

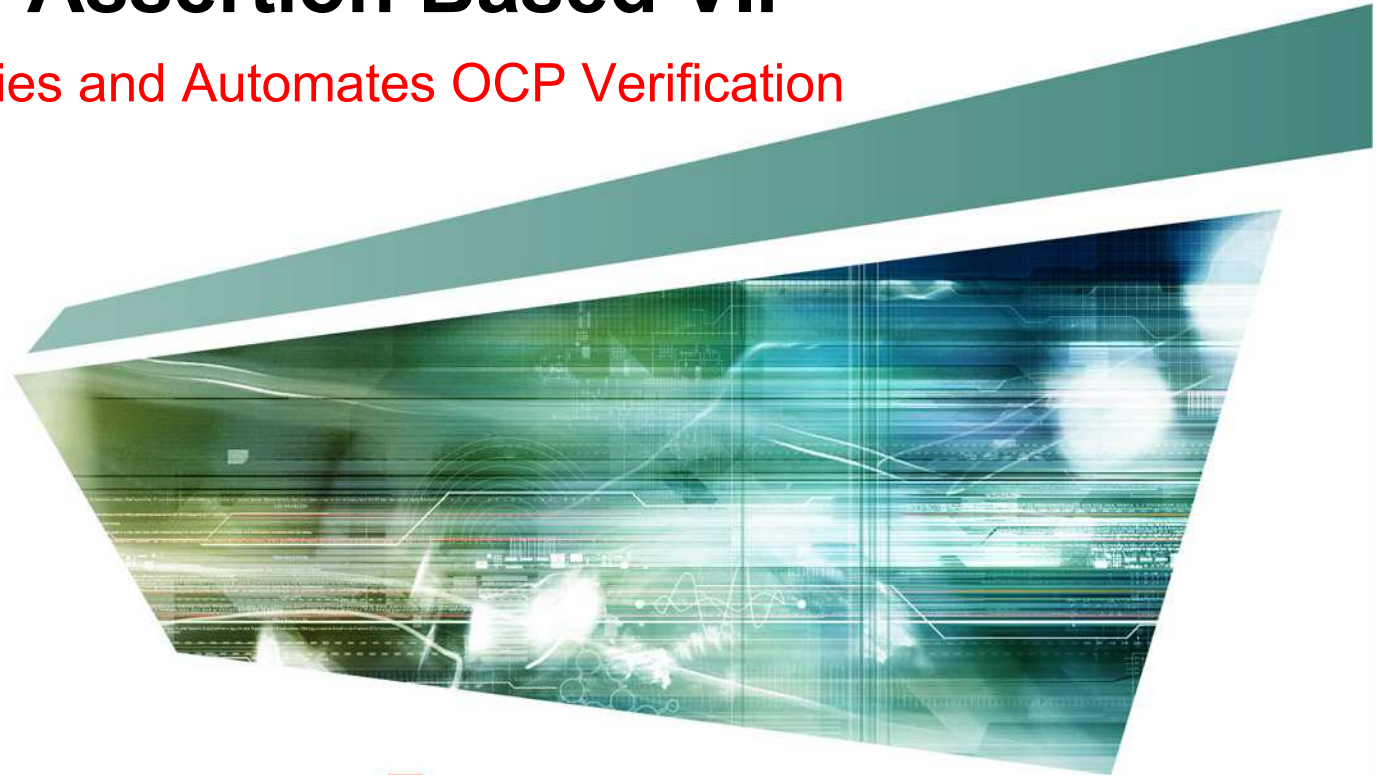
INVENTIVE

cādence™

CONFIDENTIAL

OCP Assertion Based VIP

Simplifies and Automates OCP Verification





Agenda

- Assertion Based VIP Introduction
- Applying ABVIP to Formal Analysis
- ABVIP Status
- OCP ABVIP Applications



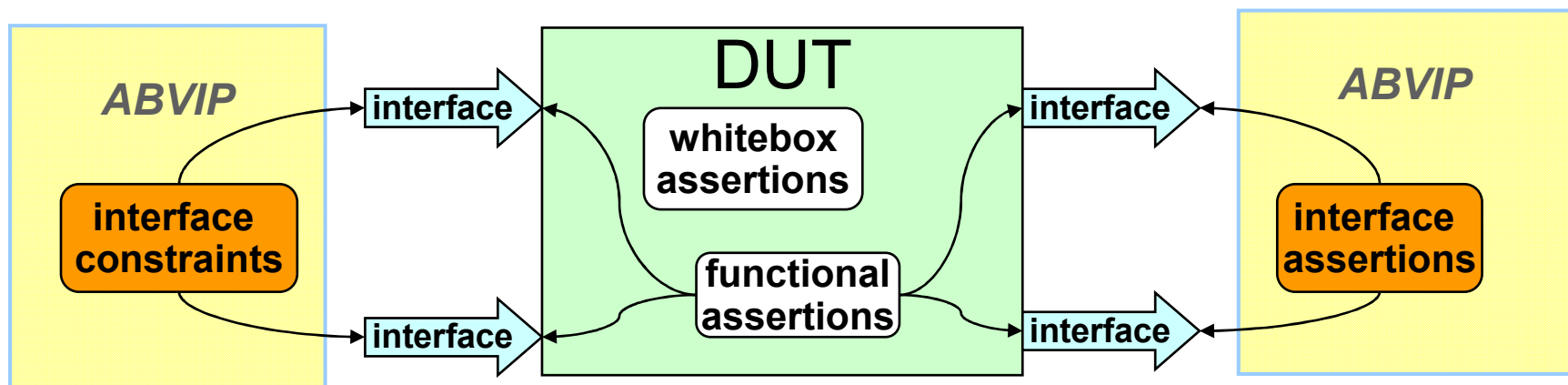
Introduction to Assertion Based VIP (ABVIP)

- Increases quality, predictability, productivity
- Supports Incisive assertion based verification engines
 - IFV
 - IUS
 - Acceleration (planned)
- Available for OCP and AMBA (AHB and AXI)
- Ideal for interface verification
- Automated and easy to use: requires no testbench or stimulus generation
- Customer proven: over 200 projects completed*

* As of Q4'2007

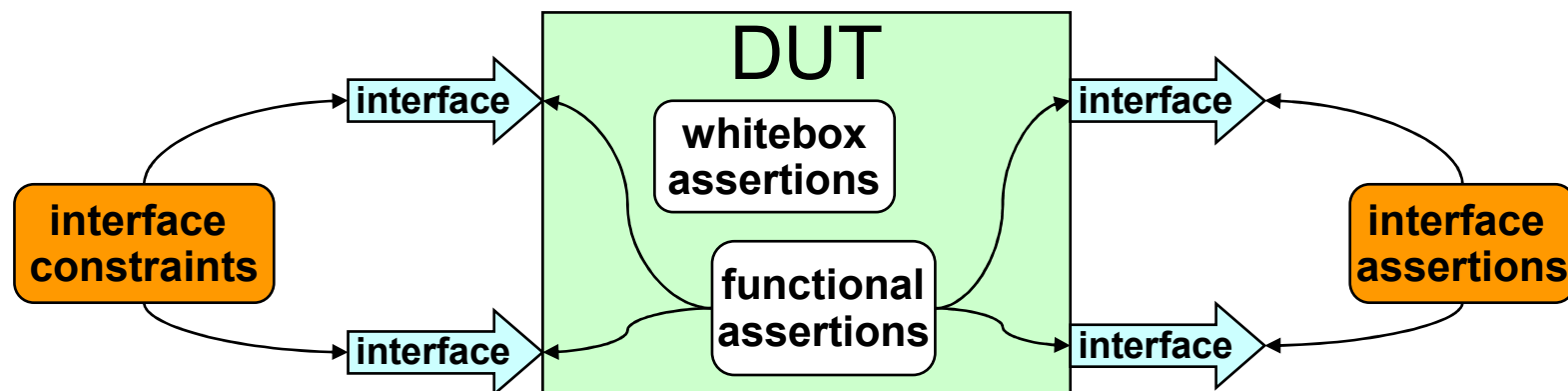
Assertion Based VIP (ABVIP)

- Pre-validated interface properties automate protocols interface verification
 - Fast bring-up
 - Minimizes debug of missing constraints
 - Quick sanity checks and coverage points
- Ensures interface consistency between blocks
- Self-consistent assertions eliminate false-negatives
- Performance optimized properties ensure rapid results



Formal Analysis Backgrounder

- Properties express desired behavior of DUT
 - Termed *assertions*
- Properties also express behavior of environment
 - Termed *constraints*
- Objective: Formal exhaustively verifies assertions in context of constraints
- Comment: Developing properties is single biggest task
 - Especially around the interfaces





Formal Analysis Definitions

- Properties express desired behavior of DUT
 - Termed *assertions*
- Properties also express behavior of environment
 - Termed *constraints*
- Objective: Exhaustively verify assertions in context of constraints.
- Comment: Developing properties is single biggest task
 - Especially around the interfaces



Assertion Based VIP Key Features

- Full complement of *self-consistent* properties
 - Formal methodology ensures self-consistency
 - No false-negatives due to insufficient or incorrect properties
- Full set of coverage points and debug checks
 - Detect subtle design and environment issues
- Performance optimized properties
 - Ensures best quality of results in formal analysis
 - Parameterization enables scaling to larger designs
- Incisive Formal methodology compliant
- Incisive Formal methodology aware documentation



ABVIP Status

- All ABVIP is in production
- Extensive track record of customer success
 - Over 200 projects completed*

Protocol	Status	Projects Completed*
AHB	Production	70+
AXI	Production	35+
OCP	Production	120+

* As of Q4'07



OCP Checker Features

- Supports Incisive Assertion Based Verification flows
- All assertions derived from OCP-SIG FVWG OCP 2.0 properties proposal
- Supports OCP 2.0 Configuration file for checker generation
- Generate complete formal environment including vunit and Verilog/VHDL top-level (optional)
- Functional coverage points for the OCP 2.0 protocol
- Verilog, VHDL or mixed language



OCP Monitor Features

- Number of checks depends on OCP implementation
 - Defined in OCP rtl.conf file
- Compliant to OCP-SIG FVWG Protocol Compliance Checklist
- Compliance Checks:
 - All Signal-level checks, including those for holding and retraction.
 - 16 Phase-level checks, including checks for Byte Enables
 - 28 Transaction-level checks, including checks for Burst Invariance
 - 12 Transfer-level checks, including tracking across threads and phase ordering
 - 12 sideband checks
- Coverage and Latency Checks:
 - 22 Latency checks for performance analysis
 - 105+ Debug cover checks
 - 90+ Functional cover checks

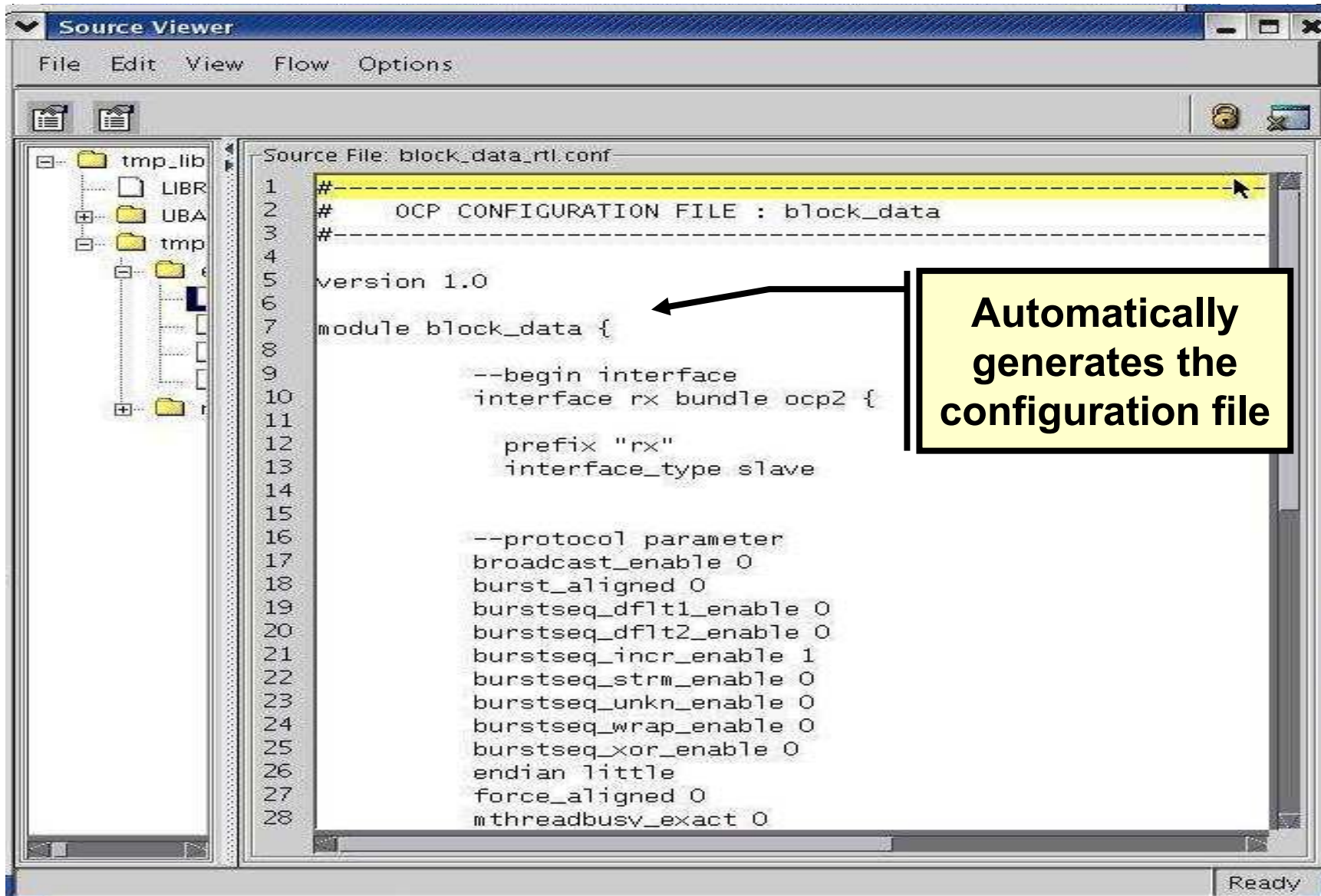


OCP Auto-Configurator

Automates configuration file creation

- Creates RTL Configuration file from scratch using GUI interface
- Builds environment to run Formal Analysis using OCP ABVIP
- Default parameter values of RTL Configuration file can either be:
 - loaded from existing Configuration file
 - set to OCP Specification values
- Provides checks to detect invalid values and invalid parameter combinations
- Implements “Configuration Compliance Checks” from OCP 2.2 Specification
- Detailed Help provided for each parameter
- Automatically bind physical ports to OCP logical nets (optional)

Auto-Configurator Example



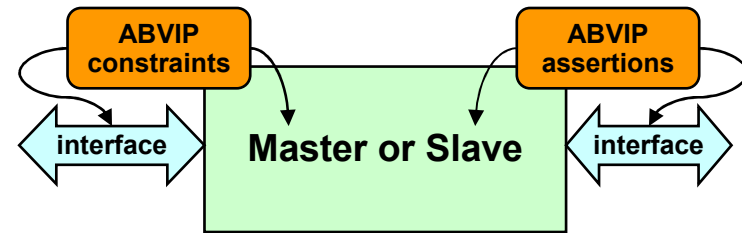
Source File: block_data_rtl.conf

```
1 #-----
2 #   OCP CONFIGURATION FILE : block_data
3 #-----
4
5 version 1.0
6
7 module block_data {
8
9     --begin interface
10    interface rx bundle ocp2 {
11
12        prefix "rx"
13        interface_type slave
14
15
16        --protocol parameter
17        broadcast_enable 0
18        burst_aligned 0
19        burstseq_dflt1_enable 0
20        burstseq_dflt2_enable 0
21        burstseq_incr_enable 1
22        burstseq_strm_enable 0
23        burstseq_unkn_enable 0
24        burstseq_wrap_enable 0
25        burstseq_xor_enable 0
26        endian little
27        force_aligned 0
28        mthreadbusv_exact 0
```

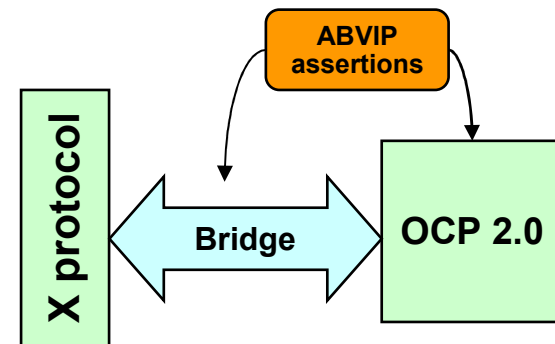
Automatically generates the configuration file

OCP ABVIP Application Examples

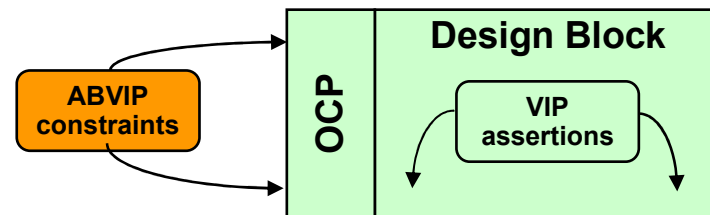
- Protocol Compliance Verification



- Protocol Conversion (bridge) Verification



- ABVIP constraints verify other functionality





OCP ABVIP Roadmap

OCP 2.1 and 2.2 functionality planned for 2008

Class	Functionality	Early Access Availability
2.1 Generator Support	Tags, tag_interleave_size, taginorder parameters	Now
2.1 Property Support	Tags, tag_interleave_size, taginorder	Now
2.2 Generator Support	Clk_enable, modified threadbusy behavior, 2-dimensional bursts parameters	Apr-08
2.2 Property Support	Clk_enable, modified threadbusy behavior, 2-dimensional bursts	Apr-08



Summary

- ABVIP increases quality, predictability, productivity
- Supports Incisive assertion based verification engines
- Available for OCP and AMBA (AHB and AXI)
- Ideal for interface verification
- Automated and easy to use: requires no testbench or stimulus generation
- Customer proven with over 200 completed projects



cadence[®]

