

Advanced Verification for FPGAs

Simone Catenacci
Application Engineer
IC Verification Solutions

April 9, 2018

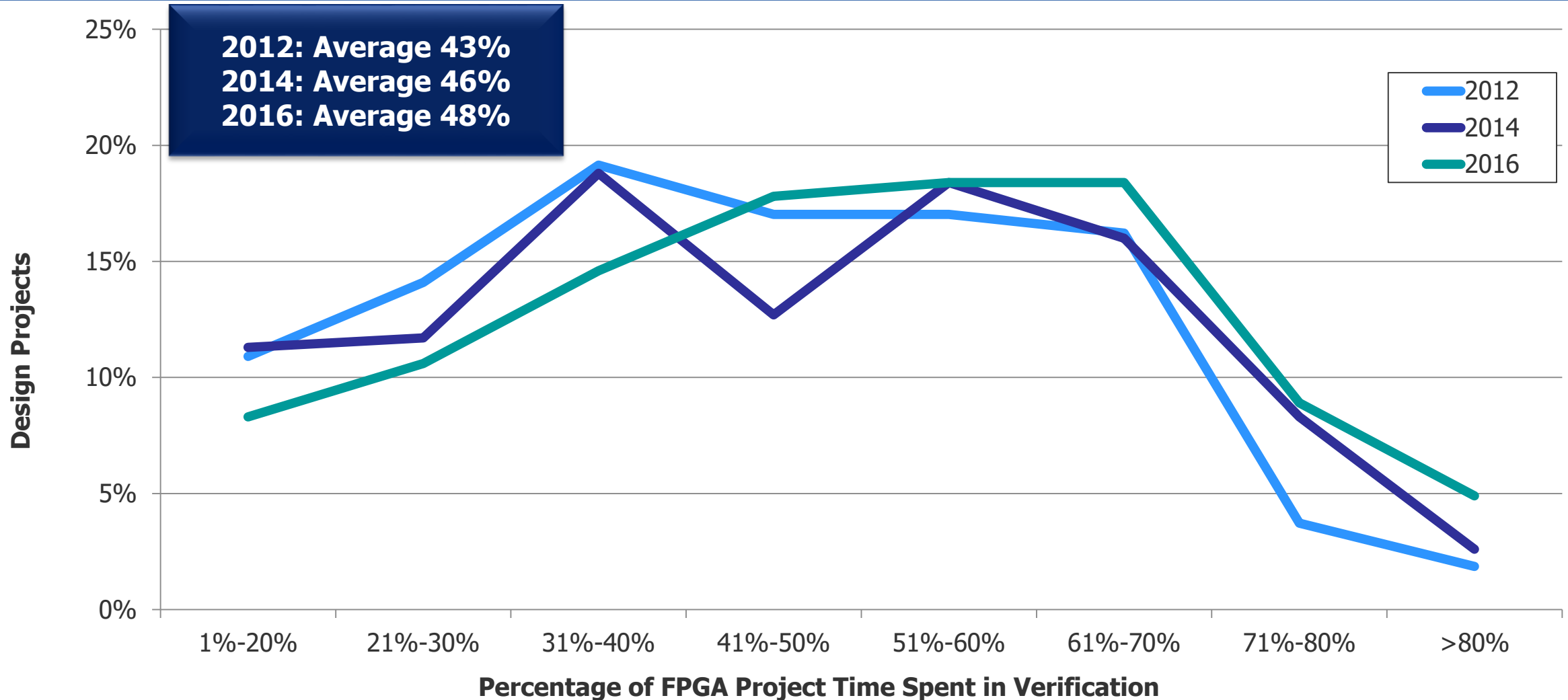
Agenda

- Industry Trends and Challenges for FPGA Development
- Staying competitive with Advanced Verification
- Final Thoughts

INDUSTRY TRENDS AND CHALLENGES FOR FPGA DEVELOPMENT

FPGA Verification Project Time

FPGA verification consumes majority of project time



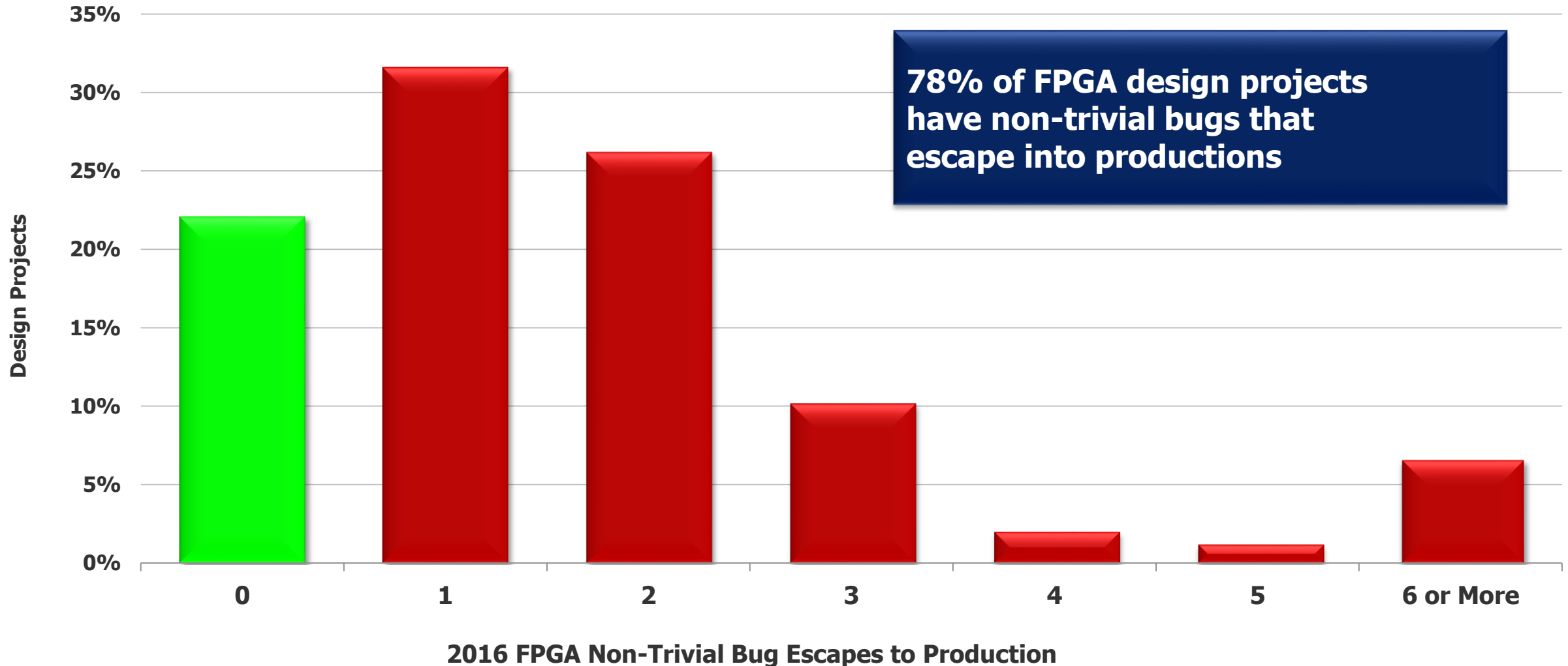
Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study

Copyright © 2018 Mentor, A Siemens Business
All Rights Reserved

Mentor
A Siemens Business

Number of FPGA Bug Escapes to Production

Bugs found late in the development cycle are exponentially more expensive



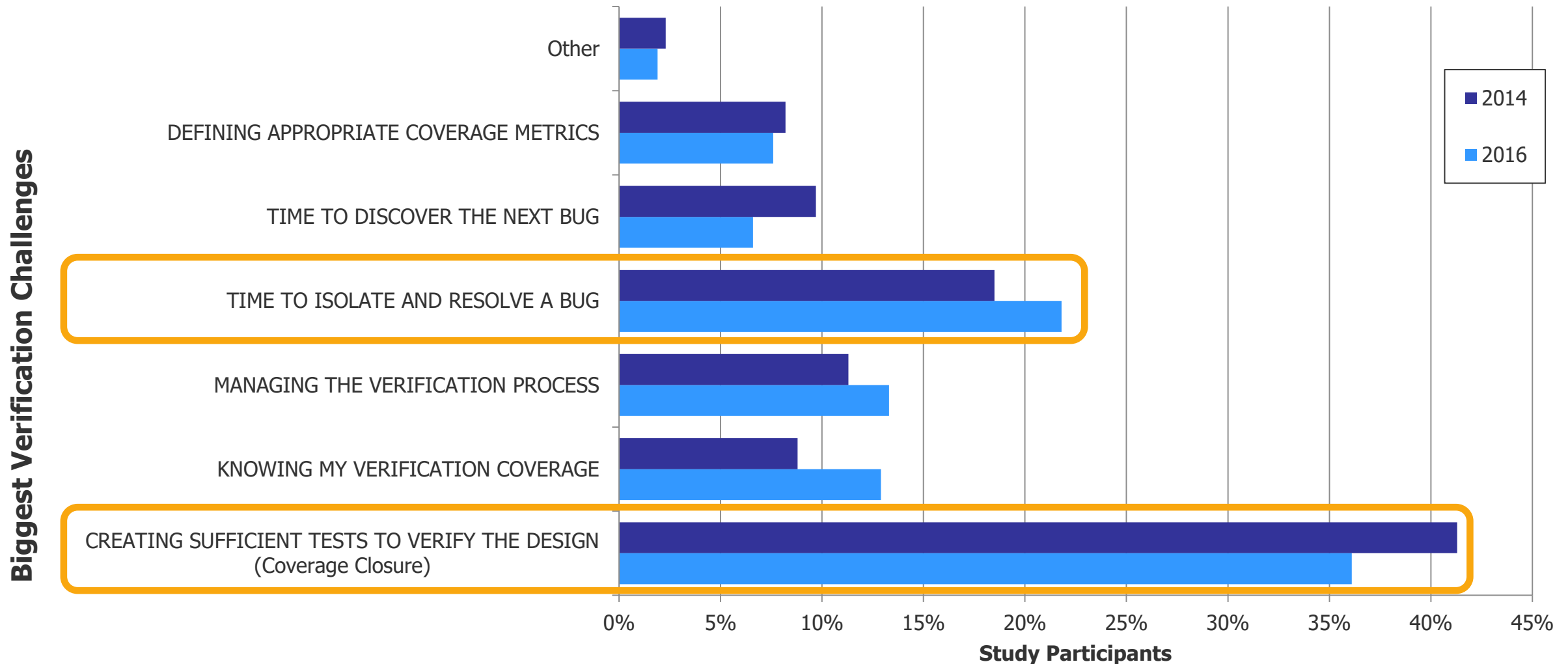
Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study

Copyright © 2018 Mentor, A Siemens Business
All Rights Reserved

Mentor
A Siemens Business

Biggest FPGA Verification Challenges

Sufficient testing and improving debug efficiency are the biggest challenges



Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study






Copyright © 2018 Mentor, A Siemens Business
All Rights Reserved

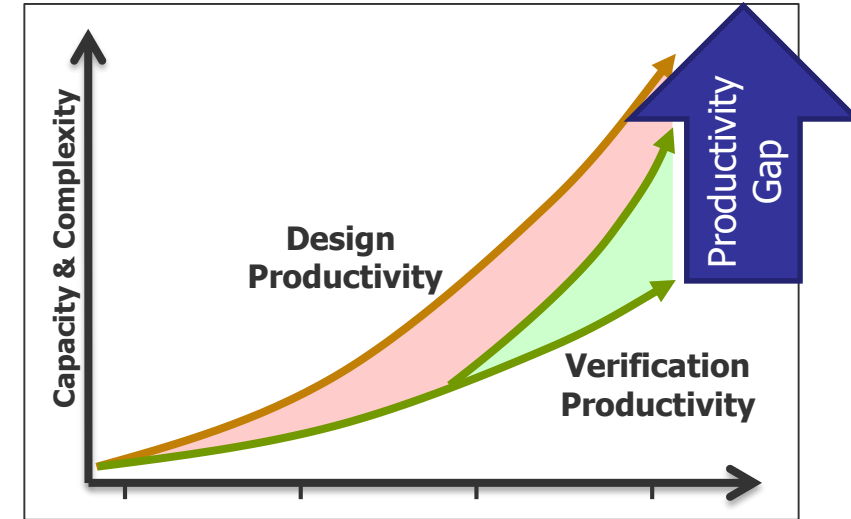
Mentor
A Siemens Business

FPGA Productivity Gap

Verification must evolve to keep pace with design innovations

■ FPGA vendors continue with design innovations




- Hardened blocks – Reuse 
- IP - Reuse 
- SOC – Reuse 
- Graphical system builders - Automation 
- HLS – Abstraction 



■ “Design Productivity” alone ≠ faster “Time to Market”

- Designs must also work and be of high quality to gain adoption
- TTM depends on maximizing design and verification productivity

■ Verification must similarly evolve to keep pace

- Maximize reuse 
- Improve automation 
- Raise the level of abstraction 

STAYING COMPETITIVE WITH ADVANCED VERIFICATION

How Do You Improve Verification Productivity?

Reuse, automation and abstraction to the rescue



Maximize Reuse

- Reuse test environments from project to project
- Minimizes time spent developing TB environments
- Minimizes TB debug time
- VIP for bus protocols & memory models



Improve Automation

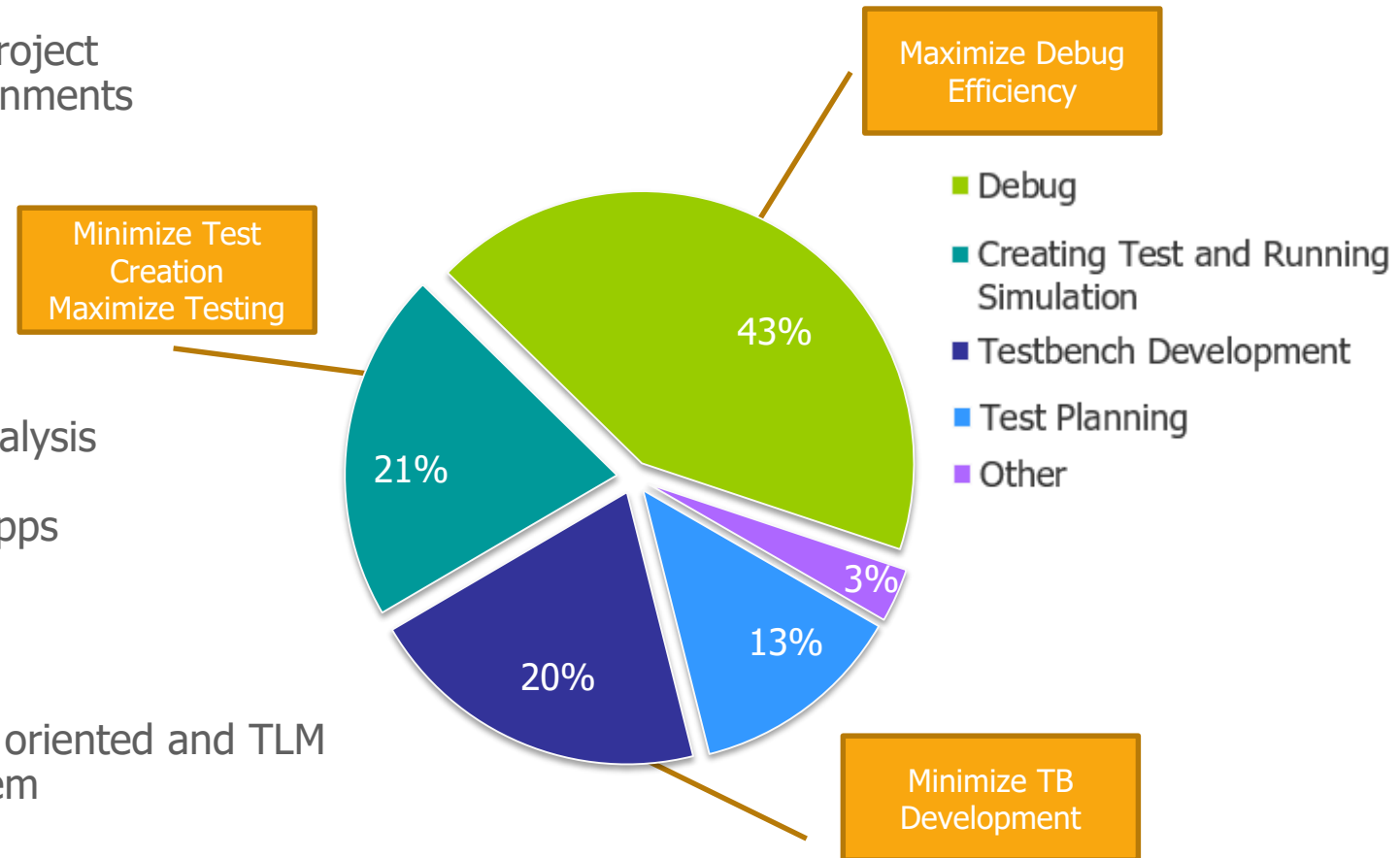
- Automate TB creation
- Automate regressions
- Automate results/metrics collection and analysis
- Automatic testing with Formal Apps
- Accelerate coverage closure with Formal Apps
- Automate requirements tracing



Raise Abstraction Level

- Improve TB creation efficiency with object oriented and TLM
- Write tests at scenario level to match system
- Debug at a higher level than pin wiggles

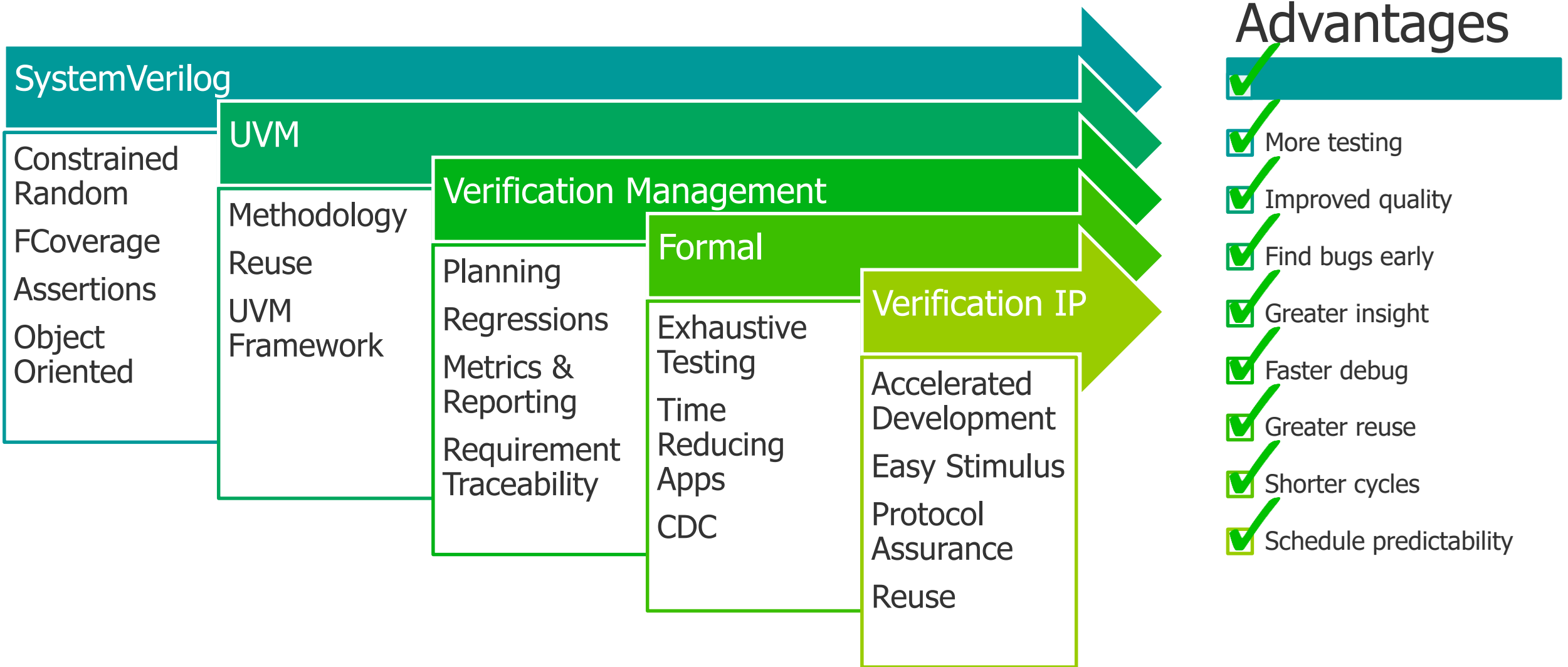
Where do spend your time for FPGA verification?



Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study

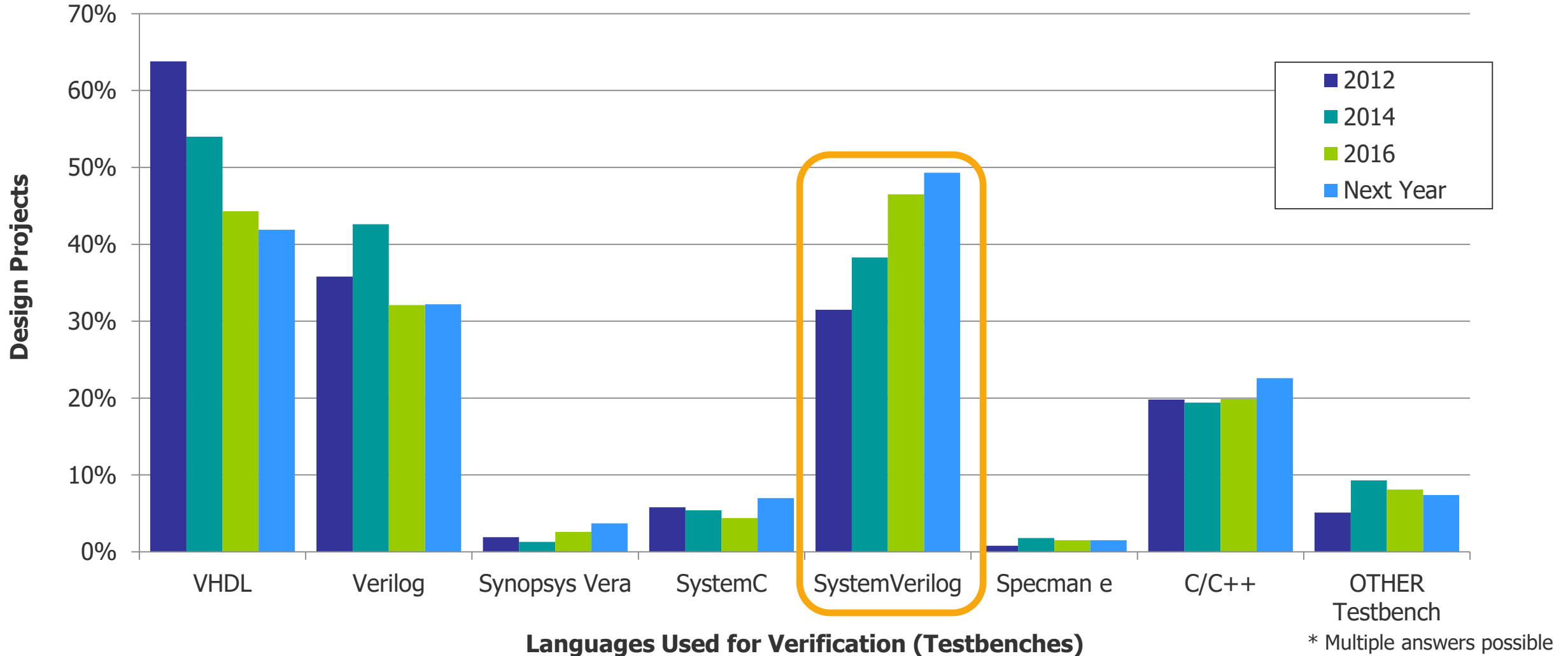
Improving with Advanced Verification

From the lab to the modern age for better quality and productivity



FPGA Verification Language Adoption Trends

SystemVerilog is now the leading FPGA verification language



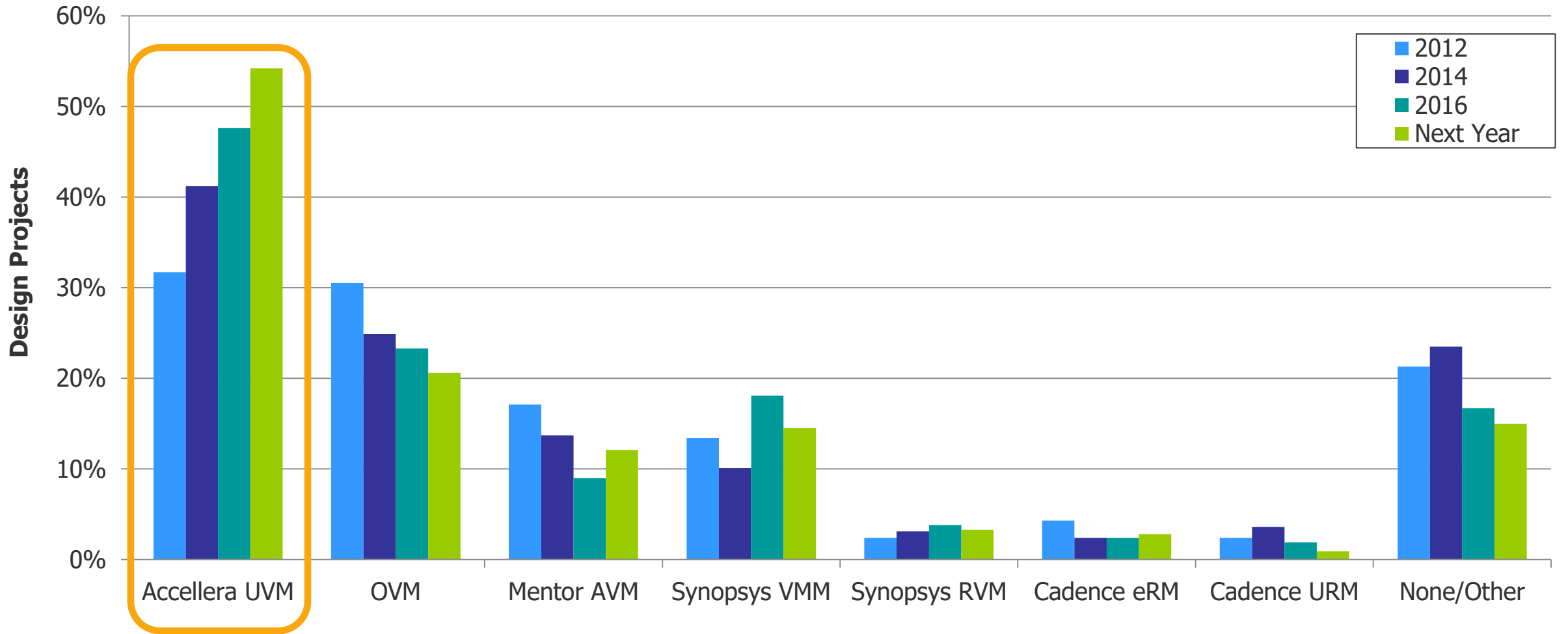
Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study

Copyright © 2018 Mentor, A Siemens Business
All Rights Reserved

Mentor
A Siemens Business

FPGA Testbench Methodology Adoption Trends

UVM is the clear leader in FPGA testbench methodologies



FPGA Methodologies and Testbench Base-Class Libraries

* Multiple answers possible

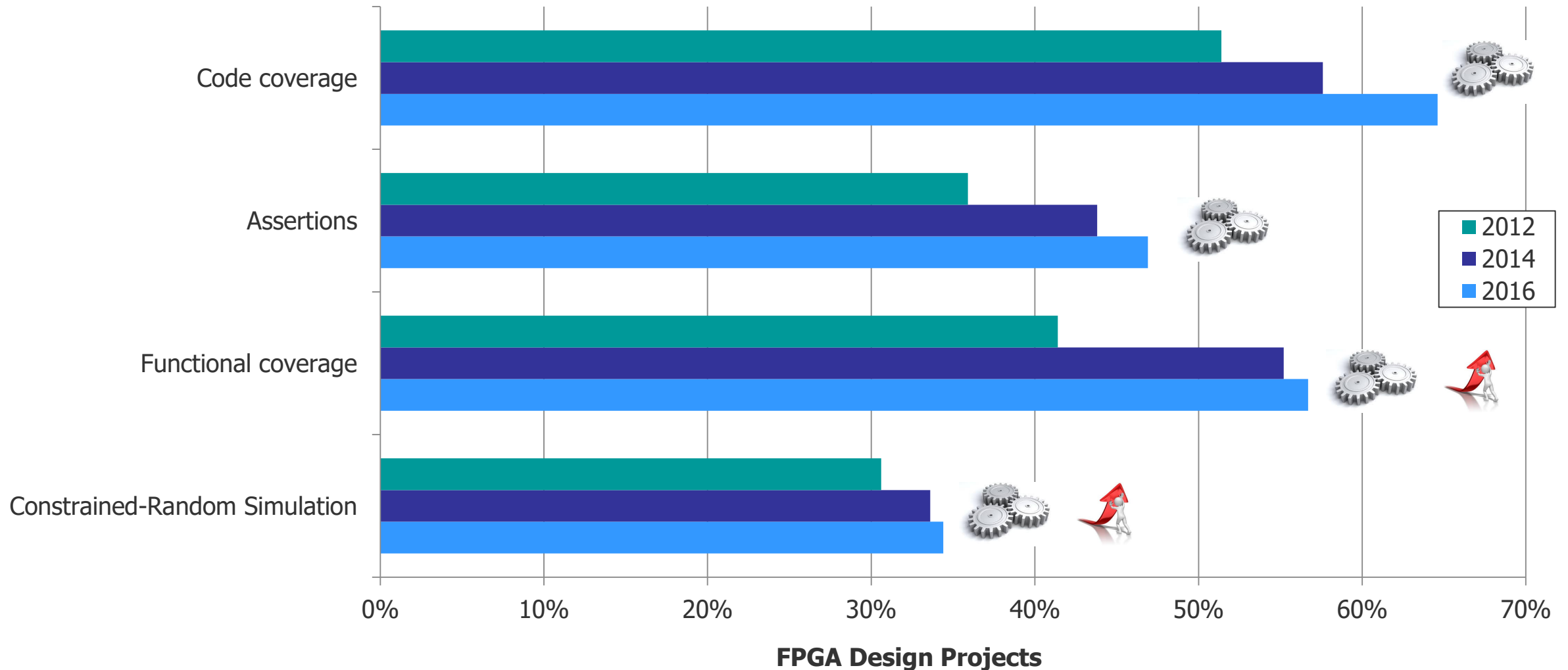
Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study

Copyright © 2018 Mentor, A Siemens Business
All Rights Reserved

Mentor
A Siemens Business

FPGA Dynamic Verification Adoption Trends

Adoption of advanced verification techniques continue to grow for FPGA verification



Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study

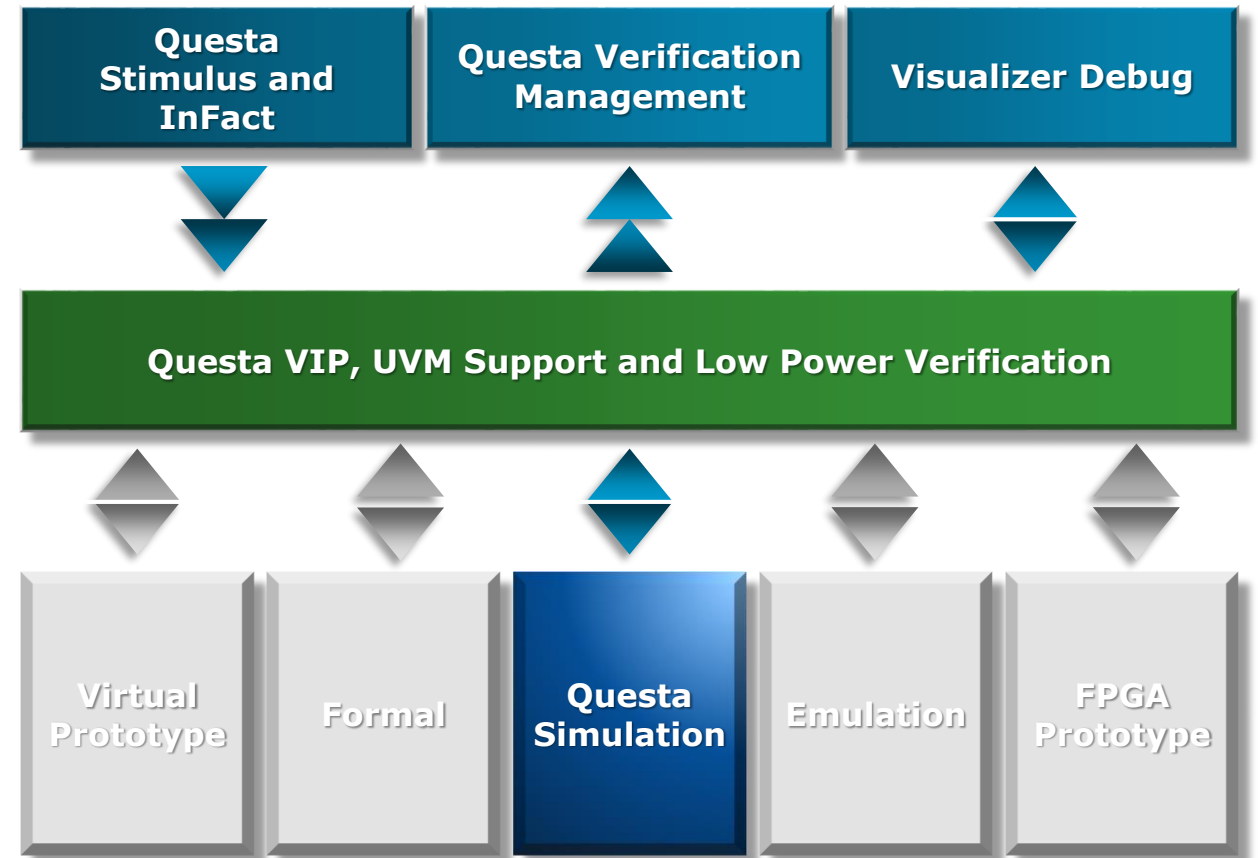
Copyright © 2018 Mentor, A Siemens Business
All Rights Reserved



Questa Simulation

Verification Productivity with High Performance Simulation at its Core

- Complete Solution with Mentor Enterprise Verification Platform
- Most Comprehensive Support of Methodologies for Increased Verification Productivity
 - Assertion-Based Verification
 - SV-UVM
 - Coverage/Plan Driven Verification
 - Regression Creation and Management
 - Test Bench Automation



Mentor UVM Framework (UVMF) for Easier UVM

Making the transition to UVM possible



UVM Jumpstart

- Immediately productive while learning UVM
- Layer on top of UVM that hides UVM details
- Allows team to focus on verifying product features
- Promotes reuse increasing productivity



UVM Testbench Generators

- Code generators to create a UVM testbench in minutes
- Saves 3-4 weeks of effort on every project
- UVMF is open source and no cost – Delivered in Questa



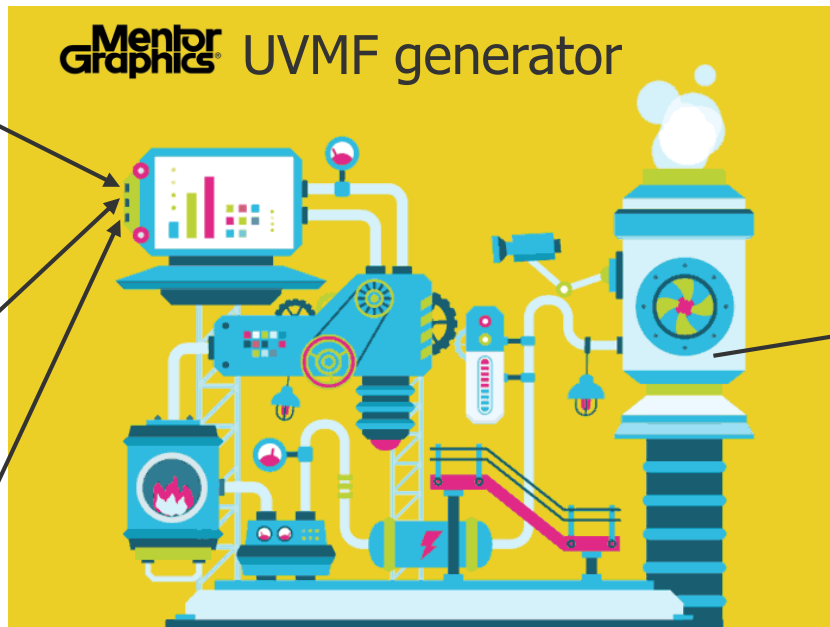
Helped over 40 companies adopt UVM

- 75% are FPGA customers
- Over half are in mil-aero industry
- Majority use VHDL for design and have no SystemVerilog experience

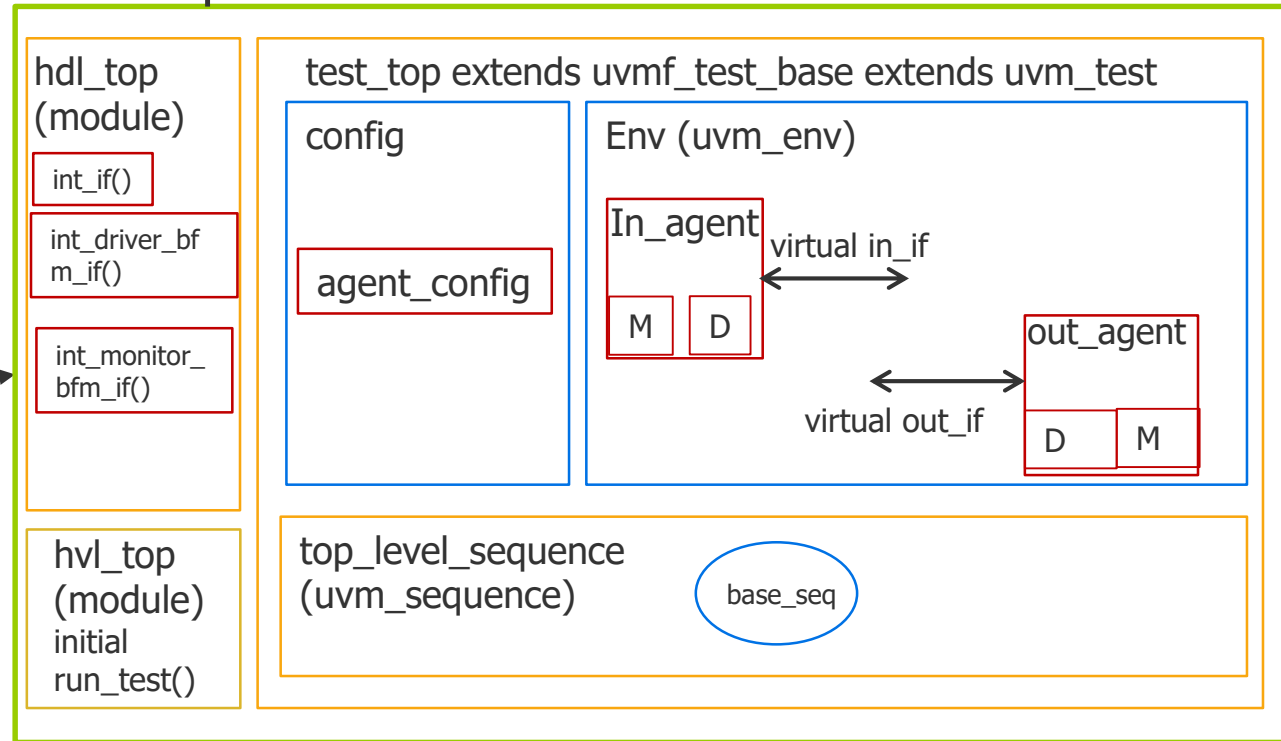
UVMF-Generator Flow



1 File for each DUT interface



UVMF production Environment



Output files includes:

Package Declaration, class definitions, interface definitions, file lists and Makefiles

Interface config

Environment config

TestBench config



Working Simulation Out Of The Box

- 'make debug' to bring up interactive simulation

The screenshot displays the UVM simulation environment with three key components highlighted by red circles with numbers:

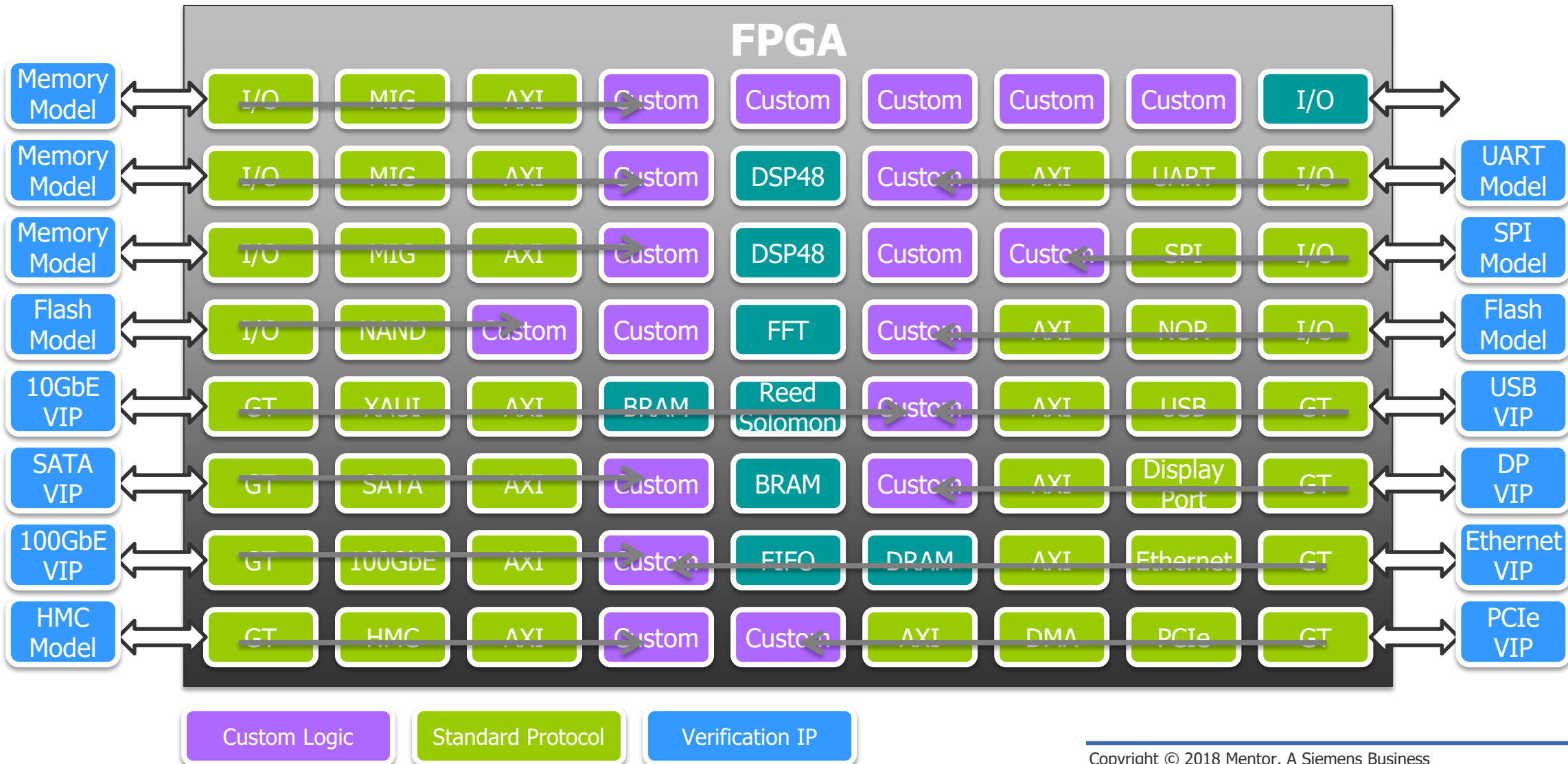
- 1**: Instance window showing the UVM environment hierarchy, including `uvm_test_top`, `environment`, `wb_U1_agent`, and `ahb_U1_agent`.
- 2**: Wave window showing a transaction stream with columns for `wb transaction`, `wb transaction`, and `wb transac...`. The wave shows signals like `clk`, `rst`, `inta`, `cyc`, `stb`, `adr`, `we`, `din`, `dout`, `ack`, `err`, and `rtv`.
- 3**: Covergroups window showing coverage data for various groups, including `/ahb_pkg/ahb_configuration`, `/wb_pkg/wb_configuration`, and `/ahb_pkg/ahb_transaction_coverage`.

Name	Coverage	Goal	% of Goal	Status
/ahb_pkg/ahb_configuration	0.0%	100	0.0%	
trps ahb_configuration_cg	0.0%	100	0.0%	
/wb_pkg/wb_configuration	22.9%	100	22.9%	
trps ahb_transaction_coverage	66.6%	100	66.6%	
trps ahb_transaction_cg::op	0.6%	100	0.6%	
CVP ahb_transaction_cg::data	1.4%	100	1.4%	
CVP ahb_transaction_cg::addr				
/wb_pkg/wb_transaction_coverage				

1. Complete Dual Top UVM environment compiles & simulates
2. Monitor transactions automatically added to the wave window
3. Covergroups automatically created and collect coverage

Verification IP Enables Rapid Testbench Development

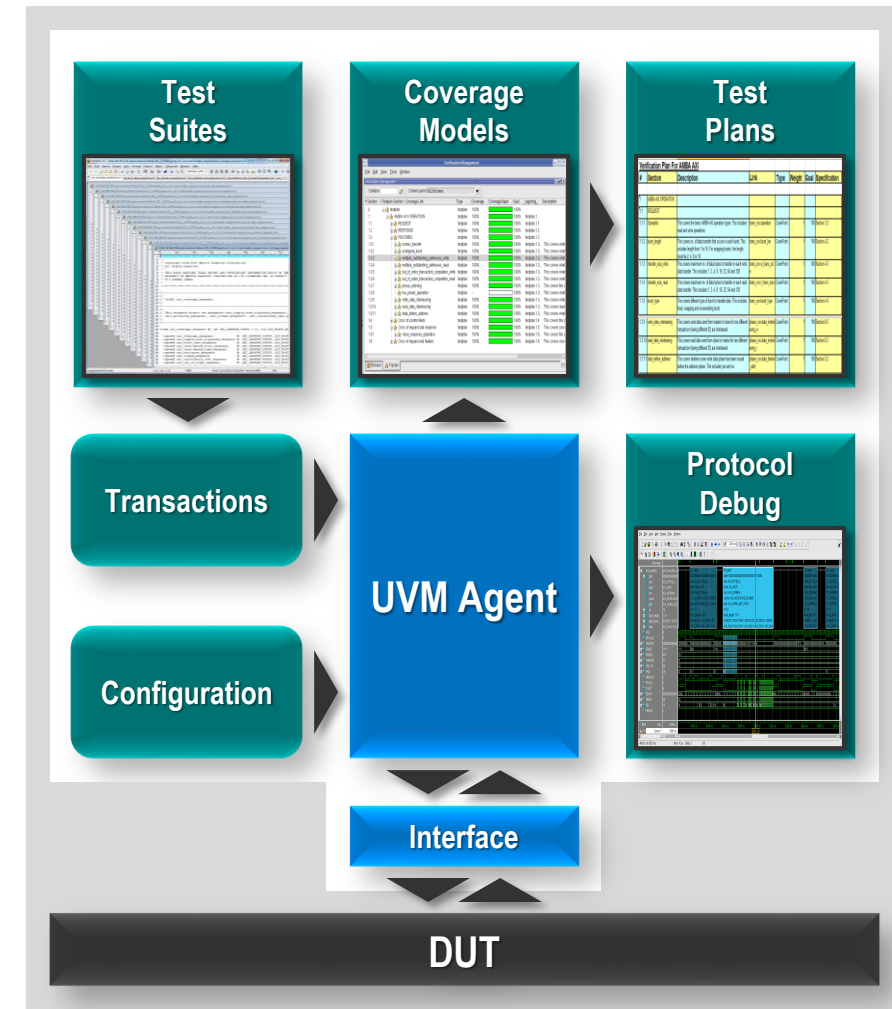
Enabling verification of your custom application without deep protocol expertise or effort



Verification IP: Off The Shelf Protocol Test Environments

"Protocol Expertise" Reduces Time, Effort & Risk for IP Centric Designs

- What is Verification IP?
 - Reusable test environments
 - Built around standard protocols
- Reduces Verification Effort
 - Provides completed TB at standard interfaces
 - Provides functionality to create and run tests
 - Less time creating TB and more time testing
- Eliminates Protocol Expertise
 - Deep protocol knowledge not needed for TB creation
 - Provides out of the box protocol stimulus and checking
 - Enhanced debug productivity with transactions and assertions
- VIP Reduces Risk
 - Re-usable VIP building blocks
 - Common design architectures & protocols
 - Complete protocol assurance



A Complete VIP Solution for FPGA Designs

Mentor Graphics Questa Verification IP Library

Questa VIP Protocols

AMBA® Family	PCIe® Family	Ethernet Family	USB Family	MIPI® Family	Serial Family	Display Family	Automotive Family
ACE	NVMe	200/400G	3.1 Pipe	I3C	JTAG	CEC	CAN
AXI4	AHCI	25/50G	3.1 Serial	UFS	SmartCard	HDCP	CAN-FD
AXI3	RMMI	100G	USB PD	Unipro	I2C	HDMI 2.1	LIN
AHB5	MRIOV	40G	SSIC	LLI	I3C	HDMI 1.4	
AHB	MPHY	10G	oHCI	CSI-2 / CSI-3	I2S	HDMI 1.3	
APB3	PCIe 4.0	2.5/5G	xHCI	DSI	SPI	DisplayPort	5G Family
	PCIe 3.1	1G	MPHY	DigRF	SPI 4.2	eDP	CPRI
	PCIe 2.1	100M	eHCI	MPHY	UART	V-by-One	JESD204B
	PCIe 1.1	10M	USB 3.0	HSI			Storage Family
	PIPE 4.3	Automotive	USB 2.0	CPHY			SATA
	PIE8	Interlaken	USB 1.1	DPHY			

Questa Memory Library

DRAM Family	Flash Family	Mil-Aero Family
LPDDR4	SDCard 4.2	Spacewire
LPDDR3	SDIO 4.1	1553b
LPDDR2	eMMC 5.1	PCI
DDR4	ONFI 4.0	SRIO
DDR3	Toggle	AMBA 5
DDR2	UFS	CHI 5
WIDEIO	ParallelNOR	Hyperbus
DFI	Serial NOR	Hyperram
HMC		Hyperflash
HBM2		NVMe over Fabric
DIMM		

Debug Faster with Questa Verification IP

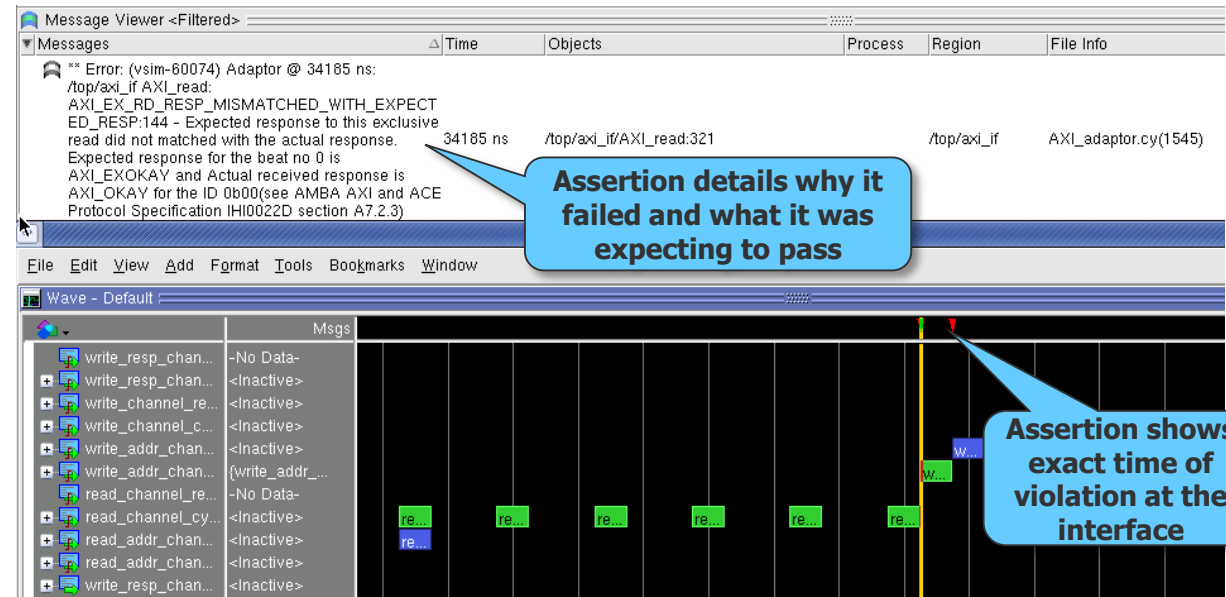
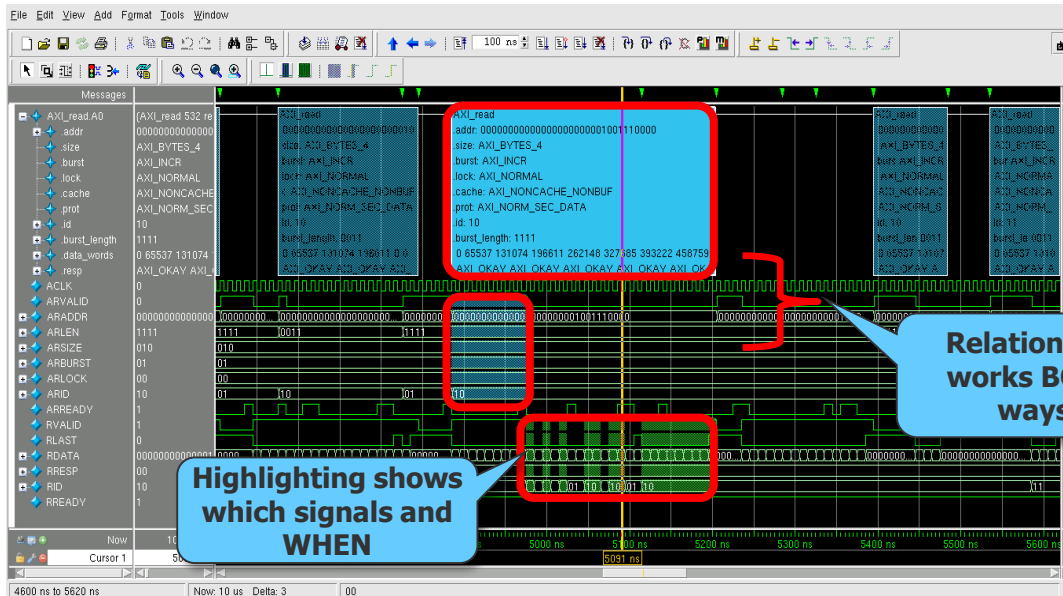
QVIP Improves debug abstraction and automation to boost productivity



- Transaction recording
 - Debug at a higher level with integrated transactions
 - Quickly understand and analyze bus activity
 - Automatically links transactions to signals



- Protocol assertions
 - Integrated assertions automate protocol checking
 - Protocol assertions immediately pinpoint source of failure
 - Quickly understand integration of your design with IP



Questa Verification Management

Productivity in Tool Execution, Data Management, Analysis and Tracking

■ Test Plan Tracking

- Lifts the lid on simulation, formal, CDC
- See progress day by day
- Have I done enough testing?
- Am I ready to freeze the RTL?

■ Trend Metrics

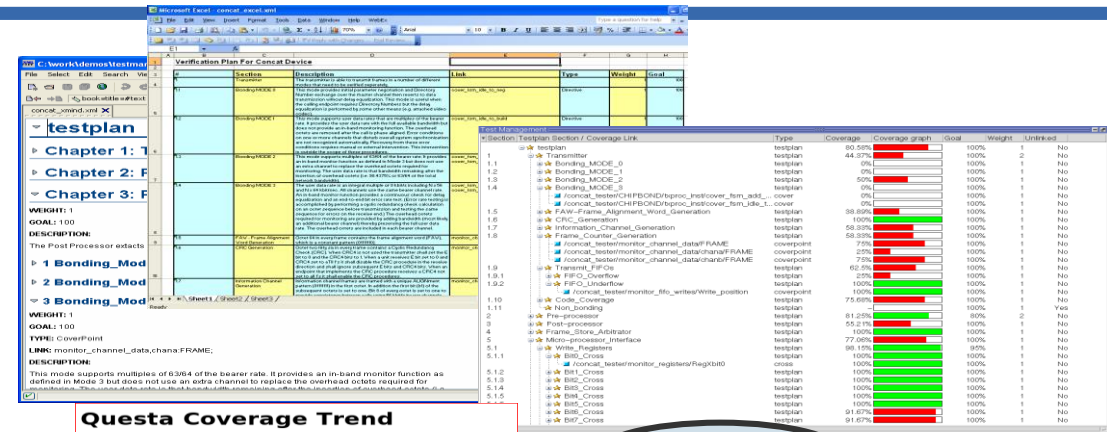
- Monitor progress throughout the project
- Mitigate verification risks
- Help improve future project timescale estimations

■ Results Analysis

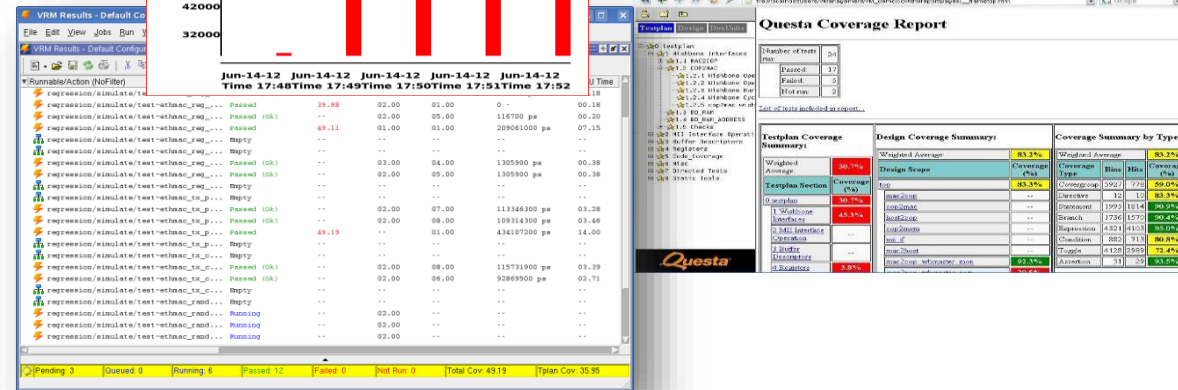
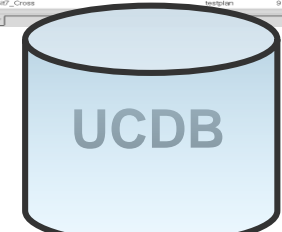
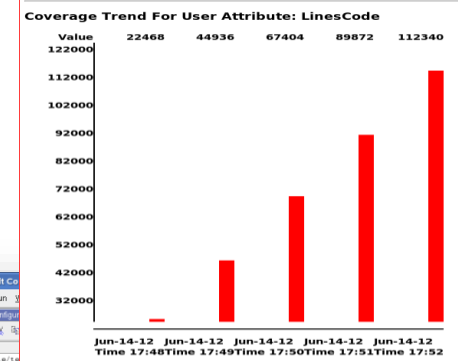
- Quickly Identify and Organize Results
- Improve debug turn-around time

■ Verification Run Manager

- Automate all regression tasks and reporting
- One interface to control all



Questa Coverage Trend



Jenkins & Questa Verification Run Management

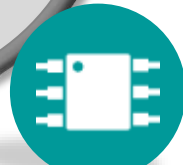
Continuous integration and reporting for advanced regressions management



Simulation



Emulation



Formal

HTML Reports
VRM & Coverage

Test Results
Native Jenkins
Full Questa VRM

Build Summary
Expandable

Project Ethmac
Ethernet MAC Controller

Latest Questa VRM Report
Latest Questa Coverage Report
Workspace
Recent Changes
Latest Test Result (3 failures / +2)
Latest Regression Result (no non-test failures)

Questa Results Summary

Build	Duration	Passed	Failed	Skipped	Total CPU Time	Total Coverage	Testplan Coverage
#213	6 min 9 sec	139	3	1	143 3 hr 4 min	69.3561	56.3910
#212	6 min 17 sec	126	1	1	128 2 hr 51 min	69.3561	57.5431
#211	4 min 13 sec	105	0	0	105 2 hr 25 min	69.3588	57.6413
#210	5 min 46 sec	88	2	0	90 2 hr 57 min	69.3605	56.4361
#209	7 min 26 sec	84	6	0	90 2 hr 22 min	69.4030	24.5018

Test Result

Questa Coverage Results

Test Statistics

All Failed Tests

All Tests

Test Result Trend

Build Time Trend

Coverage Result Trend

Attribute Graphs

Trend Graphs
Test results
Build times
Coverage code/functional

Custom Trends
Any user attribute

FINAL THOUGHTS

Proven Benefits from Advanced Verification

FPGA customers realize the benefits of evolving their verification methodologies

SystemVerilog UVM/UVMF

- Constrained random for better testing
- Functional coverage for better insight
- Consistent methodology
- Promotes Reuse

Verification IP

- Protocol expertise
- Reduces TB development effort
- Risk reduction

Formal

- Find bugs at design time
- Identify unreachable code
- Exhaustive testing

FPGA Customer Success Stories

Increased business due to shorter design cycles

400% ROI after adopting UVM and VIP

5 straight FPGA's without a bug

Zero bugs found in lab since moving to UVM

3x Speed up verification cycle with Formal Apps

Higher quality code

Where Can You Go To Learn More?

■ Verification Academy

- <https://verificationacademy.com/>
- Most comprehensive verification resource in industry
- Free online courses, verification cookbooks, discussion forums
- More than 40,000 members

■ Functional Verification at www.mentor.com

- <https://www.mentor.com/products/fv/>
- Learn more about tools to enable methodologies
- Simulation, SV/UVM, Formal, Verification IP
- Training catalog
- Success Stories
 - <https://www.mentor.com/products/fv/success/>

The image shows a screenshot of the UVM Cookbook website. The top part features a 'Methodology' diagram with a central 'DUT / Connection' block. To its left is a 'Test Environment' block containing 'params', 'config', 'register model', 'bus vip', and 'xVC'. To its right is a 'Test Stimulus' block with 'virtual sequencer' and 'Sequences'. Below the DUT is a 'Verification IP' block with 'driver', 'seq', 'mon', 'AP', 'config', and 'cov'. A vertical arrow on the left indicates 'Reuse' and 'Interop' between 'model' and 'scoreboard' blocks. The bottom part of the screenshot shows the 'Mentor Verification IP' product page, which includes a search bar, a 'Learn More' button, and a list of features: '1000x performance improvement using common verification IP', '24/7 enterprise-level datacenter resource for emulation and simulation', '25x productivity boost with unified coverage and analysis', '10x productivity improvement in debug', 'Full testbench reuse across simulation and emulation', and 'High-speed automated Formal applications for targeting verification tasks'. A navigation menu at the bottom lists 'Memory Library', 'Veloce Emulation', 'Verification IP', 'Veloce Power Application', and 'Verification Horizons'.

Mentor[®]
A Siemens Business

www.mentor.com