

# ESP

## Custom Design Formal Equivalence Checking Based on Symbolic Simulation

### High-quality equivalence checking for full-custom designs

### Overview

ESP is an equivalence checker for full custom designs. It enables efficient comparison of a Verilog reference design against other Verilog models or a transistor-level SPICE netlist.

ESP provides fast and extensive coverage, enabling users to quickly find bugs and have the confidence that the Verilog reference design is functionally identical to other Verilog models or its transistor-level implementation.

ESP improves overall verification productivity by simplifying the testing process. It directly verifies the SPICE netlist, eliminating the need to manually extract the transistor network into a gate-level representation.

### Verification Scope

ESP verifies that two different design representations are functionally equivalent. These designs may be described as Verilog behavioral models, RTL, UDP's, gates, transistors, or SPICE netlist views (Figure 1).

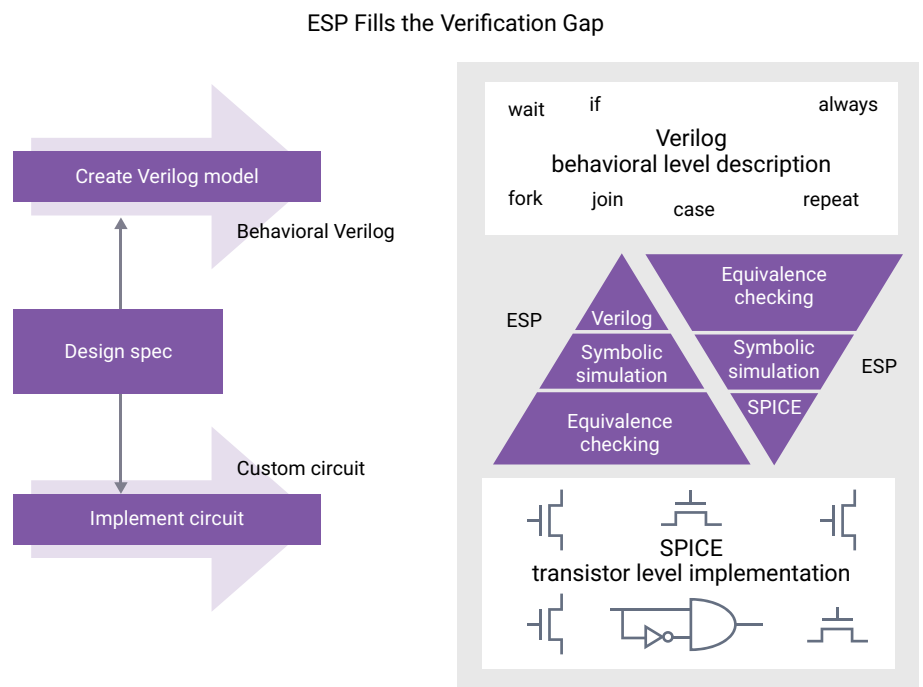


Figure 1: Bridging the verification gap between Verilog and SPICE

# Benefits

## Higher Quality

ESP provides fast and complete coverage, enabling you to quickly find bugs and have the confidence that the reference model is functionally identical to the transistor model.

Increased Productivity With ESP, you no longer have to derive directed and random tests or have a long delay in releasing models while you complete your verification suite.

## Easy to Use

ESP directly verifies the SPICE netlist, eliminating the need to manually extract the transistor network into a gate-level representation.

## Memories are Changing

Over 50 percent of the total silicon real estate in today's SoC is consumed by memories. And as designs move toward sub-nanometer process technology, functionalities such as redundancy, ECC, BIST, pipelining, etc. are being added to these designs, resulting in significantly higher functional complexity.

With few standards and many degrees of freedom, functional verification of embedded memories has become a critical need in the SoC design verification process. A key requirement of a successful design is that the behavioral reference model used for SoC full chip simulation is functionally identical to the transistor-level netlist that represents the actual implementation.

As the quantity and complexity of memory designs continues to increase, and the project schedules and available resources continues to shrink, designers are faced with the challenges of delivering high quality memories on-time and with limited resources.

## High Coverage Verification

Traditional methods used to verify memories such as SPICE simulation or cell-based formal verification have their limitations. SPICE simulation provides circuit-level accuracy but its coverage is dependent on the vector set created by the engineers and the time available for running the simulation. Likewise, cellbased formal verification may provide complete coverage but cannot accurately represent the behavior of the transistorlevel netlist.

ESP is based on a patented symbolic simulation technology that combines the power of formal methods with proven event-driven simulation technology. ESP leverages symbolic simulation to perform sequential equivalence checking, dramatically increasing the quality of functional verification (Figure 2).

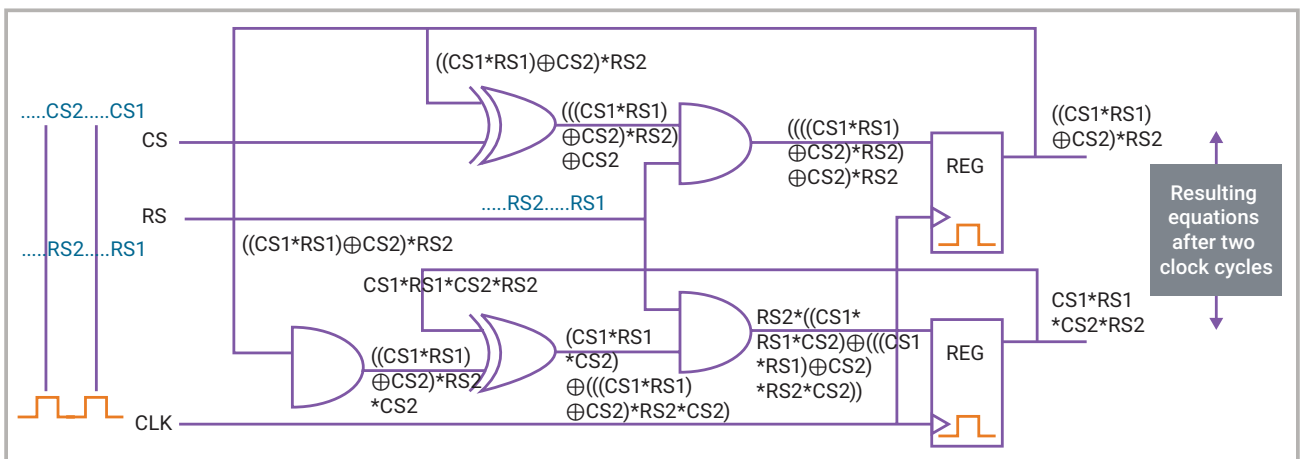


Figure 2: Symbolic simulation propagation—ESP performs formal analysis on output equations to establish equivalence

ESP simultaneously simulates two different design representations using symbolic inputs while observing the outputs of each representation to assure equivalent responses. Instead of applying all possible combinations of binary states, ESP applies a symbol that represents all possible input states. This results in coverage of  $2^N$  possible states with only  $N$  number of symbols (Figure 3A).

ESP is capable of applying formal verification to the circuit-level designs, delivering an easy-to-use verification solution that is circuit-aware.

Unlike other methods that require modeling of transistor-level designs to cell-based gate equivalent netlists, ESP directly verifies the functional equivalence of the SPICE-level netlist against a behavioral or RTL representation of the design. ESP greatly simplifies the inclusion of transistor parasitic effects in the functional model by automatically calculating the RC value based on transistor length, width, and process technology (Figure 3B).

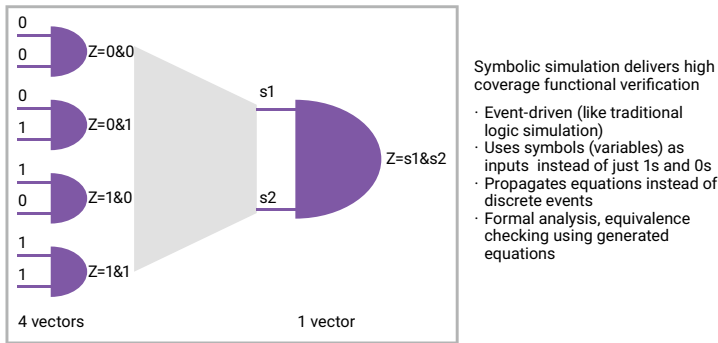


Figure 3A: Using symbols as inputs delivers high coverage

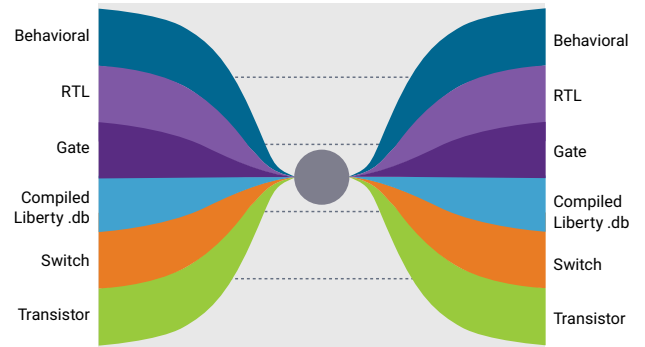


Figure 3B: Direct verification of behavioral Verilog, RTL, gate, Liberty .DB, switch and transistor-level design netlists

## Differentiating Features

### Power Integrity Verification

The Power Integrity Verification flow detects common low-power design errors and power-down/sleep-mode failures. When errors are found, this flow also generates SPICE vectors for analysis and debug (Figure 4).

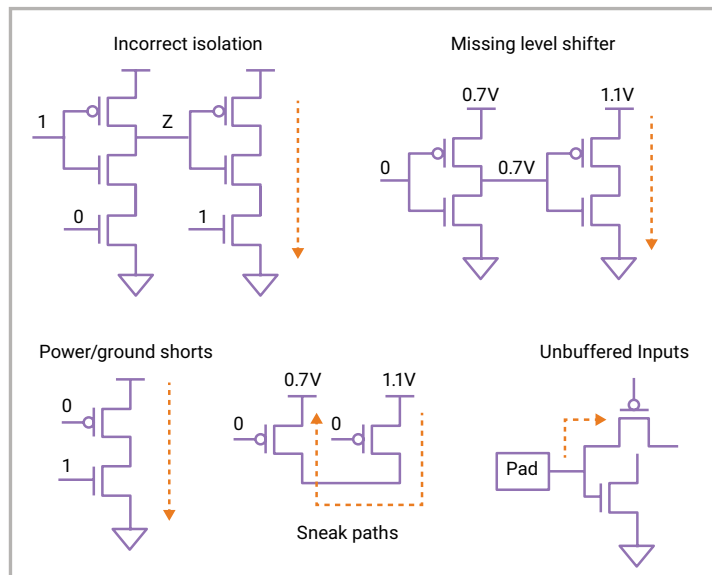


Figure 4: ESP power-integrity verification

## Redundancy Verification Flow

ESP uses formal techniques to quickly verify that the redundancy logic added to the memory array to replace defective cells and improve yields is performing correctly (Figure 5).

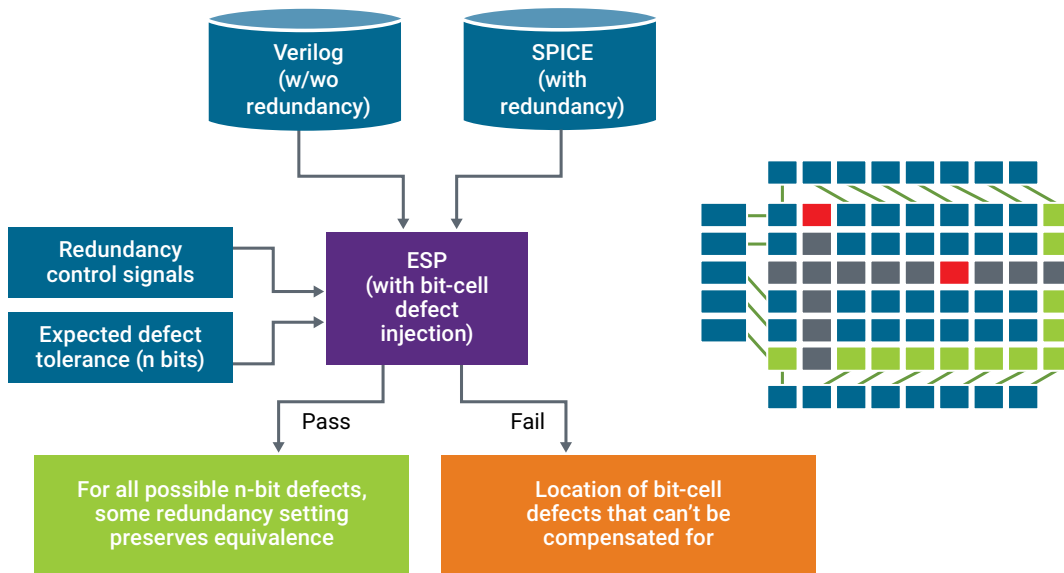


Figure 5: Redundancy verification flow

## Verdi Interface

Using the Verdi3 advanced debug platform environment already used by most RTL designers, ESP provides a graphical way to debug mismatches in transistor-level design using Interactive Signal Tracing (IST). ESP also provides resistance, capacitance and delay information for events on schematic nets as well as traditional Verilog "1", "0", "X" and "Z" logic values. Using Verdi3 dramatically reduces the overall verification debug time and improves productivity by providing a common debug platform for logic and circuit designers alike (Figure 6).

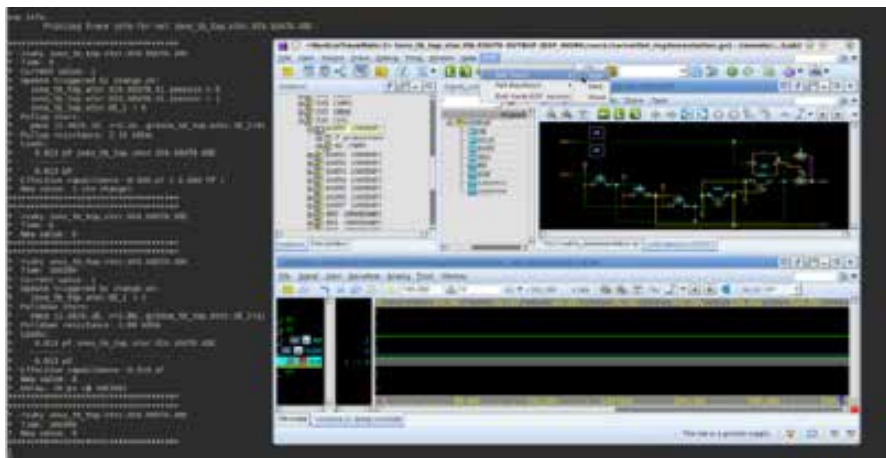


Figure 6: Verdi3 interface provides graphical debug environment

## Device Model Simulation

ESP Device Model Simulation allows you to perform ESP verification for new device technologies, e.g. FinFET and FDSOI, as soon as your SPICE circuit simulator supports that technology. It supports multiple circuit simulators and transparently handles CMI and TMI encrypted models (Figure 7).

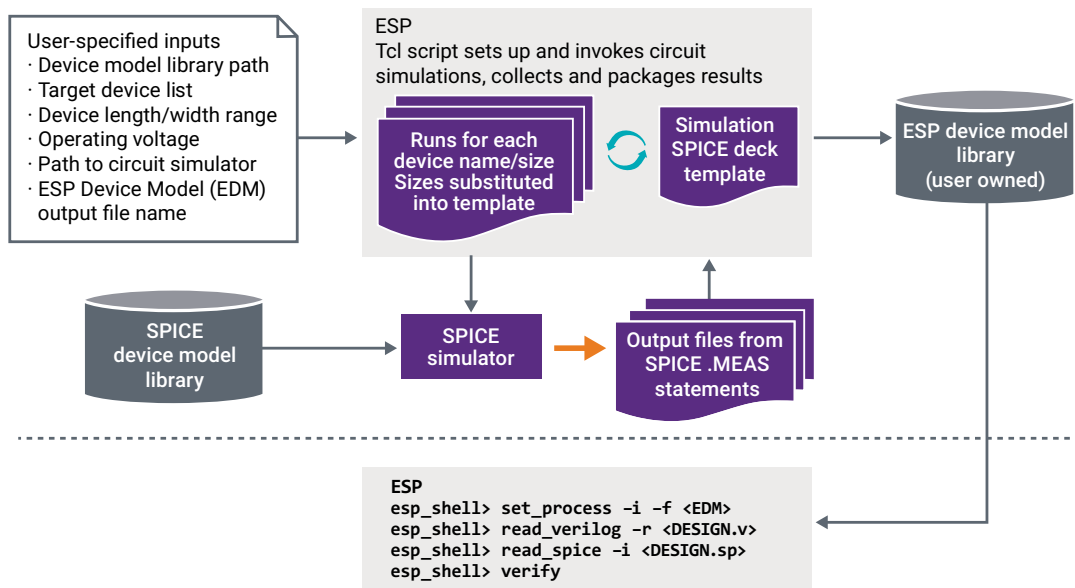


Figure 7: Generating ESP device models from SPICE models

## ESP Additional Features

Full Verilog (RTL, Structural and Behavioral) Language

The tool supports all constructs from the Verilog language, including behavioral structures, such as fork, join, and task, etc.

### Timing Dependent Functionality

It calculates transition time as a function of symbolic states and transistor widths and lengths, and therefore supports timing dependent functionality.

### Automated Error Validation

It spawns SPICE simulation to validate verification differences and provides generated waveforms for debugging.

### Distributing Testbench Execution

The tool supports simulation of testbenches in parallel using LSF and Oracle GRID (SGE), and therefore reducing time-to-results.

### Integrated Test Suite Management

A designers' productivity is improved by automating multiple test bench simulations and producing a merged coverage report.

### Interactive Signal Tracing (IST)

IST provides unique features to help debug verification errors. By using a simple command-based interface or Verdi schematic display, a designer can back trace incorrect or suspicious values in the design. It also tracks internal signal transitions and reports calculated transistor strengths.

### Transistor Level Verification

Transistors are natively supported by ESP and it does not depend upon gate extraction or pattern matching techniques.

### Asynchronous Timing

It supports asynchronous clocks, pulsed logic and self-timed logic.

## No State Point Mapping

It verifies functional equivalency of designs with different structures, without any dependency on isomorphic state mapping.

## High Capacity

It has the ability to verify over one billion transistor designs on a single workstation.

## Typical Applications

- Compiled memories
- Custom memories (i.e. CAM, SRAM, ROM, etc.)
- Datapath blocks
- Programmable Logic (FPGA)
- Problematic / Dynamic circuits
- Standard cell and I/O libraries

## Conclusion

With the increasing complexity and the importance of memories in modern ICs, there is a clear need for specialized tools and techniques for the design and verification of embedded memory blocks. ESP brings formal technology into the memory designer's hands and raises their confidence in the quality of the design while simplifying the testing process and increasing overall verification productivity.