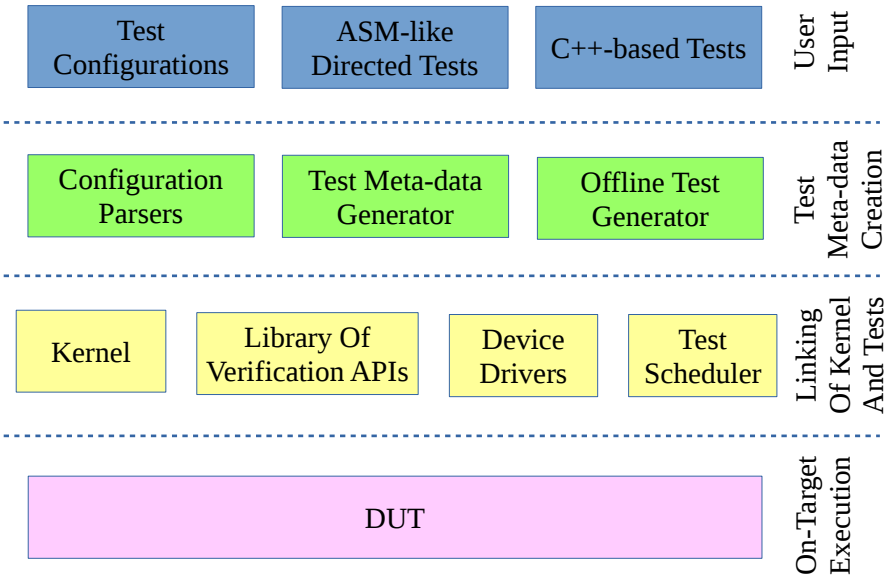
STING

PRODUCT DATASHEET

SOFTWARE SYSTEM FOR DESIGN VERIFICATION OF IP, CPU AND SoC IMPLEMENTATIONS

STING enables verification engineers to address the complexities and challenges involved in functional testing of the advanced SoC designs. It brings together a sofware stack of test generators, light weight device drivers of different classes of peripherals, result checkers and a micro-kernel into a single ready-to-use solution, which can be flexibly configured to create a portable bare-metal program as per the needs of the verification environmet.

Developed with a vision to solve problems commonly seen in design verification and system validation, STING embodies the best methodologies and practices in the industry whilst providing innovative solutions for the unique challenges in specific ecosystem. Designed for scalability and extensibility, companies can make full use of it across the spectrum of embedded, client and server SoCs.



STING tranforms user input into portable programs which can be executed seamlessly on any device under test (DUT)

LAYERED SOFTWARE ARCHITECTURE

STING can be viewed as an operating system (OS) specially designed to serve as a platform for the design verification of IP/SoC implementations. The kernel and library provides an execution environment for the functional tests bundled together with it. The tests comprises of constrained random, directed and coverage based stimulus for all the IPs included in testing.

Even though the intent is around functional testing, the software execution based methodology is very much like the real-world OS/applications, thus bridging the gap between verification and validation at large.

KEY FEATURES

Run the **same portable stimulus/program on simulation, emulation, FPGA prototypes and silicon** without any change

Stable and deterministic kernel with tiny memory and instruction footprints

Generate **extremely tight sequences of code** for faster closure on coverage

Configuration file based input to control kernel setup, test generation and execution

Support for **management of clock, power, memory and interrupt resources**

Extensive hardware support

Interspersed constrained random and directed testing for better coverage under different levels of stress

Support for standard verification scenarios available with the library of test stimulus

Architecture agnostic kernel/library APIs for building functional tests available to test developers

BENEFITS

Reduces redundancy by getting rid of duplicate engineering efforts put in each verification environment

Faster time-to-market by shift-left of system validation efforts and early enabling of software

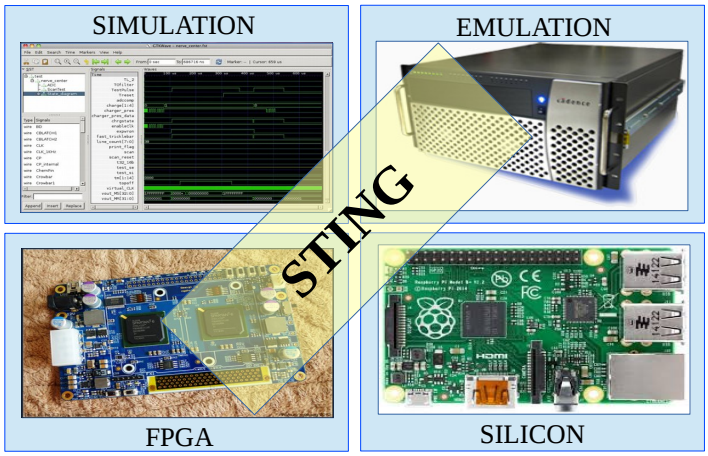
Promotes easy reuse of test stimulus across projects and verification environments

SHIFT-LEFT YOUR SOFTWARE BASED STIMULUS

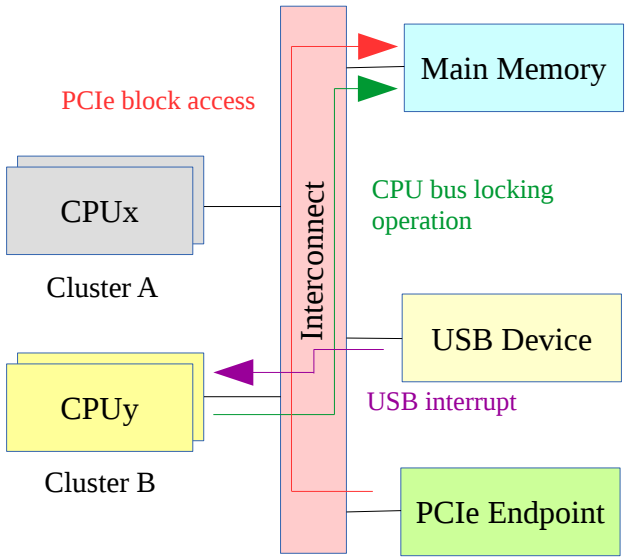
The tests generated by STING can be seamlessly executed across different verification environments (simulation, in-circut emulation, FPGA prototypes and silicon) without any external dependencies or code change.

The test stimulus gets enabled very early on the design and there is little or no effort for SoC teams to ramp up from one verification environment to another. Additionally, there is no need to maintain redundant and duplicate test stimulus for different environments.

Failing code sequences can also be easily migrated to simulations for higher visibility and faster debug.



Enable STING seamlessly and consistently across all verification environments



Programmed concurrent accesses flowing across the SoC during the test execution

CONCURRENCY TESTING SIMPLIFIED

STING provides coverage for all the IPs at the same time. Concurrent execution of traffic from masters (started with usecs of each other as a result of tight CPU scheduling) results in throttling of resources, maximum system stress and thorough testing of IP cross products.

Verification engineers write each scenario as a discrete unit. STING enables cross product of the same with other resulting into an exponential number of scenarios with very little effort from the users.

Complex scenarios which take a lot of time and effort to cover using hand written test can be easily generated.

HIGHLY INTUITIVE AND CONFIGURABLE TEST SPECIFICATION MECHANISM

STING supports rich test specification schemes which allows even complex scenario to get mapped to a test with sufficient ease which is then portable across all verification environments. It enables scenario driven test content development and reuse and automatic scaling of test stimulus across multiple SoC generations and heterogenous hardware configurations respectively.

Companies can retain existing legacy tests as well by easily integrating them within STING's software framework and get benefits of its execution methodology.

EXTREMELY FAST TEST GENERATION AND EXECUTION

STING test generators and kernel are highly optimized for extremely fast test generation and execution. It allows execution of maximum possible tests covering a large amount of validation space in very short time. For example, STING was able to get a throughput of close to 200 tests/sec on a ARMv8 based 16-CPU silicon for the cache coherency verification algorithm. As a result close to 17 million tests were executed on a single system in a day's time.

Focused test development framework and advanced scheduling schemes gives a tight control on generation of test stream and resources required by it. The resulting tight sequences of code ensures that the simulation/emulation cycles are efficiently used and a closure on coverage is achieved faster.

MULTIPLE MODES OF EXECUTION

STING supports multiple modes of execution for achieving the best test throughput for a particular verification environment.

The difference lies in the way the different components are stiched together to create the portable test program. For example, the offline mode of execution is for slow top-level simulations where only the test section is executed on target with setup and check done offline. Whereas for faster environments like FPGA and silicon, back to back tests are generated, executed and checked on the target.

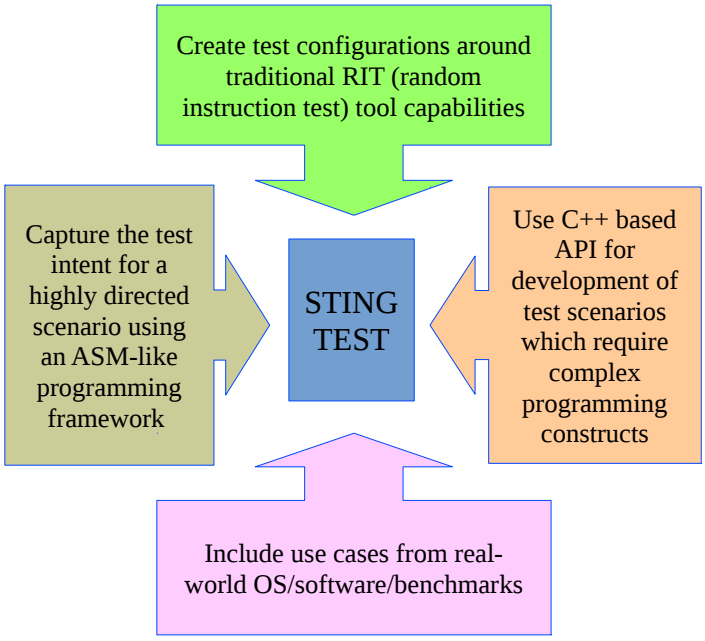
This improves the overall test efficiency by removal of operations from the execution flow which do not contribute to test stimulus in slower verification environments.

SUPPORT FOR DIFFERENT TEST METHODOLOGIES

STING supports different testing methodologies which included contrained random, directed, coverage-based and use-case based test stimulus generation.

In addition to the constrained random stimulus, the test is interspersed with directed and algorithmic tests. Intelligent resource sharing and constraint resolution allows all the different classes of stimulus to be present inside the test at once.

The versatile input specification mechanism also makes it easy to port benchmarks/standalone tests and implement real-world use cases in the test framework of STING.



Support for different test methodologies in STING

TEST STIMULUS ANALYSIS AND EVALUATION

STING implements lot of custom mechanisms to analysis and evaluation of test stimulus, which include methods based on log message, hardware performance counter etc. A novel hardware counter and interrupt based mechanism for analysis of run-time concurrency is in works which can be used for SoC characterization as well.

Coverage based feedback enables users to create high quality stimulus. Existing mechanisms can be extended easily to meet customer's requirements.

CHECKING AND DEBUG MECHANISMS

Several mechanisms that vastly improves the througput of failure disposition are implemented in STING. Checkers are ingrained at different stages of execution (both inside and outside the test) to catch any behavioral anomaly as quickly as possible.

Ability to start the test at a later point to enable faster failure reproduction, along with a host of debug features for test stimulus reduction and generation of verbose information around the failure scenario, makes it very easy for the verification engineers to reduces the time taken to triage any silicon issue.

ACCESS TO A LARGE LIBRARY OF TEST GENERATORS, DEVICE DRIVERS AND DIRECTED TESTS

A large library of test stimulus is available with STING. It provides a starting base for validation activities. The library includes different types of test generators, device drivers for peripherals and a large number of directed tests. The device driver framework allows integration of new devices easily.

Some of the examples are – (1) algorithmic tests for cache coherency checking, (2) directed tests for checking memory consistency and ordering, (3) device drivers for USB, SD, DMA controllers etc., (4) tests for branch prediction, (5) True and false sharing of cache-lines between different masters, (6) Use cases for memory management unit e.g virtual page size change, translation fault etc.

For the latest product information, write to us or visit our homepage

Registered Office:
Valtrix Technologies Pvt. Ltd.
#No 1462, 25th Cross,
HSR Layout Sector 2,
Bangalore – 560102, India

Sales and Support:
Phone: +91 9845735129
Email: contact@valtrix.in

Homepage:
www.valtrix.in

Valtrix SYSTEMS