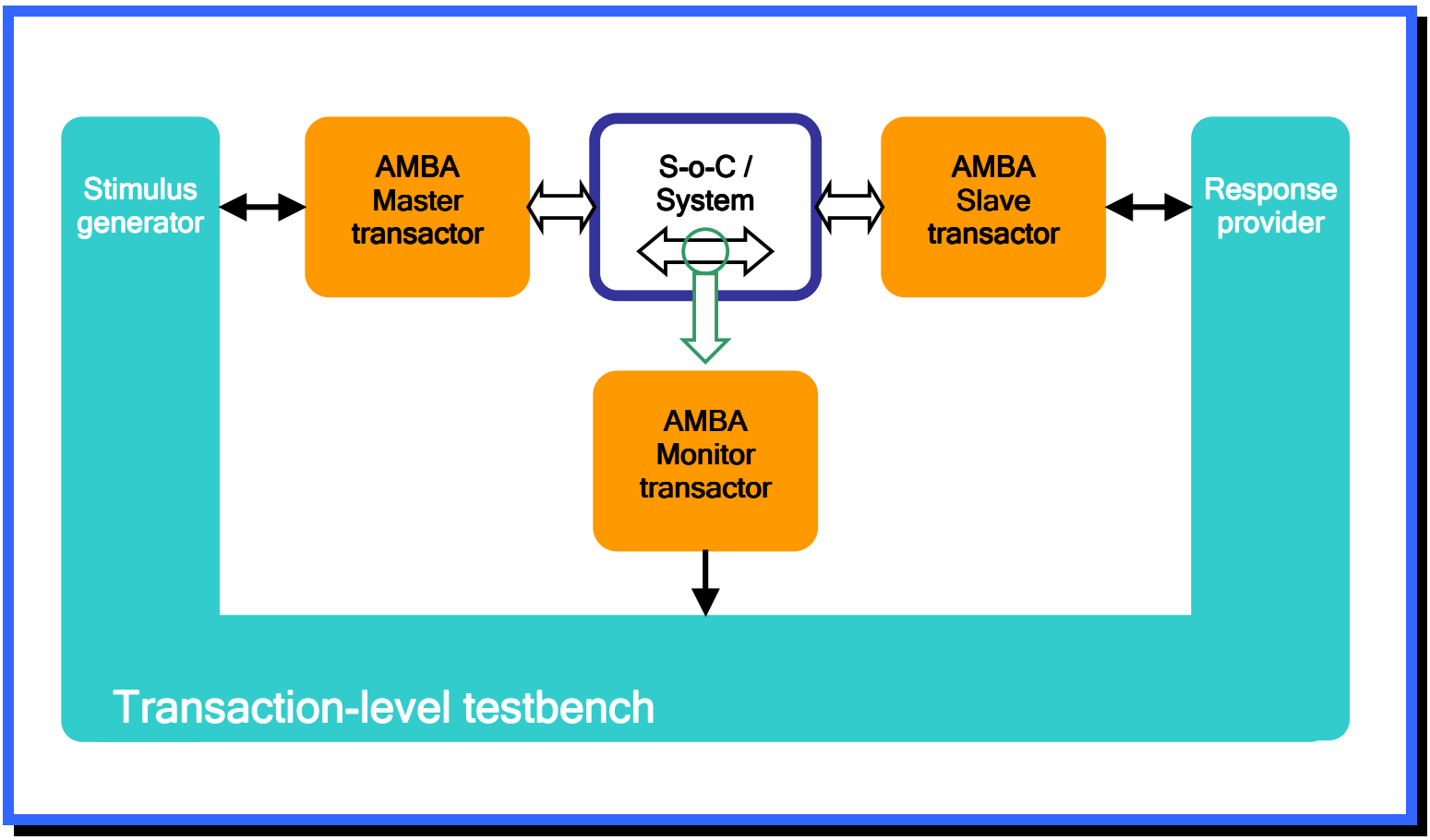
SDV Structured Design Verification



Verification IP for AMBA protocols

Focusing on test scenario's at the Transaction-level is a vital step to addressing the exploding cost of functional verification, yielding better testcases with less effort. Proven, re-useable SDV AMBA transactors connect transaction-level records to AMBA-compliant bus signals, while automatically detecting protocol violations.

- ***'ready-to-use' transactors, source licenses, tools & services***
- ***detects AMBA APB, AHB & AXI protocol violations***
- ***full range of verification methodologies***
- ***common interface and testcases from System to RTL-level***
- ***transaction recording in searchable database***
- ***automatic Functional Protocol Coverage***

Flexible support for your verification methodology

SDV's unique, flexible approach to Verification IP is not built on expensive per-simulation licenses of object-code BFM's, instead it fully supports your design methodology at every step of your functional verification process.

The key to the SDV approach is efficient source-code configuration in place of conventional object-code switches, because building just what you need yields the smallest, fastest interfaces with the most effective coverage statistics.

Ready-to-use object-code

Pre-configured, fully-featured transactors, supplied as object libraries, ready-to-use in your environment

Source-code licenses

SystemC or HDL source code for fully-featured transactors, ready to compile and instantiate in your testbench

Unique Transactor generation tools

The unique value of the SDV approach is derived from its ***TransactorWizard*** IP generation technology, that automatically creates transactors from Formal Protocol descriptions based on ***PSL***.

Many designs, including internal buses within S-o-C's, are configured to support limited protocol features (***AMBA-lite*** & ***Multi-layer AHB***), or they don't utilize the full range of Protocol features, for example, many AHB applications don't contain devices capable of requesting SPLIT transactions.

The unique ***TransactorWizard*** approach enables users to rapidly sub-set protocols, generating exactly the required Verification IP, with the bus width and endian charcteristics for their application.

Precise configuration ensures early detection of faults where IP blocks continue to use protocol features that should have been disabled, plus configuration guarantees meaningful Coverage statistics, and eliminates time wasted hunting false negatives.

IP generation services

SDV supports "black-box" compliance checking through the delivery of independently configured, validated IP to exactly match your application.

Universal Transaction-level interfacing

The unique ***TransactorWizard*** UTLI ensures easy integration within self-checking, golden-model, adaptive, response-checker based, pseudo-random traffic and other popular verification methodologies.

Unlike other BFM's the ultra-flexible UTLI fits your methodology without change, because its universal interfacing allows you to instantiate SDV transactors with the interface to exactly match your application.

Reference methodologies

The ***TransactorWizard*** UTLI supports reference design flows including the Cadence Incisive platform.

"Push" or "Pull" interfacing

Choose the simplicity of doTransaction() calls ("push") or the responsiveness of callback ("pull") interfacing.

Seemless re-use strategy

Derive from an abstract class for simplicity, connect (bind) stand-alone components to enhance re-use, or exploit flexible callbacks within registerable Publisher/Subscriber schemes.

"Plug & Play" architectural investigation

Common interface 'look & feel' simplifies switching between protocol styles, supporting easy architectural investigation.

The transactor trio

Master and Slave transactors inherently check protocol compliance without need for separate monitors in most simulations.

Master transactor

Choice of highly responsive "pull" interfacing or FIFO "push" interfacing for pipelined AHB & AXI protocols.

Sample pseudo-random traffic generators

Easy integration with directed and adaptive test strategies

Slave transactor

Recognition of any legal transaction in any order

Universal "response provider" interface for emulation of status / feedback systems

Source code for simple RAM/ROM emulation.

Monitor transactor

Detects protocol violations

Recognition and recording of any legal transaction in any order, including coverage recording

Reconstructs AHB burst-mode transactions for recording, including undefined length INCR mode transactions

Flexible application support & environment

General Configuration for AMBA

Bus width

Cycle-accurate or timed signal setting

Big / little-endian operation

APB protocol

AMBA 2.0 compliance with configurable support for PREADY signal

AHB protocol

AMBA 2.0 compliance

Automatic bus-requesting

Optional automatic reconstruction of burst-mode transactions on loss of HGRANT

Master configurable for burst termination on error

Easy sub-set of protocol to disable SPLIT / RETRY operation for multi-layer & AHB-lite protocols

AXI protocol

Multi-transaction concurrency

Out-of-order processing, including support for, and width of, ID signals

Control of read and write transfer interleaving

Simulation environments

SDV AMBA transactors operate in a wide variety of simulation environments

SystemC cycle-accurate simulations

System-level & architectural simulation

HDL model with C++ testbench

SystemC Verification Extensions (SCV)

Testbuilder 1.3

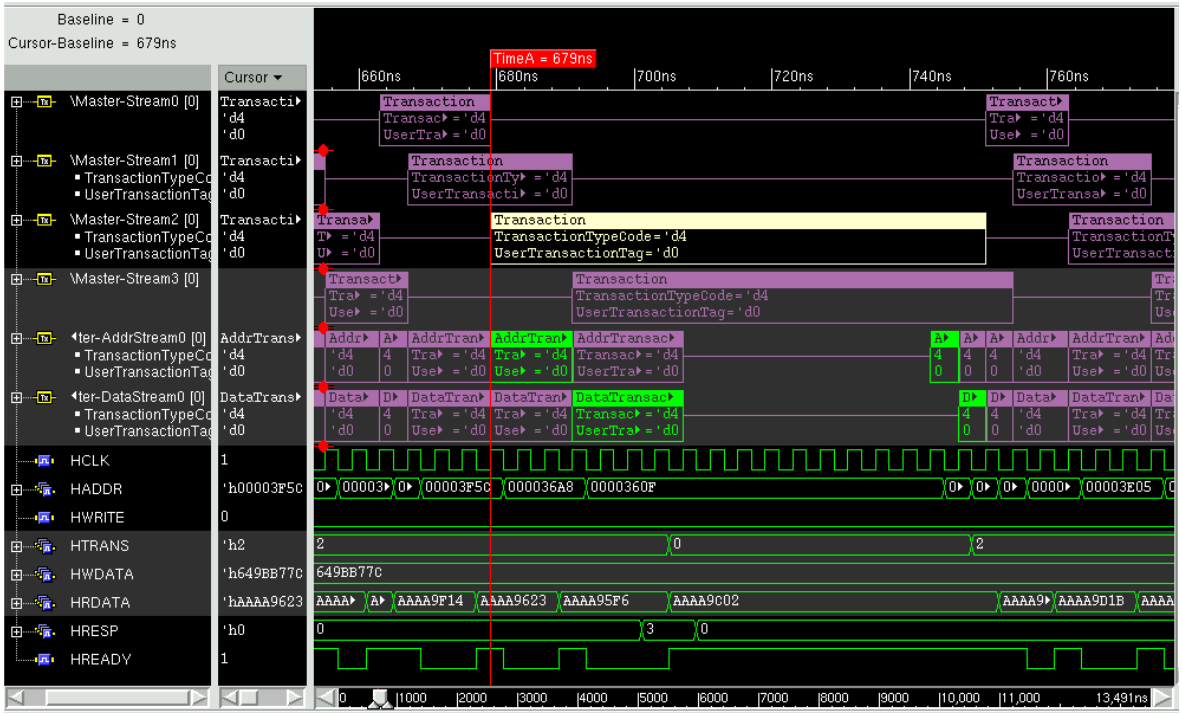
Easy adaptation to custom C++ environments

HDL model with other HVL

HDL transactors interfaced with HVL's such as OpenVera and Specman / 'e'

HDL simulation

HDL transactors interfaced directly with HDL testbenches



Example of AMBA-bus 'SPLIT' transaction record with interface waveform

Automatic Functional Protocol Coverage

Automatic coverage recoding without additional programming or API's confirms how effectively interface protocols have been exercised in simulation.

Reporting is by color text PDF file, or using a results navigation GUI

Transaction Coverage

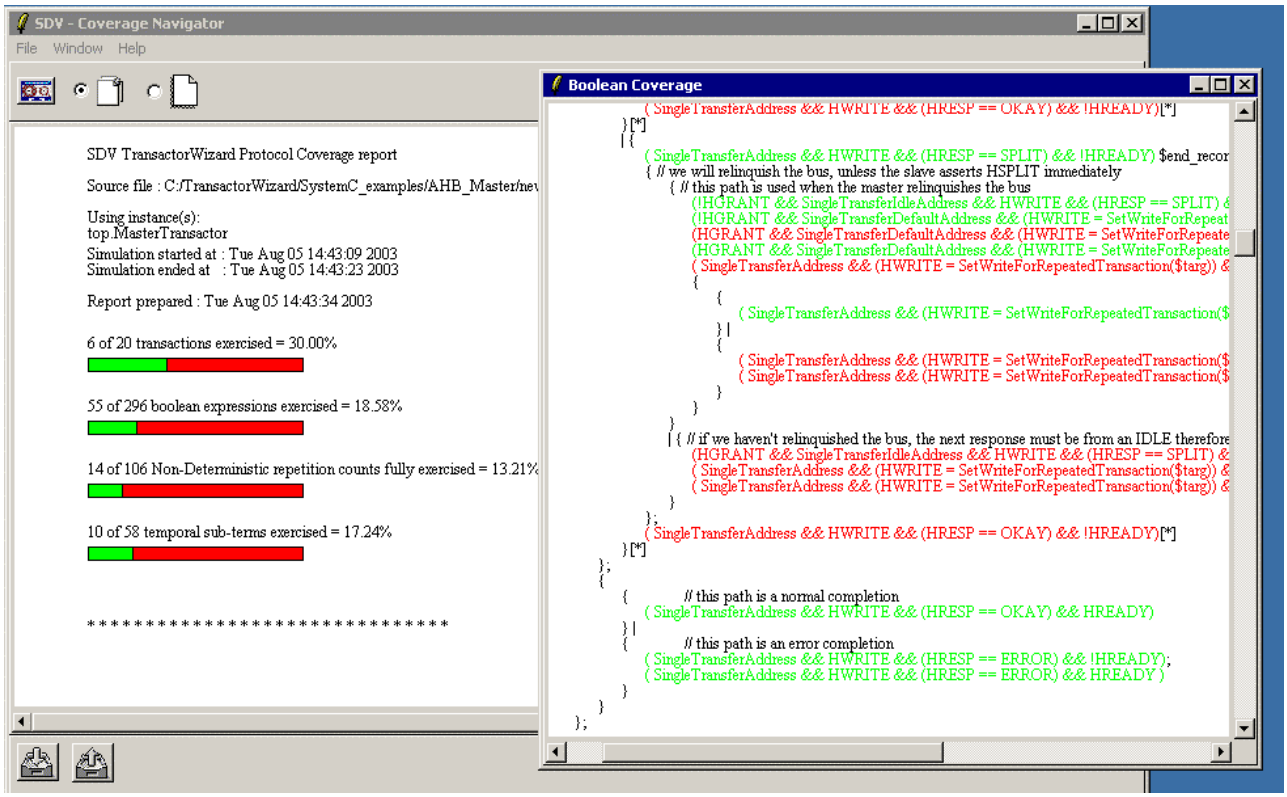
Confirms all possible transactions have been exercised

Boolean Coverage

Confirms all stages in the PSL/Sugar Protocol description has been exercised (Equivalent to FSM state coverage)

Transition Coverage

Confirms all possible pairs of sequences have been exercised, for example that a SPLIT transaction has been followed by both OKAY & ERROR completions



Example coverage report for an AMBA-bus simulation

SDV Inc.
Austin TX

Tel : **(512) 431-5126**

EMAIL : **info@sdvinc.com**

April 2005