Software Fault Isolation Using HALT and HASS

The following paper from Allied Telesis (formerly Allied Telesyn) Labs in New Zealand illustrates how HALT and HASS can help detect software issues in digital systems.

Introduction

In 2005, Allied Telesyn established a Highly Accelerated Life Testing (HALT) facility at its New Zealand research and development Centre. This facility enables rapid diagnosis of product problems during the development phase of the product lifecycle. The company also undertakes Highly Accelerated Stress Screening (HASS) testing at its factory sites. While the techniques of HALT and HASS are commonly reported, it is difficult to find specific examples of faults identified during testing. Allied Telesyn categorizes faults found during HALT and HASS by their root cause, either mechanical, electrical, or software. This white paper focuses on the discovery of software failures in products and the techniques used for their isolation.

HALT testing is conducted throughout the design stage of product development to highlight major product problems so they can be isolated, analyzed, and corrected promptly. HASS is conducted on a sample basis at the company's three factories. The screens created during HASS development are used for No Trouble Found (NTF) debugging, component qualification, and software patch verification.

Until 2005, all of Allied Telesyn's HALT and HASS testing had been conducted offsite using an independent lab in San Jose, California. This testing process has been extremely useful for Allied Telesyn staff to gain experience and knowledge. Still, it has also been expensive due to the company's remote location on New Zealand's South Island and the cost of travel to California.

In 2004, Dr. Gregg Hobbs was invited to Allied Telesyn's New Zealand design center to teach his Mastering HALT and HASS seminar. This led to an experimental HALT conducted locally under the guidance of Dr. Hobbs and ultimately to the implementation of a comprehensive HALT and HASS program in Christchurch. Allied Telesyn encountered multiple obstacles and challenges inherent to setting up a HALT program. Initially, many engineers were skeptical about the relevance of failures found at the high stress levels applied during the HALT process. This response is common when first adjusting to over-stress techniques, especially.
When testing is conducted remotely, it is difficult to directly isolate and diagnose software faults, let alone debug them from the other side of the world.

The cooperation of Allied Telesyn's Engineering team, coupled with extensive feedback through comprehensive debug reports, has overcome these initial frustrations. Top-level management support is imperative for implementing any HALT and HASS program.
What is a Software Fault?
The term "software fault" is defined at Allied Telesyn as a fault found in:
•The firmware of a product, such as code in a Programmable Logic Device (PLD)
•The boot code of a product, such as EPROM boot code
•The operating system of a product

These software faults may occur because of changes in the performance of the associated hardware.

Testing and Monitoring

The sequence of tests applied to a product during HALT and HASS plays a critical role in uncovering software faults. The monitoring process provides a snapshot of each product's status shortly before failure.
The test sequence enables us to identify a myriad of software faults related to the fundamental operation of products, including clock signals, voltage rail monitoring, and environmental factors, which can affect product stability. Other factors that may also be monitored during testing include read/write timing, chip selects, reset pulses, and signal integrity.

A considerable amount of debug information must be extracted from the testing process to highlight failures; without it, fault isolation would be extremely laborious. The debug information may include memory dumps, voltage measurements, clock observations, and other product-specific measurements.

Testing and Monitoring for HALT
HALT testing and monitoring need to be as comprehensive as possible. Experience shows that greater test coverage, coupled with exhaustive product monitoring, yields a wealth of diagnostic information that can be used when debugging each failure mode. Test developers need to put considerable thought into developing a broad range of tests that will reveal the required information, coupled with an all-encompassing monitoring program, before starting the HALT.

Testing and Monitoring for HASS
Highly Accelerated Stress Screening (HASS) is performed on multiple sample units of each product during mass production. Appropriate design measures are important for HASS to ensure that external monitoring equipment can be utilized, and should include provisions for both hardware monitoring and built-in software testing in conjunction with data logging.
Typical Test and Monitoring Process

Allied Telesyn maintains a 15-minute dwell time at each step during the HALT process. During this time, the unit is functionality tested and monitored for failure. Some examples of the testing conducted include:

1. External traffic test
    a. Using industry standard equipment
2. Power cycling
    a. Voltage and frequency margining
3. Internal memory test
    a. RAM test
    b. NVS test
4. Internal packet generator test
    a. CPU test
    b. Encryption engine test
    c. RAM test
5. Other product-specific tests

In addition to this test sequence, the product is monitored in real time to ensure that the slightest change in operating specifications is observed and recorded. The product monitoring includes:

1. Voltage rails
2. Frequency
3. Temperature
4. Critical system signals
5. Self-diagnostics

All of these exert some influence on whether a fault will be discovered and corrected or remain unnoticed and hamper a product throughout its life in the field.

Examples of Fault Isolation Techniques

HALT is a creative process with many paths to a reliable product and many innovative approaches to fault isolation. Using hardware and software tools, fault isolation can be as rudimentary as or as complex as the Engineer’s creativity allows.

The following examples demonstrate how to identify the specific cause of a failure simply and efficiently.

Note: During thermal isolation, it is important to monitor the temperature of the component or circuit using a thermocouple or similar device.

Freeze Spray Example

The controlled application of freeze spray to a suspect component or circuit identifies the area where a failure is generated. This keeps the suspected cause of failure at room temperature while the rest of the product is heated, providing a straightforward and cost-effective way to isolate heat-related faults.

Figure A2.1 Freeze Spray is Applied to a Suspect Component

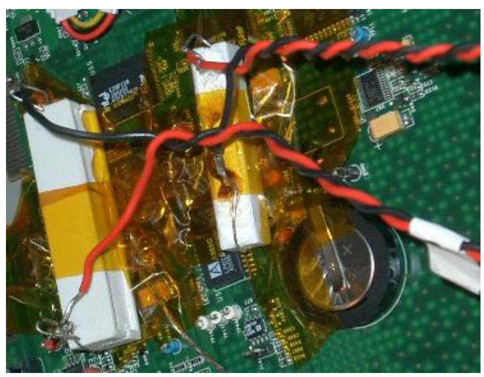

Figure A2.2 Power resistors are applied to a suspect circuit

Heat Application Example

The controlled application of heat to a suspect component via a power resistor or Peltier device identifies components that are failing due to cold temperature. Keeping this component isolated from surrounding environmental conditions can provide a simple way to identify the root cause of a failure.

Note: In both of these examples, only the location of the failure has been identified. The cause of failure must be determined before deciding upon an appropriate course of action. The combination of testing, monitoring, fault isolation, and effective debug information will, in most cases, lead to the root cause of the failure. Once the root cause is determined, the fault is analyzed, and an appropriate corrective action is implemented, allowing testing to continue.

Allied Telesyn Fault Examples

This section focuses on software failures found by Allied Telesyn during HALT and HASS testing. Figure A2.3 shows that almost one-third of all failures found during Allied Telesyn's HALT testing are attributed to software. Two percent of failures are yet to be determined.

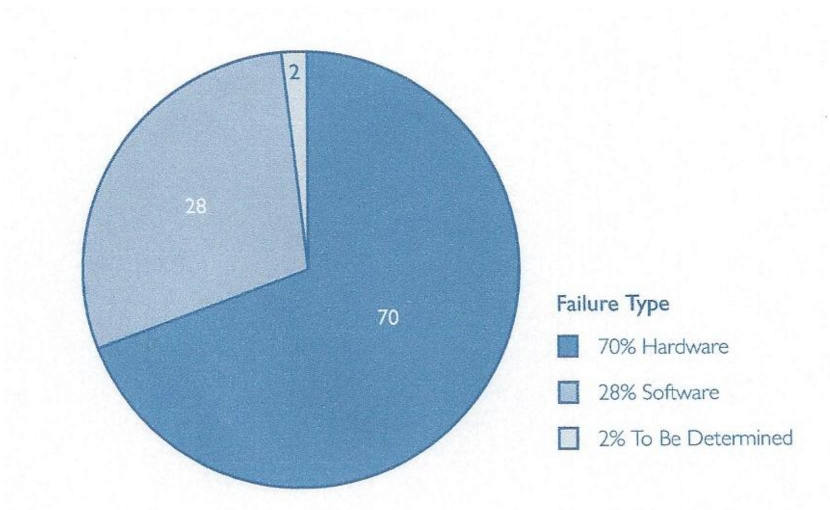

Figure A2.3 Fault Type Summary by Percentage

Figure A2.4 compares the failure type with the applied stress, illustrating that all software-related failures occur within the first three steps of the HALT process. In most cases, the problem is identified and eliminated before the product undergoes combined environment testing.

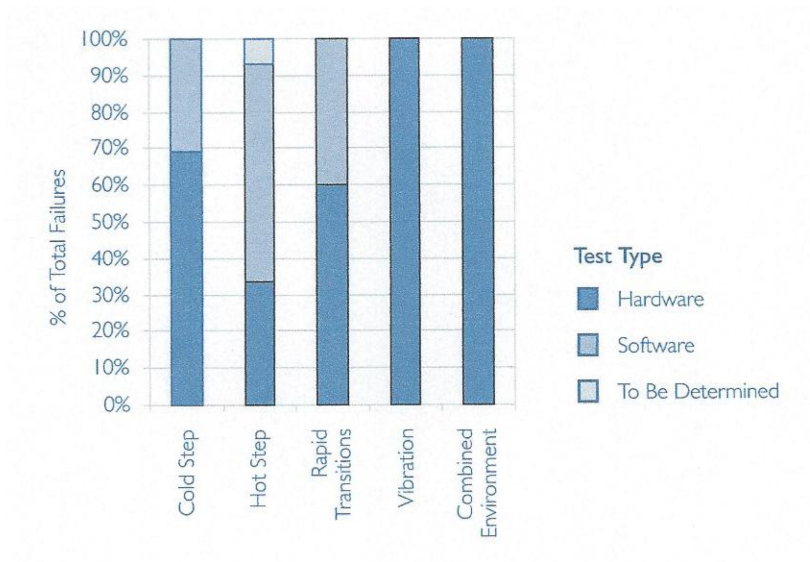

Figure A2.5 Fault Type Summary by Failure Type

Common Software Faults Identified During HALT and HASS

The following examples demonstrate the variety of software failures discovered by Allied. Telesyn during HALT and HASS.

Abnormal LED Activity
Light Emitting Diodes (LEDs) are used on network products such as routers and switches to indicate their current status. LEDs are often driven directly from the physical interface for basic functionality or, in more complex designs, through one or more Programmable Logic Devices (PLDs). This logic includes specific timing signals for status display, reset, and chip selection.

One fault occurred after a hard power cycle, but not after a soft reset. The failure caused random LED activity upon powering up, which persisted until the temperature was raised or the soft reset was activated. This anomaly was found during cold step testing at-10°C and was attributed to the reset pulse timing in the PLD code.

The fault was isolated by applying heat to the suspect component, leading to a fix implemented within 30 minutes. The mechanics of this failure lay in the PLD's code. During a hard power cycle, the PLD reset pulse duration was insufficient, causing the board to power up in an unknown state. After the PLD code was updated, there were no further LED indication errors at temperatures as low as 50 °C.

A similar Allied Telesyn product that had been in mass production for three years had also exhibited this fault during standard production testing. The fault had slipped through the traditional design qualification testing when Allied Telesyn did not have a HALT program in place.

Switch Tuning

Network switches are built around silicon switching components that divert packets via hardware. Some components require tuning over an extended temperature range at various input voltages; one such device is the Marvell Prestera 98EX115D.

While developing a new switch product, it was necessary to tune the silicon switch to ensure proper operation across a broad temperature range. At the start of the HALT test, the unit had an Upper Operating Limit (UOL) of 70°C and a Lower Operating Limit (LOL) of -20°C.

Step stressing revealed that the switch was not tuned correctly, and over a few days, new software versions were created and tested, ultimately fixing the errors. Corrective action via software enhancements resulted in a UOL exceeding 100°C and a LOL below -60°C.

Power-Up Sequencing

Power-up sequencing is a common requirement in today’s complex electronics. While many dedicated power sequencing components are available, it is occasionally more efficient to use onboard logic to control the voltage rail sequencing.

A product that underwent HALT testing failed when various components were not functional after a power cycle at-20°C. Applying thermal isolation techniques identified the PLD device as the faulty component. The code that controlled the power-up sequencing proved unreliable at low temperatures.

This problem had previously been identified and fixed during prototype development at Allied Telesyn's design center. However, HALT revealed that the applied fix had shifted the failure, making it less replicable, and allowed Allied Telesyn to implement a more robust corrective action. The code inside the PLD was modified, allowing the product to reach temperatures below -50°C without failure.

System Crash

System crashes during the design phase are inevitable when developing prototypes that contain onboard CPUs. The key is to stimulate these crashes during development and allow a fix to be implemented at minimal resource and time cost.

A product that had been in the field for six months was taken to a HALT lab to investigate its operating margins. At the beginning of the HALT, the product attained a UOL of 70°C, and a system crash was observed at this temperature. By changing a register setting in the boot code to initialize a particular memory interface, the unit was able to operate at temperatures above 100°C.

In addition to the software fault, a flaw in the CPU silicon was revealed, amplifying its effects. The complete solution came with the release of a new CPU revision, coupled with the original boot code change.

The same problem eventually appeared on three separate products, two of which occurred in the field environment. No further failures occurred after the boot code was modified and the CPU flaw corrected.

System Silent Reboot

A silent reboot occurs when the product reboots without displaying any error or debug messages. These silent reboots are notoriously frustrating to debug and often require exhaustive troubleshooting.

A major Allied Telesyn customer with 22,000 units of one product type in a major network was experiencing 8 silent reboots per day, representing a 0.036% failure rate. Allied Telesyn faced the problem of how to replicate a failure that occurred in only 0.036% of units. The same fault took weeks to replicate intermittently with traditional methods, prompting the use of a simplified HALT to identify the problem. The same failure mode was repeatedly replicated in less than one day of testing, enabling software engineers to isolate and remedy the cause of the failure in a short time.

Rapid Thermal Transitions exposed a flaw in the software during temperature ramps, even though the initial failure occurred in a moderate climate inside a server room. The failure mode was only apparent when running one particular test. A software patch was released to fix this problem.

Summary

Software is an integral part of today’s electronics, and ensuring the software on our products is reliable is vital to delivering quality products to our customers. Reliability failures in hardware are usually caused by wear. However, software does not wear out and may continue to function after an initial failure, making the fault harder to replicate, isolate, and analyze. HALT and HASS are excellent tools for uncovering dormant defects in both hardware and software. Still, without an exhaustive test and monitoring plan, many software faults will continue to go undetected.

A comprehensive HALT program needs to include relevant monitoring and test suites that uncover more than just hardware faults. Having a comprehensive test plan yields substantial information that can be used to assess the relevance of each failure and the corrective action required.

Experience at Allied Telesyn has shown that accurate fault isolation is a fundamental aspect of HALT and HASS testing, without which analyzing faults and implementing corrective actions may be an exercise in trial and error. Thorough planning and consultation before conducting HALT should provide an overall picture of the products' dependencies and highlight the steps necessary to test and monitor these critical aspects. The HALT chamber is a very useful tool for increasing a product's reliability; it is not a magic box and will only provide the outlined results if used to its full capabilities.