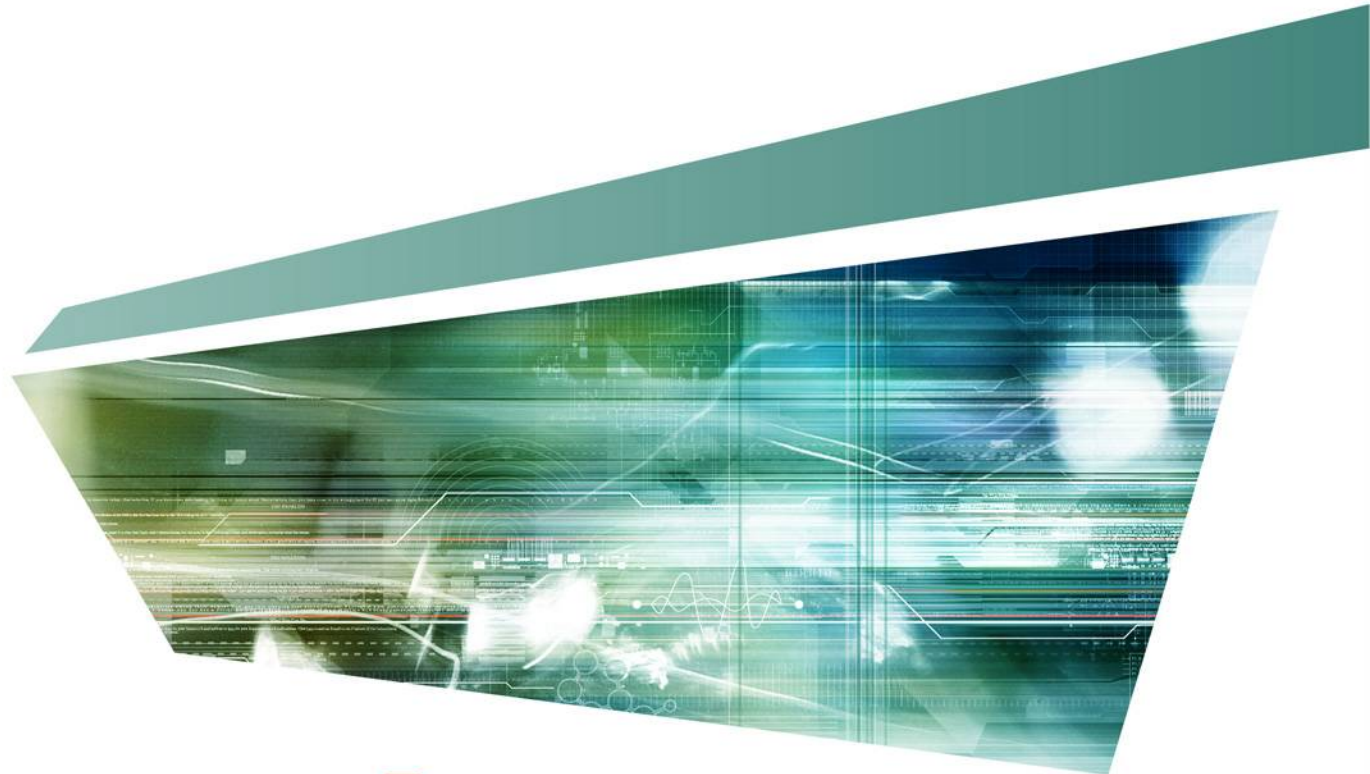


Incisive OCP Verification Family

2009



Agenda

- Cadence Verification IP Strategy
- OCP Protocol and OCP-IP Consortium
- OCP UVC Overview and Differentiation
- CMS Overview
- OCP UVC Technical Details
- Deliverables
- OCP UVC Applications and Usage
- Summary

Incisive VIP Strategy

Cadence VIP spans full metric-driven verification process

Compliance Management System (CMS)

UVC

OVM Interface | e Interface

Advanced Testbench Core

Assertion Based VIP

Transaction Based Acceleration

SpeedBridge™



- OVM Compliant
- Production proven in over 2,000 projects
- Supports full spectrum of verification needs
- Available for most demanded protocols

ARM
AHB

ARM
AXI

PCI
EXPRESS™

mipi™ **New!**
mobile industry
processor interface

Ethernet



OCP
International Partners

SERIAL
ATA

PCI
CONVENTIONAL

- Unique Compliance Management System (CMS) automates compliance verification

Market Leading OCP VIP

Cadence uniquely provides assertion based and testbench VIP

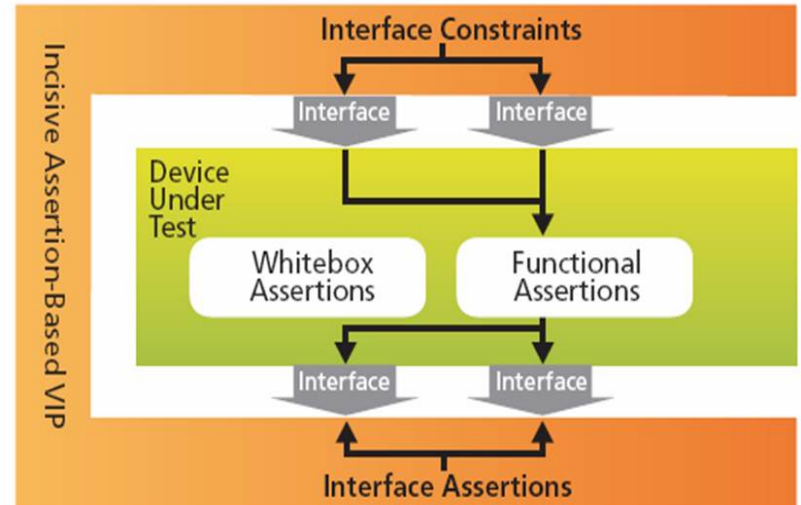


Testbench VIP (UVC)

- Compliant to OCP 2.2 now, 3.0 coming
- Mature, proven on over 270 verification projects*
- Metric-driven verification for SystemVerilog and/or *e*
 - CMS coming 1H'09
 - Functional Coverage Guidelines included
 - Predefined OCP Sequences/Tasks with user selectable randomization levels



- **Assertion Based VIP (ABVIP)**
- Compliant to OCP 2.2 now, 3.0 coming
- Mature, proven on over 150 projects*
- Easy to use
 - No testbench or stimuli generation required
 - Push button config file generation with unique Auto-Configurator
- Optimized for formal verification with IFV
- Rapid block level verification




* As of Q4'08



Cadence's Role in OCP-IP

- Cadence is OCP-IP Sponsor Member
- Cadence chairs the compliance Functional Verification Working Group



OCP UVC Overview and Differentiation

UVC: Much More Than a Simple BFM

***Generates* Constrained
Random Stimulus**

UVC: Much More Than a Simple BFM

***Generates* Constrained
Random Stimulus**

***Checks* for
Protocol Compliance**

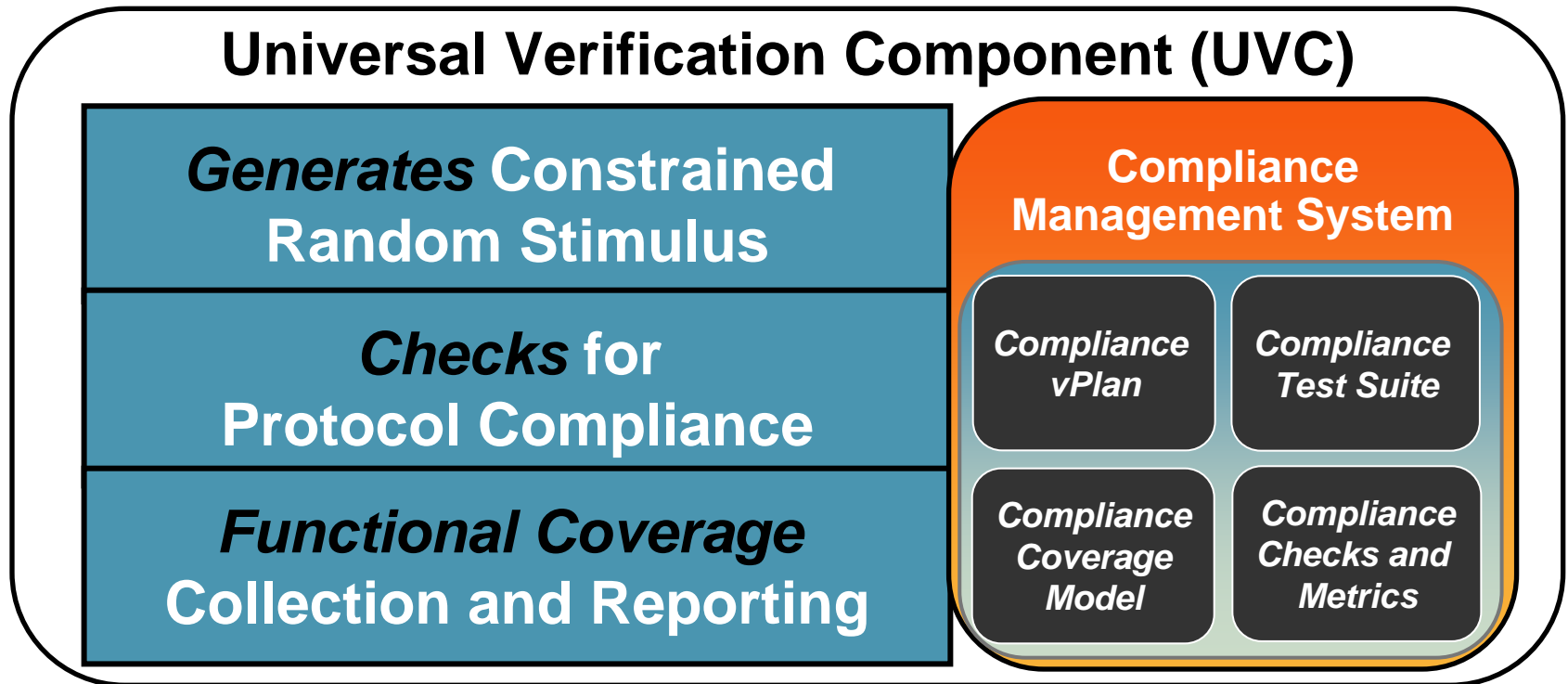
UVC: Much More Than a Simple BFM

***Generates* Constrained
Random Stimulus**

***Checks* for
Protocol Compliance**

***Functional Coverage*
Collection and Reporting**

UVC: Much More Than a Simple BFM



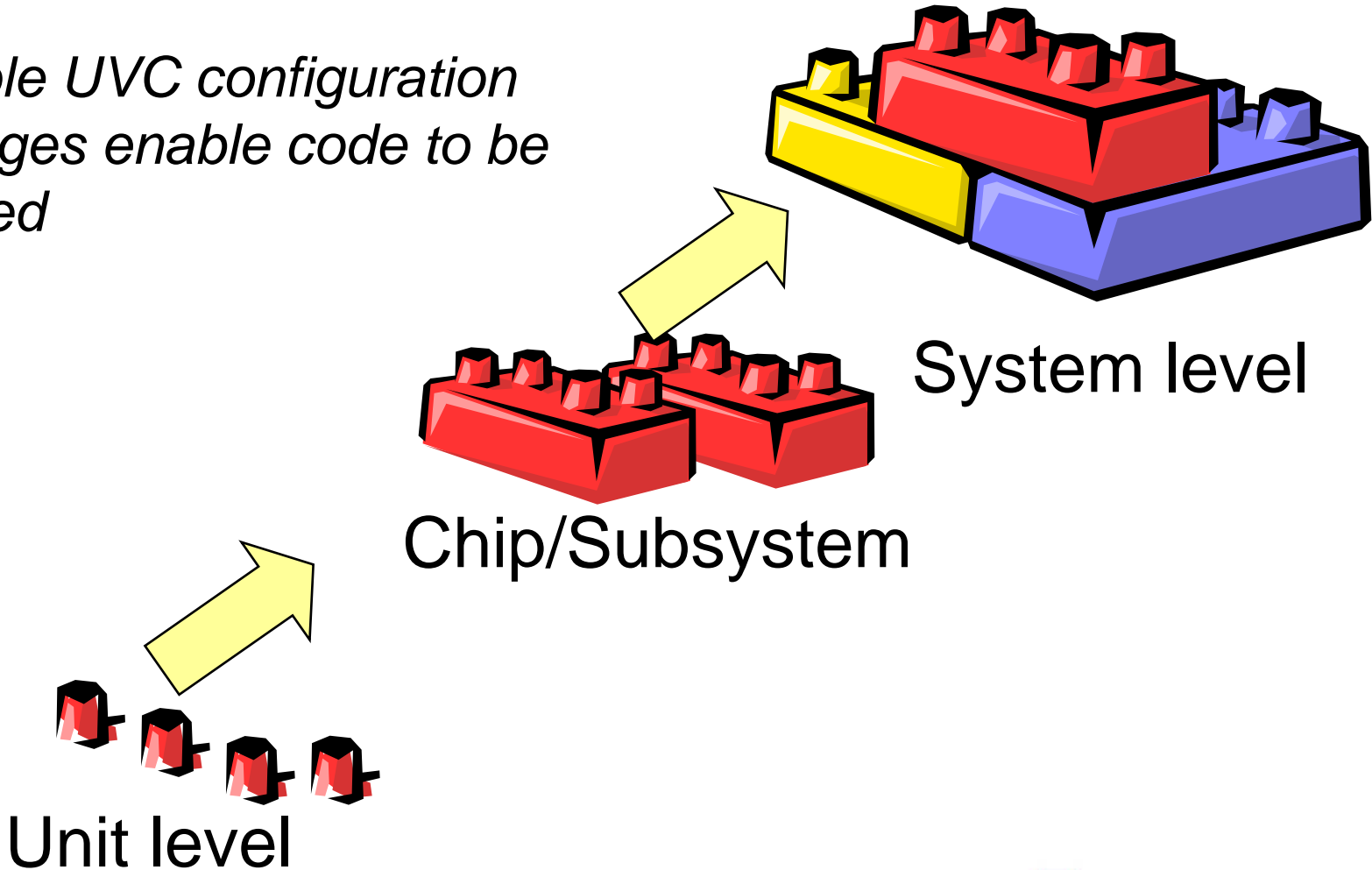
- Reusable from block to chip to system level
- URM and OVM compliant
- Applicable to both Design IP and SOC verification
- CMS included at *no* additional charge

Cadence UVC Advantages

- Maximizes **quality**
 - uncover more bugs, faster
- Maximizes **predictability**
 - CMS provides metrics to determine completion status
- Maximizes **productivity**
 - Save weeks of environment bringup time
 - Focus on the most important verification issues
 - Fully reusable from module level to system level verification
 - OVM compliant to ensure interoperability
 - Reduces protocol expertise required

Cadence UVCs Architected for Reuse

Simple UVC configuration changes enable code to be reused



OCP UVC Differentiation

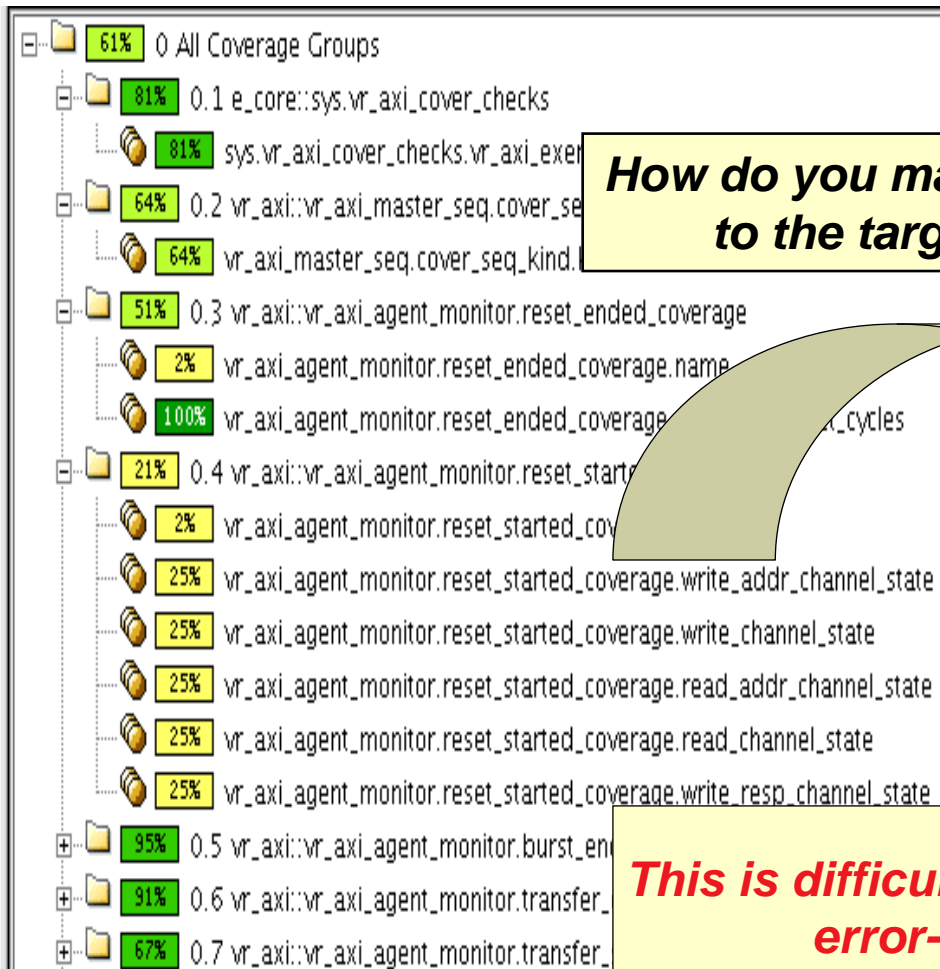
- Multi-language OVM compliant
 - Supports SystemVerilog and *e* testbenches
- Automates Protocol Compliance Verification with Compliance Management System (CMS*)
 - Achieves high functional coverage from the start
 - Shortens the time to compliance coverage closure
- Provides Metric Driven Verification
 - Supplies relevant metrics
 - Automates reporting of status versus metrics
- Drives and verifies all OCP 2.2 features

* CMS for OCP Available 1H'09



CMS Overview

Without Compliance vPlan, Difficult to Interpret Coverage (AXI example)



AMBA AXI Protocol Specification

How do you map collected coverage to the target specification?

1.1	About the AXI protocol
	Architecture
1.2	Basic transactions
1.4	Additional features

Signal Descriptions

2.1	Global signals
2.2	Write address channel signals .
2.3	Write data channel signals
2.4	Write response channel signals
	Read data channel signals .
	Read response channel signals
	Interface signals

This is difficult, time-consuming, and error-prone process!

Coverage Assessment and Analysis with vPlan

1 AXI Interface vPlan

Details: The AXI protocol is a burst based protocol. The address along with information on the nature of the data is passed via a read or write address channel from the master to the slave. The data is transferred either on the write channel from the master to the slave or on the read channel from the slave to the master. In write transaction where all data flows from the master to the slave, there is an additional channel, the write response channel which allows the slave to signal the master on the completion and status of the transaction. The AXI allows for several transactions to take place in parallel using the id tag mechanism.

Required Samples: 10

1.1 Data Items

Details: Related vsif: vr_axi_data_items_test_suite.vsif
This section is related to the data items on the bus. The coverage that is been collected in this section relate to the burst fields and the relationships between them.

1.1.1 Burst Attributes

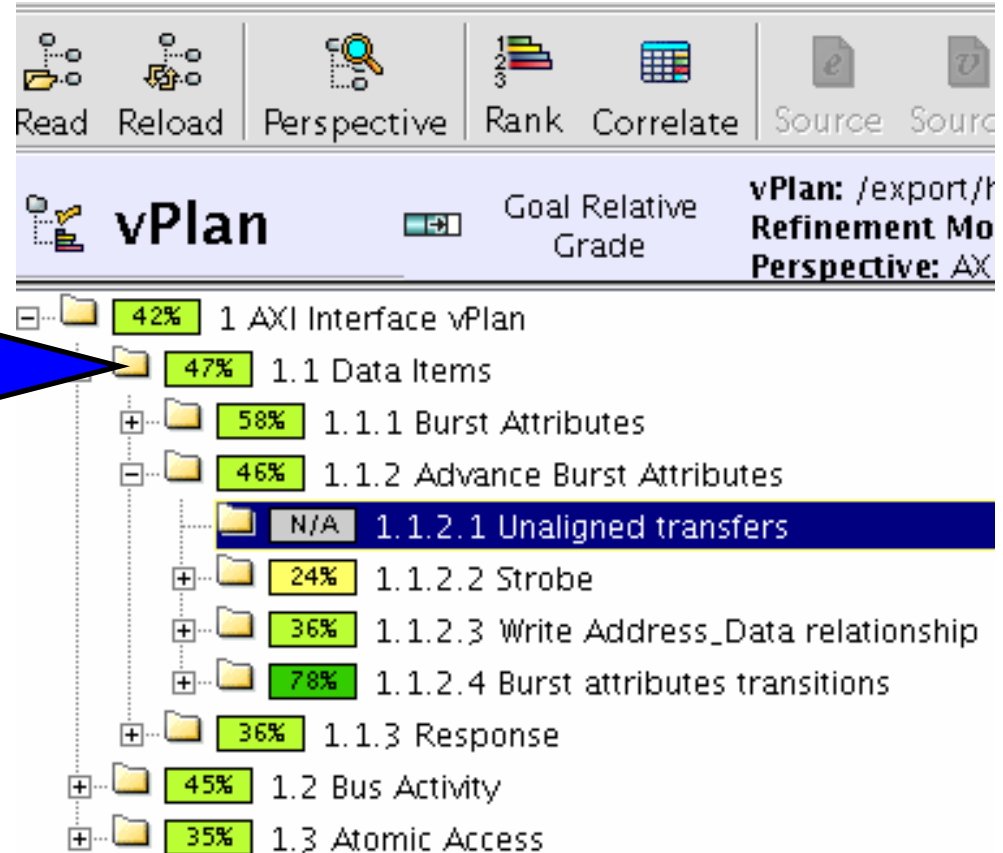
Details: Related vsif: vr_axi_burst_attributes_test_suite.vsif
This section collect coverage on basic burst attributes.

1.1.1.1 NORMAL Access Burst Attributes

Details: Related vsif: vr_axi_burst_attributes_normal_access_test_suite.vsif
This section collect coverage on basic burst attributes using only "normal access" bursts.

1.1.1.1.1 FIXED Burst Attributes

Name	Item Pattern	Bucket Filter	Parameters
Burst Id Tag	vr_axi_agent_monitor_burst_end ed_cross_access_kind_id	(access==NORMAL and kind==FIXED)	
Burst Length	vr_axi_agent_monitor_burst_end ed_cross_access_kind_length	(access==NORMAL and kind==FIXED)	
Burst Size	vr_axi_agent_monitor_burst_end ed_cross_access_kind_size	(access==NORMAL and kind==FIXED)	



Easy Corner Case Identification

Verification Plan Tree (Sessions*)

File Edit View Analysis Refinement Filter Options

Export Info Views Runs Report

Reload a verification plan (vplan) file

Read Reload Perspective Rank Correlate Source Source Buckets Holes Project

vPlan Goal Relative Grade

vPlan: /export/home/dimitry/eVC/vr_ahb_2.4
Refinement Mode: local
Perspective: AHB

- 11% 2 AHB
 - 9% 2.1 AHB_instance_M0
 - 9% 2.1.1 Master Compliance VPlan
 - 5% 2.1.1.1 01 First transfer responses
 - 6% Zero wait state OKAY response on the first transfer
 - 39% vr_ahb_agent_monitor.tr_ended_master_cover(per_type) d**
 - 6% vr_ahb_agent_monitor.tr_ended_master_cover(master_nar
 - 31% vr_ahb_agent_monitor.tr_ended_master_cover(master_nar
 - 30% vr_ahb_agent_monitor.tr_ended_master_cover(master_nar
 - 0% vr_ahb_agent_monitor.tr_ended_master_cover(master_nar

Bucket Analysis [0]

File Edit View Analysis Options

Prev Next Export Buckets Holes Project Lock Close

Buckets of Item:
vr_ahb_agent_monitor.tr_ended_master_cover(per_type).cross_transfer_res
ponses_and_wait

At least : 1

Buckets Table

direction	burst_k	wait_state	Count	Tests
READ	SINGLE		719	1
READ	INCR		16	1
READ	WRAP4		5	1
READ	INCR4	FIRST	0	0
READ	WRAP8	FIRST	1	1
READ	INCR8	FIRST	2	1
READ	WRAP16	FIRST	4	1
READ	INCR16	FIRST	0	0
WRITE	SINGLE	FIRST	334	1
WRITE	INCR	FIRST	21	1
WRITE	WRAP4	FIRST	3	1
WRITE	INCR4	FIRST	1	1
WRITE	WRAP8	FIRST	0	0
WRITE	INCR8	FIRST	0	0
WRITE	WRAP16	FIRST	6	1
WRITE	INCR16	FIRST	1	1

Missing test scenarios

cadence

CMS Summary

Compliance Management System Advantages and Benefits

- Easy to Use, verification plan and stimulus provided
- Automates protocol compliance verification for IP and SOC verification
- Maximizes controllability; stimuli generation controllable at any granularity
 - Full specification
 - Per section
 - Specific functionality
- Maximizes predictability
 - Provides visibility into effort required to achieve full compliance
 - Status is quantified and always available
- Maximizes productivity
 - Eliminates writing and maintaining large numbers of tests
 - Generates unanticipated test scenarios
 - Enables parallel verification on multiple protocol sections
 - Provides language-neutral user interface
 - Minimizes protocol knowledge expertise required
- Maximizes management's visibility, control and communication
 - Clear, metrics-based progress measurement
 - Automatic status report generation



Addressing OCP Verification Challenges

OCP UVC Solves Major Verification Challenges

- OCP UVC implements a flexible verification environment
 - Scalable and configurable architecture
 - Modular test cases ranging from top level scenarios to individual transactions
- Covers multiple abstraction layers
- Measures functional coverage and system performance
- Checks interface compliance with the protocol
- Provides a reusable reference model

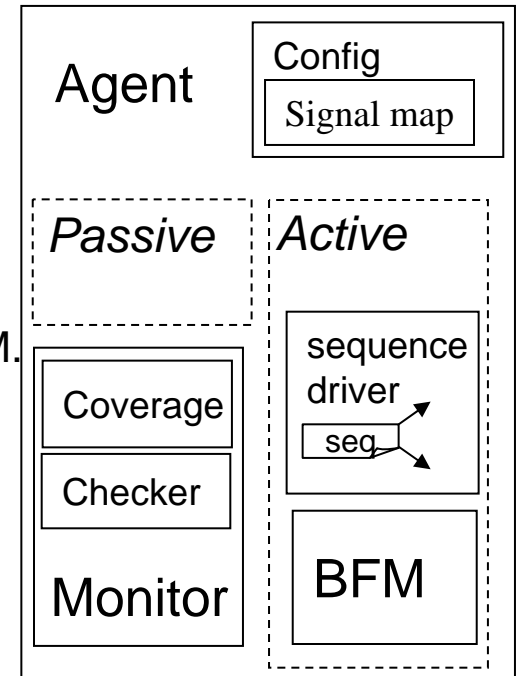
OCP UVC Functionality

Much more than a simple BFM

- Automated Compliance Verification using CMS
- OVM compliant
 - Enables verification language independent block → chip → system level and project to project reuse
 - Fully and easily configurable to specific DUT needs
- Multi-language constrained-random generation (SV and e)
 - Interoperable with VHDL/Verilog devices, SV/SystemC models and all HDL simulator
- Metric-driven verification environment
 - Includes constrained-random generation, assertion checking and functional coverage
- Fully compliant to OCP specification
 - Multi-threading, new burst and new tagging models supported
- Easily configurable as Master, Slave or Monitor
 - Large quantity of configurable fields/parameters
 - Master can drive the full set of commands
 - Slave can react to all transactions
 - Monitor can log all the bus traffic

Agent Architecture

- Active/Passive subtype
 - Active agents reads and drives bus signals
 - Passive Only reads bus signals
- Active agents include :
 - BFM – drives the data item to the bus
 - Sequence driver – move data items from sequences to BFM.
 - Sequences – generate data items
 - Monitor – tracing, checking, logging, and coverage
- Passive agents include monitor



Built In Metric Driven Verification

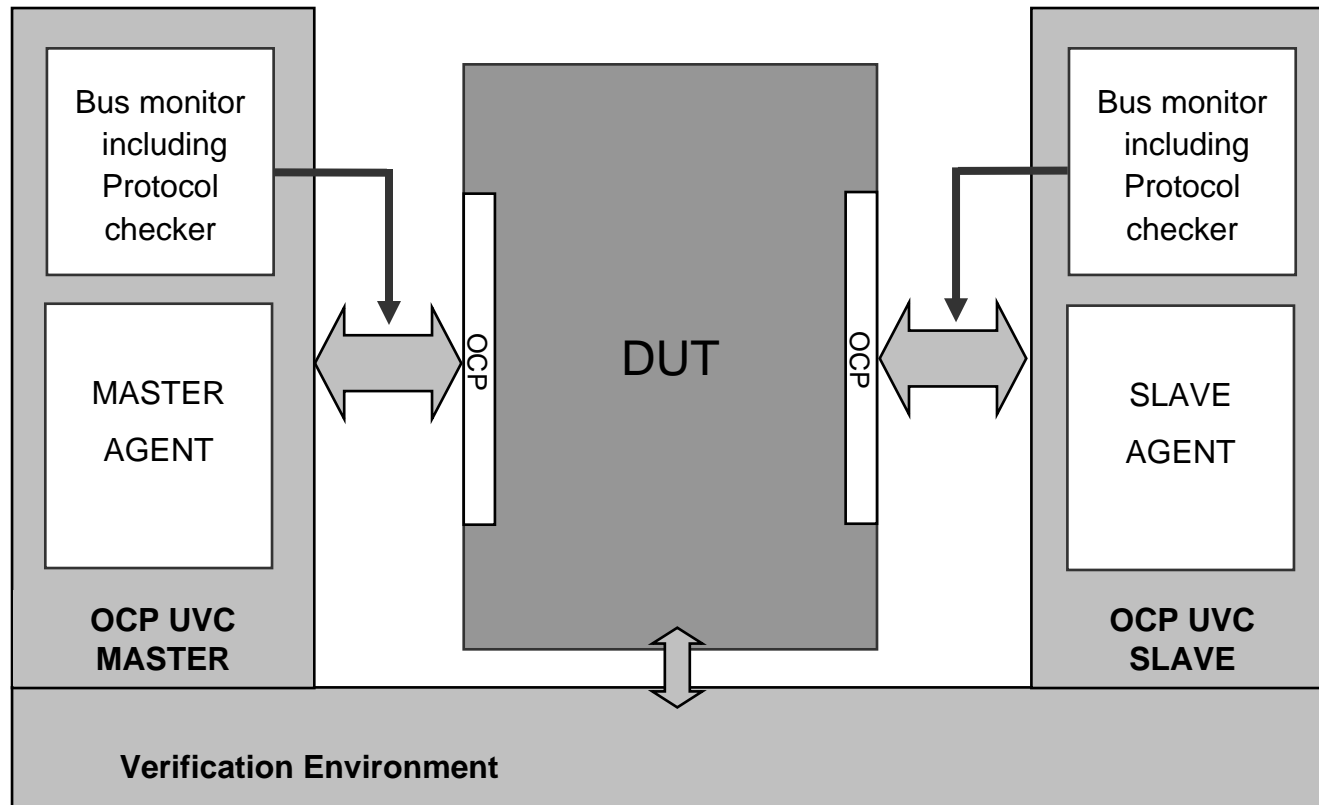
- Metric-driven verification process
 - Defines the target for complete verification
 - Used as feedback to control stimulus generation
 - Built in coverage points to exercise the protocol fully
 - Coverage hooks enable user to extend coverage for module/system verification
- Metrics provide management guidance
 - Verification completeness
 - Remaining work estimation
 - Verification productivity
 - Tool utilization

Using the Monitor API for Coverage and Checking

- User may want to extend built in coverage and checking for design specific issues
 - Cover specific IP-related state when different bursts occur
 - Build design specific scoreboard ...
- Comprehensive API to monitor for user extensions
 - E.g. ~15 events on all interesting bus event triggers {burst_started/ended, address_phase_started/ended, handshake_phase_started/ended, response_phase_started/ended etc...}
 - Events related to stored data samples from the bus

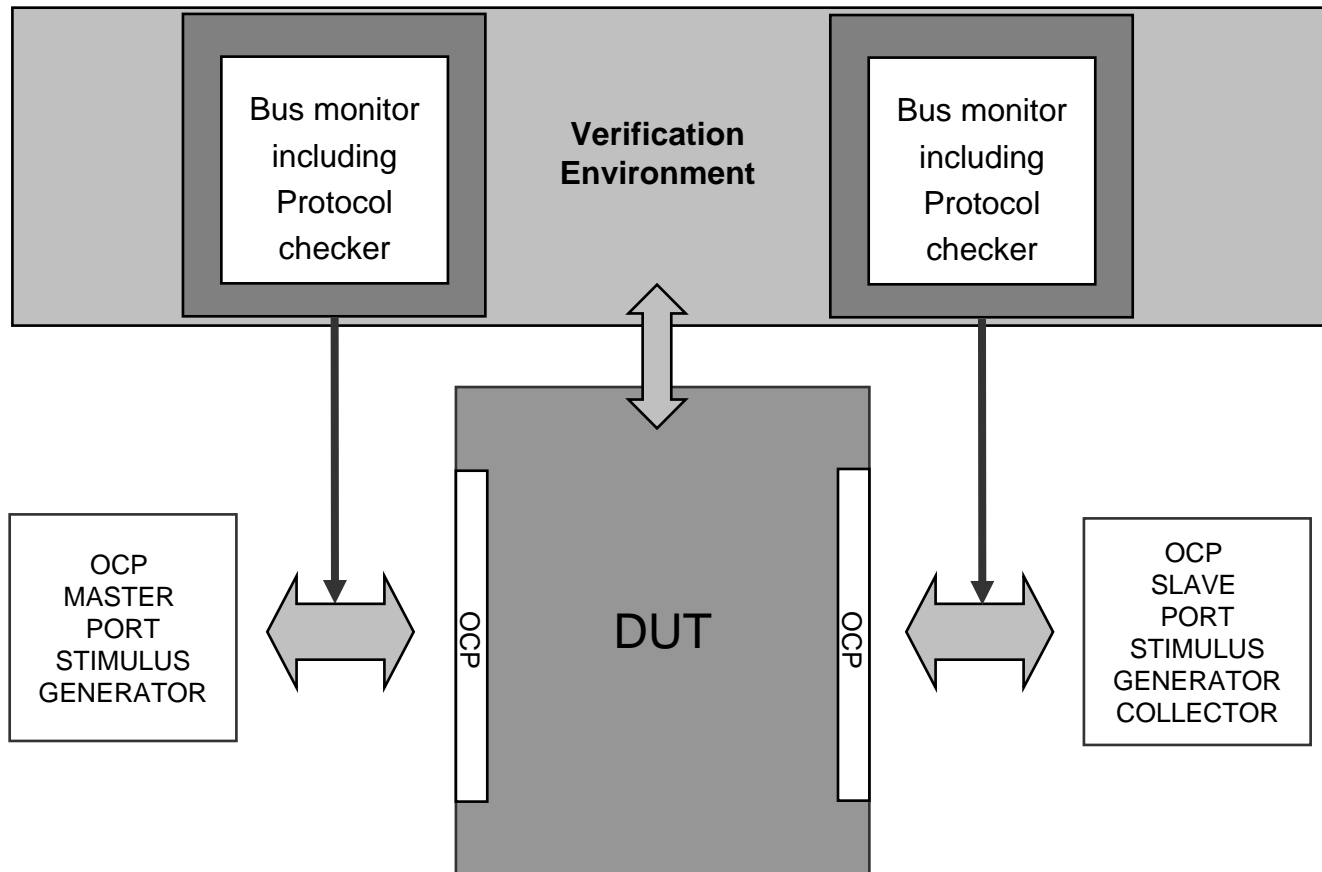
Module Level Verification example

- OCP UVC MASTER instance to emulate Micro actions
- OCP UVC SLAVE instance to emulate external memory



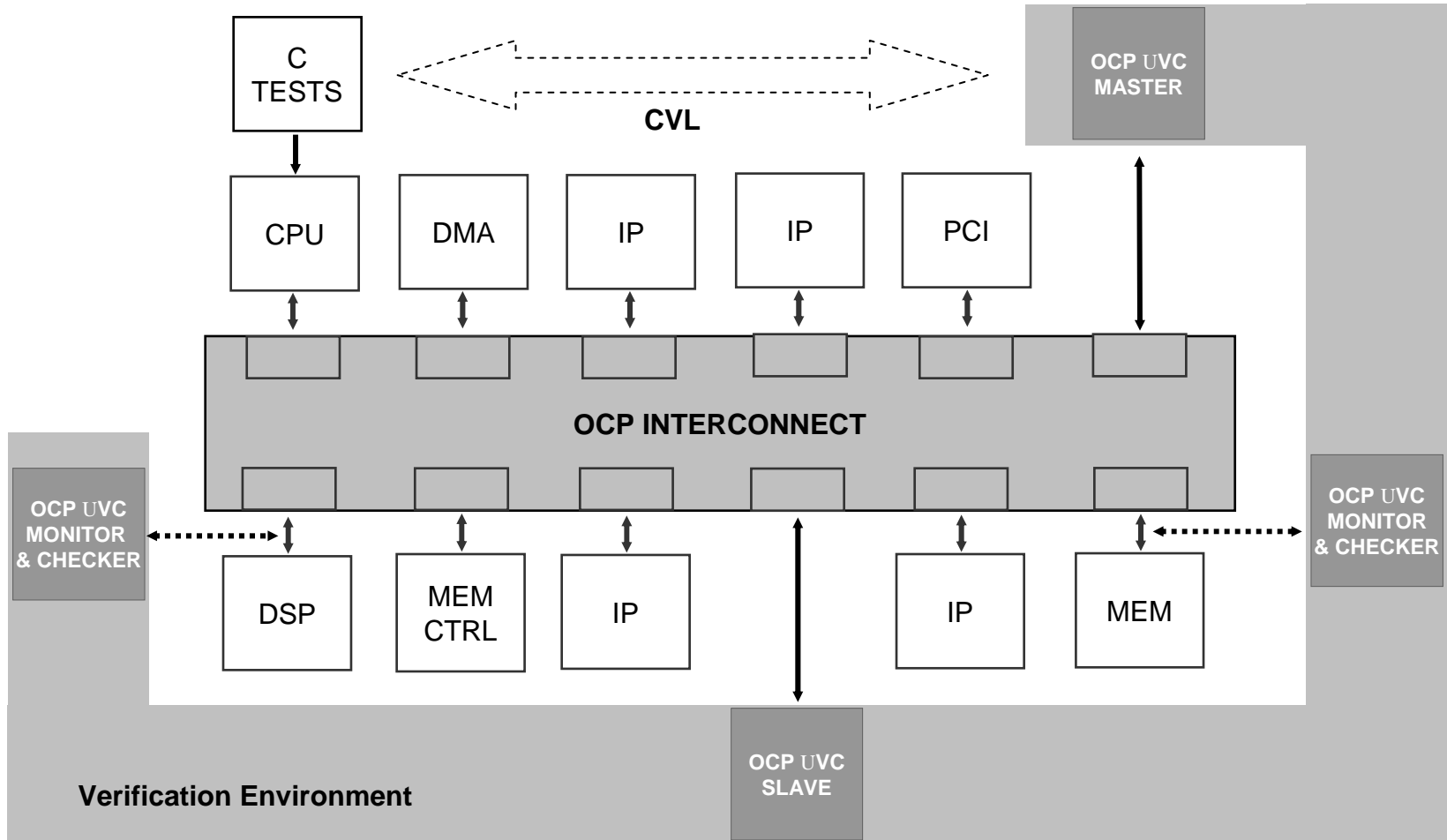
Module Level Verification (Cont.)

- HDL-based verification environment (complete test suite)
- OCP UVC MONITOR instances to check OCP protocol



OCP Based System Level Verification example

- Multiple instances of the OCP UVC (Master, Slave, Monitor)
- Support for different environments (i.e. HDL, C tests, etc)



Test Example

INCR Read Sequence Using OVM-SV Interface

```
class ocp_simple_read extends ovm_sequence;
  `ovm_sequence_utils(ocp_simple_read,cdn_ocp_master_sequencer)
  cdn_ocp_api_burst burst;
  ...
  constraint seq_constraints {
    address < 1000;
    address>0;
  }
  ...
  virtual task body();
    `ovm_do_with(burst, {kind_timing == READ; use_kind_timing == 1;
      start_addr == address; use_start_address == 1;
      mburstseq == INCR; use_mburstseq == 1;})
  endtask
endclass
```

Test Example

Write Sequence Using OVM-*e* Interface

// Send a sequence of 2 burst, writing a location and reading it back.

```
extend READ_AFTER_WRITE cdn_ocp_master_sequence {  
  
    !write_cmd : MASTER WR SINGLE cdn_ocp_trans_s;  
    !read_cmd  : MASTER RD SINGLE cdn_ocp_trans_s;  
    addr      : uint(bits: ADDR_WIDTH_MAX);  
    keep addr[1:0] == 2'b00; // keep address aligned  
  
    body() @driver.clock is only {  
        do write_cmd keeping {  
            it.addr == addr;  
            it.maddrspace == 0;  
        };  
        do read_cmd keeping {  
            it.addr == write_cmd.addr;  
            it.maddrspace == 0;  
        };  
    };  
};
```

OCP UVC Deliverables

- OCP UVC
- Compliance Management System (CMS)*
- Documentation
 - User Manual
 - Release notes
 - ElectronicDOCs
- Sample verification environments

* CMS for OCP Available 1H'09

Cadence OCP UVC Summary

- Mature, proven to increase quality, productivity and predictability
 - Metric Driven Verification with CMS*
 - Identifies DUT bugs that ad hoc methodologies miss
 - Reusable across designs and hierarchies
- Robust and proven technology and methodology
 - SiliconBackplane, SonicsMX, MIPS & ARM based designs
 - Standards defined extensible protocol checker
- Multi-language support
 - SystemVerilog, *e*
- Scalable and configurable architecture
 - Core, subsystem and top-level
 - From a single transaction to an entire system

* CMS for OCP Available 1H'09



cādence™