Accelerating Automotive Random Error Analysis Through PSS

Adnan Hamid, CEO
Breker Verification Systems

Functional verification is hard enough, but when placed in the context of automotive, it takes on several additional dimensions. Much has been written about addressing the Systematic aspects of ISO 26262 which is concerned with ensuring that the requirements for the hardware design are specified correctly, have been fully considered with safety in mind, and that the implementation of these requirements has been fully and rigorously verified. Less has been written about the Random aspect, which is concerned with ensuring that the correct operation of the device is maintained even if internal components are affected due to environmental or other effects.

Safety mechanisms designed to correct errors introduced during the operation of the device (due mainly to environmental effects) are included in the design itself. The Systematic Verification Process must ensure that these safety mechanisms do not interfere with the correct operation of the device.

In addition, a significant analysis of the device is performed to ensure that it will self-correct in the event of a Random error. This involves injecting faults at points in the design during a special verification process, and then observing the continued operation of the device. Once tests have been derived for the Systematic phase, they may also be leveraged for this Random analysis process. Being able to leverage the same tests from the block level right up to full system can save a lot of time during this phase.

Random fault analysis consists of running tools such as fault simulation, sometimes combined with formal verification, to observe the effect on device operation of inserting a fault at specific locations in the design. Fault simulation is inherently a slow process. In effect, a simulation of the entire test suite must be run for every potential fault in the design. While special simulators have been developed that can handle multiple faults in parallel (or concurrently), this is not possible when using emulation – a tool often needed to accelerate the execution of these large designs and enable more reasonable execution times.

Fault simulation technology has been around for a long time and was in general usage in the early 1980's for determining the effectiveness or vector sets for the purpose of manufacturing test. This was before scan technology took over and made most fault simulation unnecessary. There is, however, a significant difference between fault simulation for manufacturing test and fault simulation being used for verifying functionality in the presence of faults.

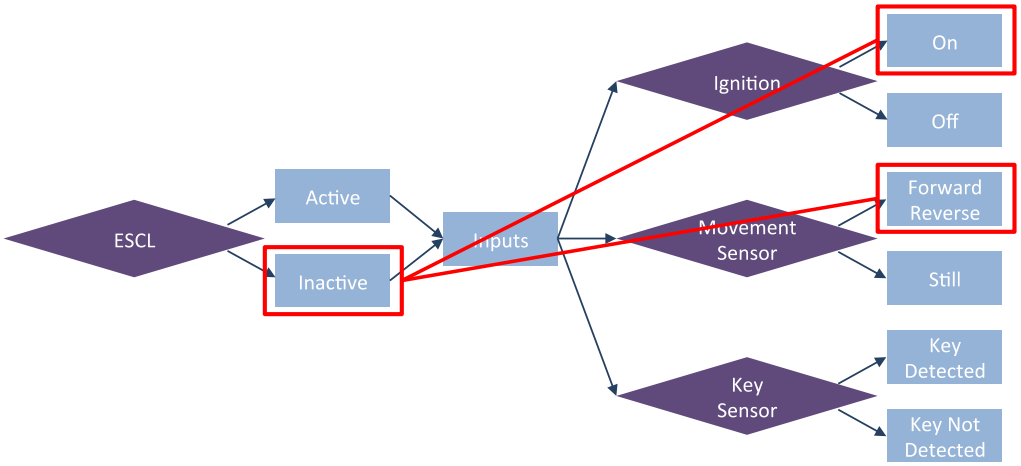
In the case of manufacturing test, the objective is only to demonstrate that the effect of a fault can be observed. As soon as that happens, the fault simulation run, for that fault, can usually be terminated

because continuing the run would provide no additional information. But with ensuring continued safety under a random error, the test has to be continued until no effect of the fault remains in the design.

Given a normal UVM type of methodology, this is impossible to determine and so all tests would have to run to completion. With a PSS flow, which employs continuous checking of intermediate results, it is likely that methodologies could be developed that would allow tests to terminate earlier, thus speeding up the analysis.

The total number of faults, even if using a simple stuck-at fault model is huge. It is effectively impossible to perform a fault simulation for every fault. Many faults can be collapsed, and it can be proven that if one particular fault can be detected, then a number of other faults would also exhibit exactly the same symptoms and thus only a single example has to be run. The number of faults, however, remains too large to be considered.

In the 2000s, technology was developed to assess the quality of testbenches. It was argued that a statistical sample could be used with high confidence, and techniques were developed to find the high-quality faults that, if those were found, would be a good indicator that many other simpler faults would also be found. In this manner, high confidence could be achieved with a much smaller total number of faults being simulated.



Path Constraints Applied Across a PSS Graph

The other side of this is ensuring that the stimulus used for these runs is as compact as possible. This means that every step in every test has to be as unique as possible. The better this stimulus is at targeting faults, the more efficient the analysis will be. PSS has a significant advantage over UVM in that tests synthesized from a PSS model understand sequentiality in the design, and scenarios define actual activity that are meant to be observed in the design. There is thus no stimulus activity that does not cause something important to happen in the design, unlike tests generated from a UVM generator.

There is another aspect of a PSS methodology that is important – that of hierarchical composition. To achieve an ASIL D rating, greater than 99% fault coverage must be achieved right across the final design. To achieve these results in practice, a design team will perform various analyses at different verification levels, starting at the block level and processing through subsystem and full system test. For example, a team might perform a verification step at the block level just to make sure a single fault will correctly trigger an alarm signal, waiting for the subsystem level to run rigorous fault analysis to fully exercise the safety mechanisms, and then perhaps perform a Monte Carlo statistical analysis at the full system level. Alternatively, a large number of faults could be run at the block level with the goal of showing that all faults raise an alarm. Then, at the system level, it is not necessary to simulate all of the block-level faults because it has been shown that they can be collapsed down to any fault that triggers an alarm.

Being able to reuse the same stimulus, as well as checkers and coverage models, at all levels saves time in both crafting the tests to best target the fault analysis and comparing the coverage at the different abstraction levels.

It should be noted that Portable Stimulus also provides significant improvements to the ISO 26262 Systematic flow as well. Modeling requirements for verification is normally not easy, but the PSS actually allows for a relatively clean mapping from requirements to test scenarios that allows for coverage to be easily assessed, especially if Breker's Path Coverage overlay is applied. This simplified completion of the classic V-Model saves time and increases accuracy. More on this in a future article.

Tools will continue to be enhanced that reduce the total number of faults necessary, the quality of the stimulus, and the performance of the fault simulation technology. It will not be possible to do any of this using brute force techniques as was done in the past. Significant progress has already been made in all of these areas. Some of it builds on the technology of the past, while other pieces are completely new.