



Certify Faster with Simulation

WHEN IT MATTERS, IT RUNS ON WIND RIVER

EXECUTIVE SUMMARY

Let’s agree: software safety certification is a good thing. It helps prevent planes from falling out of the sky, trains from crashing into each other, and industrial robots from running amok. That said, it is an arduous, time-consuming, and costly process for software developers and builders of components. The price of failure is high. Subcontractors who can’t get certified run the risk of contract cancellation and non-payment. And there is no partial credit.

Developers can mitigate that risk by focusing on the preparation process for certification testing. Problems can be found up to 97% faster, which means they can be fixed sooner and at a lower cost through testing during development rather than waiting for the formal test run. Using Wind River® Simics® for simulation, developers of DO-178C-compliant and other safety critical software can streamline the process of getting certification-ready—saving time, reducing costs, and greatly increasing the likelihood of testing successfully on the first pass.

This document explains how Simics delivers development efficiency to help you create higher quality code and get it certified more quickly.

TABLE OF CONTENTS

Executive Summary	2
The Virtues of Virtual Platforms	3
Using Simics in Pre-certification	3
Simulating and Testing Complete Systems	4
Cleared for Takeoff: A Real-Life Example	4
Conclusion	4

THE VIRTUES OF VIRTUAL PLATFORMS

Simics provides a virtual platform for simulating target hardware. It can be used to simulate anything from a single processor to large, complex, and connected electronic systems. You can, in fact, replicate the entire system in which your software will eventually operate.

Simulation is widely used in the development of embedded software, and allows engineers to test designs more quickly, at less cost, and more reliably than using physical hardware alone. Simics provides functional simulation defined by the hardware design, not by the implementation. Hardware models can be built as soon as the design is complete and available, in advance of physical hardware prototypes or final systems, so you can start software development earlier without waiting for the actual hardware. For example, if different contractors are developing software and hardware, the software contractor can keep the project on track and demonstrate the software without a critical dependency on the hardware developer.

Simulating the functional behavior of target hardware allows engineering, integration, and test teams to work simultaneously and collaboratively, reducing development time and improving the quality of safety critical systems. You can't put a hardware prototype on every engineer's desk, but you can put a virtual model on everyone's screen. Using Simics virtual platforms allows engineers to work on their own schedules and makes it easy for them to track and share what they are working on. This flexibility and ease of collaboration encourages productivity, which in turn results in better code.

Moreover, virtual platforms provide software debugging and analysis features that are simply not possible to implement on physical hardware. Unlike a traditional test run on hardware, in which you drain time away rerunning the software and hoping to spot the bug, Simics enables you to run time in reverse to locate the precise breakpoint. It's easy to see that a problem has occurred after it occurred, but what if you could figure out why it occurred before it occurs? Simics' reverse execution enables you to do just that—in effect, to go back in time to find the source of a bug and fix it on the spot.

Developing on Simics has been proven to enhance developer team collaboration, reduce development time and costs, and accelerate time-to-market while ensuring the quality and performance of the end product. Development can begin earlier, more defects are found and removed during development, and quality is achieved sooner (see Figure 1).

DRIVE QUALITY SOONER

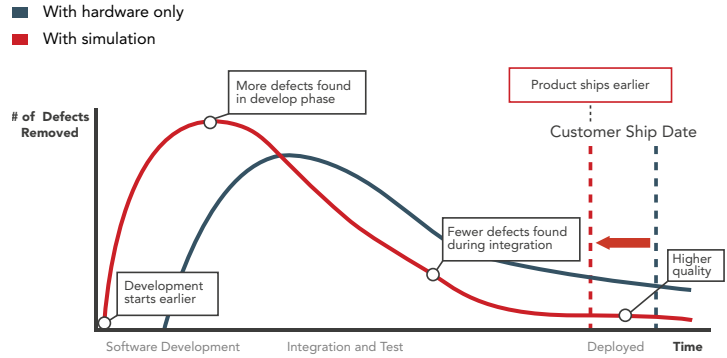


Figure 1. With a virtual platform, more defects are discovered and removed earlier in the cycle; development begins earlier and the software achieves the quality required for deployment sooner

USING SIMICS IN PRE-CERTIFICATION

System developers and component manufacturers tend to focus on certification testing costs, and often overlook the costs of preparation. Yet pre-certification costs can be enormous, and must be factored into the total cost of obtaining certification. Use of physical hardware for development and pre-certification testing impedes your agility, slows the development cycle, and drives up costs. Simply put, problems take longer to find and longer to fix.

Simics makes it possible to perform virtually non-stop testing throughout the development process. You can reduce the time required to run a complete regression test from weeks to overnight. That means you can do more frequent regression testing and catch regressions earlier, reducing the number of regressions introduced to the system and the cost to fix them.

Simics is a more reliable target platform for running batch tests than physical hardware platforms. It also removes the host-to-target communication bottleneck and reliability issues. Tests can fail because of communication glitches between target hardware and the host, in particular when moving large amounts of data. With Simics, this friction is removed and tests can run more reliably.

Finally, finding bugs during certification testing and having to repair and retest is extremely costly and can delay delivery. Simics' superior debugging abilities make it a very good tool for running and debugging the tests that will eventually be run for credit on physical hardware using qualified tools.

SIMULATING AND TESTING COMPLETE SYSTEMS

Because Simics can simulate a complete system, it is an ideal platform for developing requirements-based system tests, which normally requires a lab with physical hardware and supporting infrastructure. Testing can be performed even before the entire system is fully functional. Problems are detected and fixed earlier in the development cycle, helping assure quality and performance at the time of actual certification testing.

Fault injection with actual hardware is difficult, risky, and expensive. Digital fault injection with Simics makes it easier to test for robustness and fault tolerance. There is no need to destroy hardware just to see if the software can detect a stuck pin or some other abnormal condition, and tests can be repeated and scripted.

Requirements-based testing at the system level identifies the most critical problems. However, it often takes place late in the process due to the limited availability of target hardware, and ends up getting rushed. Simics overcomes hardware constraints, allowing more timely and thorough testing, which in turn results in a higher quality system.

Consider how much it would cost and how long it would take to develop a lab of a complete system for testing and integration. Then imagine that you could have a simulation of that complete system on the desktops of everyone working on the project, months if not years in advance. The value of Simics becomes readily apparent.

CLEARED FOR TAKEOFF: A REAL-LIFE EXAMPLE

For an electronics subcontractor on a major new aircraft project, full DO-178C testing was taking three weeks, primarily due to the limited availability of flight hardware test systems. Moreover, the test runs usually found defects, meaning no test credit and time lost doing the repairs and retesting.

You can't put a hardware prototype on every engineer's desk, but you can put a virtual model on everyone's screen.

Simics made it possible to perform a test run overnight. The developer could not get credit for a test run on a virtual platform—but the average time from code commit to defect detection dropped dramatically from four and half weeks to 24 hours. That meant enormous cost savings and helped ensure that the product would not fail the actual certification test again.

The happy ending: The component was delivered on time and under budget, and has not been subject to recalls or FAA directives.

CONCLUSION

Ultimately, for safety certification, final software has to be tested on final hardware—you have to be able to “test what you fly and fly what you test.” Simics is not a substitute for genuine hardware certification. What it can do, however, is ensure that your software is ready to meet the final test requirements—and in the process save time, cut costs, and reduce risks. The ability to detect and fix defects during development rather than during testing makes the entire process far more efficient. And you're not reliant on test hardware, which can be expensive and limited in availability.

Using Simics reduces total certification costs as well as performance and program risks, and helps ensure timely delivery of a quality product. Considering the importance of certification and the high price of failure, it is better to go into it with confidence that you can pass with flying colors.

