HyperFault

Mixed-Level Fault Simulator

HyperFault is a Verilog IEEE-1364-2001 compliant fault simulator that analyzes a test vectors' ability to detect faults. Supports mixed levels of gate, behavioral, and switch with SDF timing.

- **Verilog HDL IEEE 1364-2001 compliant fault simulator**
- **Uses standard Verilog source files and libraries for mixed-level fault simulation with gate, behavioral and switch devices**
- **Complements BIST and ATPG in finding interconnect faults**
- **Efficient multi-pass concurrent fault simulation algorithm with iterative fault collapsing gives optimal memory allocation and excellent runtime performance**
- **Automatic design partitioning supports distributed CPUs with load balancing for fast grading of large designs**
- **Fault grading models include stuck-at high/low output and input faults**
- **Full timing fault simulation supports SDF back annotation for post-route delay analysis**
- **Silvaco's strong encryption is available to protect valuable customer and third party intellectual property**

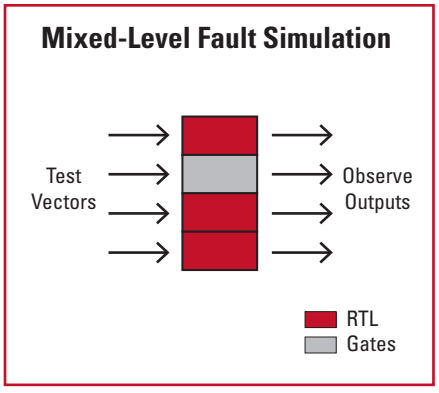
SILVACO

Accurate Fault Detection for Mission Critical Products

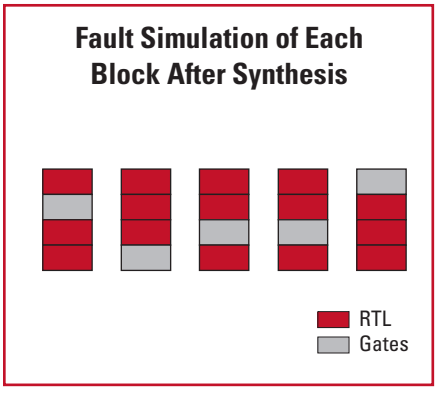
- Full timing fault simulation encompasses SDF back annotation for post-route delay analysis
- Accurate fault grading models include stuck-at high/low output and input faults
- Manageable fault simulation runtimes are achieved using fault sampling
- Random sampling algorithm provides for accurate fault grading
- Because there is no need to modify any design files or libraries, no new errors are introduced

Verilog Compliance

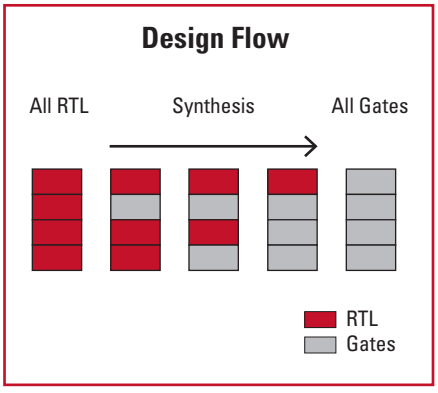
- IEEE 1364-2001 Verilog language compliant for designs, libraries and test benches
- Compliant Standard Delay Format (SDF) for back annotation
- Mixed-level fault simulation with gate, behavioral and switch devices
- Accepts standard Verilog cell, I/O, memory, and other IP libraries



Mixed level fault simulation analyses a gate level block within an RTL design.



Multiple designers can analyze their blocks independently.



Merged results yield system fault coverage.

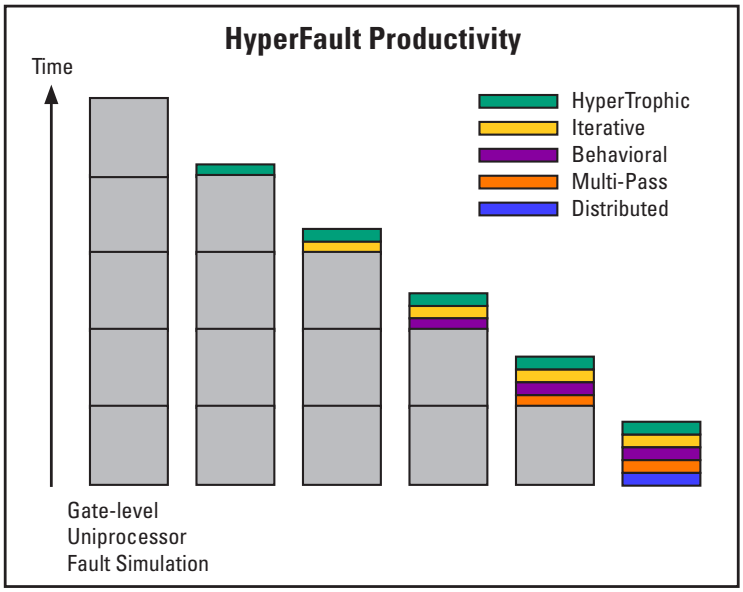
Ease of Use

- HyperFault is used by both design engineers and test engineers
- No need to modify any design files and possibly introduce errors
- No need to create or modify any library files
- Choice of intuitive graphic user interface, or Unix shell prompt for batch operation
- Regular expressions can be used to select faults from specific design blocks, or all of the faults in the device under test (DUT) can be selected
- Supports PLI driven test benches with appropriate code to support the multiple pass strategy of fault simulation
- Runs on Windows and Redhat Linux platforms

Scaleable Performance

- Automatic design partitioning supports distributed CPUs with load balancing for fast grading of large designs
- Iterative Fault Simulation accumulates fault coverage as successive test patterns are applied
- Distributed Fault Simulation divides up the fault simulation job among network processors for a linear decrease in fault simulation time (ten CPUs reduce fault simulation time by a factor of 10)
- Multi-Pass assures all faults will be processed in the memory available. Faults that do not "fit into memory" are deferred from fault simulation into a subsequent pass
- Large design simulations finish overnight instead of a week or more

Five HyperFault algorithms work together to drastically reduce simulation time.



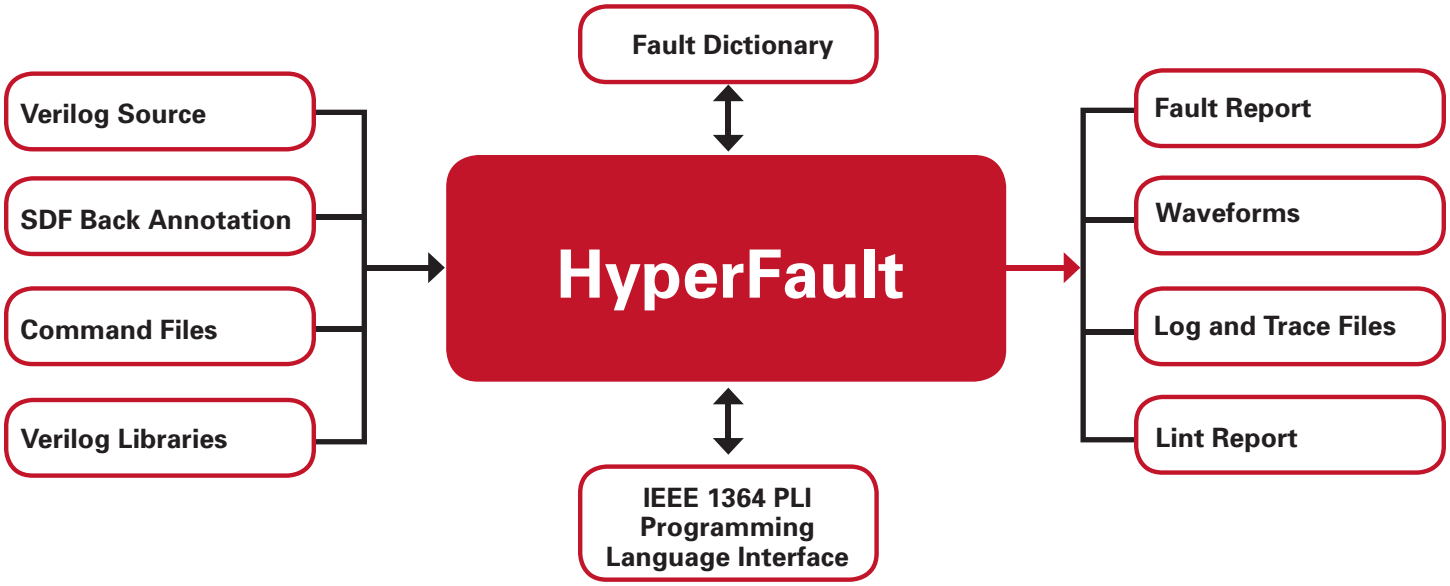
Fault Simulation Algorithms

- HyperTrophic fault simulation algorithm minimizes processing of fault effects—significantly reducing runtime by 10X
- Efficient multi-pass concurrent fault simulation algorithm with fault collapsing gives optimal memory allocation, excellent runtime performance, and accurate fault detection
- Value Change Dump (VCD) input of test vectors is supported
- Fault reporting is listed by instance in hierarchy for easy recognition
- Fault Injection can be scheduled after DUT is powered up, eliminating false detections

Applications for Fault Simulation When BIST and ATPG is Not Enough

- Critical paths hand coded for speed that cannot have scan chains inserted
- Designs for low power consumption that cannot tolerate additional gates
- Legacy designs that were not built with BIST or scan chains
- Asynchronous paths that are not static such as an input port that is also an asynchronous reset or an input used as both a data signal and clock
- When full post-route timing is required to correctly simulate fault behavior
- Feedback paths such as an internal ring oscillator, where inserting control to break the feedback loops would disturb the element balancing
- Libraries that do not support ATPG
- Cost of ATPG software is greater than cost of fault simulation software
- Finding interconnect faults in mission critical designs between individual IP blocks constructed with BIST and ATPG

HyperFault Inputs/Outputs



SILVACO

HEADQUARTERS
4701 Patrick Henry Drive, Bldg. 2
Santa Clara, CA 95054 USA
Phone: 408-567-1000
Fax: 408-496-6080

CALIFORNIA	sales@silvaco.com 408-567-1000
MASSACHUSETTS	masales@silvaco.com 978-323-7901
TEXAS	txsales@silvaco.com 512-418-2929

JAPAN	jpsales@silvaco.com
EUROPE	eusales@silvaco.com
KOREA	krsales@silvaco.com
TAIWAN	twsales@silvaco.com
SINGAPORE	sgsales@silvaco.com

WWW.SILVACO.COM

Rev 042513_14