Exhaustive solutions for complex verification challenges

Even the most carefully designed testbench is inherently incomplete since constrained-random methods cannot hit every corner case. Unfortunately, this means that even after 100% functional coverage is achieved there can still be showstopper bugs hiding in unimagined state spaces. Questa® Formal apps statically analyze a design's behavior with respect to a given set of properties; then exhaustively explore all possible input sequences in a breadth-first search manner. This uncovers design errors that would otherwise be missed or are impractical to find with simulation-based methods.

Three categories of verification technologies and tools

The Questa® Formal-based Technology suite offers users three categories of formal verification technologies and tools, ranging from push-button apps to advanced model checking: CDC solutions, automated applications, and model checking and Verification IP.

CDC solutions: All properties and design intent are inferred by the software. There are four fully automatic tools in the Questa CDC suite: Questa Clock-Domain Crossing Verification; Questa Signoff CDC; Questa CDC-FX,
which ties formal-based CDC results into QuestaSim analyses; Questa Power Aware CDC, for when low power circuitry is incorporated into the DUT; and Questa Reset Check, for reset signaling network analysis.

**Automated applications:** Assertions are synthesized from a combination of automatic RTL design analysis and a high-level specification of design intent. The properties are then exhaustively verified with formal analysis. The Questa Formal suite includes applications to address issues such as code coverage closure, X-states analysis, and register verification. Additionally, apps like Questa Sequential Logic Equivalency Check (SLEC) use formal methods to perform exhaustive comparisons between inputs to reveal any behavioral discrepancies, expediting ECO and fault analysis verification.

**Model checking and Verification IP:** Users write properties and constraints and then run the Questa Formal suite’s property checking engines, which are among the industry’s most powerful. This allows for advanced methods, such as abstraction, decomposition, assume-guarantee, and waypoints. Manual property checking can also address issues of I/F protocols, functional coverage, control logic, data integrity, and post-silicon debug, which in sum provide the most exhaustive possible analysis of a design. Formal-optimized Verification IP is available for popular standard protocols.

**Benefits and Highlights**

**Find bugs early** – Questa Formal applications enable verification to be started early in the design phase, before a simulation testbench is ready.

**Improve verification quality** – The exhaustive approach of Questa Formal apps enables your analysis to traverse the entire state space, revealing unexpected corner-cases that can be missed by other verification methods.

**Increase verification throughput** – Questa Formal provide complete, accurate, push-button automatic verification of specific design aspects that are difficult or impossible to verify using traditional methods.

**High performance analysis** – Dedicated applied research and engineering investment in Questa Formal core technologies have produced continuous improvements in wall clock performance, memory usage, and storage consumption; together with optimized engines under-the-hood, this means Questa Formal regularly exceeds demanding scalability and compute resource expectations.

**An Integral Part of the Enterprise Verification Platform**

Built upon several powerful technologies and tightly integrated with Veloce® emulation, the Enterprise Verification Platform transforms verification, dramatically increasing productivity and more efficiently managing resources. The Questa CDC and Formal solutions are integrated with simulation and emulation, sharing common features such as verification management, compilers, debuggers, and language support for SystemVerilog, Verilog, VHDL, UPF, and more. This enables users to select the best application or tool for the job, and then combine results from all the engines to dynamically track the progress of the entire verification program.