Boosting Debug Productivity

Practical Applications of Debug Innovations in a UVM World

Joerg Richter
Director R&D, Verification Group

SYNOPSYS®

accellera SYSTEMS INITIATIVE

2017 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

Agenda

Motivation

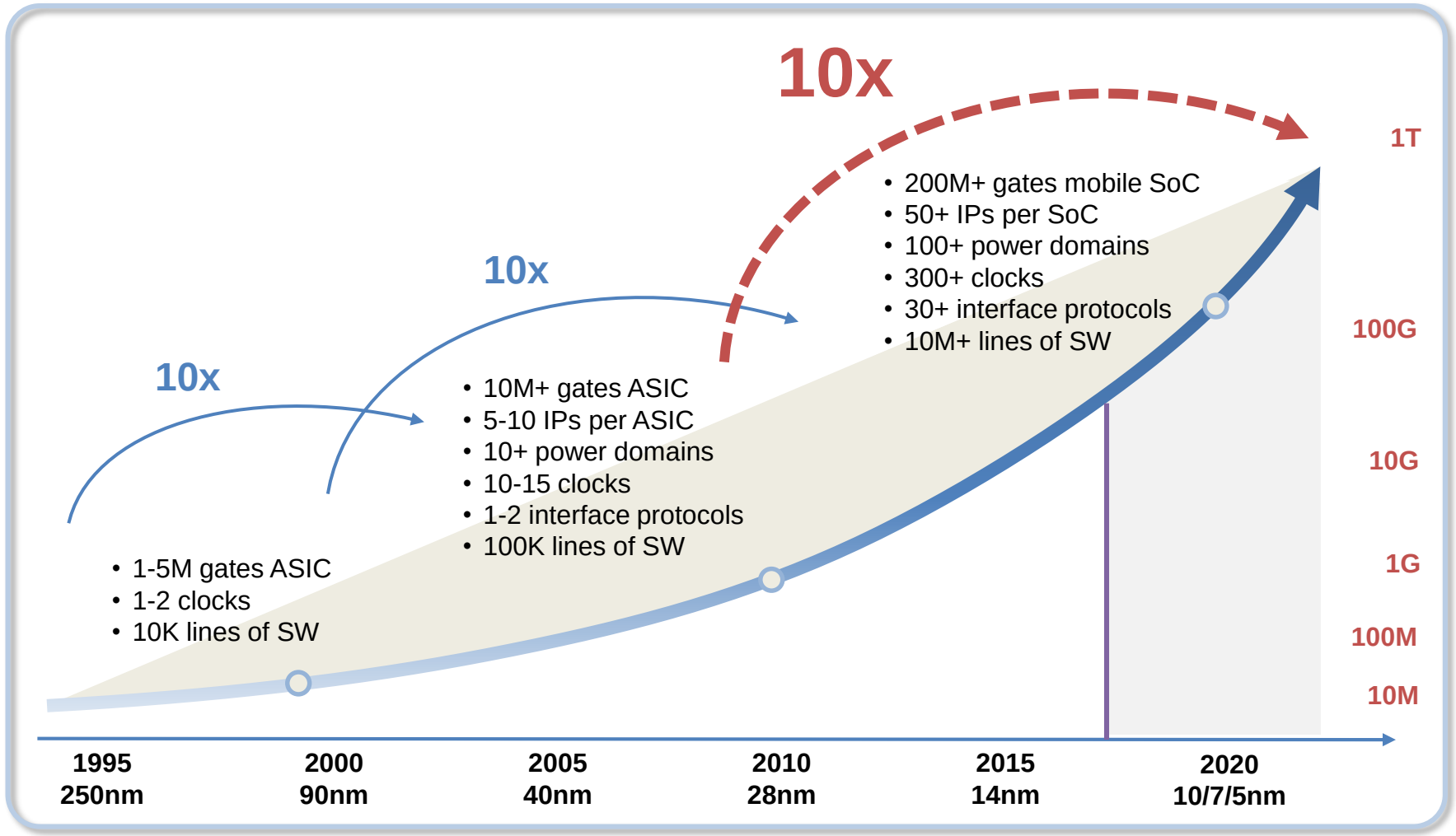
Efficient SV/UVM Testbench Debug

Rapid Root-Cause Analysis with Reverse Debug

Search-centric Debug with OneSearch

Embedded SW Debug

Verification Complexity Exponentially Increasing



Billion+ Gates SoCs

+

Functionality Power Performance

+

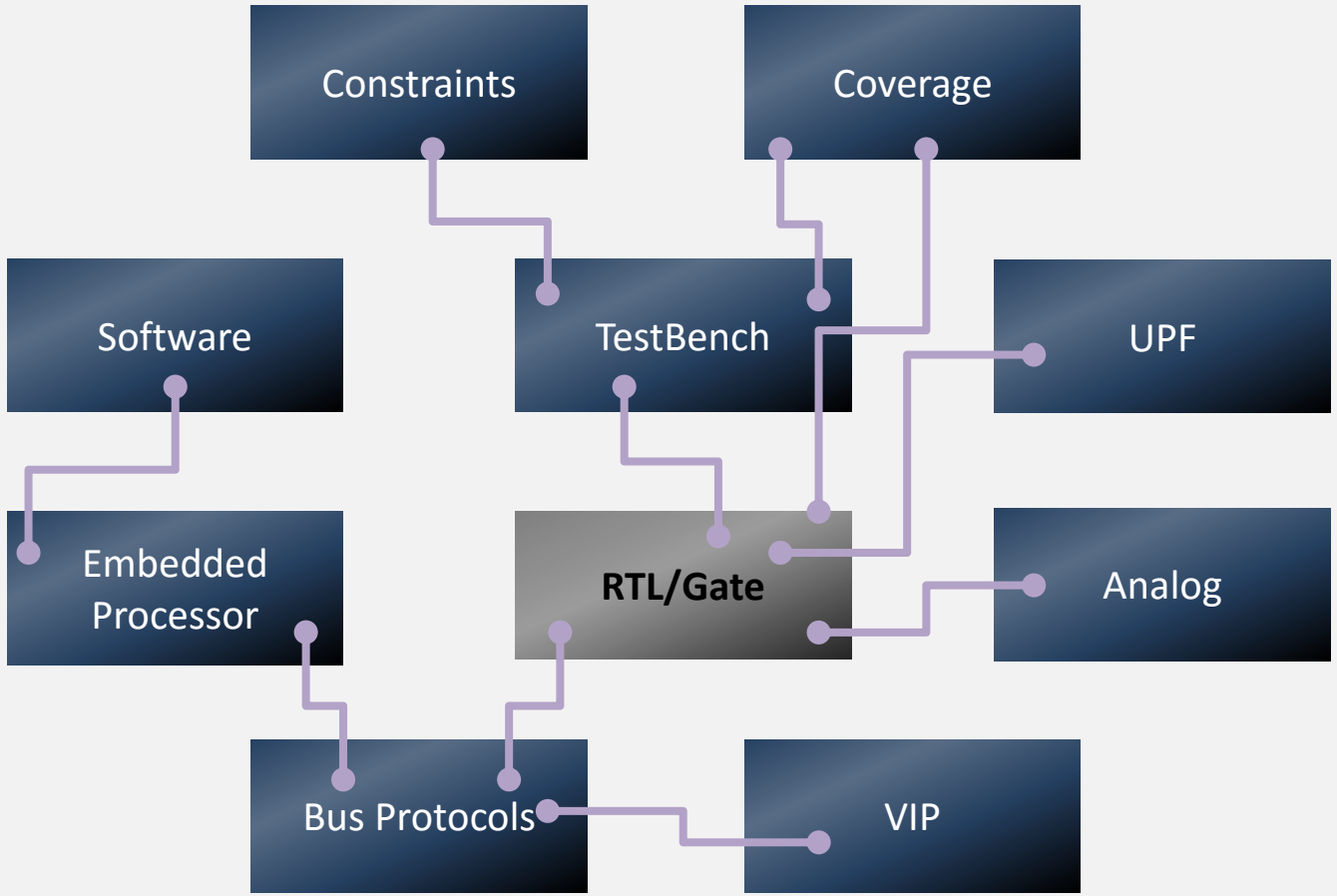
Block-level testing to SW Bring-up

+

Shrinking Time-to-Market Windows

Source: Synopsys

Scope of Debug Expanding



Agenda

Motivation

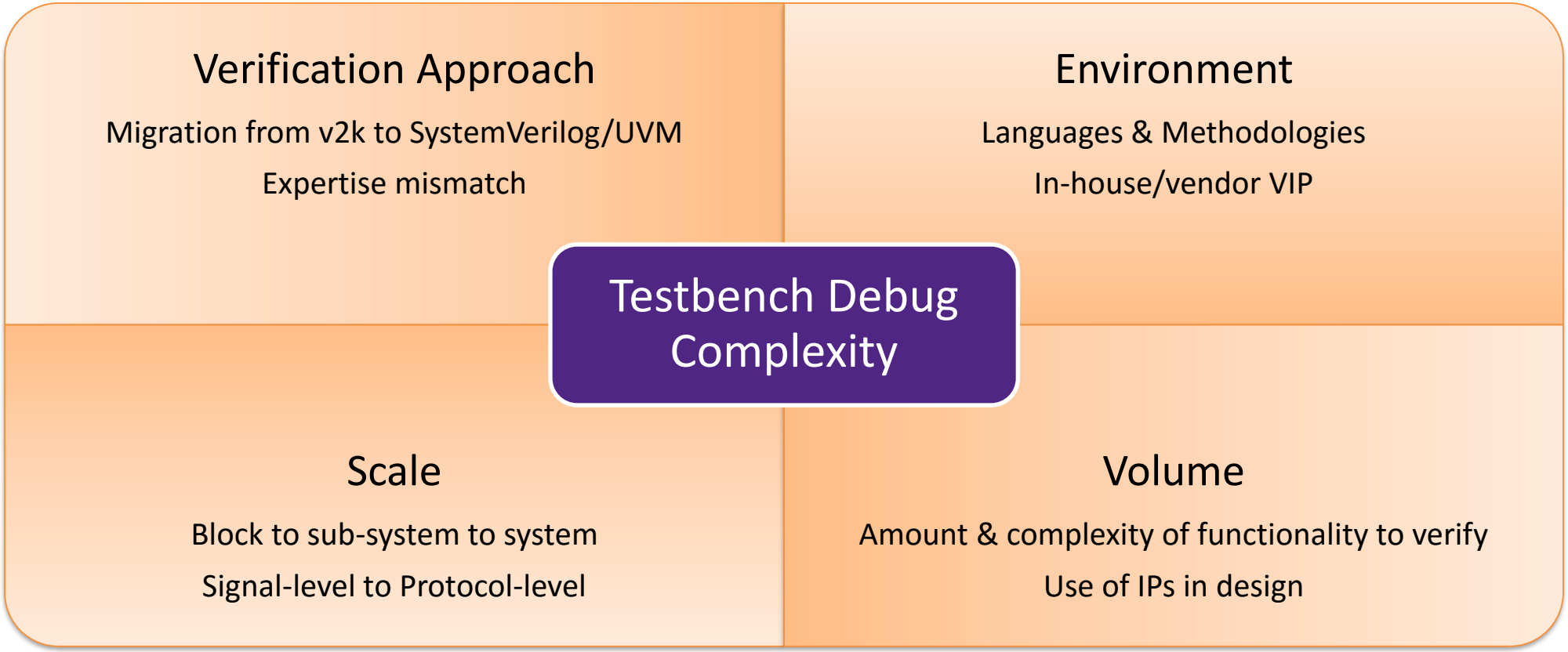
Efficient SV/UVM Testbench Debug

Rapid Root-Cause Analysis with Reverse Debug

Search-centric Debug with OneSearch

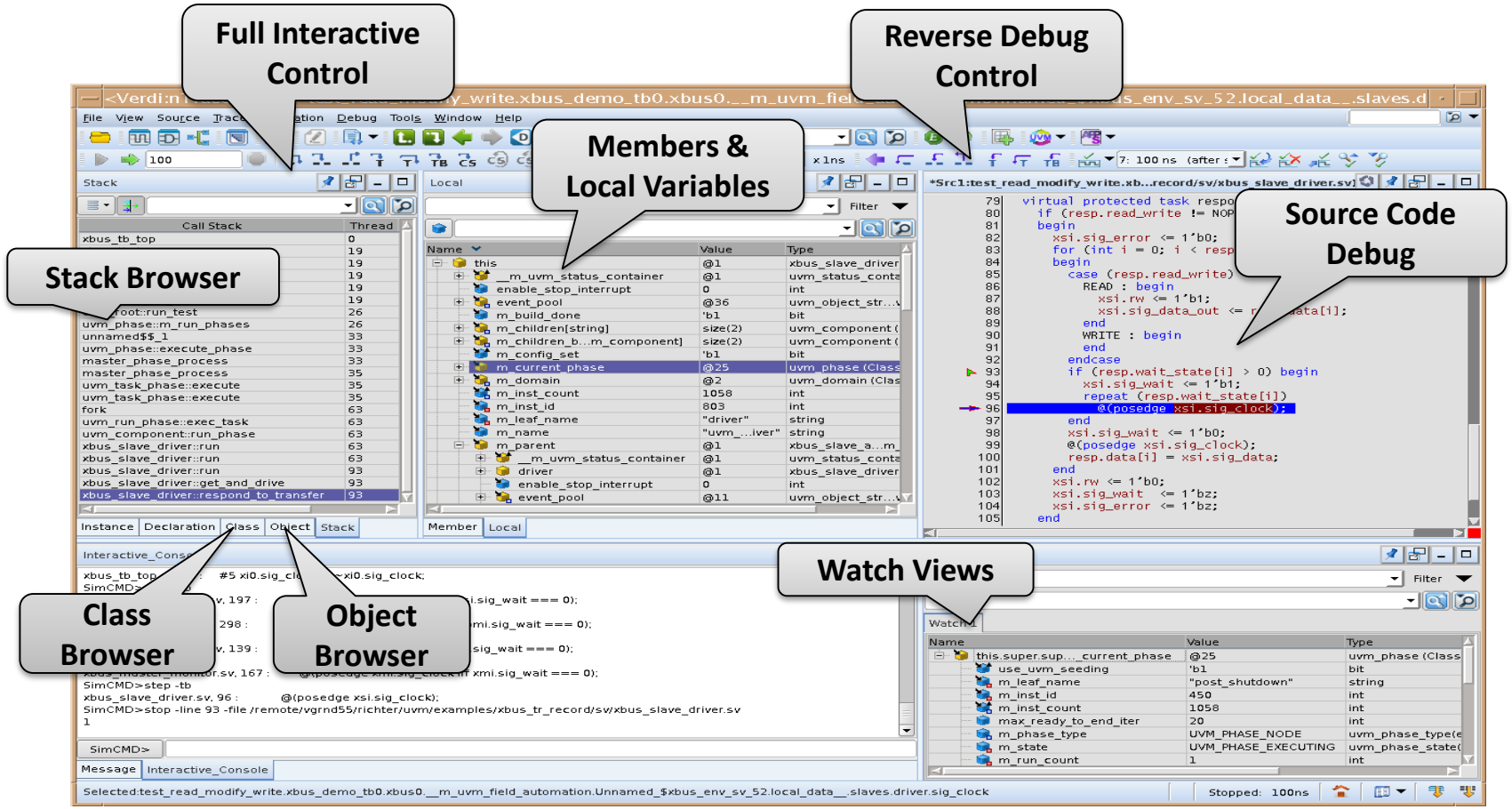
Embedded SW Debug

Testbench Debug Complexity



Debug Testbenches Interactively

Full-visibility Testbench Debug in Verdi & VCS

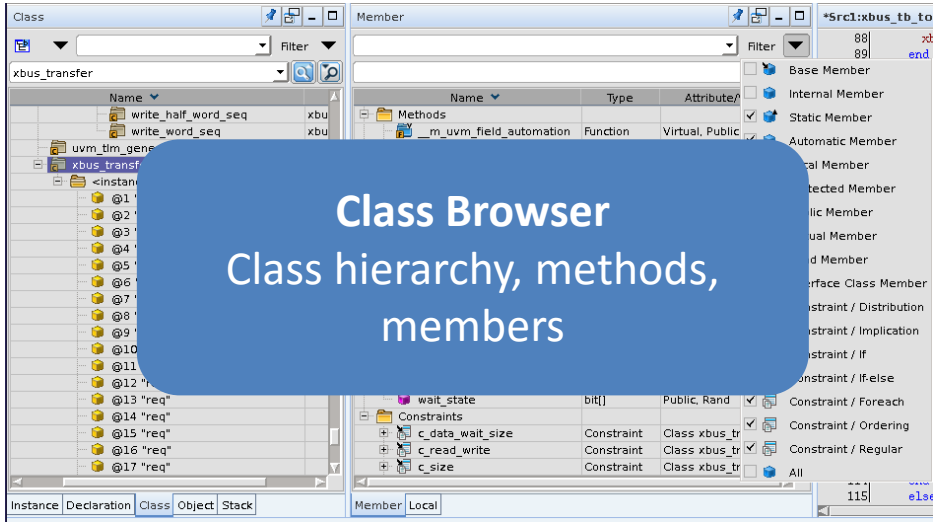


- **On-demand Debug**
- Step forward/back/into testbench constructs
- Go back in time without restarting simulation OR setting checkpoints

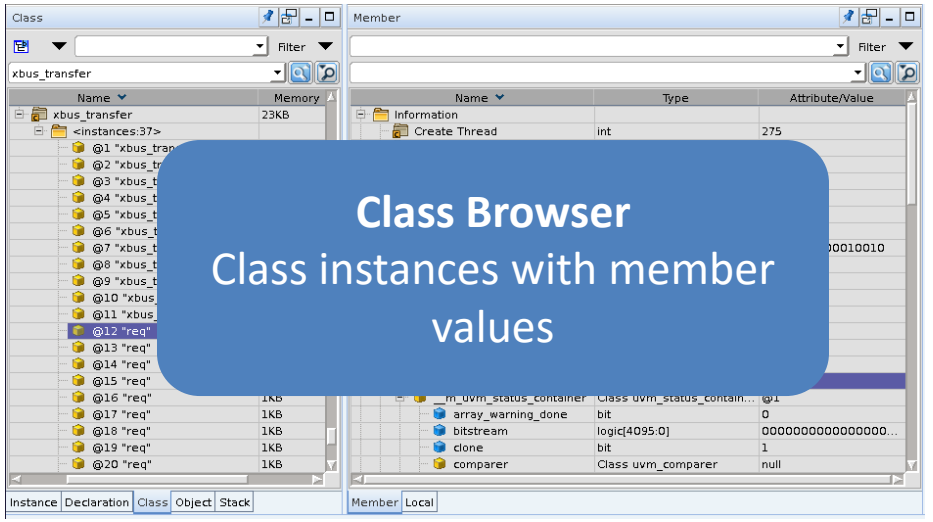
- **Full Visibility**
- Source code, dynamic objects, watch list etc.
- Call stack, members, local variables etc.

- **Powerful Capabilities**
- Trace value assignment into testbench
- Perform what-if analysis
- Enable day-to-day and batch simulation debug

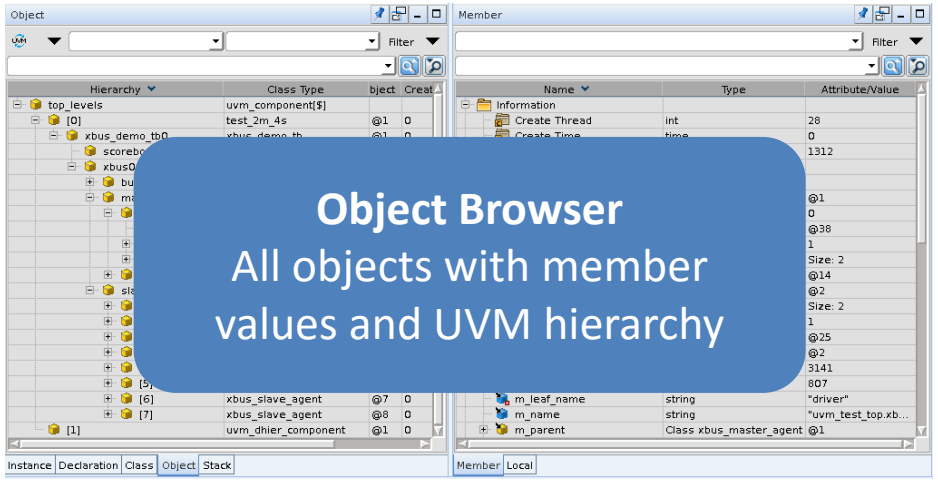
Key Elements for Efficient SystemVerilog TB Debug



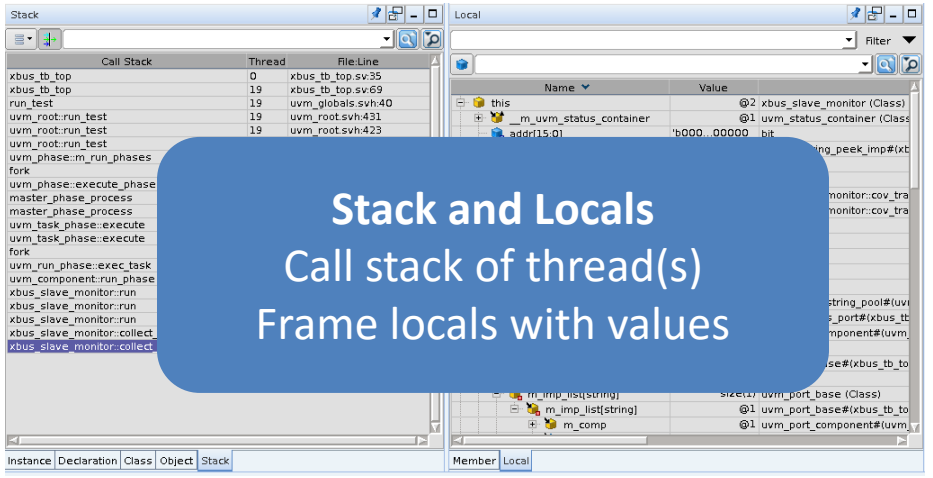
Class Browser
Class hierarchy, methods, members



Class Browser
Class instances with member values

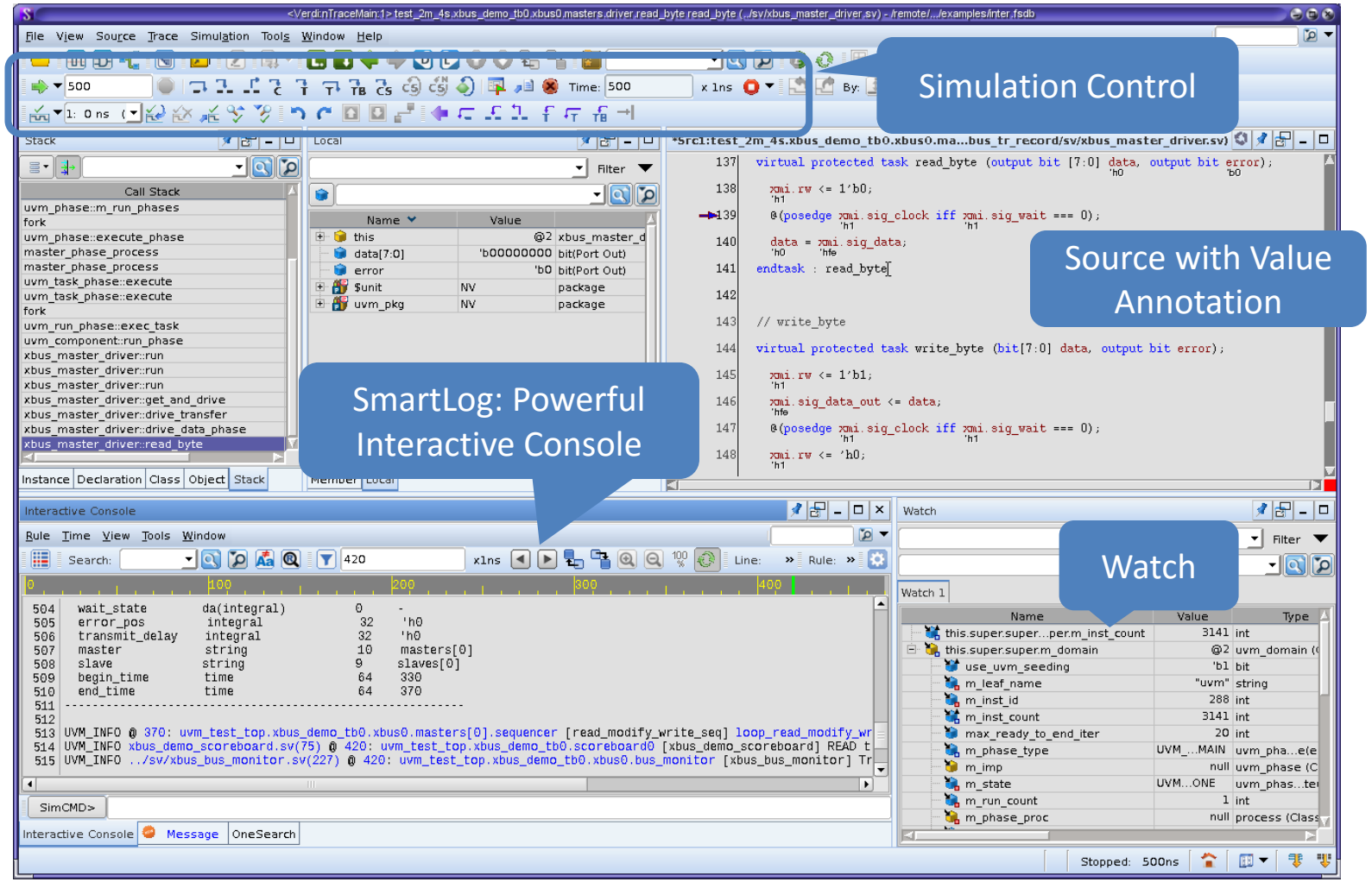


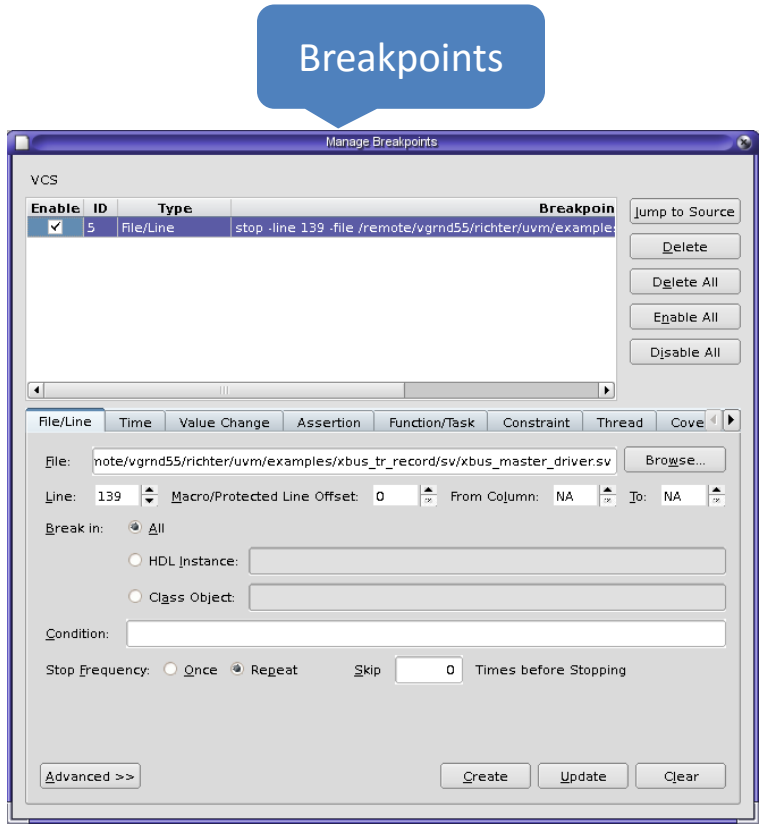
Object Browser
All objects with member values and UVM hierarchy



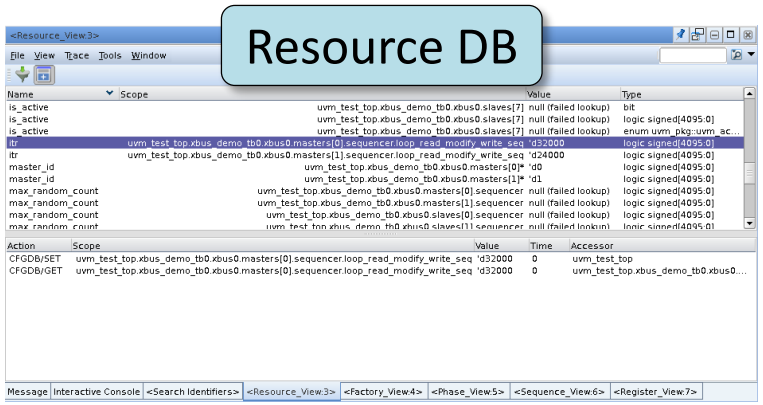
Stack and Locals
Call stack of thread(s)
Frame locals with values

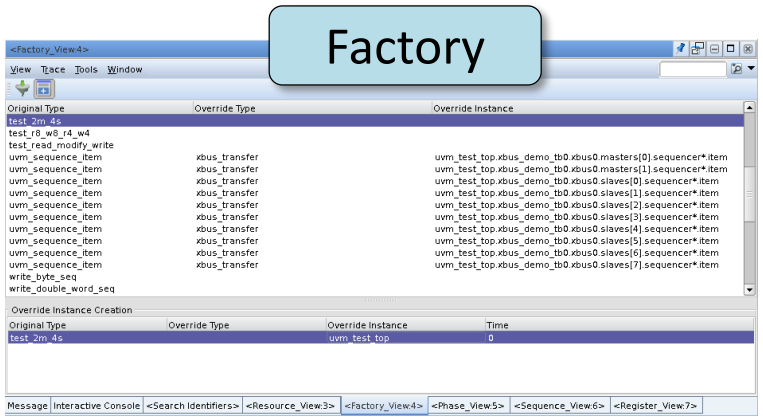
Simulation Control & TB Behavior Tracking

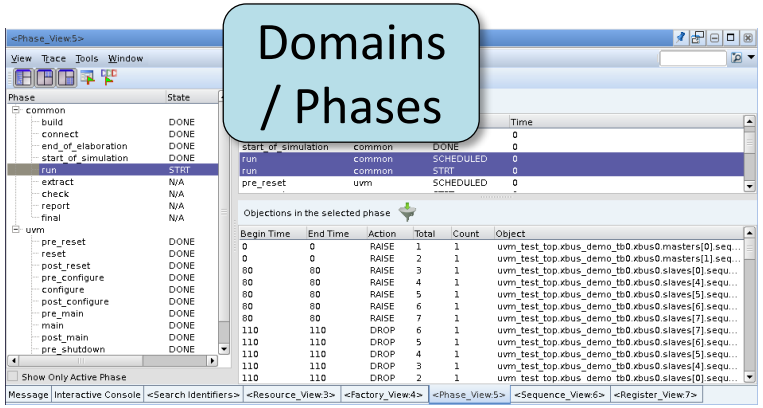


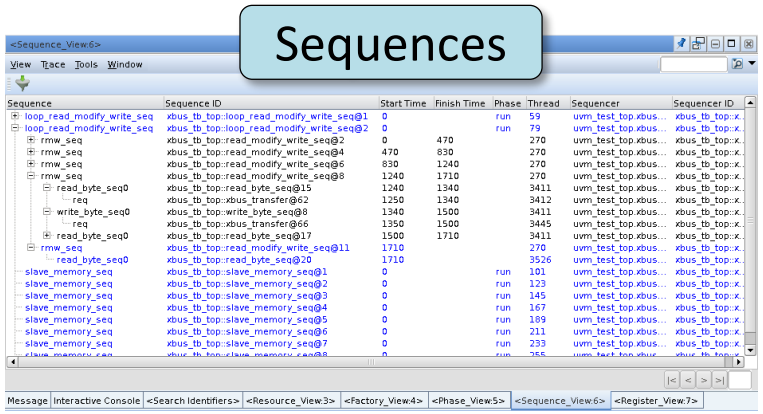


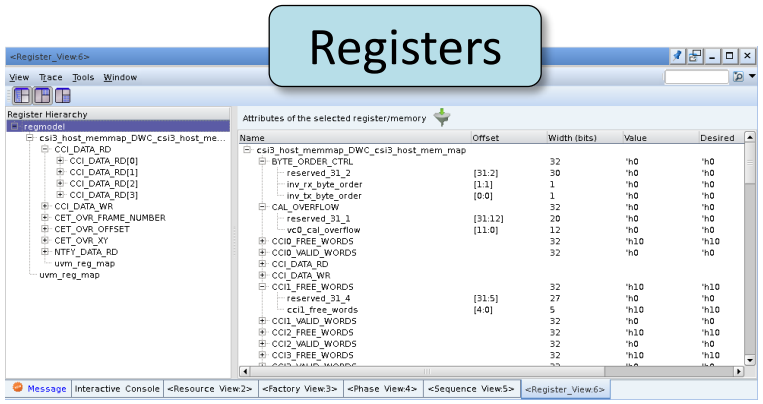
Making Debug UVM Aware

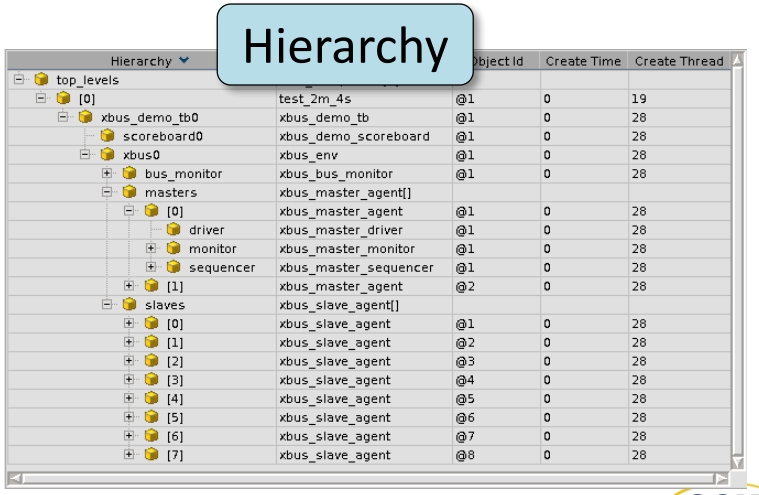












Buried in logfiles?

Logfile is too big! How do I find the UVM Fatal?

Emacs search not enough anymore…

Manually correlating log messages is so annoying!

I want just the errors around the 10,000 nsec mark

What's the source code that created this message?

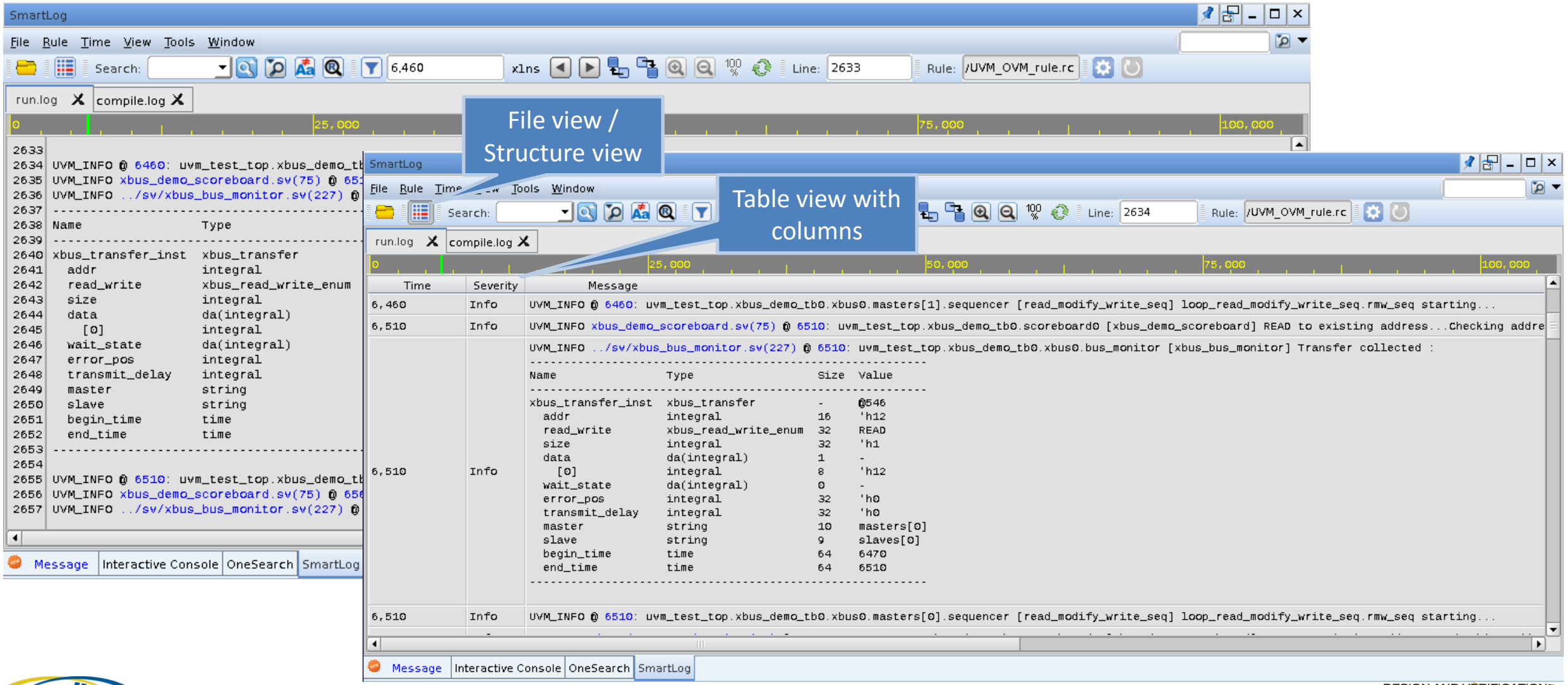
How can I browse a log with custom messages?

There has to be a better way to browse data across all these logs!

SmartLog to the Rescue

- Smart browsing of log files
  - Hyperlinks
  - Message filtering: UVM, VCS, Severity…
  - Time visualization and navigation
  - Message navigation
  - File view or structure view
  - Searching and text based filtering
  - Customize and add own log parsing rules (for custom log files)

SmartLog – File View



Agenda

Motivation

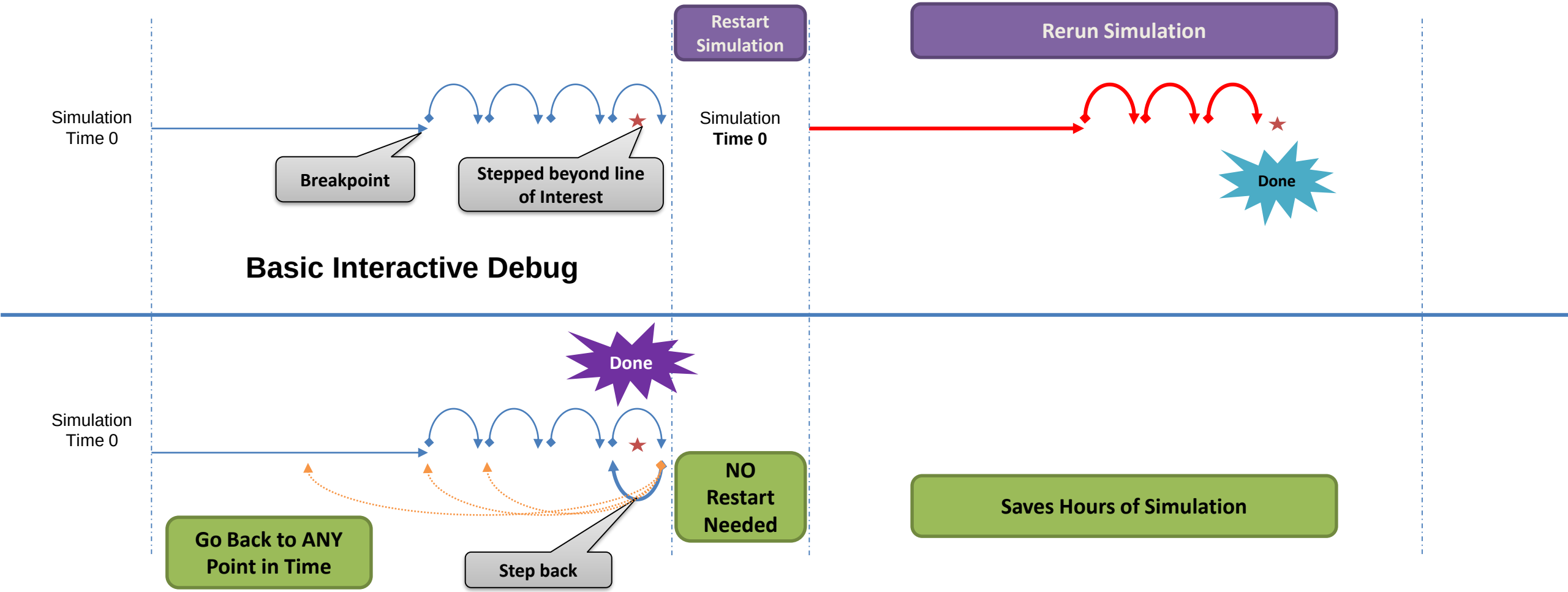
Efficient SV/UVM Testbench Debug

Rapid Root-Cause Analysis with Reverse Debug

Search-centric Debug with OneSearch

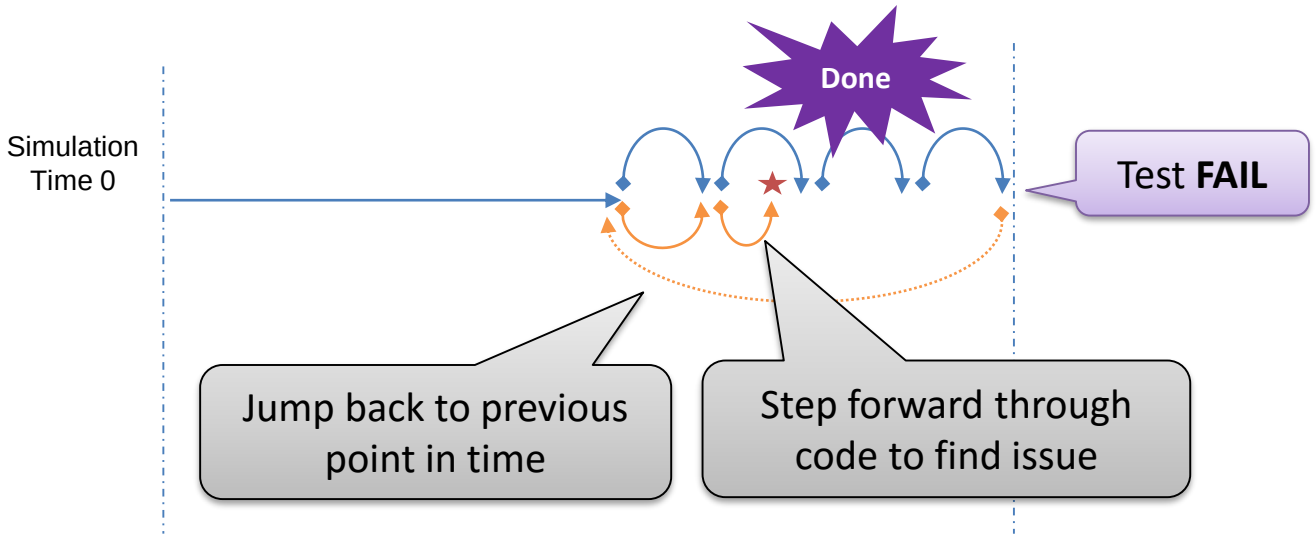
Embedded SW Debug

Productivity with Reverse Debug



Finding Failures

Diagnose and Debug - Moving forward and backward in time



Advance forward to later time without re-simulation

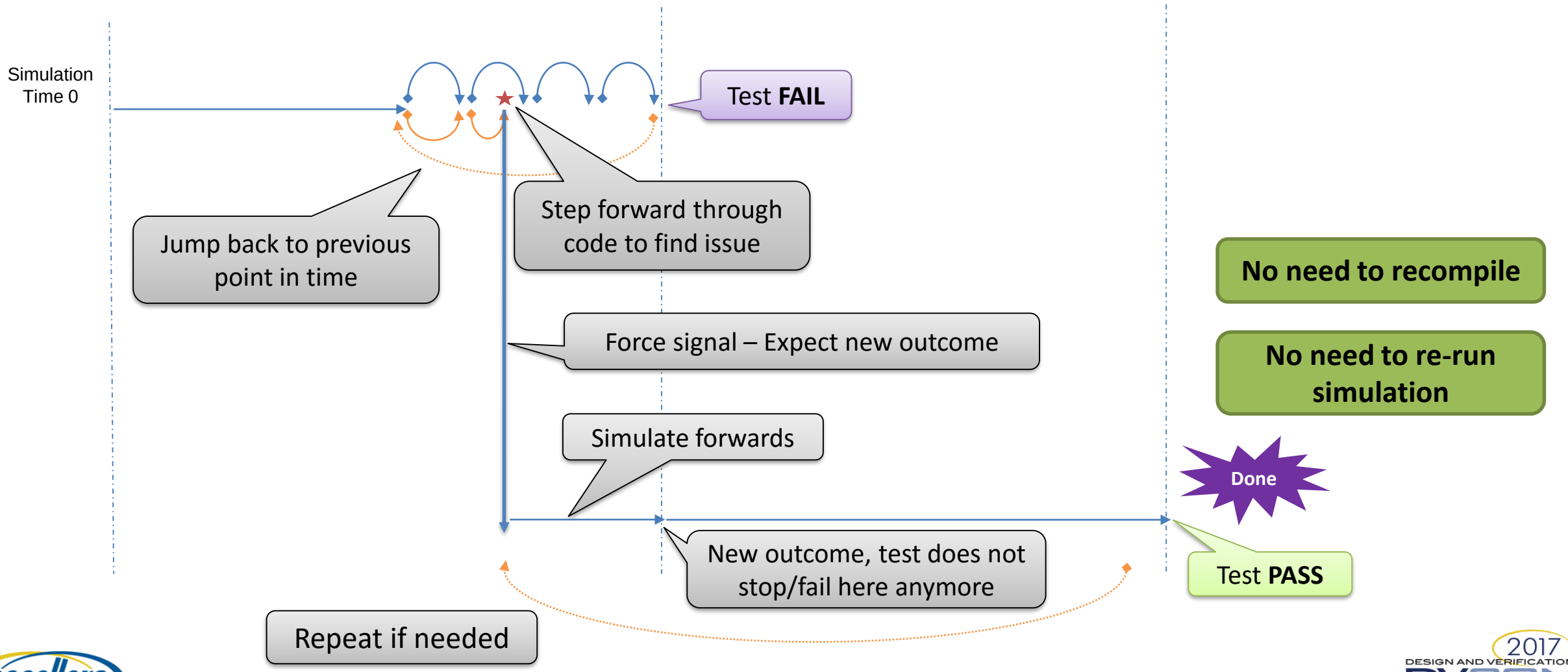
Undo/Redo
Last simulation control command
Forward & Backward

Go To
Previous/Next
Value Assignment

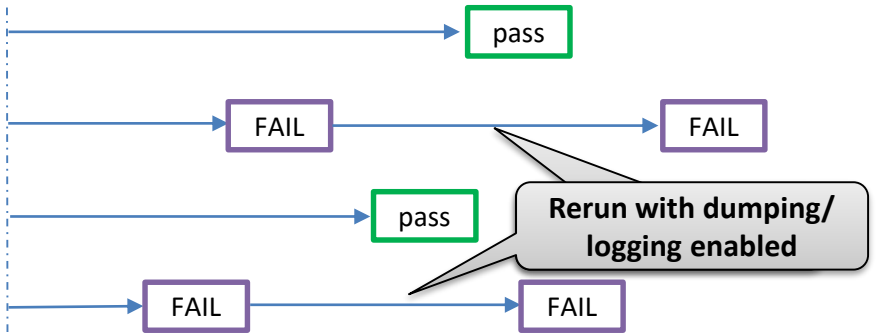
Reverse Controls
Run - Next – Step – Step Out – Step in
Thread – Next in Thread – Step in TB

What-if Analysis

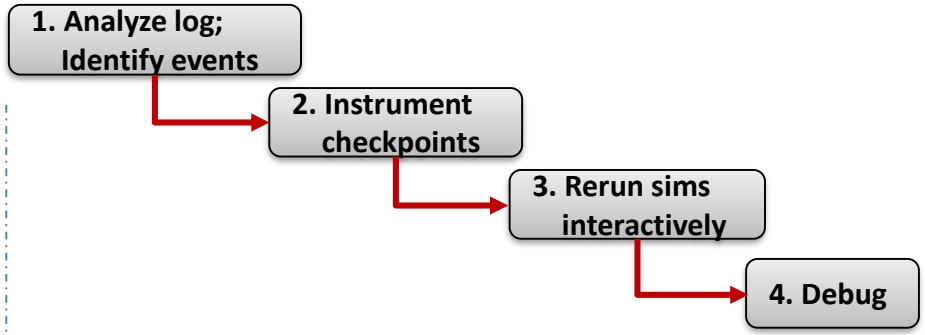
Fix and Validate in One Simulation

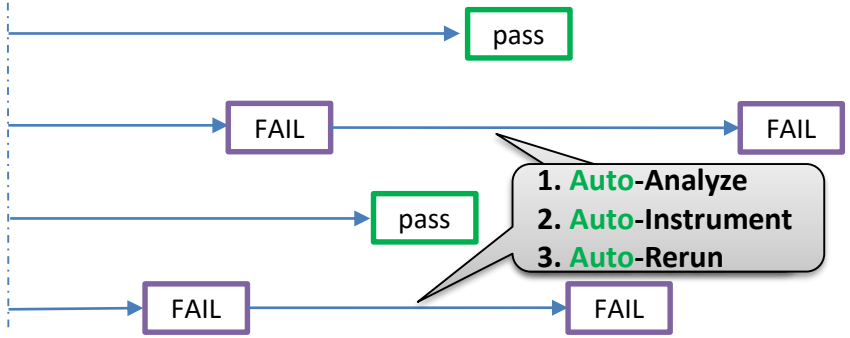


Regression Debug



Overnight Regression





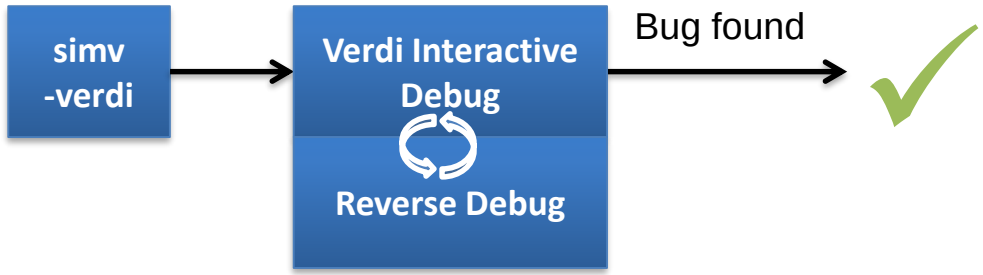
w/ Regression Debug Enabled

Reduce debug TAT significantly
Maximize regression utilization
Focus manual effort on debug

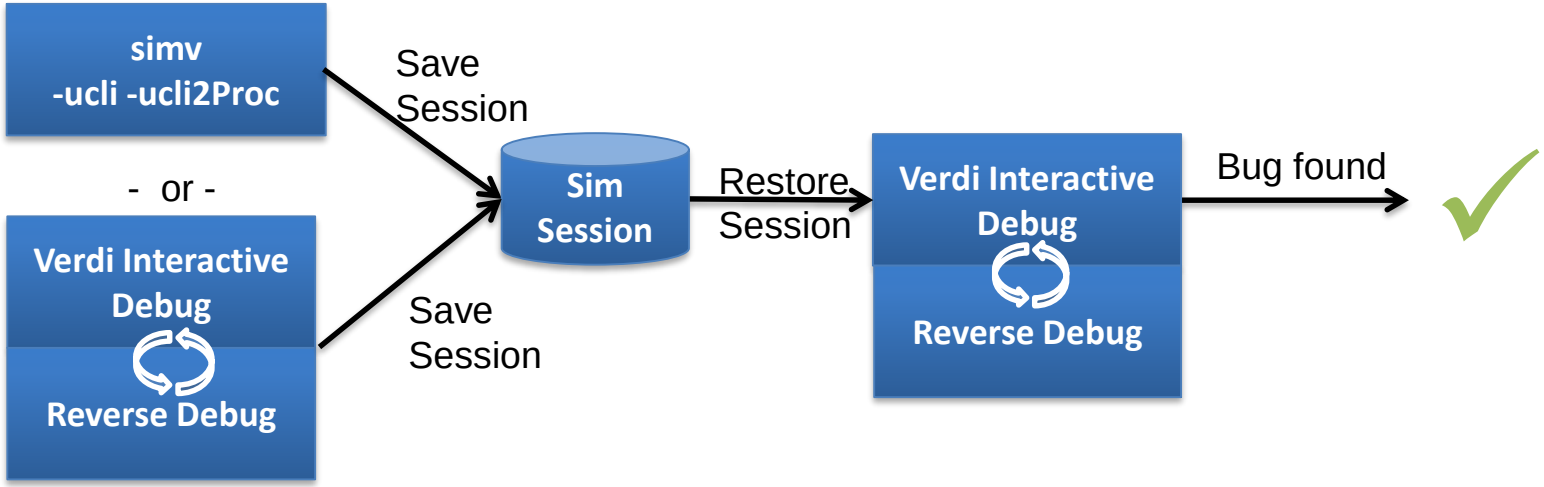
Interactive Debug Flows

Interactive debug for both day-to-day and batch simulation tasks

1. User runs and debugs interactive simulation in Verdi



2. User runs batch/interactive simulation, saves sessions, later restores sessions and debugs interactively



Verdi Reverse Debug

Demo

Agenda

Motivation

Efficient SV/UVM Testbench Debug

Rapid Root-Cause Analysis with Reverse Debug

Search-centric Debug with OneSearch

Embedded SW Debug

Search Queries…

Can you imagine life without Google?

What do you use Google for?

Any kind of Information! The only place!

What do you enter?

How do you search?

Google Search I'm Feeling Lucky

People just enter plain words – no syntax

Do you still use bookmarks?

It changes your need of other features

Now, how would designers and verification engineers like to search?

OneSearch

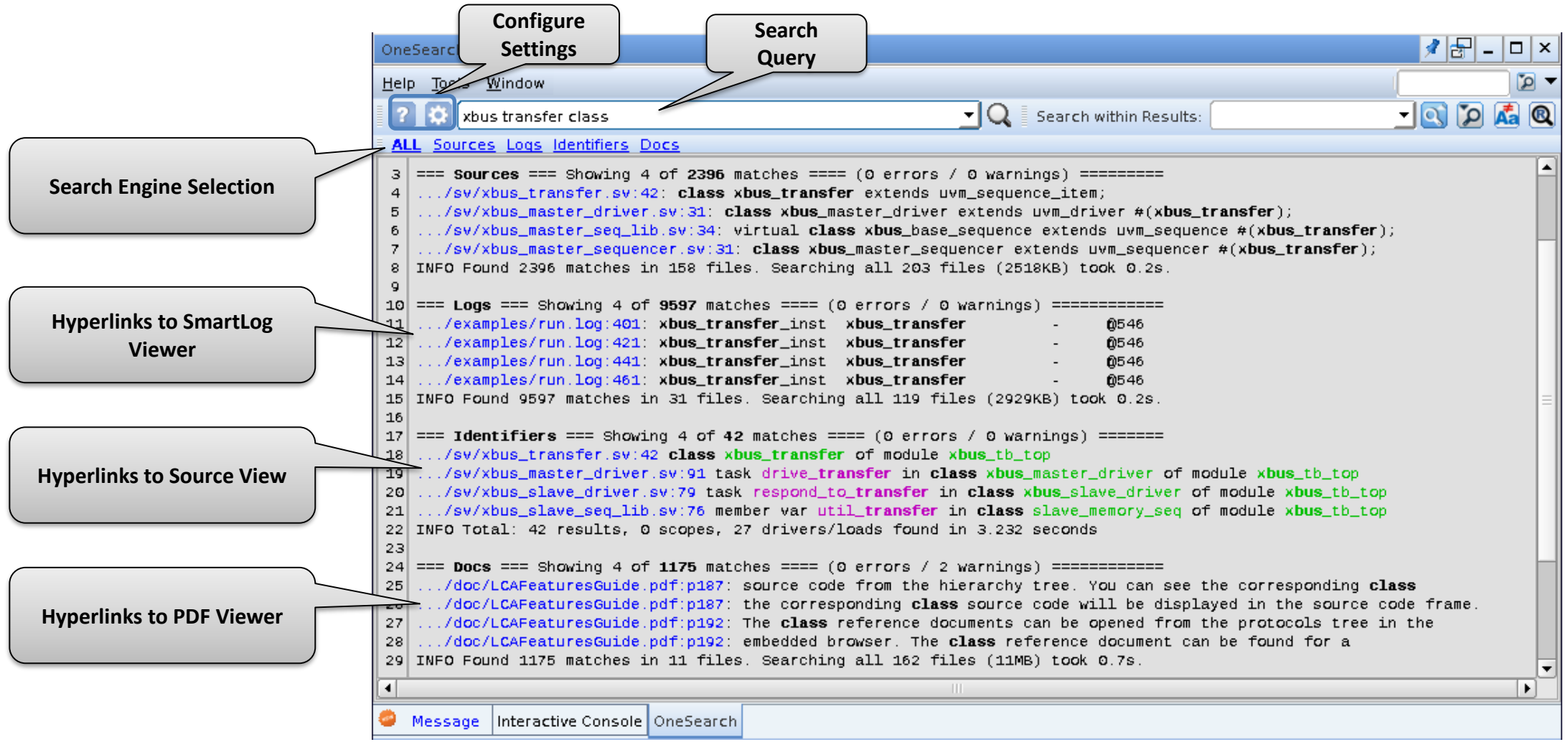
A single search for the entire verification environment

- Single and simple user interface
- Fast search engines

Sources	searches within **source files** and files used for the design
Logs	searches within **log files** from compilation and execution
Identifiers	searches within the compiled design for any **Identifier**
Docs	searches within **Verdi tool documentation PDF files**
Your Own	searches fully customized (custom DBs, specific files, environment, …)

Custom PDFs:
FAQ available

Using Verdi OneSearch



Verdi OneSearch

Demo

Agenda

Motivation

Efficient SV/UVM Testbench Debug

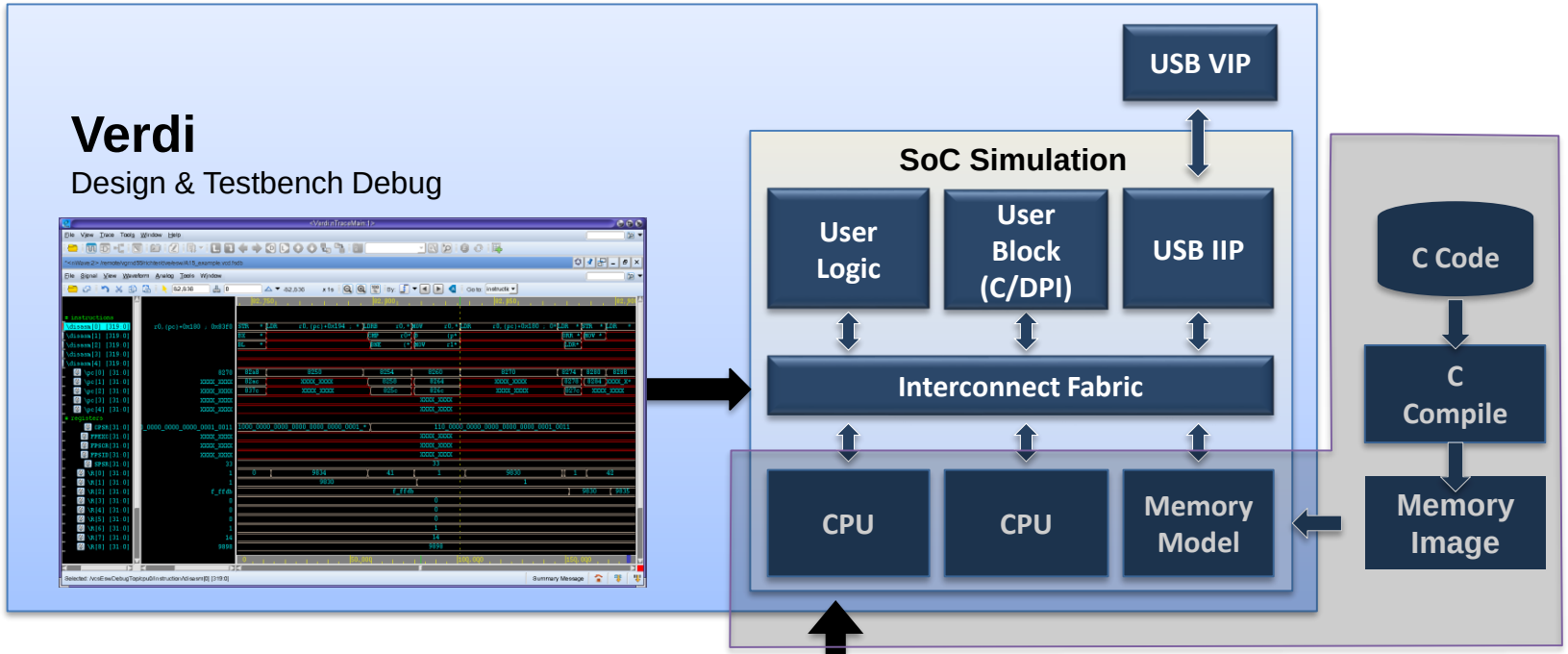
Rapid Root-Cause Analysis with Reverse Debug

Search-centric Debug with OneSearch

Embedded SW Debug

SoC Debug Challenges

Lack of Visibility on the Software Side



BUT… What about the embedded software?

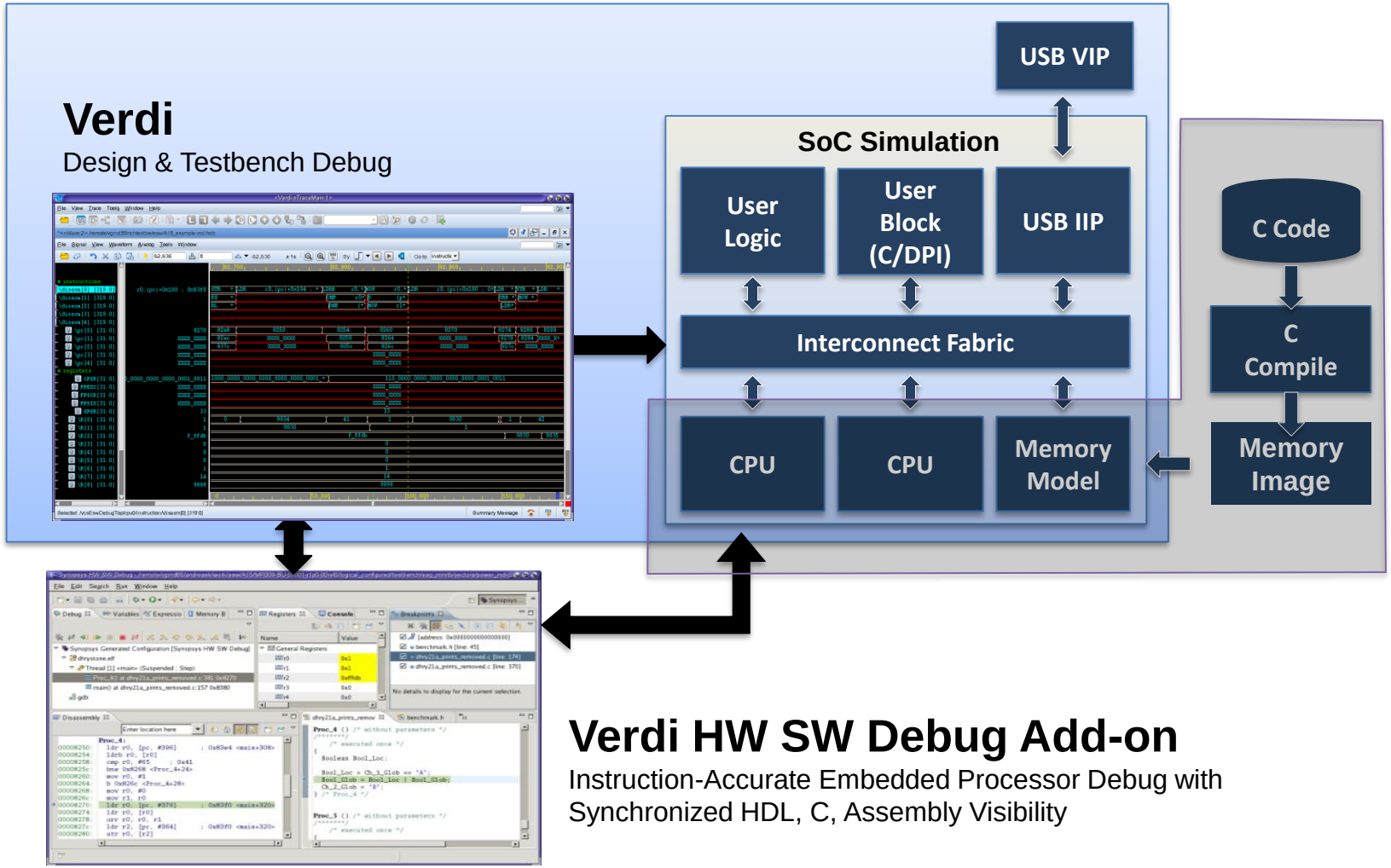
CPU Software Debug
- Encrypted Model ?
- Registers + Nets ?
- No or Low-Visibility
- Looks like a Black Box

Time Consuming Manual Debug:
- Log-files, Waveforms
- ELF Dump, Disassembly
- Decoding Stack, Variables

At every point in time…

SoC Debug Challenges

Lack of Visibility on the Software Side

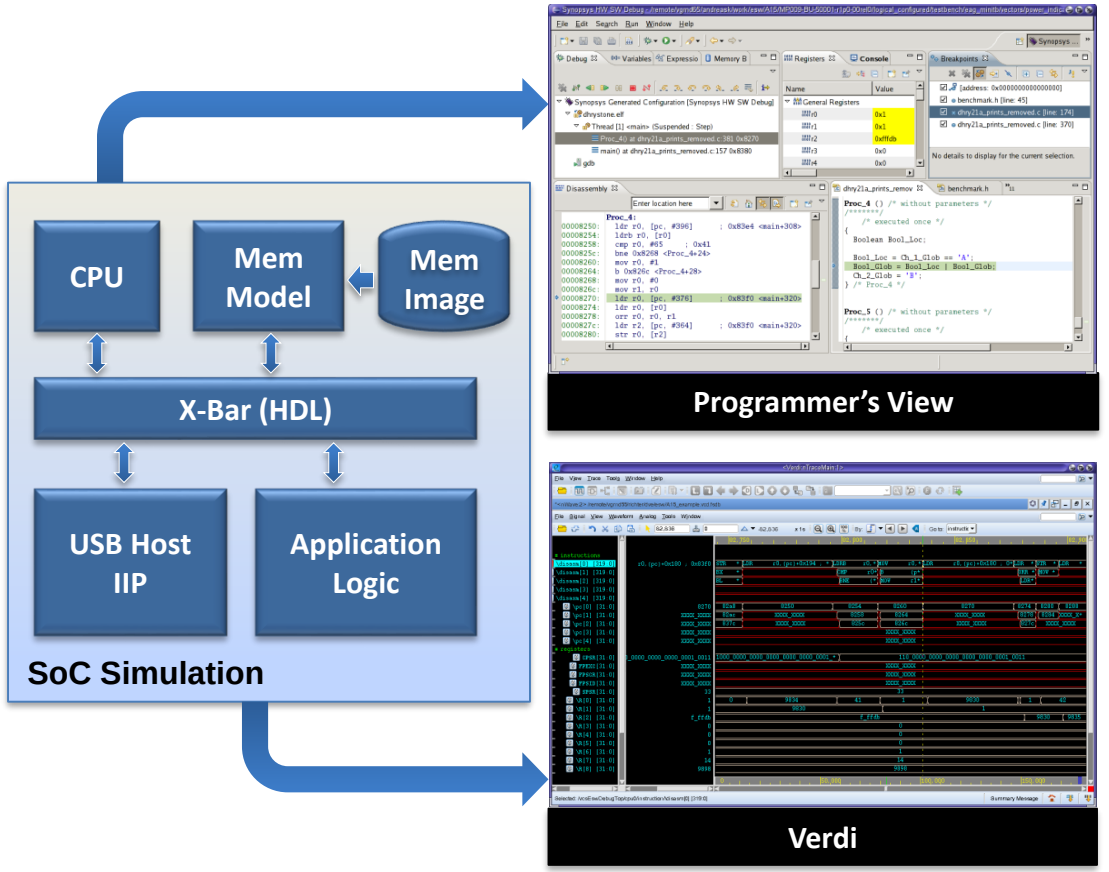


Verdi HW SW Debug Add-on

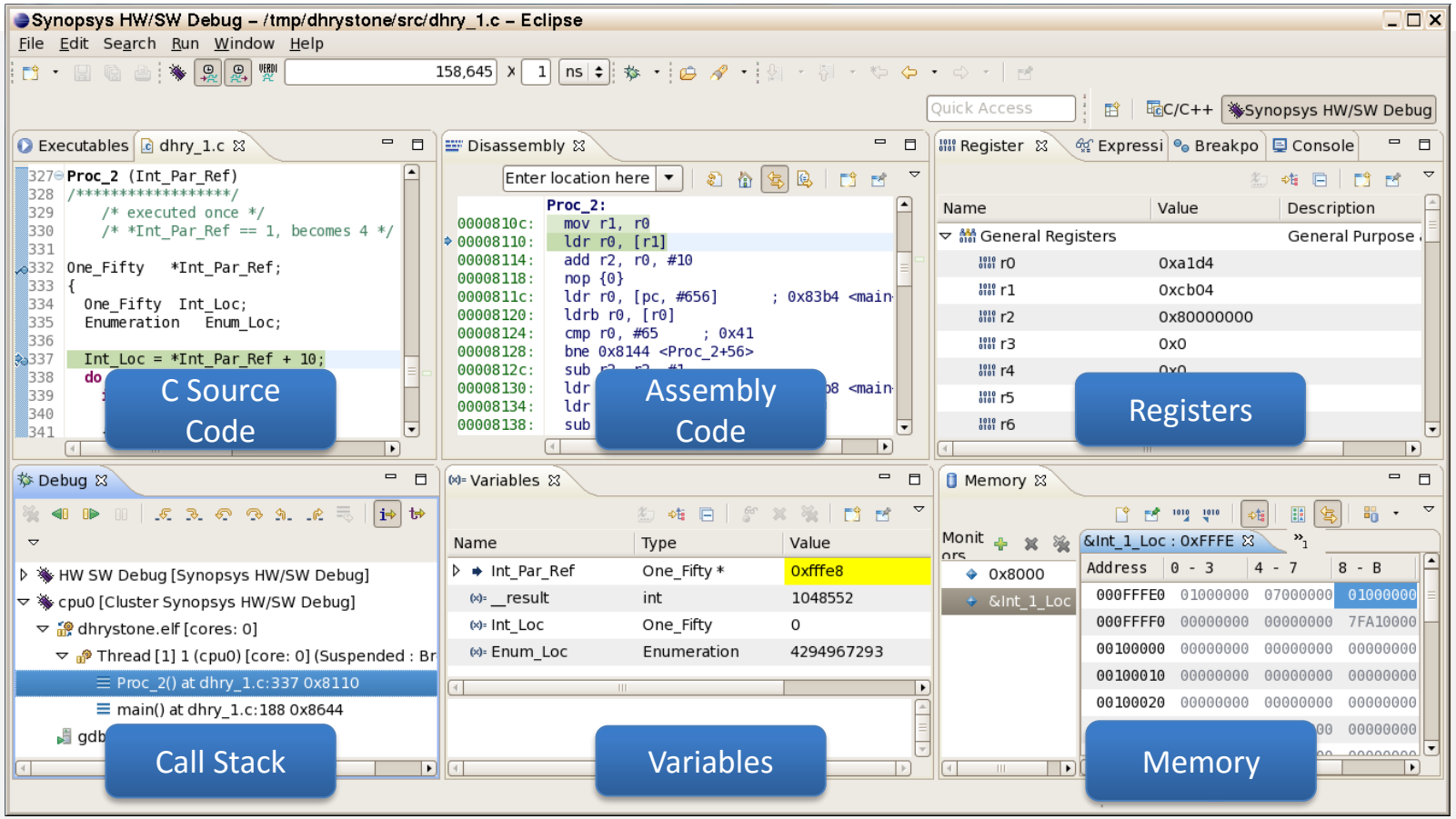
Instruction-Accurate Embedded Processor Debug with Synchronized HDL, C, Assembly Visibility

Verdi HW SW Debug

- Enables co-debug between HW and SW
- HW and SW debug synchronized in time
- View C/Assembly source,
  C variables, stack, memory
- Debugs multiple cores simultaneously
- Simulation supports all ARM® Cortex® cores
- Easy to support additional cores or custom cores



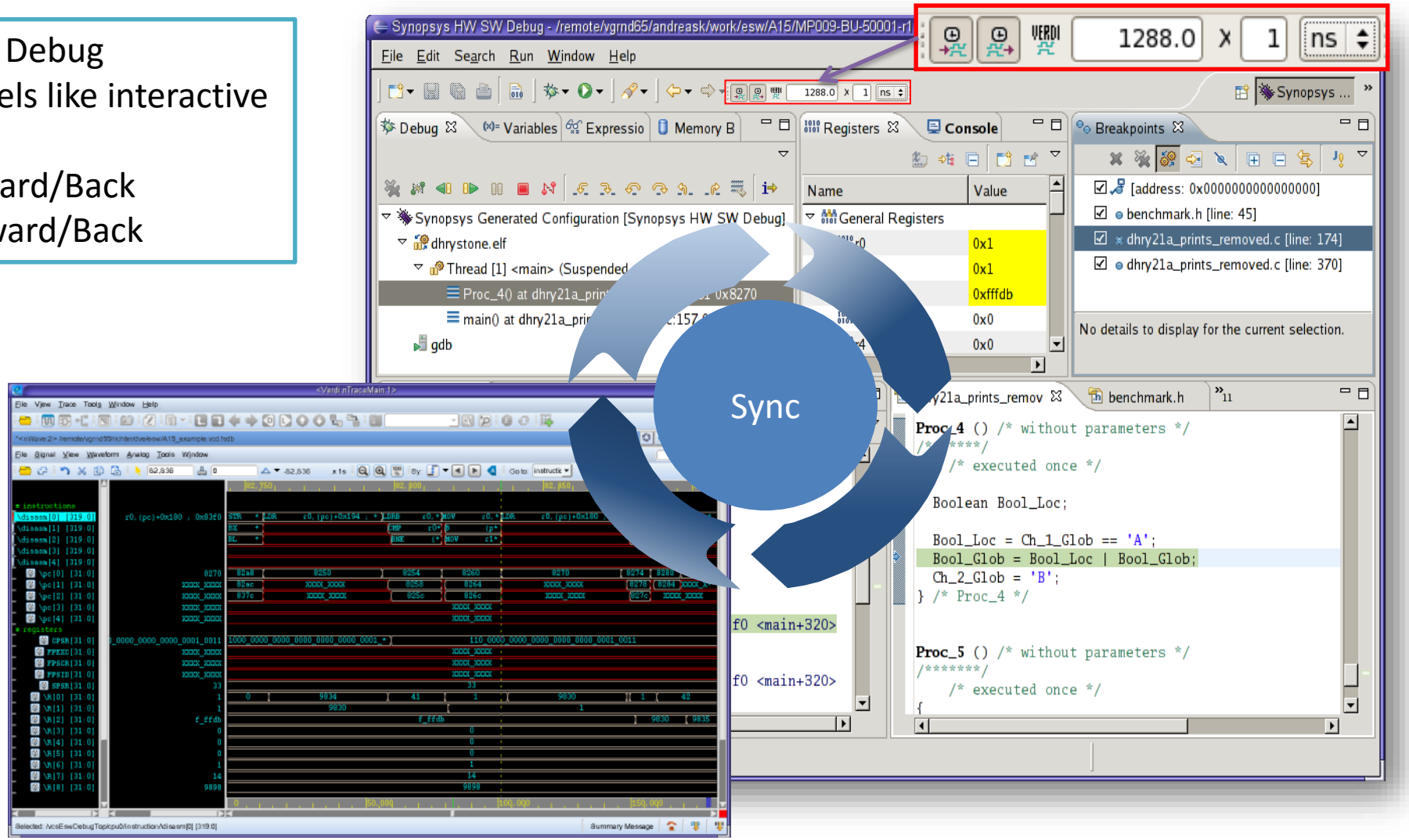
Embedded Software Debug in Eclipse



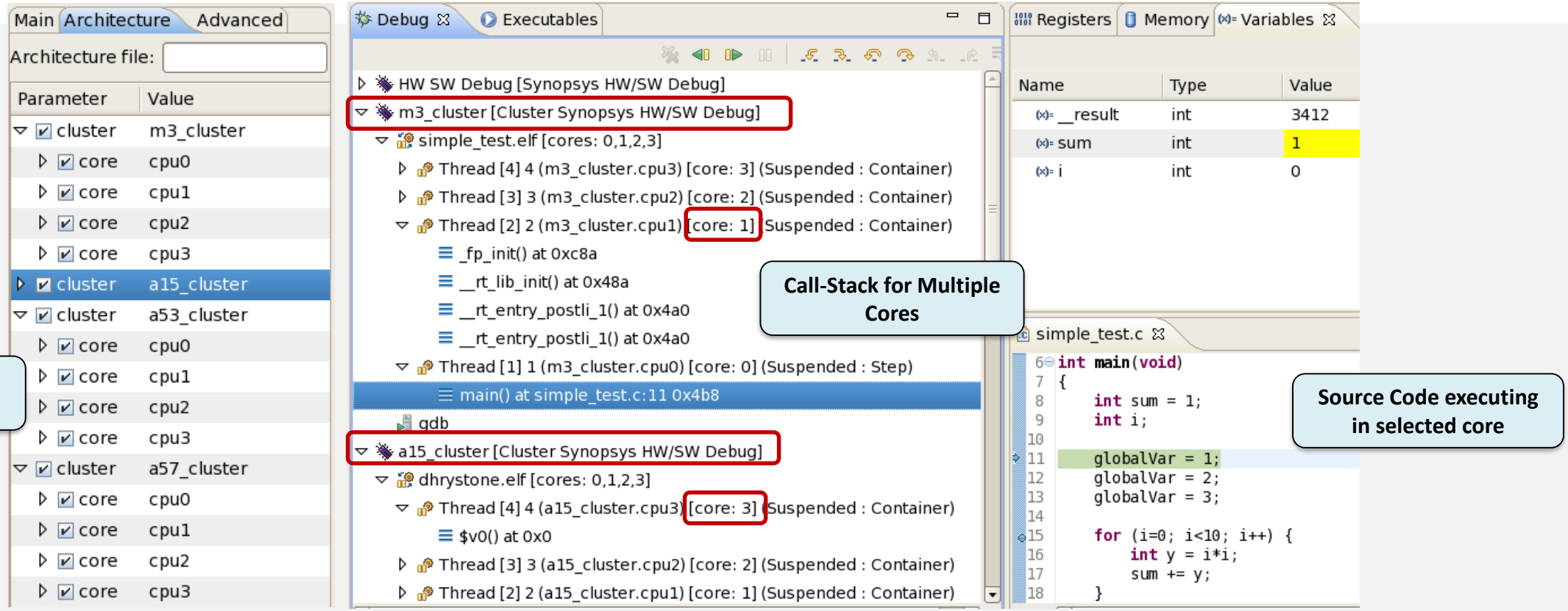
Provides programmer's view of code running on simulated processor

Combined HW and SW Debug

Post Process Debug
- Eclipse feels like interactive debug
- Run Forward/Back
- Step Forward/Back

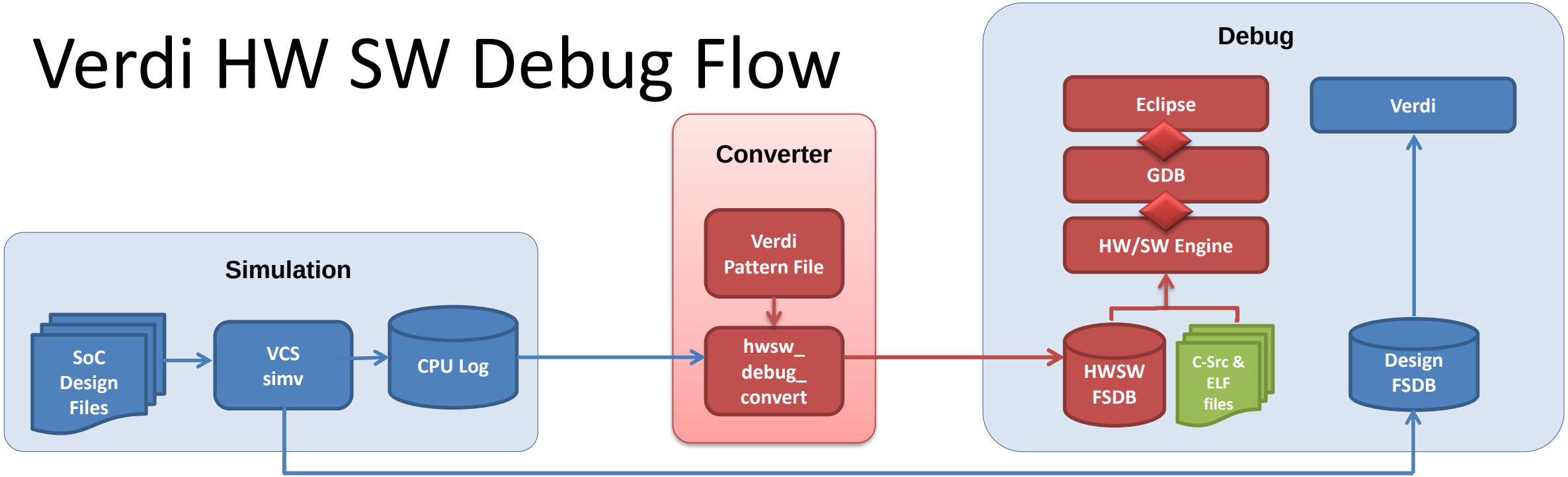


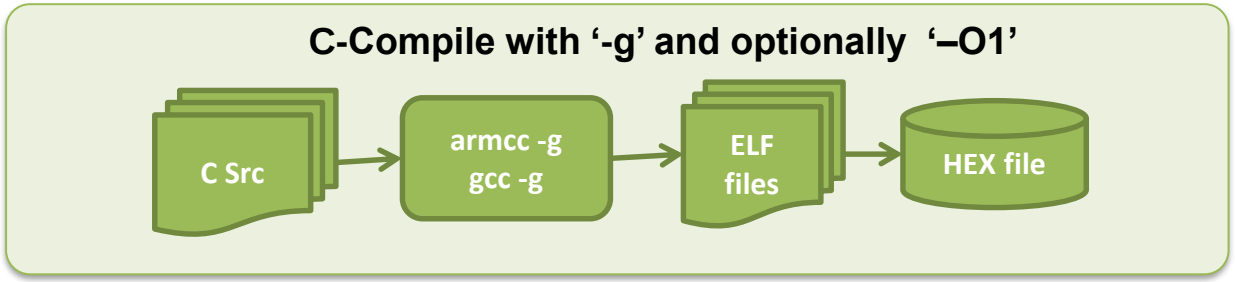
Debugging Multi-Core

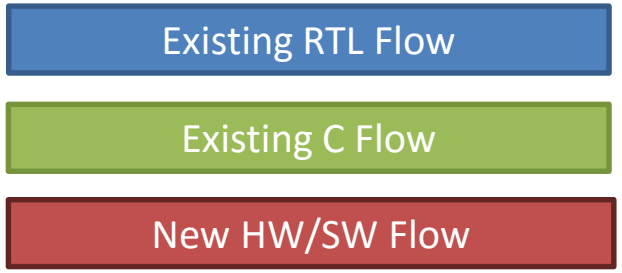


Provides programmer's views into multiple cores simultaneously

Verdi HW SW Debug Flow







No modifications needed - Enable CPU log during simulation

Verdi HW SW Debug

Demo

Verdi HW SW: Getting Started

Install

- Download and install latest Verdi Release
- Run Dhrystone Example to verify installation

Prepare

- Identify the CPU cores in your design
- Dump CPU log from your simulation

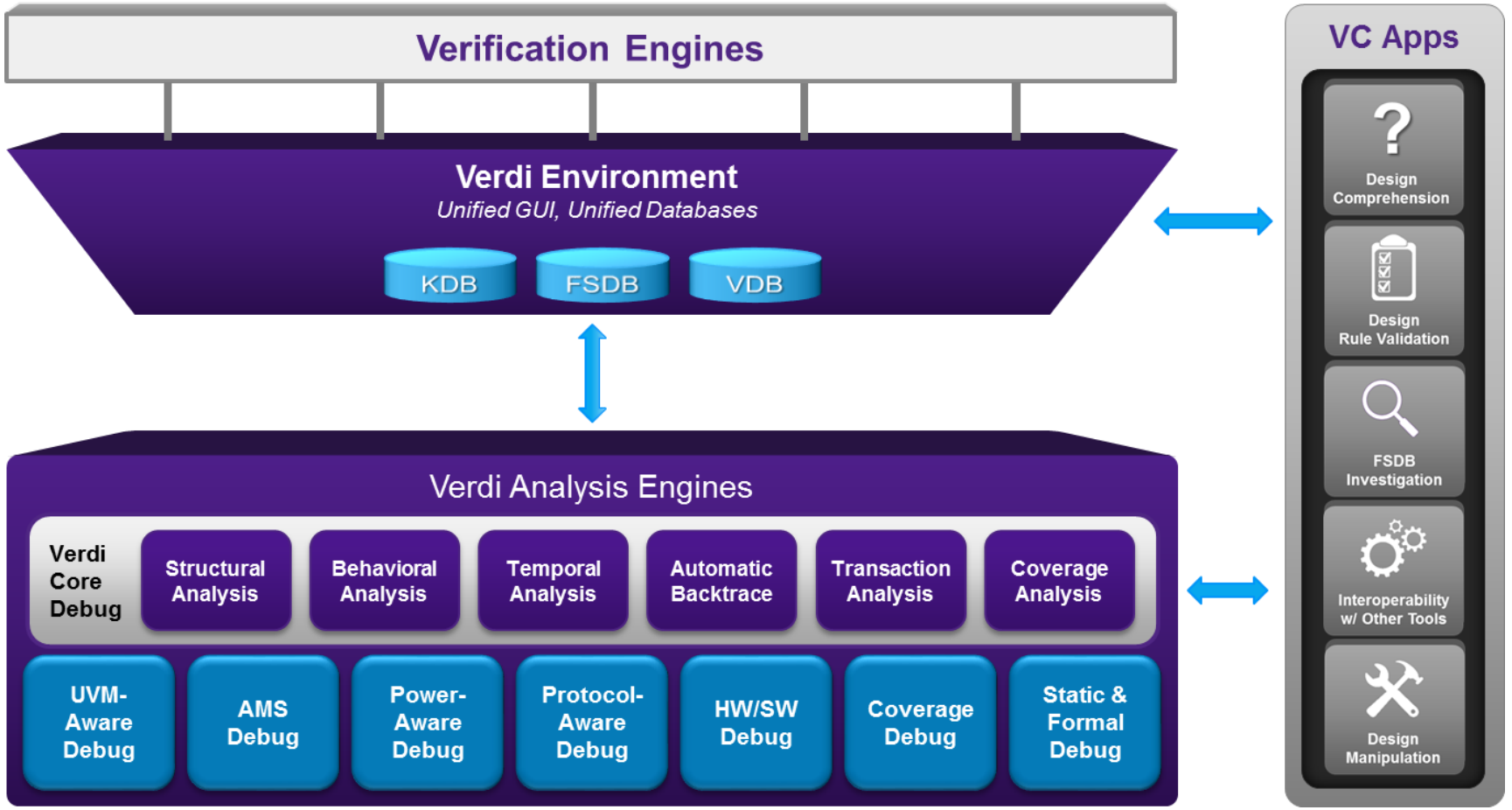
Convert

- Run Converter on CPU log to generate FSDB

Debug

- Invoke Verdi and HW SW Debug/Eclipse
- Debug your Embedded Software and Design HDL

Summary



Interactive Debug

- Full-featured, full visibility debug for SV/UVM testbench
- Run forward and backwards for improved debug efficiency

OneSearch

- Fast and efficient search
- Search entire verification environment at once

SmartLog

- Organized logs in table format with time and other filters
- Parse logs with hyperlinks to source

HW SW Debug

- Synchronized HW and SW debug
- Multi-core support
- Performance profiling and embedded SW code coverage

Questions?

Motivation

Efficient SV/UVM Testbench Debug

Rapid Root-Cause Analysis with Reverse Debug

Search-centric Debug with OneSearch

Embedded SW Debug

References

- Synopsys Verification Home
  https://www.synopsys.com/verification.html

- Synopsys Verdi Debug Platform
  https://www.synopsys.com/verification/debug.html

- Synopsys Verdi HW/SW Debug
  https://www.synopsys.com/verification/debug/verdi-hw-sw-debug.html