VERIFICATION ACADEMY

UVM Basics

Introduction to UVM

John Aynsley
CTO, Doulos

academy@mentor.com
www.verificationacademy.com

Mentor Graphics

Sessions in this Module

Introduction to UVM

- For managers and engineers

UVM "Hello World"
Connecting Env to DUT
Connecting Components
Introducing Transactions
Sequences and Tests
Monitors and Subscribers
Reporting

- For engineers
- Simple, step-by-step
- Actual runnable code
- Suitable if you know an HDL
- No OOP or CRV required

What is UVM?

- The Universal Verification Methodology
- Test benches for designs in SystemVerilog/VHDL/SystemC
- Accellera standard with wide industry support
- SystemVerilog UVM BCL (Base Class Library)
- Open source (Apache licence)
- Near-backward compatible with OVM

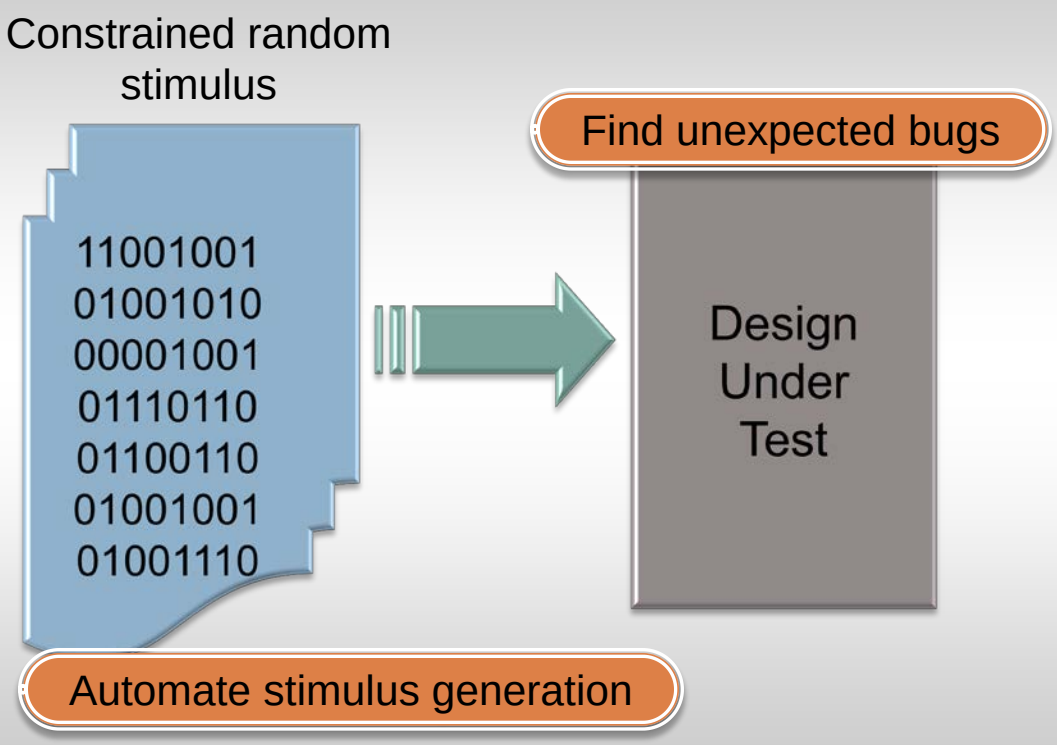
UVM Highlights

- Constrained random verification
- Configurable, flexible, test benches
- Verification IP reuse

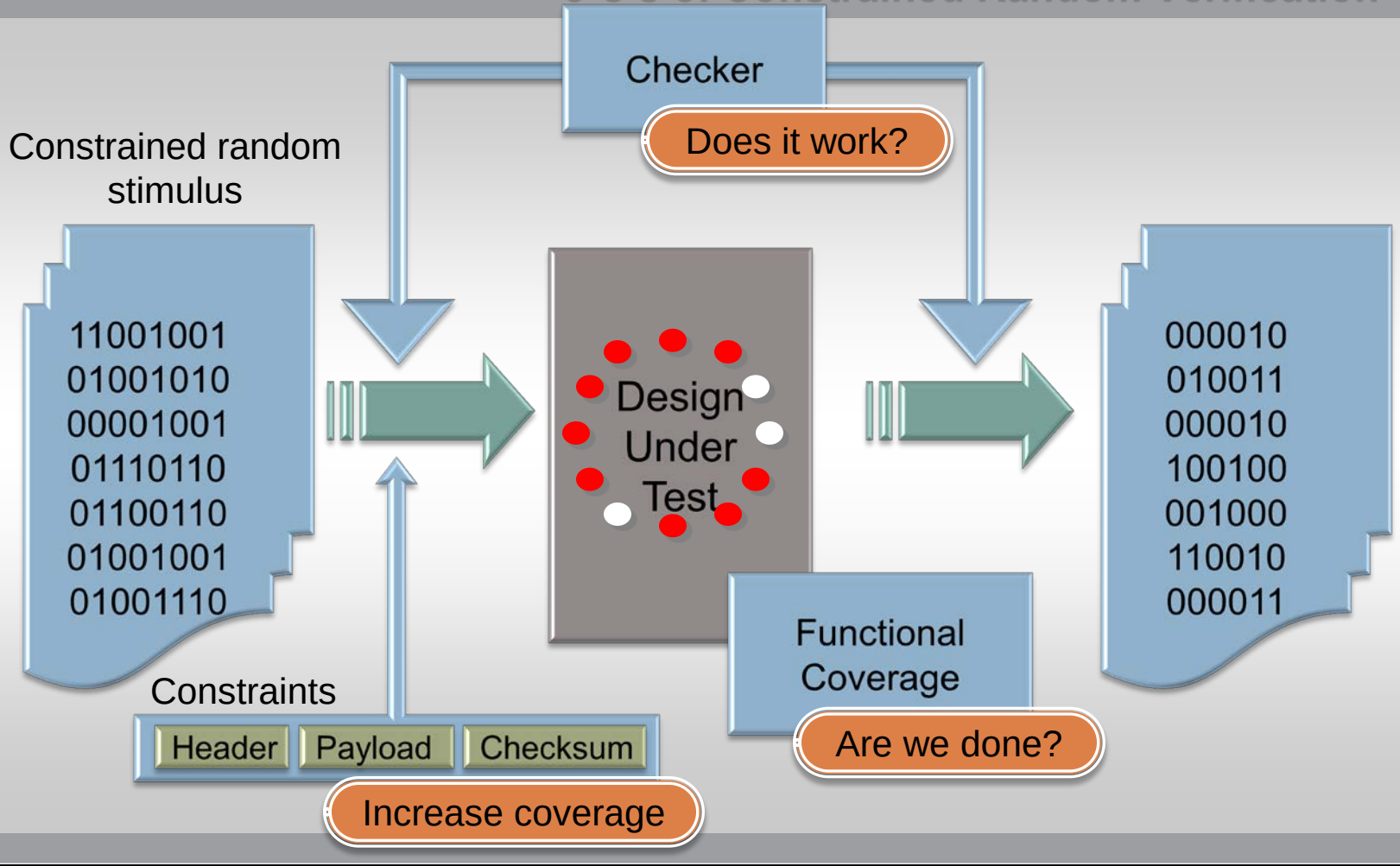
A standard approach – consistency and uniformity

- Separation of tests from test bench
- Transaction-level communication (TLM)
- Layered sequential stimulus
- Standardized messaging
- Register layer

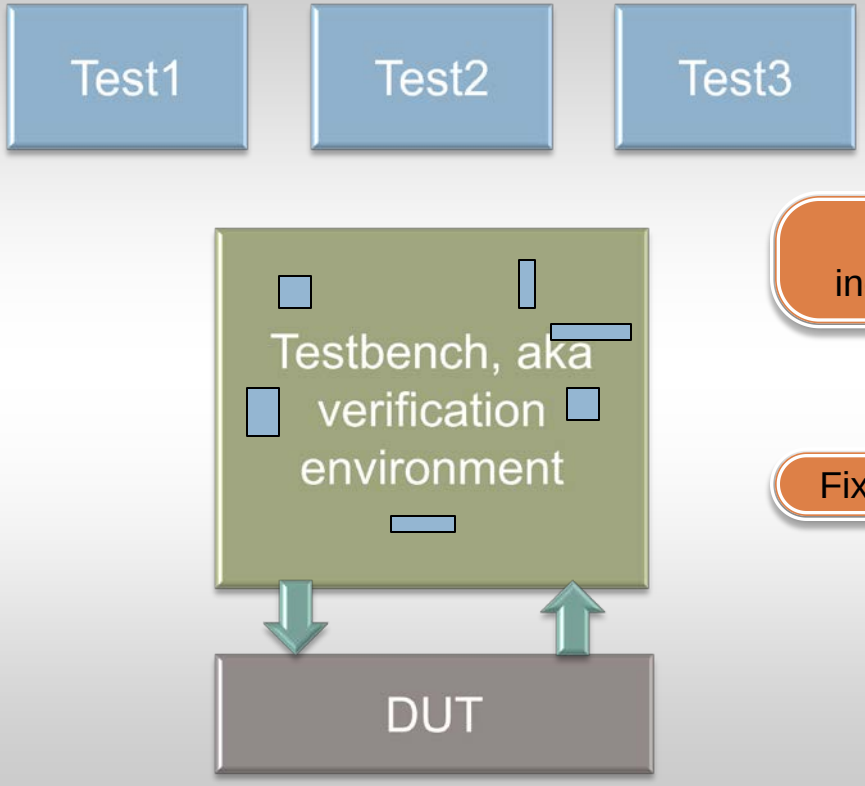
Benefits of Constrained Random Stimulus



3 C's of Constrained Random Verification



Test versus Testbench



Each test makes incremental changes

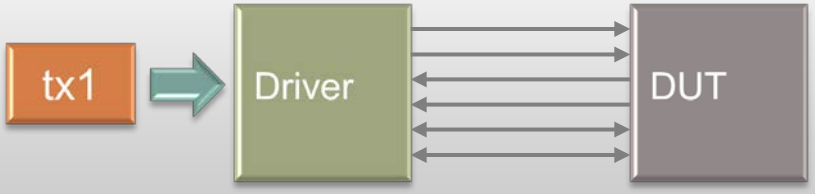
Fixed, stable, reusable

Layered Sequential Stimulus

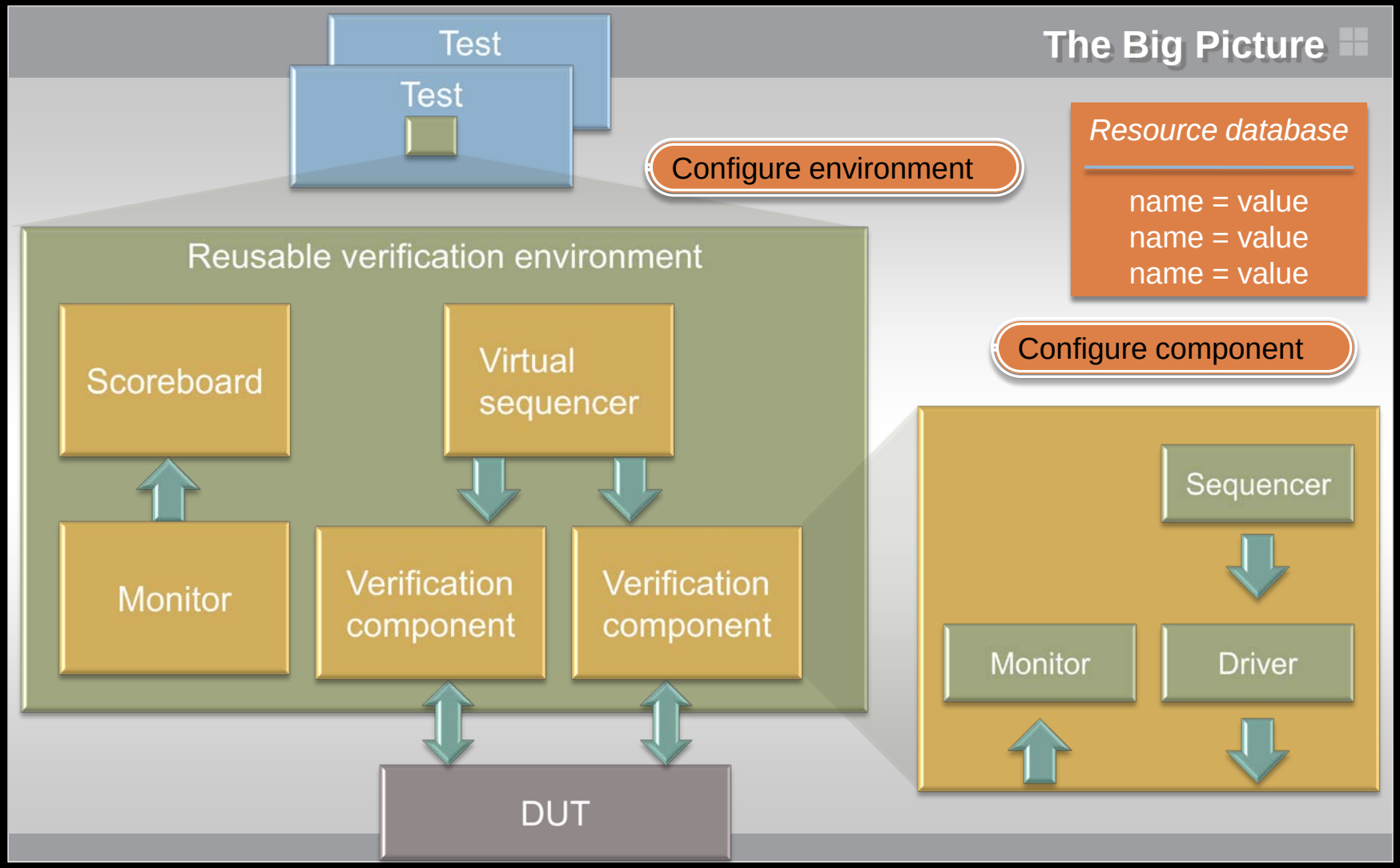
Nested, layered or virtual sequences

Constrained random sequence of transactions

Transaction = command to driver



The Big Picture



HTML and PDF Documentation

accellera Universal Verification Methodology

Classes and Utilities

- BASE
- REPORTING
- FACTORY
- CONFIGURATION AND RESOURC
- SEQUENCERS
- SEQUENCES
- SYNCHRONIZATION
- CONTAINERS
- TLM
- COMPONENTS
- MACROS
- POLICIES
- REGISTER LAYER
- COMMAND LINE PROCESSOR
- GLOBALS
- INDEX

Search

UVM Class Reference

The UVM Class Library provides the building blocks needed to quickly develop well-constructed and reusable verification components and test environments in SystemVerilog.

This UVM Class Reference provides detailed reference information for each user-visible class in the UVM library. For additional information on using UVM, see the UVM User's Guide located in the top level directory within the UVM kit.

We divide the UVM classes and utilities into categories pertaining to their role or function. A more detailed overview of each category-- and the classes comprising them-- can be found in the menu at left.

Globals	This category defines a small list of types, variables, functions, and tasks defined in the *uvm_pkg* scope. These items are accessible from any scope that imports the *uvm_pkg*. See Types and Enumerations and Globals for details.
Base	This basic building blocks for all environments are components, which do the actual work, transactions, which convey information between components, and ports, which provide the interfaces used to convey transactions. The UVM's core *base* classes provide these building blocks. See Core Base

What You Need to Learn

Verification Planning and Management

Constrained Random Verification

UVM BCL (& OOP & TLM)

classes

SystemVerilog

assertions
coverage
constraints
interfaces

Summary

Introduction to UVM

UVM "Hello World"

Connecting Env to DUT

Connecting Components

Introducing Transactions

Sequences and Tests

Monitors and Subscribers

Reporting

VERIFICATION ACADEMY

UVM Basics

Introduction to UVM

John Aynsley
CTO, Doulos

academy@mentor.com
www.verificationacademy.com

Mentor Graphics