle types.

This is possible on the majority of deployed systems including those with special test partitions. Other interfaces provide different challenges, but standard responses can be elicited from a few interfaces (for instance Ethernet ping and SCSI ID replies).

Edge node coverage is also increased by tests that transmit data and expect it to be echoed back in the same or an altered form. Use of the altered form requires a minimal integration effort. If some of these tests are inappropriate to certain connections during deployment, they need not be enabled. However, within the more flexible environment of an acceptance tester, the full range of options can usually be applied, obviating the need to develop separate acceptance test software.

**Conclusion**

In summary, Radstone’s approach to deployed test is an extremely comprehensive one, providing maximum assurance both at start-up (via BIT), and with application execution (via BCS). It is designed to be flexible and adaptable to specific customer or application needs.

Test coverage can be configured for the optimal trade-off with execution time, and can even be by-passed altogether for emergency bring-up situations. Moreover, proof of test coverage is geared to the realities of COTS and the real world, where more failures may be caused by damage to edge connectors than are caused by processor failures. Similarly, more faults can be induced by handling or upgrading than by component malfunction. Traditional failure mode algorithms do not take this into account, yet a fault on a single connector pin can bring down an entire system – unless BIT has the capability to find it first.

Radstone deployed test firmware provides the maximum assurance of reliable operation in mission critical environments, allowing customers to take full advantage of the unique cost-benefits that COTS hardware and software components offer, while maintaining the system reliability and integrity of the total solution.

For more information, contact Andy West at:

Reprinted from [Embedded Computing Design](#) / Winter 2004