ISO 26262 and You

Why Automotive electronics suppliers will make increasing use of formal tools to meet the standard's strict requirements for verification and satisfy supply chain demand.

By Jörg Grosse, OneSpin Solutions

Although standards are in play for many of the electronic products that consumers buy and use, it is rare for anyone except experts to know the details. There are partial exceptions such as USB, where users at least pay attention to which version of the standard is supported in their host and peripheral devices to ensure compatibility. This has not been the case for automobiles, although important standards exist. Until quite recently, ISO 26262 was a relatively obscure specification for the development of safety-related electrical and electronic systems within road vehicles.

> "Formal tools don't just find hardware design bugs; they can provide proof that no further bugs exist."

Self-driving vehicles are changing many aspects of the status quo, raising a host of questions about liability, massive changes in infrastructure, and creation or loss of entire categories of jobs. ISO 26262 is right in the middle of these changes. Automotive manufacturers and their electronics components suppliers must follow this standard and will make a big deal about compliance. Even consumers might become more aware of safety requirements and demand that their new vehicles conform.

VERIFICATION CHOICE

ISO 26262 imposes stringent requirements that encompass the entire life cycle of a system, from concept phase to development, production, and decommissioning. It addresses the overall safety management process. The standard specifies two types of faults in electronic components, both of which must be fully verified. Systematic faults are introduced during development, either through human error or tool malfunction. Random faults occur during the actual operation of the device due to external effects.

Systematic faults are handled through rigorous verification and the careful tracking of specific device requirements. Formal methods are more important than ever, since only they can provide mathematical certainty of correctness. A key characteristic of formal tools is the ability to examine design behavior exhaustively, without the need for input stimuli, and to prove that the design never deviates from its intended function. Formal tools don't just find hardware design bugs; they can provide proof that no further bugs exist.

Simulation tools cannot achieve this level of precision, although there are some types of behavior that are best verified with simulation or emulation. The choice of which behaviors to verify with simulation and which with formal methods is one of the elements of a verification plan, the heart of any electronic development project. Chips for automotive electronics can be highly complex, making verification a long and difficult process. Per ISO 26262, this process must be both rigorous and effective at eliminating systematic bugs.

VERIFICATION FLOW

Figure 1 shows a typical verification flow for large, complex designs. The project begins with the requirements for the end product, typically written by the architecture and product management teams. As

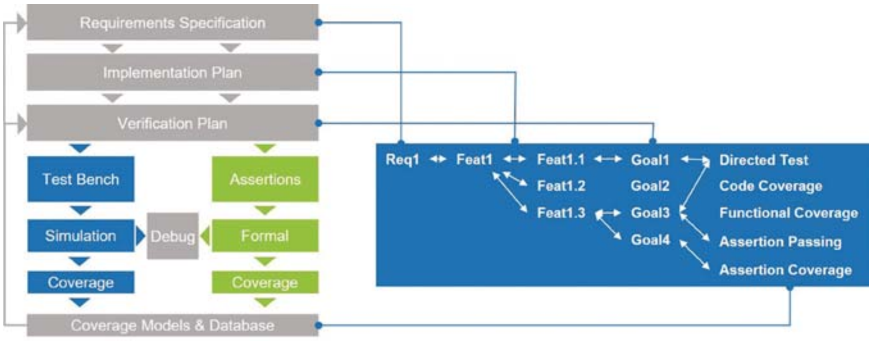

Figure 1: Development of chips compliant to ISO 26262 requires a well-organized process.

the design commences, each requirement is implemented by a series of features. In turn, the verification of each feature is broken down into a series of goals that must be met during the verification process. The list of goals is the basis for the project verification plan.

As part of the plan, the verification team decides which methods and tools are best suited to verify each feature and satisfy associated goals. For simulation, a testbench is developed, usually compliant with the Universal Verification Methodology (UVM) standard. Most of the verification is performed using pseudo-random tests that require functional coverage metrics to gauge their effectiveness. In an application such as automotive electronics with strong safety requirements, the verification team must achieve a very high degree of coverage.

Sometimes engineers will hand-write some directed tests in order to hit coverage points not easy to exercise with pseudo-random stimulus. They also are likely to measure code coverage on the design while running all automated and directed tests. Code coverage does not tie directly to intended functionality, but uncovered portions of the design may indicate redundant or spurious hardware logic, logic that cannot be exercised due to a design bug, or gaps in the verification plan.

For the portions of the chip verified by formal means, the team develops a set of assertions describing how the logic should behave. A formal model checker analyzes the design against the assertions and either reports bugs or proves agreement. For some types of analysis, formal applications ("apps") require no assertions from the user at all. Formal tools also produce coverage metrics, some similar to those from simulation and some unique. As shown in Figure 1, the verification flow must support a range of coverage metrics to judge verification thoroughness.

It must be possible to roll up all the coverage information into a single view of verification completeness. Only a thorough verification plan, a rigorous verification process, and comprehensive coverage metrics can give the verification team the confidence that systematic faults have been eliminated and meet the high bar set by the ISO 26262 standard. This is clearly important for the success of self-driving vehicles; a missed design error could easily result in an accident causing serious injury or death.

HANDLING RANDOM FAULTS

Eradication of systemic faults is not enough. Consumers also expect their vehicles to operate robustly over a long period of time, even when random faults occur due to the harsh environment. A corroded wire might break, or an alpha particle might flip a memory bit. Such faults can and do occur. ISO 26262 requires that such faults don't affect the safeness of the vehicle and therefore be handled by circuitry within the chip or through software. Such safety mechanisms must be verified to ensure that they will catch the vast majority of possible random faults.

There are two acceptable actions when a random fault occurs. One possibility is to correct the fault so that normal operation of the vehicle can continue uninterrupted. Several popular classes of error-correcting codes (ECCs) can compensate for one (or more) bit being flipped. Faults that cannot be corrected must be detected and an appropriate level of alarm must be raised. One can easily imagine classes of failure in self-driving vehicles where the best course of action to an alarm is to move slowly and safely into the breakdown lane and then stop.

Determining how well a chip design will handle random faults is not a trivial problem. Once again, formal methods provide a solution. Another key characteristic of formal tools, particularly relevant to safety-critical applications, is the ability to finely control the injection of faults into hardware models and analyze their sequential effects. Crucially, formal tools can perform this task efficiently, in terms of both user effort and computational demands, and non-invasively (no need for manual instrumentation of the design description). Figure 2 shows how this process works.

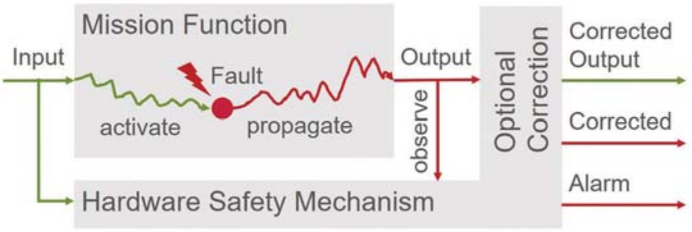

Figure 2: A hardware safety mechanism must either correct or detect random faults.

Formal tools can inject random faults, analyze whether faults can propagate to cause trouble, analyze whether the effects of faults can be observed, and provide metrics for hardware safety coverage. If a fault simulator is available, the work can be split with the formal tools, and results can be combined. Only with thorough application of these techniques can the verification team know that it has satisfied all aspects of critical safety standards.

Most consumers may not know or care about automotive safety requirements today, but this is likely to change as they look at moving to self-driving vehicles. Consumers may not know the details, but they will expect manufacturers to adhere to best practices. Compliance to ISO 26262 will be a badge of honor for manufacturers. Automotive electronics suppliers will make increasing use of formal tools to meet the standard's strict requirements for verification and satisfy supply chain demand.

Jörg Grosse is product manager for functional safety at OneSpin Solutions GmbH. He has more than 20 years of experience in electronic design automation (EDA), functional verification, and ASIC design, having served at companies in Europe, the United States, and New Zealand. Grosse holds a Diplom-Ingenieur (FH) in electrical engineering from Anhalt University of Applied Sciences in Germany.