

TWENTY-FOURTH ANNUAL



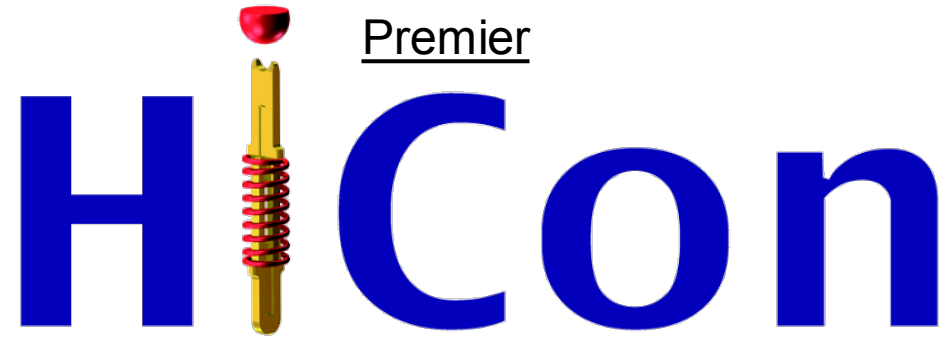
TestConX™

Archive

DoubleTree by Hilton
Mesa, Arizona
March 5-8, 2023

With Thanks to Our Sponsors!

Premier



Honored



With Thanks to Our Sponsors!

Distinguished



Industry Partners



COPYRIGHT NOTICE

The presentation(s) / poster(s) in this publication comprise the Proceedings of the TestConX 2023 workshop. The content reflects the opinion of the authors and their respective companies. They are reproduced here as they were presented at the TestConX 2023 workshop. This version of the presentation or poster may differ from the version that was distributed at or prior to the TestConX 2023 workshop.

The inclusion of the presentations/posters in this publication does not constitute an endorsement by TestConX or the workshop's sponsors. There is NO copyright protection claimed on the presentation/poster content by TestConX. However, each presentation / poster is the work of the authors and their respective companies: as such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

“TestConX”, the TestConX logo, and the TestConX China logo are trademarks of TestConX. All rights reserved.

www.testconx.org

Portable Stimulus Test Development and Execution on ATE Systems

Marcus Schulze Westenhorst [Advantest]

Markus Buecker [Advantest]

Klaus Dieter Hilliges [Advantest]

Mate Burazin [Advantest]

Michael Braun [Advantest]

Jörg Simon [Cadence]

Dave Armstrong [Advantest]



Mesa, Arizona • March 5–8, 2023



Agenda

- Motivation - Why do we need a new type of methodology on ATE Systems?
- Next generation ATE system
- PSS test case generation
- PSS Test Case Execution on ATE
- Analysis and Debugging
- PSS Test Development in Action on DUT via ATE
- Conclusion

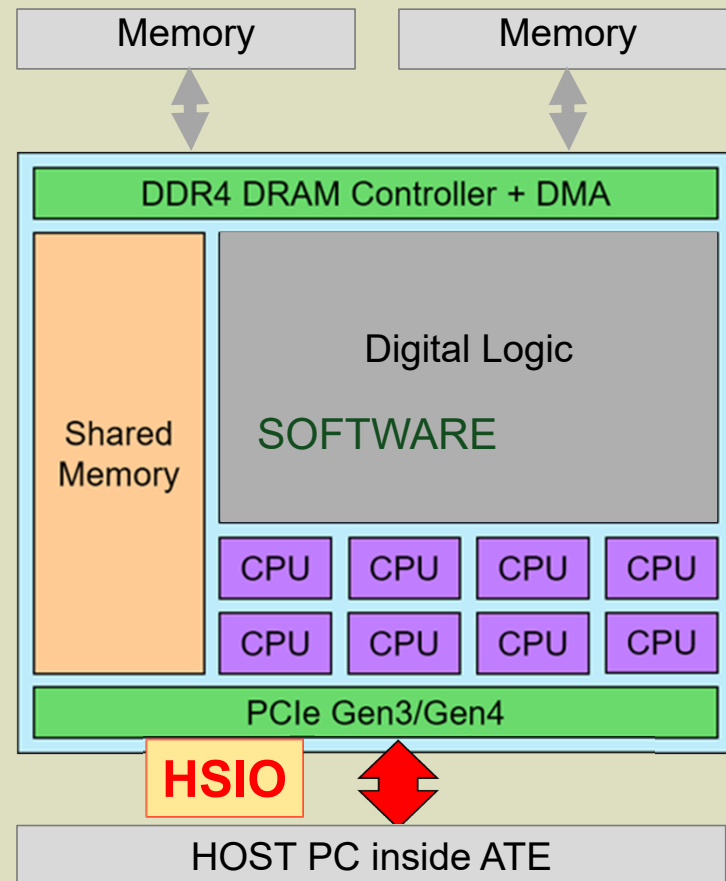


Portable Stimulus Test Development and Execution
on ATE Systems

2023

TestConX 2023

The Expanding Role of HSIO in Device Testing



Many devices today have HSIO interfaces on them.

External loopback testing of HSIO is wasteful:

- It does little confirm test margins.
- It delays testing of HSIO enumeration until SLT causing nearly completed parts to be scraped.
- It thwarts a high-speed path into the DUT to perform other testing – forcing used of inferior interfaces.

Using the HSIO interface in its native way, three different types of tests are possible:

- **SW-based, bare-metal functional test (PSS Enabled)**
- **SLT-Like functional testing on ATE (Shift Left)**
- **Scan-over-HSIO**



Portable Stimulus Test Development and Execution
on ATE Systems

2023

MOTIVATION

SW-BASED, BARE-METAL FUNCTIONAL TEST (PSS)

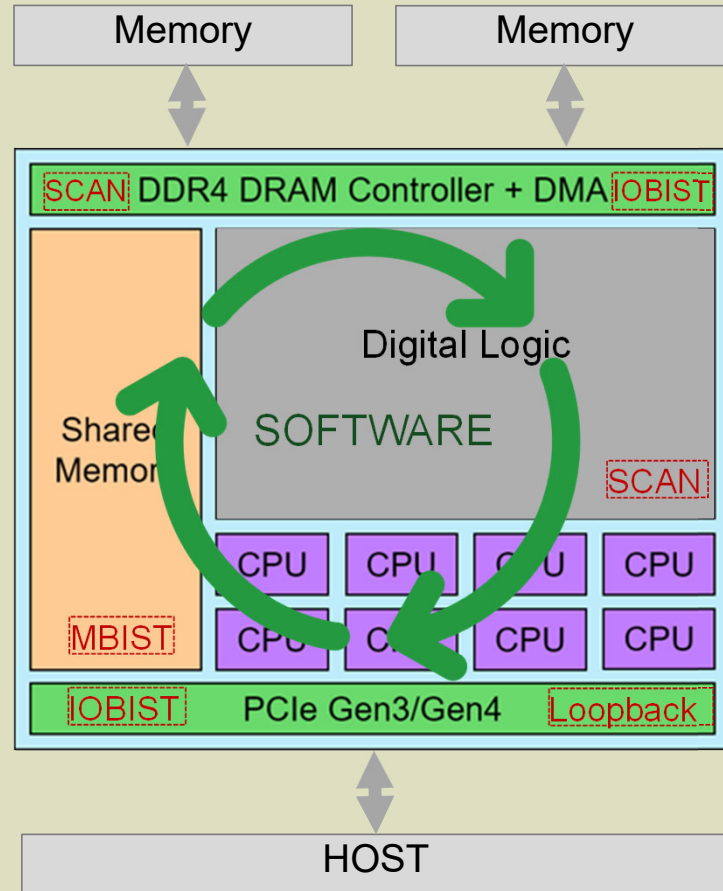


Portable Stimulus Test Development and Execution
on ATE Systems

4 **2023**

TestConX 2023

Need to Re-invent Functional Testing on Production ATE



Complex devices have many cores

- They work together in the application
- They do not work together on the tester
- SCAN, BIST etc. are core-based tests

Good defect coverage of the cores, but:

- Device software is not part of the test
 - No interaction between the cores and host
- ⇒ Test Coverage Gap for functional mode

How to close this gap ?

- Complex functional test cases across cores
- System-like interaction with the external Host
- Software – Device Firmware and Host SW

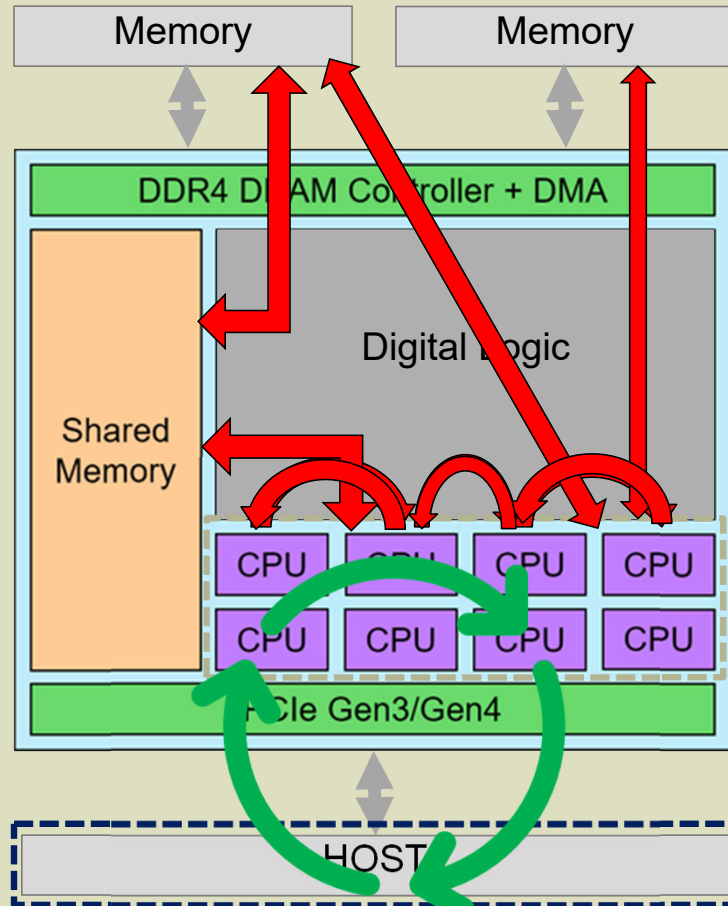


Portable Stimulus Test Development and Execution
on ATE Systems

2023

TestConX 2023

Software-Based Functional Test Closes the Coverage Gap



Software for a real functional test is

- (1) Device firmware which runs on the DUT
- (2) Supporting software which runs on the HOST

Both need to work together for full operation

(1) Device firmware

- Enables functional operation
- Drives interaction of cores and host
- Typically booted from the host

(2) HOST software

- Operating system with custom drivers
- User application software



Portable Stimulus Test Development and Execution
on ATE Systems

2023

Software-Driven Functional Test: The Big Picture

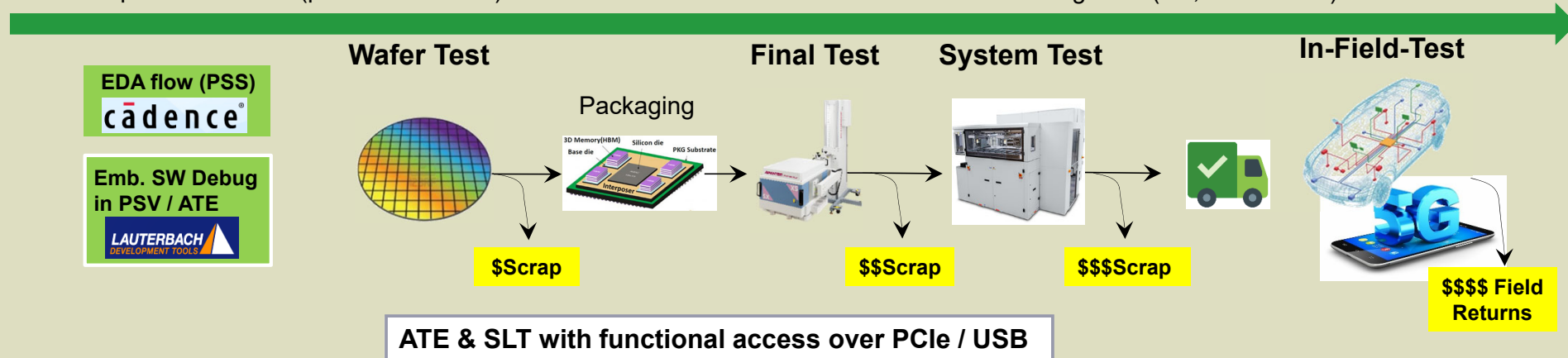
Forward flow – systematic approach for predictable TTM and TTQ

Leverage pre-silicon test content & methodology:

- ⤵ Software test cases from EDA (i.e., Cadence)
- ⤵ Example format: PSS (portable stimulus)

Bring-up & debug of full software stack ON WAFER

- ⤵ Boot device, run (modified) application software
- ⤵ Use standard debug tools (i.e., Lauterbach)



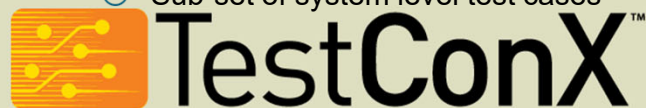
Backward flow – if required due to excessive SLT yield-loss and field returns

Add test coverage as early as possible:

- ⤵ Targeted “Bare Metal” (no OS) tests
- ⤵ Sub-set of system level test cases

Diagnose SLT and field returns:

- ⤵ Correlation between different insertions now possible
- ⤵ Example: structural / functional (SCAN vs SW-based)



Portable Stimulus Test Development and Execution on ATE Systems

2023

APPROACH - NEXT GENERATION ATE SYSTEM

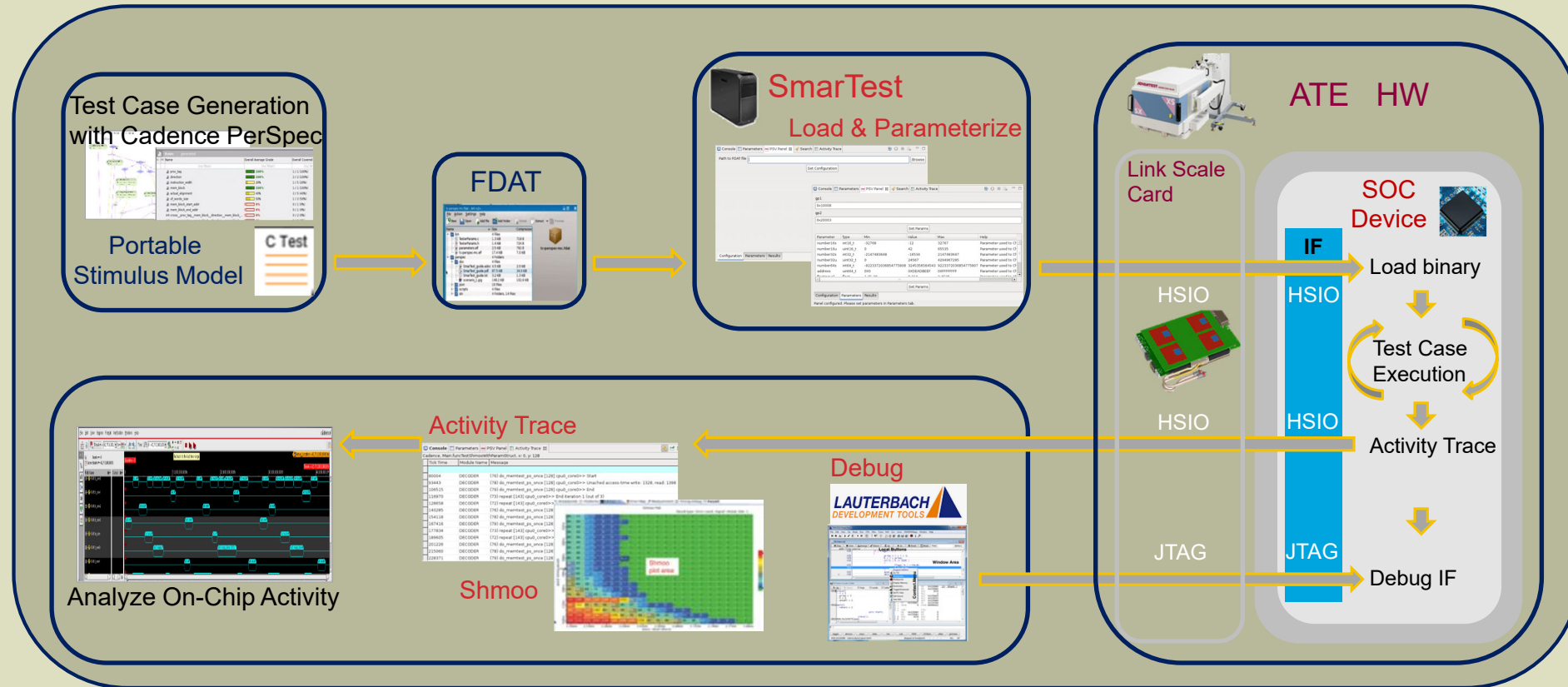
*BOOTING AND EXECUTING PORTABLE STIMULUS STANDARD BASED
FUNCTIONAL TESTS ON AN ATE (AUTOMATED TEST EQUIPMENT) SYSTEM*



Portable Stimulus Test Development and Execution
on ATE Systems

8 **2023**

Full Integrated Functional PSS Driven SoC Validation Flow



Portable Stimulus Test Development and Execution on ATE Systems

PSS Model Overview



Portable Stimulus Standard (PSS) defined accellera standard



- **Platform** platform independent description of test scenario intend for improved re-use
- **User** abstract modelling of test intention to create valid use cases / scenarios
- **Software** functional software driven approach for SoC verification, validation and testing

PSS Tooling

- **Language** supporting accellera standard
- **Test Case Creation** create valid test cases with constraint-random capabilities out of the PSS model
- **Coverage** enable coverage of complex SoC beyond individual blocks
- **Portability** powerful test generation engines creating test cases e.g. for UVM or embedded test use cases

Multiple aspects to consider planning the PSS model

- **Framework** user / compiler, toolchains / available HW / internal policies / debug / model maintenance
- **Message** output handling (fast-messages concept) – interface (streaming/dump required to be clarified)
- **Functional add-ons** (cache coherency, stress tests, power modeling) compared to pure traditional testing
- **Technology library** integration e.g. coherency, power ... and more
- **Post-Si control knobs** for tester infrastructure to enable variants of functional behavior
- **Peripheral control** combinations of external physical and internal functional parameters
- **Coverage** metric introduction – generation time coverage and runtime-coverage addons
- **Documentation** of model and use instructions
- **Regression** preparation/setup and execution
- **Statistical** data collection preparation – alignment with messaging



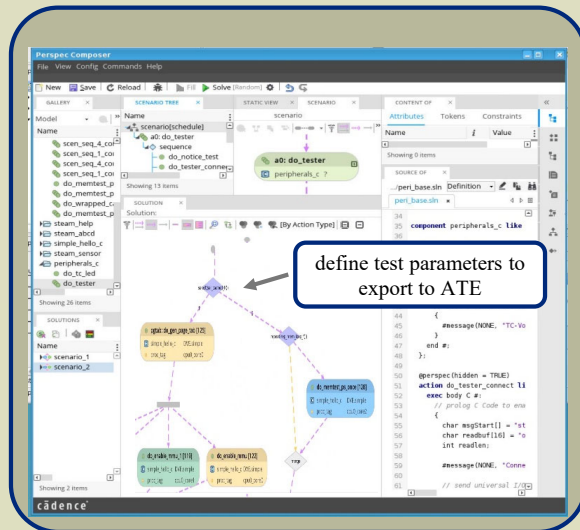
Portable Stimulus Test Development and Execution
on ATE Systems

2023

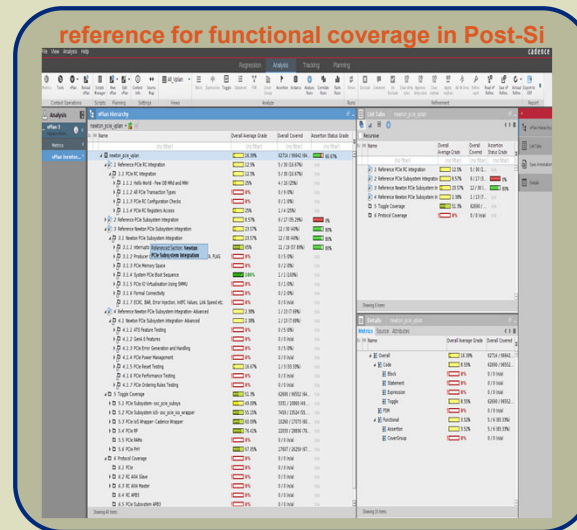
Interactive Approach - PSS Model Controls Post-Silicon

- PSS model: operator-based control infrastructure will be provided for post-si
- Typical control knobs: address, iteration counts, un/cache decisions ... and more
- Could be simple value deployment or complex conditional statements (runtime)

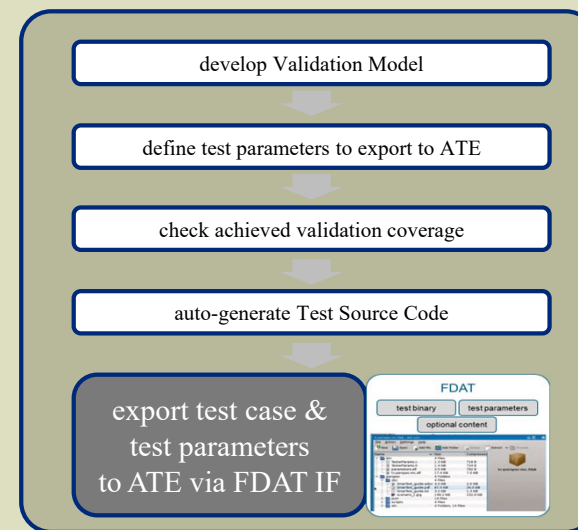
PSS model developed in PSS tool



Coverage Metric gen-time or runtime



Workflow in PSS tooling



Portable Stimulus Test Development and Execution on ATE Systems

2023

TestConX 2023

Key Capabilities of The New ATE Instrument

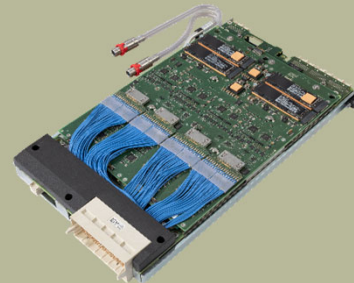
What is needed ?

- Functional HSIO support
- Local processing power (Host)
- Enough local memory
- HW integrated in the test head
- Open SW environment

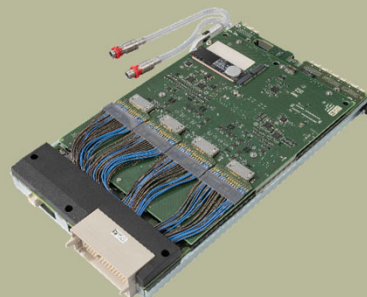
Target capabilities

- USB / PCIe full protocol stack
- Host processor for local processing
- Local memory for test data
- Multi-site capable
- Fully integrated

„Link Scale USB“



„Link Scale PCIe“



V93000 Smart Scale
V93000 EXA Scale



Portable Stimulus Test Development and Execution
on ATE Systems

12

2023

APPROACH – PSS TEST CASE GENERATION

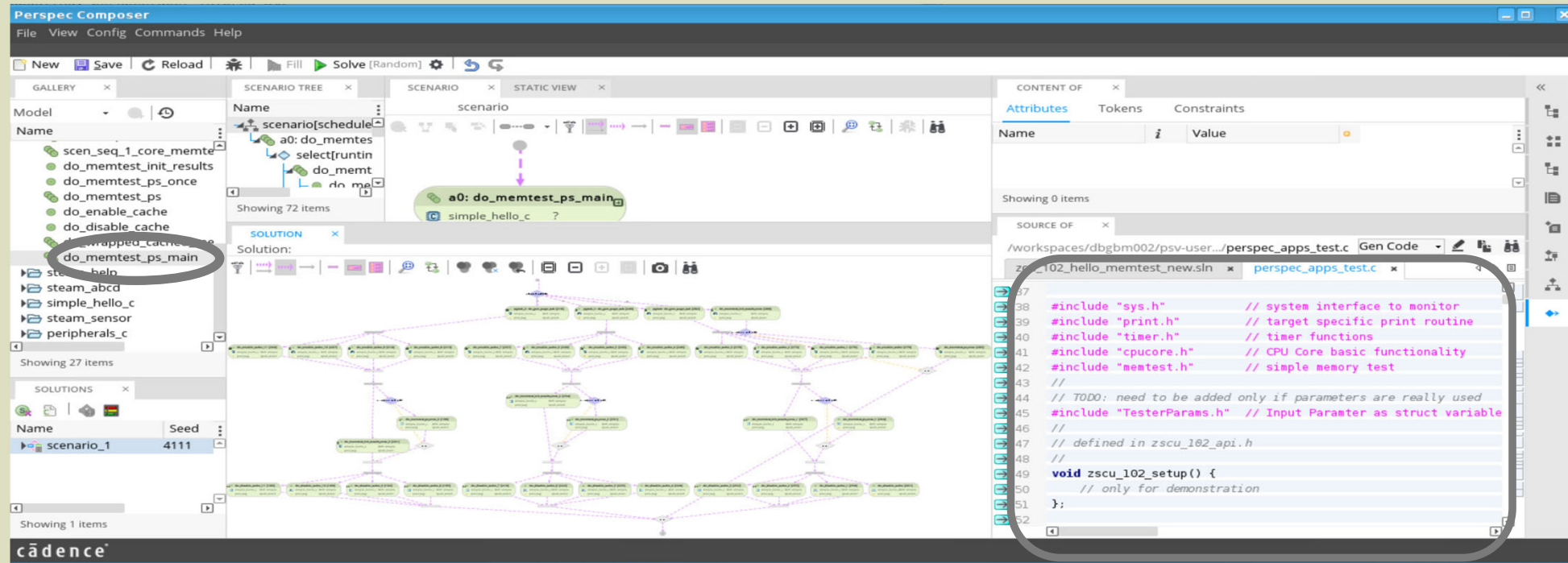
LEVERAGING POWERFUL TEST CONTENT FROM SILICON VALIDATION



Portable Stimulus Test Development and Execution
on ATE Systems

13 **2023**

Portable Stimulus Standard with Cadence Perspec System Verifier



Test intent – captured in abstract PSS model (library) already
C-Code (instrumented for Post-Si) – generated in batch/gui mode



Portable Stimulus Test Development and Execution
on ATE Systems

APPROACH – PSS TEST CASE EXECUTION ON ATE

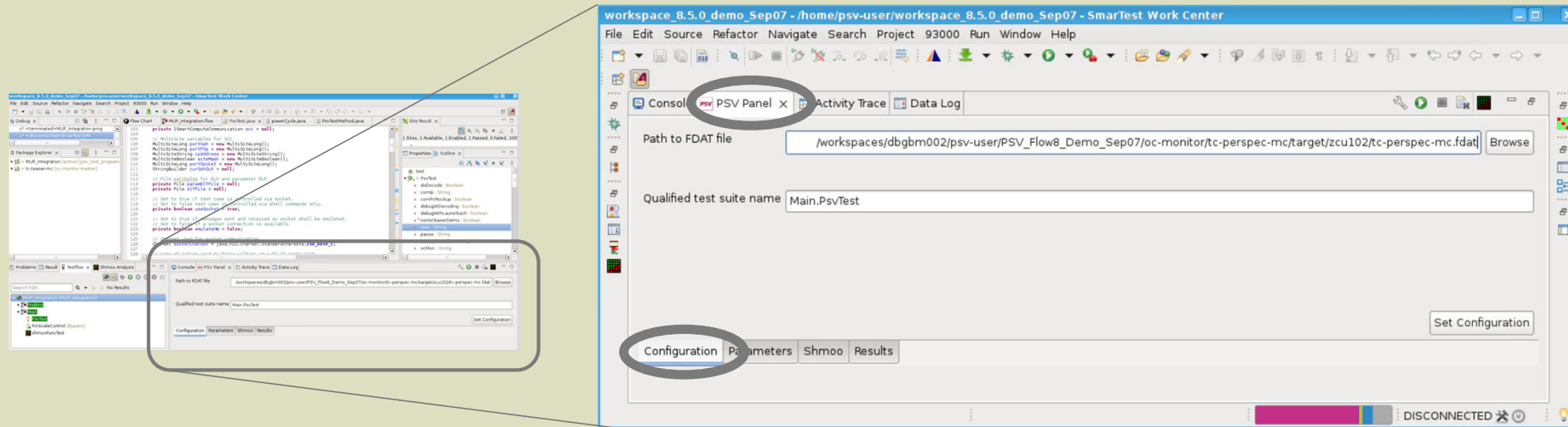
A SMARTTEST PLUG-IN FOR PSS TEST CASE EXECUTION AND RESULT RETRIEVAL



Portable Stimulus Test Development and Execution
on ATE Systems

15 **2023**

Test Control Panel as Prototype Plugin



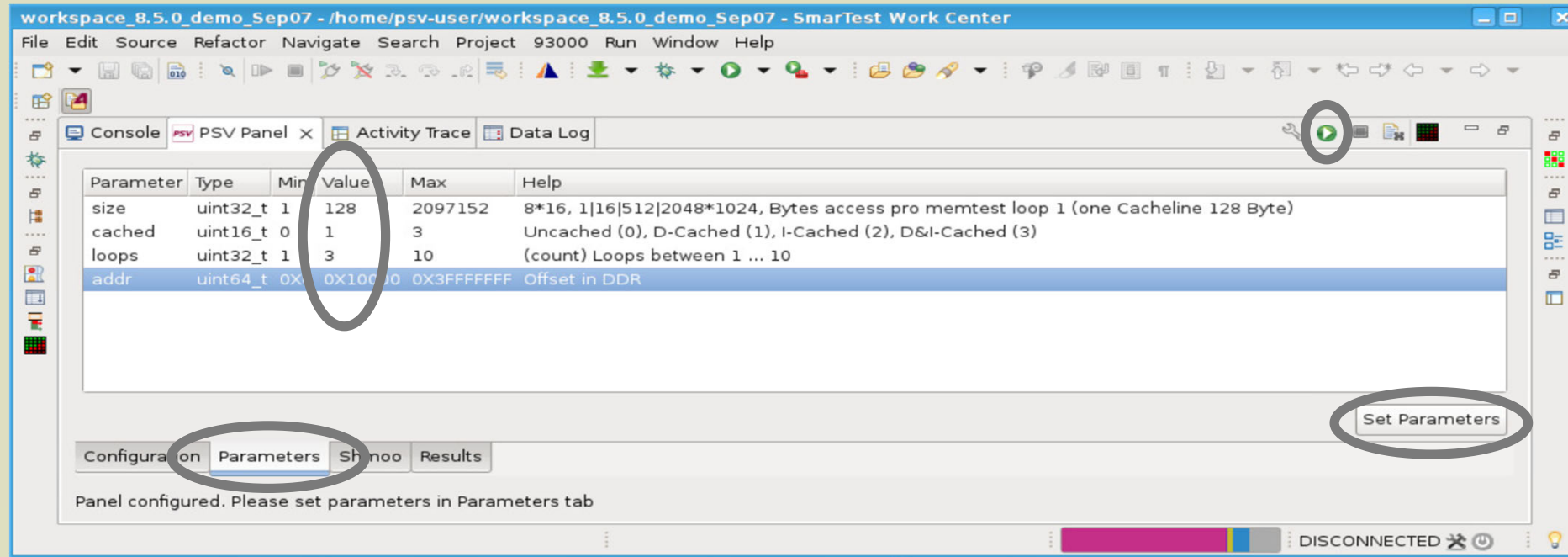
No Pattern conversion anymore – instead, use of FDAT that contains PSS test case executable.
Simplified GUI, for test case handling and execution of FDAT content.



Portable Stimulus Test Development and Execution
on ATE Systems

16 **2023**

Dynamic Parameter Passing to Test Case



On the fly parameter modification and execution without test case re-compilation. PSS model was prepared accordingly.



Portable Stimulus Test Development and Execution
on ATE Systems

Results and Activity Trace – Events Executed on DUT

The screenshot shows the SmarTest Work Center interface. The 'Results' tab is active, displaying a green bar with the word 'PASS' circled in red. Below this, the 'Outputs' section shows detailed log information for an Open On-Chip Debugger. The 'Activity Trace' tab is also visible, showing a table of events with columns for Tick Time, Module Name, and Message.

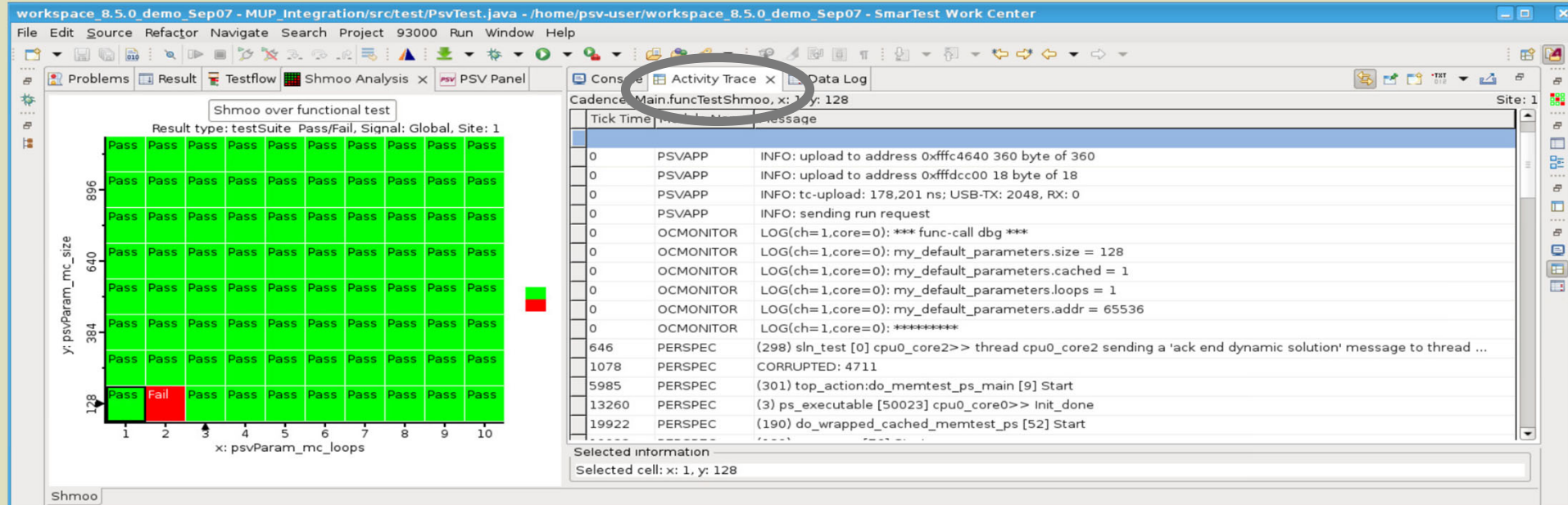
Tick Time	Module Name	Message
72137	PERSPEC	(198) do_enable_cache [279] cpu0_core0>> *** Enable D-Cache ***
75279	PERSPEC	(225) do_enable_cache_1:do_enable_cache [276] cpu0_core1>> *** Enable D-Cache ***
79348	PERSPEC	(260) do_enable_cache_3:do_enable_cache [270] cpu0_core3>> *** Enable D-Cache ***
83170	PERSPEC	(208) do_memtest_ps [123] Start
83170	PERSPEC	(248) do_memtest_init_results_once_1:do_memtest_init_results_once [267] cpu0_core2>> Start
87358	PERSPEC	(201) do_enable_cache [279] cpu0_core0>> End
90943	PERSPEC	(263) do_enable_cache_3:do_enable_cache [270] cpu0_core3>> End
95216	PERSPEC	(228) do_enable_cache_1:do_enable_cache [276] cpu0_core1>> End
111144	PERSPEC	(233) do_disable_cache_1:do_disable_cache [258] cpu0_core1>> *** Disable D-Cache ***
111464	PERSPEC	(207) repeat [325] cpu0_core0>> Start
116652	PERSPEC	(234) do_disable_cache_1:do_disable_cache [258] cpu0_core1>> *** Disable I-Cache ***
119512	PERSPEC	(209) repeat [325] cpu0_core0>> Start iteration 1 (out of 1)
127619	PERSPEC	(276) do_memtest_ps_once_1:do_memtest_ps_once [264] cpu0_core0>> Start
470863	PERSPEC	(277) do_memtest_ps_once_1:do_memtest_ps_once [264] cpu0_core0>> Cached access-time write: 211, cache...
475930	PERSPEC	(279) do_memtest_ps_once_1:do_memtest_ps_once [264] cpu0_core0>> End

Execution and high-speed download of PSS related result log. First level of debug.



Portable Stimulus Test Development and Execution
on ATE Systems

Shmoo Results Over Parameter Variations



Fully featured Shmoo analysis tool.
Overview on DUT characteristic accross full parameter range with complete set of result logs.

Test case binary loaded only once, parameter exchange with each test execution.



Portable Stimulus Test Development and Execution on ATE Systems

APPROACH – ANALYSIS AND DEBUGGING

DIVING ONE LEVEL DEEPER



Portable Stimulus Test Development and Execution
on ATE Systems

20 **2023**

First Analysis with Cadence Perspec Analyzer Environment

Post execution analysis:

Activity Trace

Test case model

C - Code

Activity over Time on multiple cores

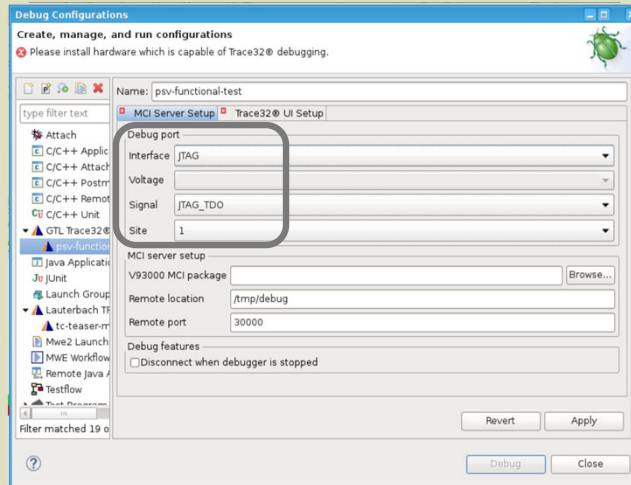
The screenshot displays the Cadence Perspec Analyzer interface. At the top, the 'Simulation Control' panel shows the current time at 72,019 ns. The 'GRAPHS' panel shows a call graph for 'do_memtest_ps_main_1'. The 'SOURCE' panel shows the C code for 'perspec_apps_test.c', with a 'while' loop and cache enable/disable logic. The 'MESSAGE' panel shows a list of system messages from the PERSPEC component, including cache enable and disable actions. The 'WAVEFORM' panel at the bottom shows a timeline of activity for four CPU cores (cpu0_core1, cpu0_core0, cpu0_core2, cpu0_core3), with bars representing the execution of various test case functions like 'do_enable_cache' and 'do_disable_cache'.



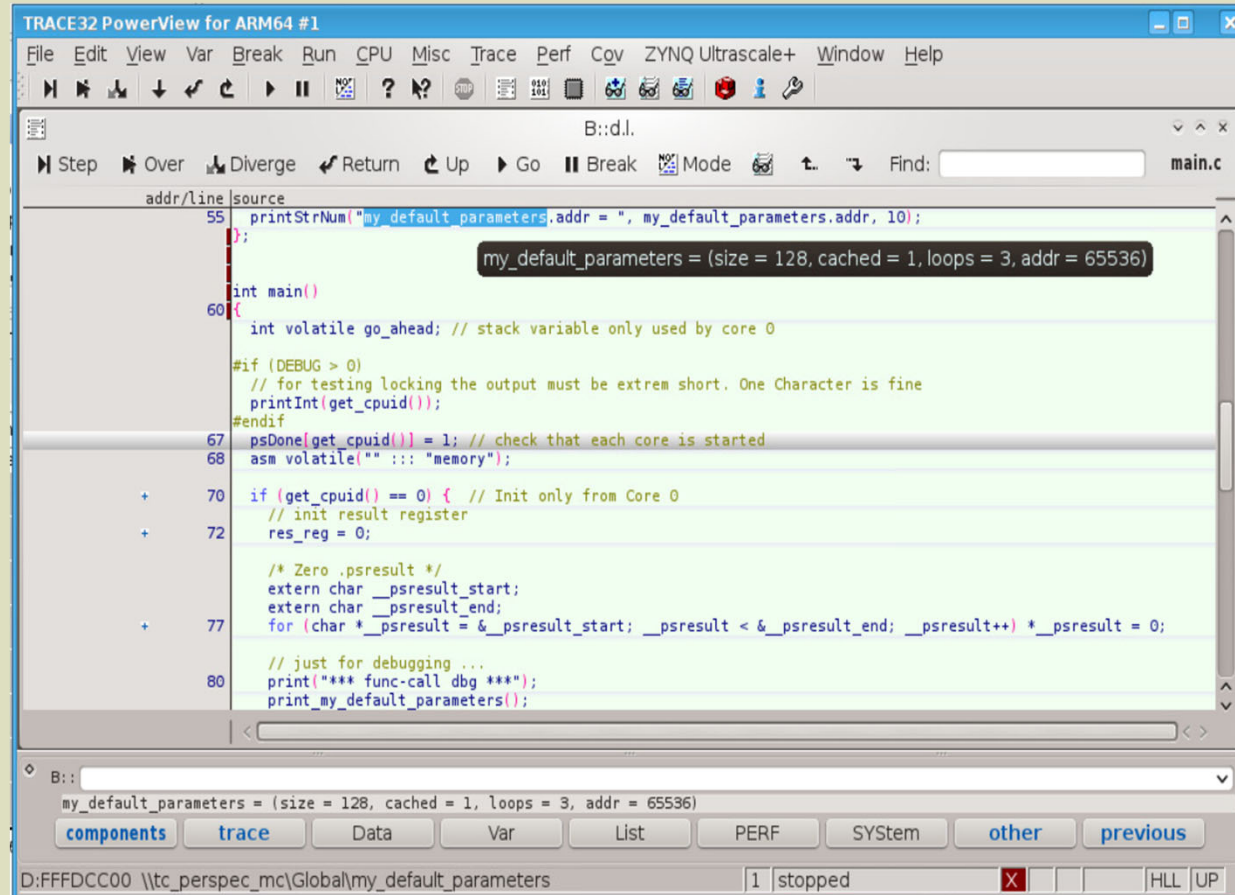
Portable Stimulus Test Development and Execution on ATE Systems

21 **2023**

Next Level of Debug – SW Debugger Configuration Panel



Integrated Lauterbach panel for instant and real time on target debugging



Portable Stimulus Test Development and Execution on ATE Systems

APPLICATION – PSS TEST DEVELOPMENT IN ACTION ON DUT VIA ATE

HW SETUP AND DUT VALIDATION CAPABILITIES



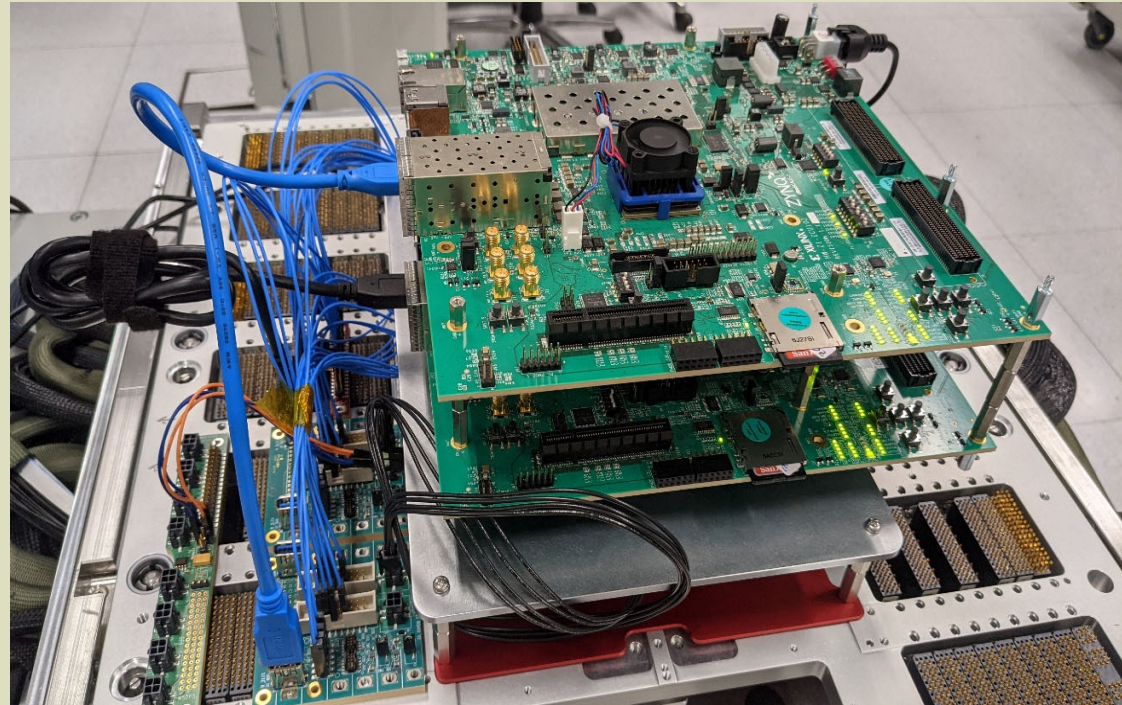
Portable Stimulus Test Development and Execution
on ATE Systems

23 **2023**

TestConX 2023

Functional Test Evaluation Setup

- Xilinx Eval Board as Validation Device on the V93000 tester
- HSIO connection to the Link Scale card in the test head
- Device boot and HSIO enumeration done by an image that is loaded via JTAG
- HSIO connection to exchange all test data (test case binary, parameter, results, traces, monitoring, ...) with fallback possibility to JTAG
- Debug capability over JTAG



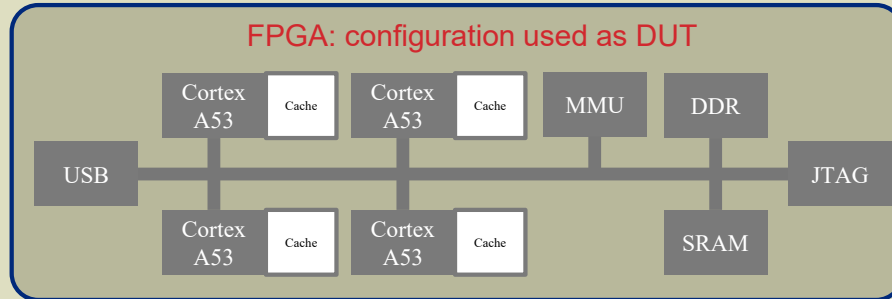
Portable Stimulus Test Development and Execution
on ATE Systems

24

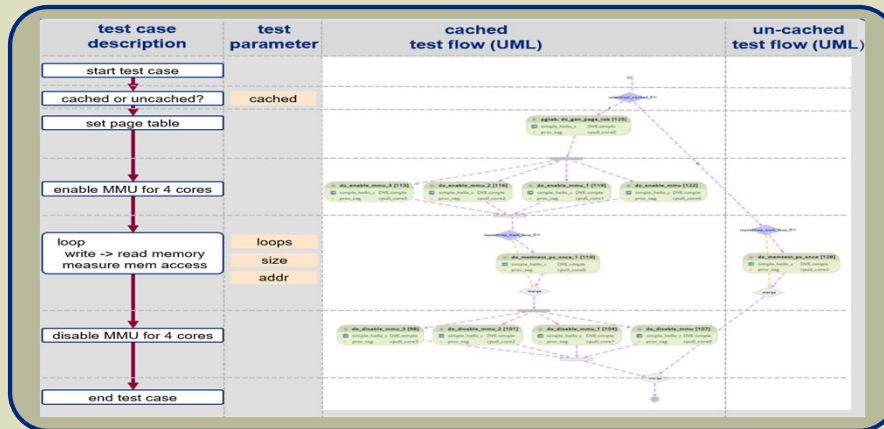
2023

Application – validation results achieved with ARM based FPGA

DUT: Xilinx Zynq-UltraScale+-MPSoC ZCU 102



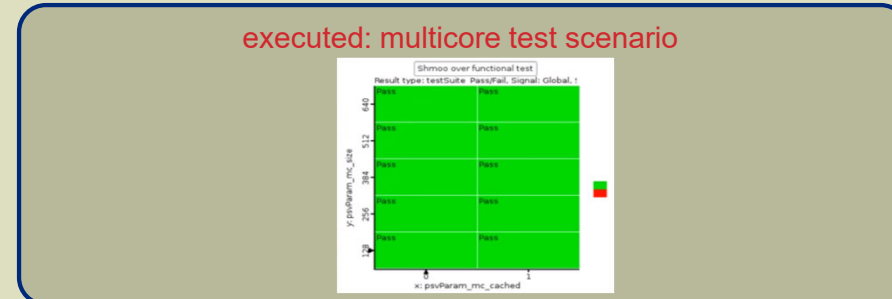
Implemented multicore test scenario



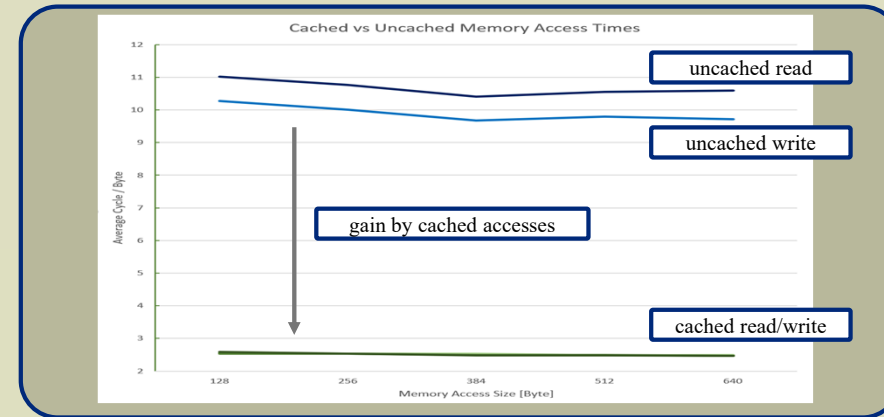
Complex Devices and Use Cases are supported



Shmoo: memory access size [byte] over cached / uncached



Memory access times post-processed from Activity Trace



Parameter variation and automation supporting design evaluation

Portable Stimulus Test Development and Execution on ATE Systems

2023

CONCLUSION

PORTABLE STIMULUS TEST DEVELOPMENT AND EXECUTION ON ATE SYSTEMS



Portable Stimulus Test Development and Execution
on ATE Systems

26 **2023**

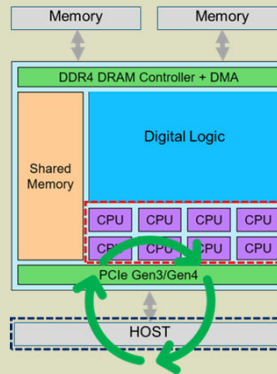
TestConX 2023

Conclusion

Link Scale cards enable new test methodologies

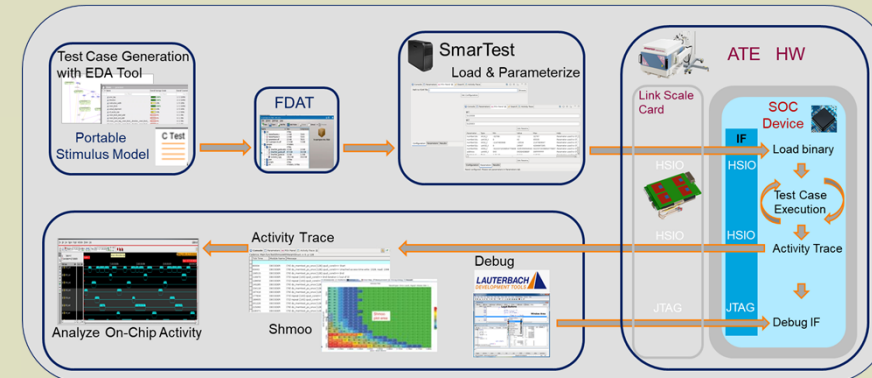
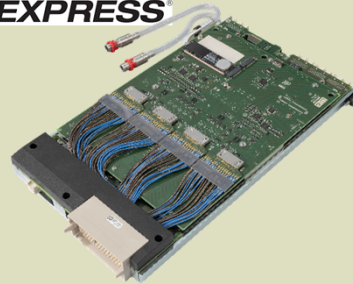
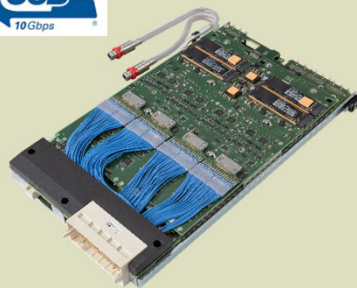
Software-based functional test

- Closing a test coverage gap
- Driving higher product quality
- Enabling known-good-die



Functional test generation with Perspec enables

- Fast turnaround time – from test case creation to execution and analysis in a matter of minutes
- No extra steps for file conversion
- Ease of test case execution
- Parameterization & Characterization
- Execution trace capture & evaluation in EDA



Portable Stimulus Test Development and Execution on ATE Systems

2023