Indago™ Debug Platform Overview

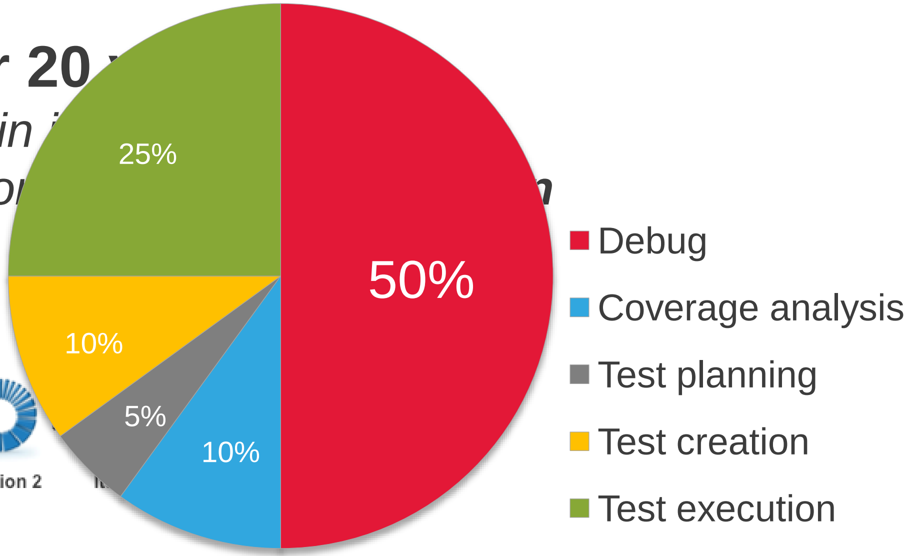
September 2015

cadence®

Debugging Continues to be the Most Time Consuming Effort by **50%**

And it’s getting worse. WHY???

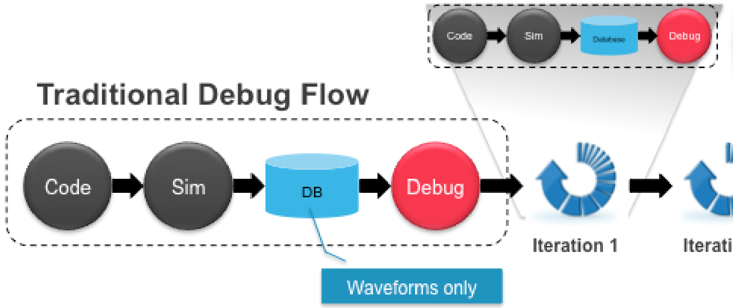
Today’s Verification Effort



Source: Verification engineer survey by Cadence

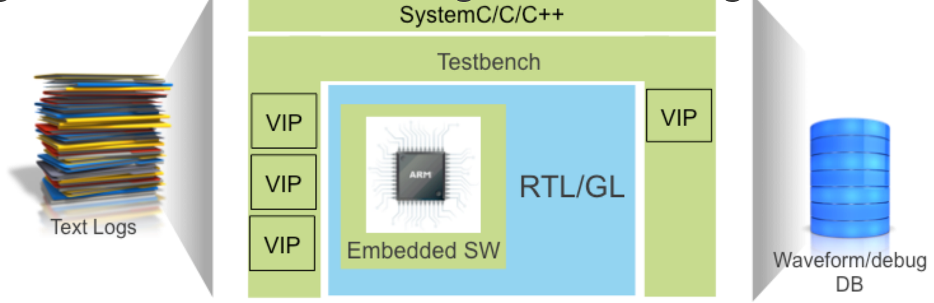
- **Same Debug Methodology for 20**

While there has been very good progress in i
areas of verification, very little has been do **n**



- **Increasingly larger SoC designs and many iterations producing Terabytes of Data**

Finding the source of the bug is becoming like finding “a needle in a hay stack”



What would you do if you were given 25% of your TIME back?

Cadence set out to do exactly that

But HOW ???

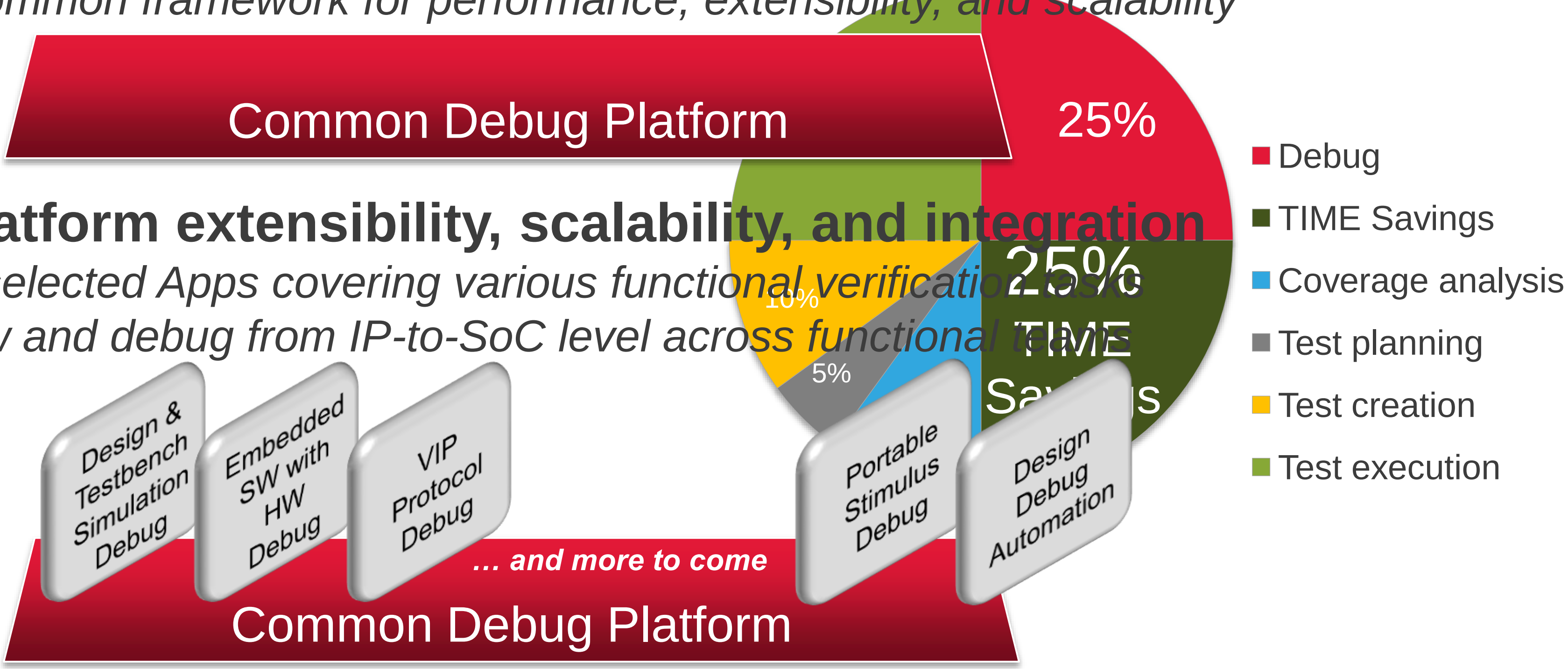
- ➢ **Debug platform architected from the ground up**

Leveraging the latest in s/w database architecture as its foundation of a common framework for performance, extensibility, and scalability

Common Debug Platform

- ➢ **Platform extensibility, scalability, and integration**

User-selected Apps covering various functional verification tasks to view and debug from IP-to-SoC level across functional teams



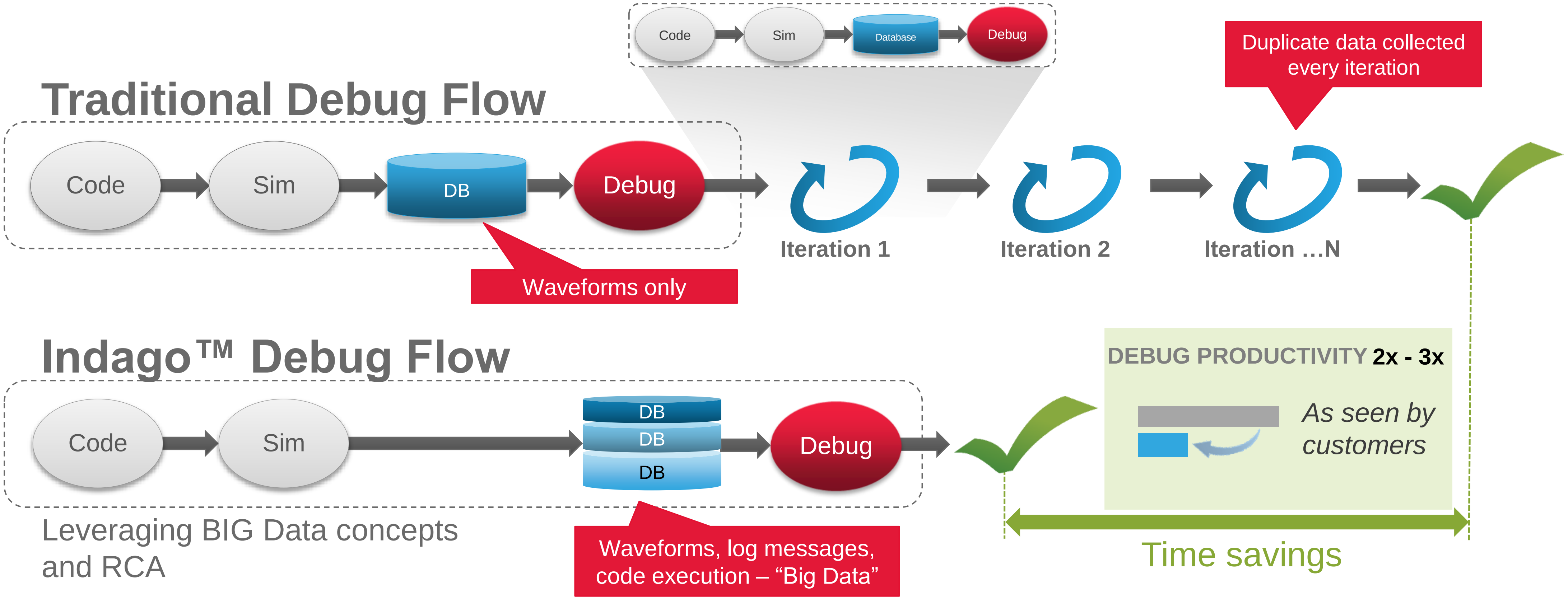
Today's Verification Effort after cutting Debug Time in ½

- ➢ **Patented Root Cause Analysis (RCA) Technology and BIG Data Techniques**

Leverage patented RCA technology together with BIG Data techniques to quickly find the source of the bug

Introducing Cadence® Indago™ Debug Platform

Finding the Source of the Bug after One Debug Run is NO Longer a Dream



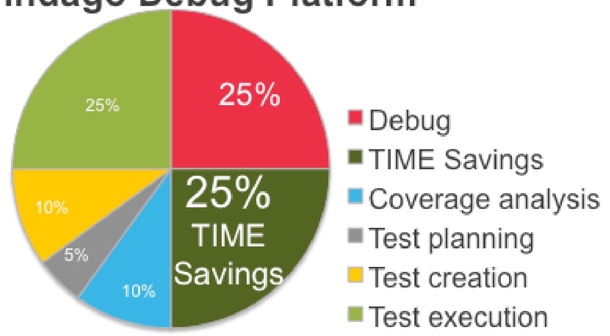
> *"Cadence's Indago Debug Analyzer App has improved our debug productivity up to 50 percent because it helps us find the root cause of the bugs faster with features like reverse debugging. We believe the Indago Debug Platform will enable us to continue to deliver for applications including consumer electronics, fitness tracking, wearables and IoT."*
> Robert Richter, Senior Expert, ASIC Development, at Bosch

Key Benefits of Cadence® Indago™ Debug Platform

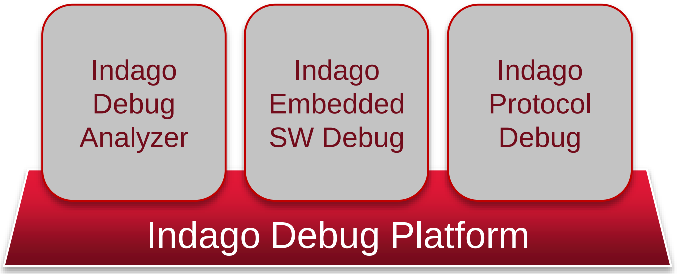
A Paradigm Shift in Debug Methodology Cutting Debug Time in ½

- 2X debug productivity improvement with Indago through:
  - Patented Root Cause Analysis technology
  - BIG Data concepts for intelligent automation
  - Integrated Analysis GUI scalable from IP-to-SoC level debug
- 3 Indago platform Apps addressing specific debug tasks
  - Debug Analyzer: RTL/GL and Testbench
  - Embedded Software Debug: Synchronized ESW/HW
  - Protocol Debug: Interface protocol functional validation
- Supports Cadence and 3rd party verification engines
  - Debug Analyzer: Phased RTL/TB support through next several releases
  - Embedded SW: Today (unmodified TARMAC trace files)
  - Protocol Debug: Today (for supported protocols)

Today's Verification Effort using **Indago Debug Platform**



Source: Verification engineer survey by Cadence



Indago Debug Analyzer

Indago Embedded SW Debug

Indago Protocol Debug

Indago Debug Platform

The Indago™ Root Cause Analysis (RCA) Engine

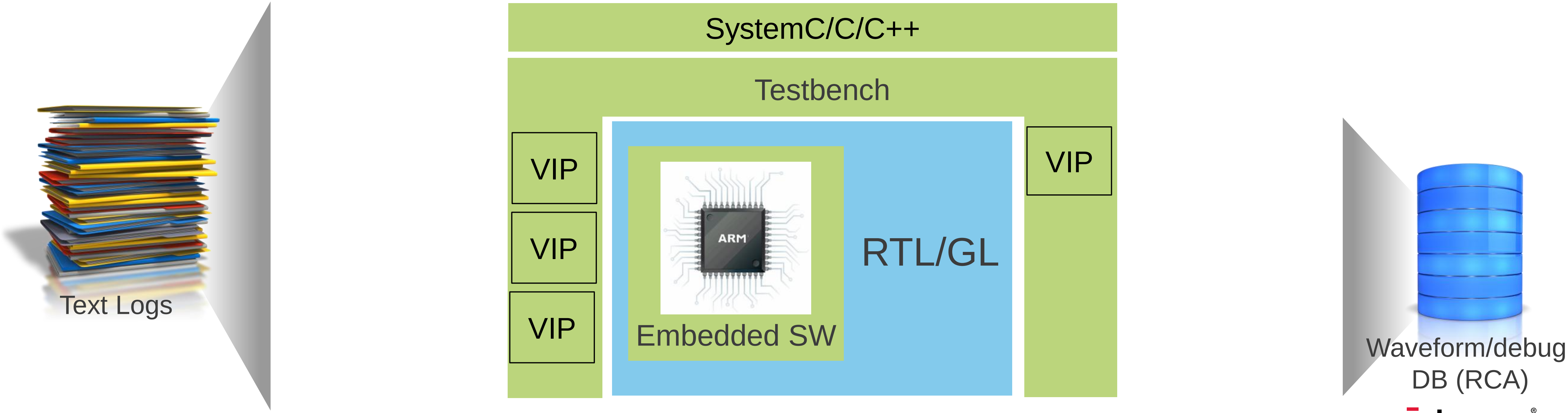
Mature, intuitive and ubiquitous

- The entire Indago Debug Platform is built on top of a mature RCA engines
- The Indago RCA Engines have existed for 8 years
  - Previously only available in advanced CAST analysis tool
  - Now being leveraged by all Debug Apps
- Access to the underlying engine appears in almost all GUI components
  - Click on any variable to traverse time and space to show cause of the variable change
  - Click on a source line to be taken through time to the last/next time that line executed
  - Intuitive RCA component to provide users with a guided tour through a bug scenario
  - Can traverse through language barriers
- Indago provides unparalleled RCA capabilities in our industry

Competitors Debug Solutions

RCA on RTL Only

- Competitors provide RCA on RTL/GL design only
  - RTL only one piece of the debug picture
- Engineers are forced to debug with severely limited visibility into other aspects of the environment



Text Logs

SystemC/C/C++

Testbench

VIP

VIP

VIP

VIP

Embedded SW

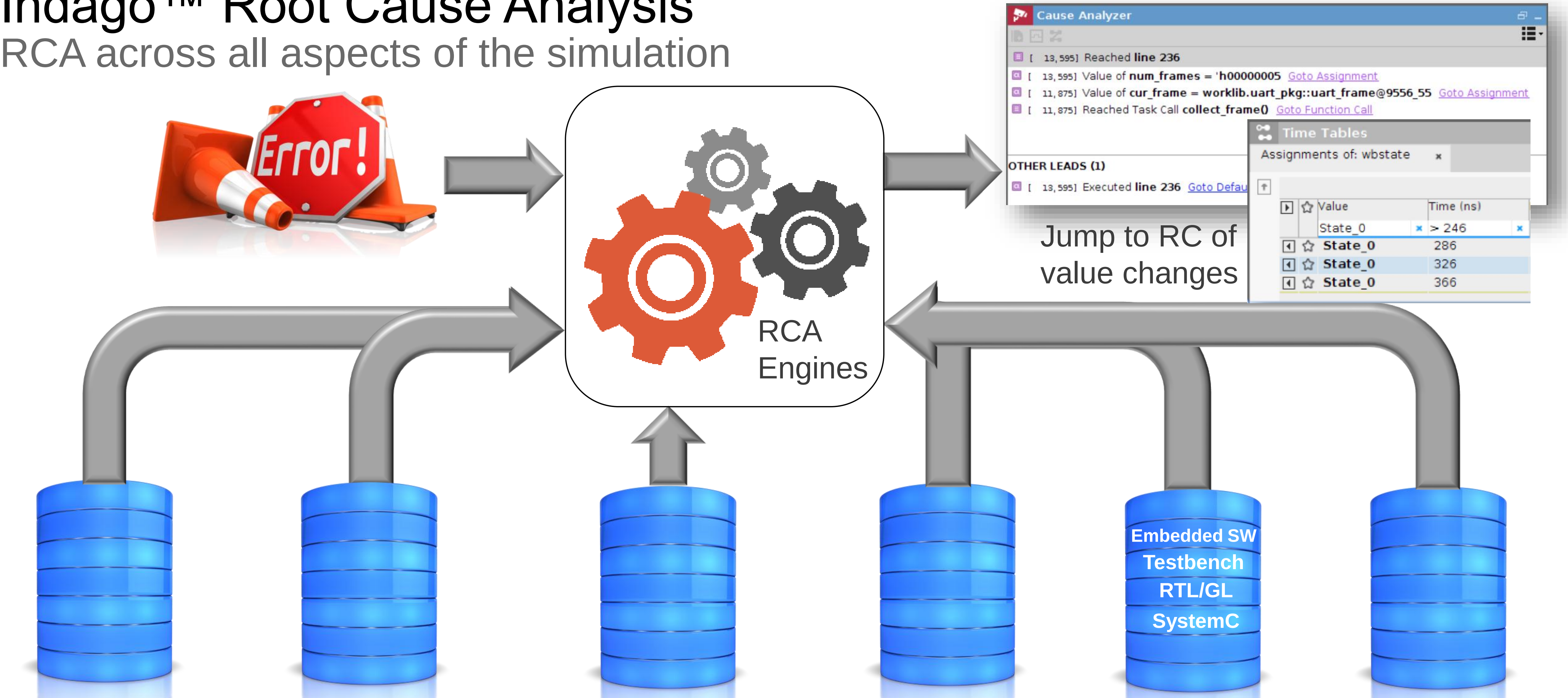
RTL/GL

Waveform/debug DB (RCA)

Indago™ Root Cause Analysis

RCA across all aspects of the simulation

Causal Relationships to explore



```
[  13,595] Reached line 236
[  13,595] Value of num_frames = 'h00000005  Goto Assignment
[  11,875] Value of cur_frame = worklib.uart_pkg::uart_frame@9556_55  Goto Assignment
[  11,875] Reached Task Call collect_frame()  Goto Function Call

OTHER LEADS (1)
[  13,595] Executed line 236  Goto Defau
```

Assignments of: wbstate

Value	Time (ns)
State_0	> 246
State_0	286
State_0	326
State_0	366

Jump to RC of value changes

Messaging DB

Embedded SW DB

TB/RTL Source Execution DB

VIP DB

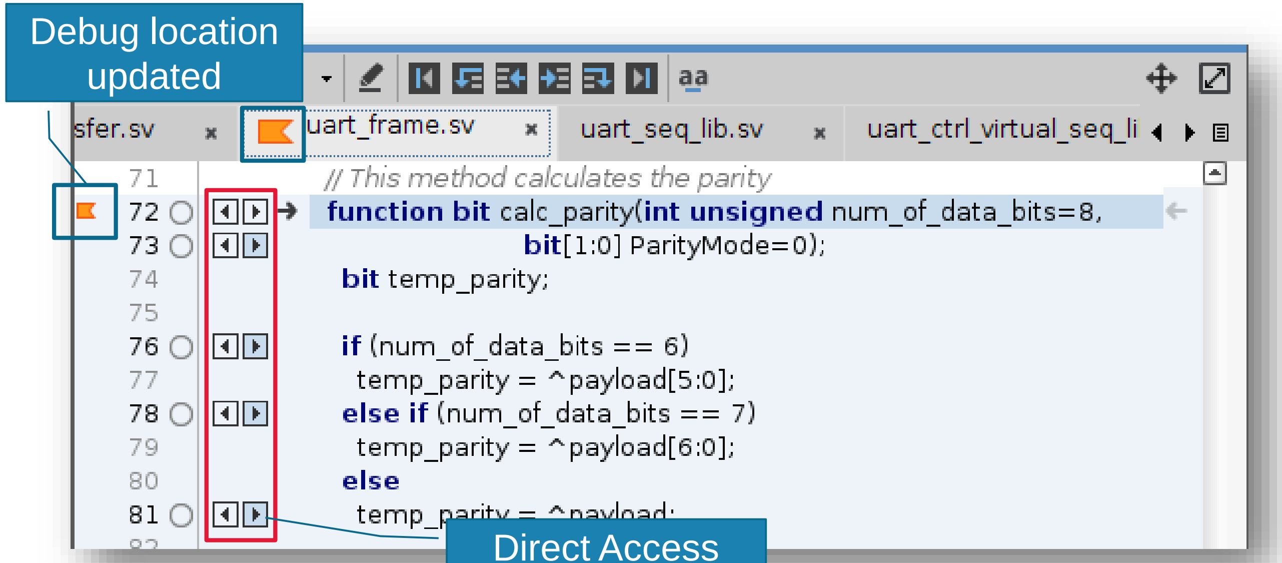
Waveform DB

Debug DB

Indago™ Root Cause Analysis

Direct Access RCA buttons

- Direct Access RCA buttons are present in many Indago debug components
  - SmartLog, Source Viewer, Variables, Active Threads, Time Tables, Search Results, etc.
- Allows immediate access to a debug point of interest
  - Variable change, last/next time a source line executed, last/next time a message was printed
  - After clicking, debug location is updated and all components update accordingly

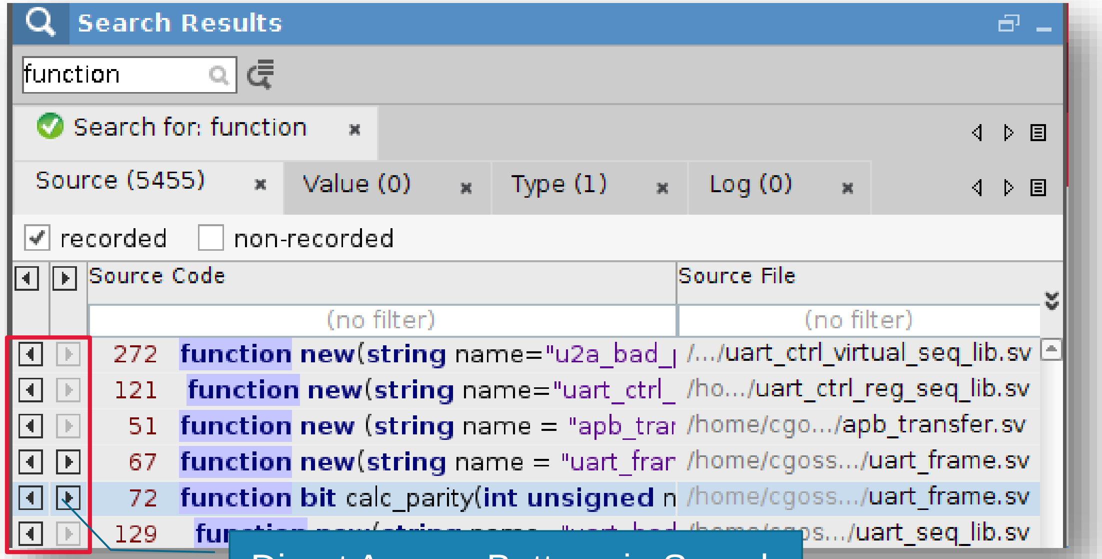


```
// This method calculates the parity
function bit calc_parity(int unsigned num_of_data_bits=8,
                         bit[1:0] ParityMode=0);
  bit temp_parity;

  if (num_of_data_bits == 6)
    temp_parity = ^payload[5:0];
  else if (num_of_data_bits == 7)
    temp_parity = ^payload[6:0];
  else
    temp_parity = ^payload;
```

Debug location updated

Direct Access Buttons in Source Viewer



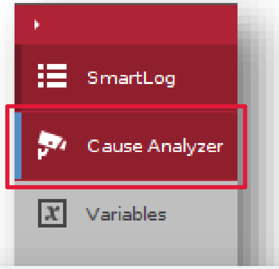
☑ recorded ☐ non-recorded

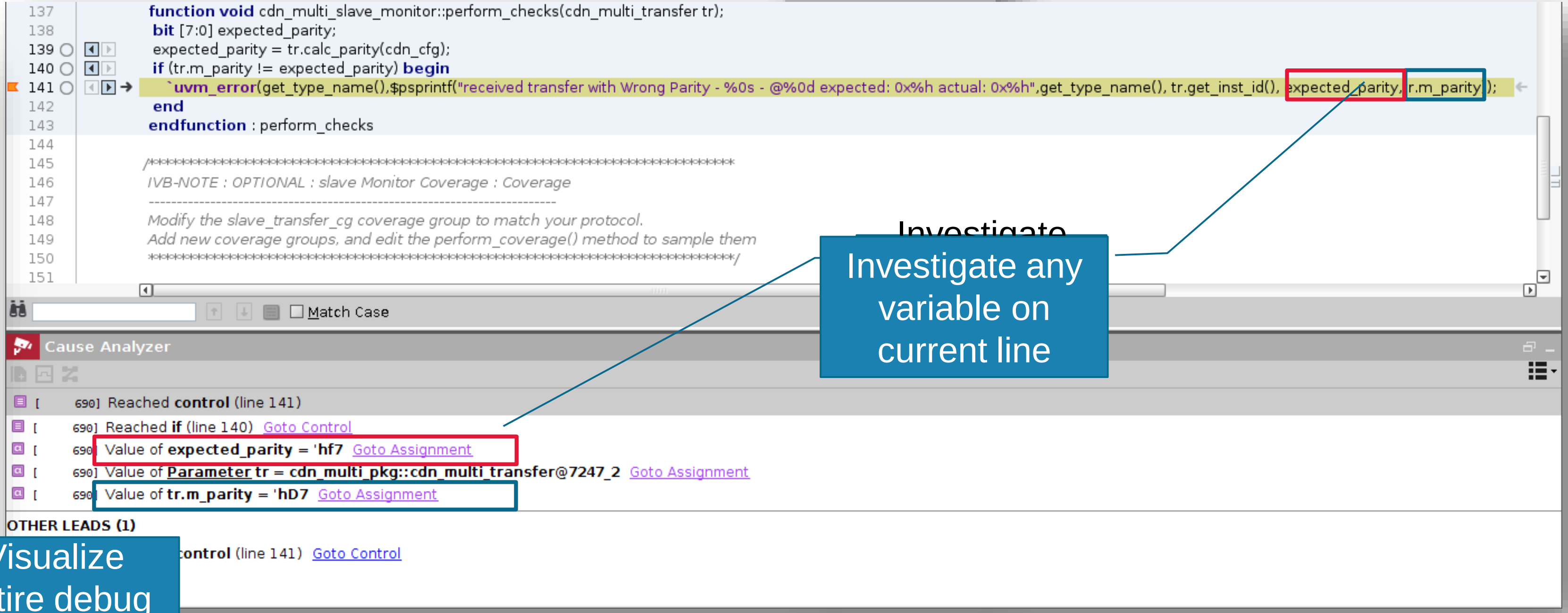
	Source Code	Source File
	(no filter)	(no filter)
272	function new(string name="u2a_bad_|	/.../uart_ctrl_virtual_seq_lib.sv
121	function new(string name="uart_ctrl_	/ho.../uart_ctrl_reg_seq_lib.sv
51	function new (string name = "apb_trar	/home/cgo.../apb_transfer.sv
67	function new(string name = "uart_frar	/home/cgoss.../uart_frame.sv
72	function bit calc_parity(int unsigned n	/home/cgoss.../uart_frame.sv
129	function new(string name="uart_bad	/home/cgoss.../uart_seq_lib.sv

Direct Access Buttons in Search Window. Clicking will open Source and set debug location

Indago™ Root Cause Analysis
Root Cause Analysis Component

- RCA Component provide a list of causal relationships to explore for any scenario
  - – Seamless language traversal from TB to RTL
  - – Saves entire debug decision tree
    - – Can revert back to previous point or launch new investigation
    - – Users no longer lost in the debug process





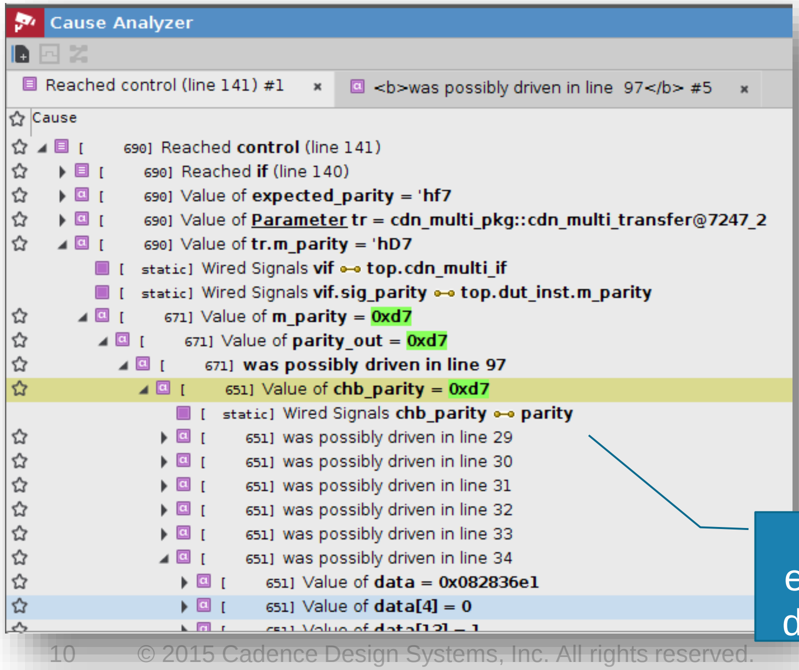
```
137        function void cdn_multi_slave_monitor::perform_checks(cdn_multi_transfer tr);
138         bit [7:0] expected_parity;
139         expected_parity = tr.calc_parity(cdn_cfg);
140         if (tr.m_parity != expected_parity) begin
141           `uvm_error(get_type_name(),$psprintf("received transfer with Wrong Parity - %0s - @%0d expected: 0x%h actual: 0x%h",get_type_name(), tr.get_inst_id(), expected_parity, tr.m_parity));
142         end
143        endfunction : perform_checks
144
145       /*******************************************************************************
146        IVB-NOTE : OPTIONAL : slave Monitor Coverage : Coverage
147        ------------------------------------------------------------------------
148        Modify the slave_transfer_cg coverage group to match your protocol.
149        Add new coverage groups, and edit the perform_coverage() method to sample them
150        *******************************************************************************/
151
```

☐ Match Case

Cause Analyzer

```
[    690] Reached control (line 141)
[    690] Reached if (line 140)  Goto Control
[    690] Value of expected_parity = 'hf7  Goto Assignment
[    690] Value of Parameter tr = cdn_multi_pkg::cdn_multi_transfer@7247_2  Goto Assignment
[    690] Value of tr.m_parity = 'hD7  Goto Assignment
OTHER LEADS (1)
         control (line 141)  Goto Control
```

Investigate any variable on current line



Cause Analyzer

Reached control (line 141) #1 × | <b>was possibly driven in line 97</b> #5 ×

```
Cause
[       690] Reached control (line 141)
  [       690] Reached if (line 140)
  [       690] Value of expected_parity = 'hf7
  [       690] Value of Parameter tr = cdn_multi_pkg::cdn_multi_transfer@7247_2
  [       690] Value of tr.m_parity = 'hD7
      [ static] Wired Signals vif ⊶ top.cdn_multi_if
      [ static] Wired Signals vif.sig_parity ⊶ top.dut_inst.m_parity
    [       671] Value of m_parity = 0xd7
      [       671] Value of parity_out = 0xd7
        [       671] was possibly driven in line 97
          [       651] Value of chb_parity = 0xd7
              [ static] Wired Signals chb_parity ⊶ parity
            [       651] was possibly driven in line 29
            [       651] was possibly driven in line 30
            [       651] was possibly driven in line 31
            [       651] was possibly driven in line 32
            [       651] was possibly driven in line 33
            [       651] was possibly driven in line 34
              [       651] Value of data = 0x082836e1
              [       651] Value of data[4] = 0
              [       651] Value of data[13] = 1
```

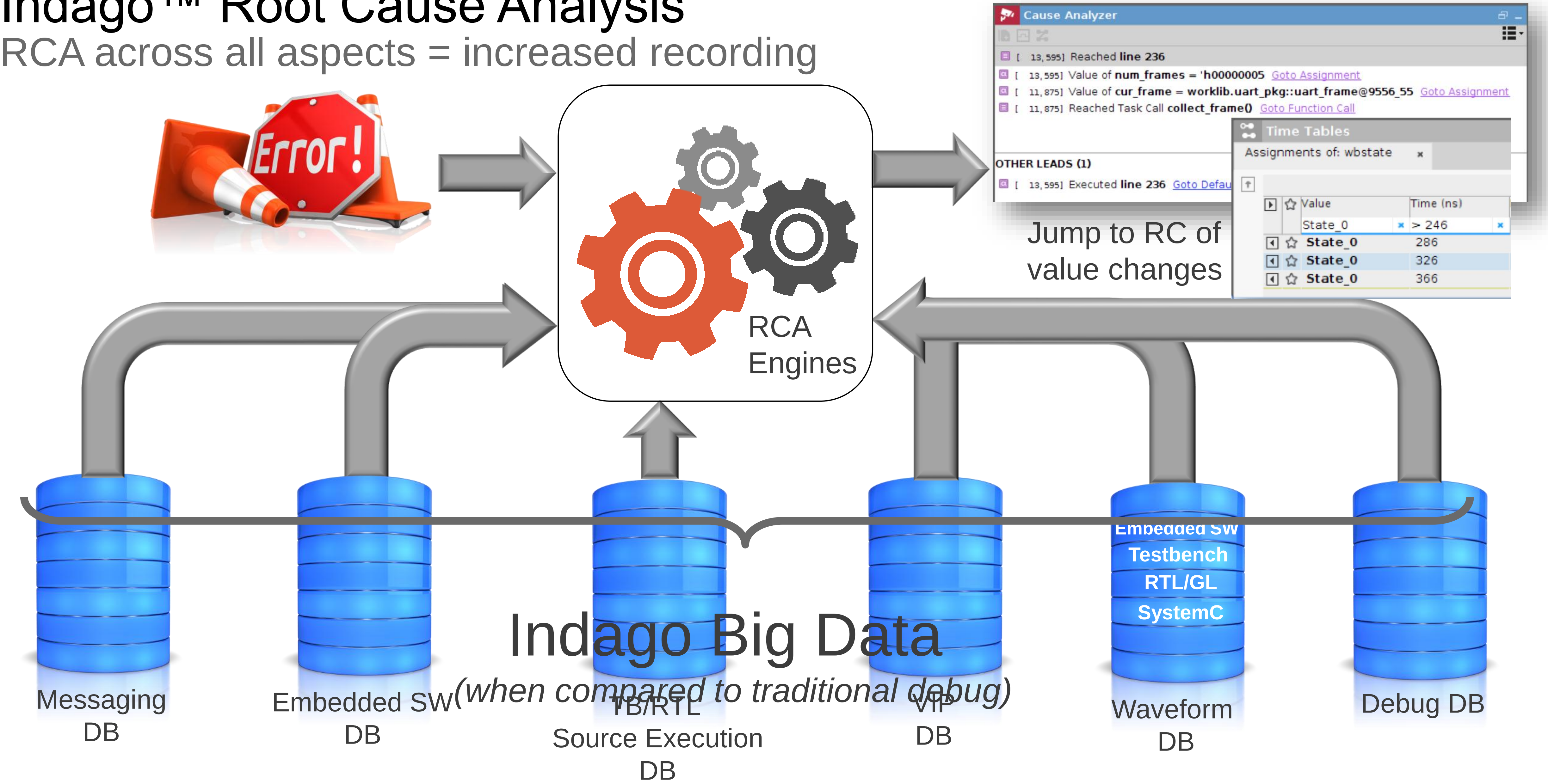
Visualize entire debug decision tree

cadence®

Indago™ Root Cause Analysis

RCA across all aspects = increased recording

Causal Relationships to explore



Cause Analyzer

```
[ 13,595] Reached line 236
[ 13,595] Value of num_frames = 'h00000005  Goto Assignment
[ 11,875] Value of cur_frame = worklib.uart_pkg::uart_frame@9556_55  Goto Assignment
[ 11,875] Reached Task Call collect_frame()  Goto Function Call

OTHER LEADS (1)
[ 13,595] Executed line 236  Goto Defau
```

Time Tables

Assignments of: wbstate

Value	Time (ns)
State_0	> 246
State_0	286
State_0	326
State_0	366

Jump to RC of value changes

Indago Big Data

(when compared to traditional debug)

Indago™ Big Data Analysis

What makes Indago unique

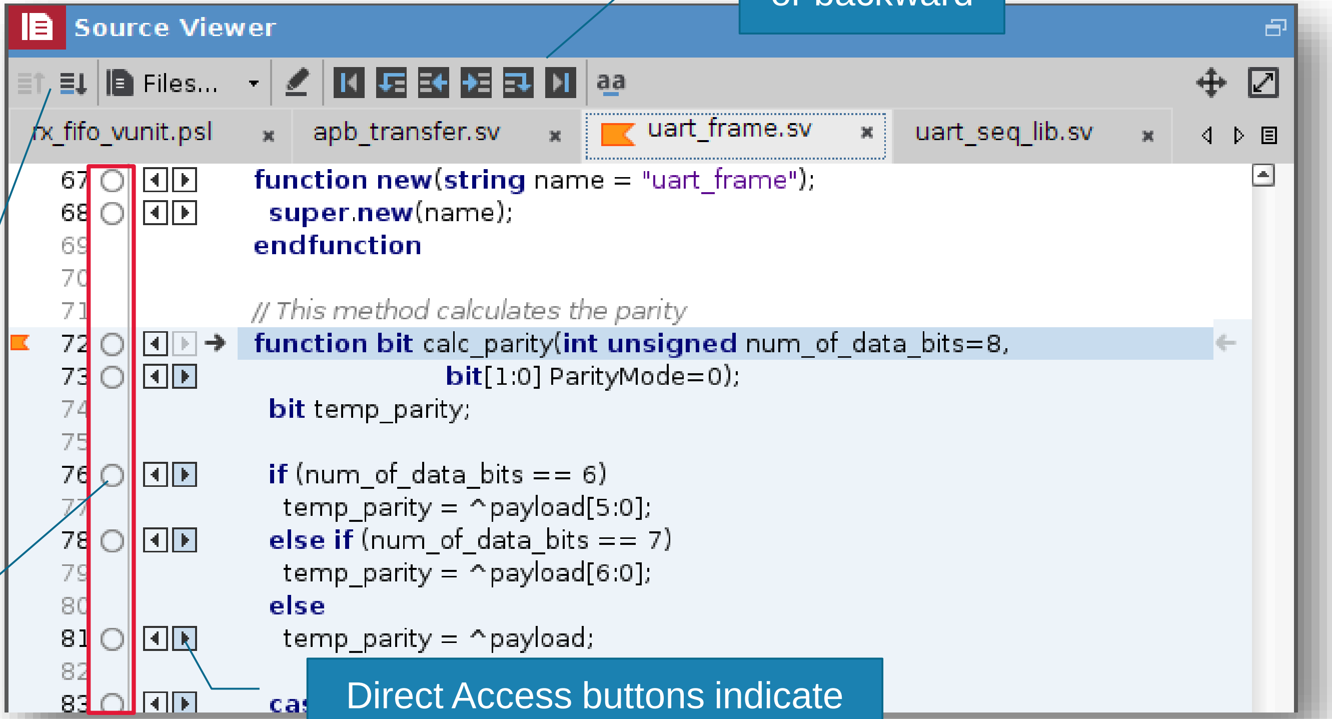
- Recording additional data allows for powerful analysis capabilities such as:
  - **Root Cause Analysis** (RCA)
    - RCA Component
    - Direct Access
  - **Playback Debugger** (forward/backward single stepping)
  - **SmartLog** (All messages saved to DB for querying/filtering as well as read/writing)
  - **SmartPrint** (write new print statement on the fly to the SmartLog DB)
  - **Time Tables** (charting of all accesses to objects over time)
  - **Powerful Searching** (organized, tabbed results)
  - **Call stack analysis** (walk through all stack frames in post process)
  - **Variables Table** (local/global variables accessible as you step)

- Let’s take a closer look at some of these features now …

*Available in ESW and IDA App

Indago™ Big Data Analysis
Playback Debugger*

- Allows for instant replay of the debug scenario without re-running
  - Step backward or forwards through time and space
  - Direct Access to any execution point
  - Code coverage visualization
    - Over recording window
  - Single click breakpoints
  - Tabs for multiple source files
  - Quick call stack walking
  - Access to all source files
  - Debug scope shaded background



```
function new(string name = "uart_frame");
  super.new(name);
endfunction

// This method calculates the parity
function bit calc_parity(int unsigned num_of_data_bits=8,
                         bit[1:0] ParityMode=0);
  bit temp_parity;

  if (num_of_data_bits == 6)
    temp_parity = ^payload[5:0];
  else if (num_of_data_bits == 7)
    temp_parity = ^payload[6:0];
  else
    temp_parity = ^payload;

  cas
```

Step forward or backward

Call stack up/down

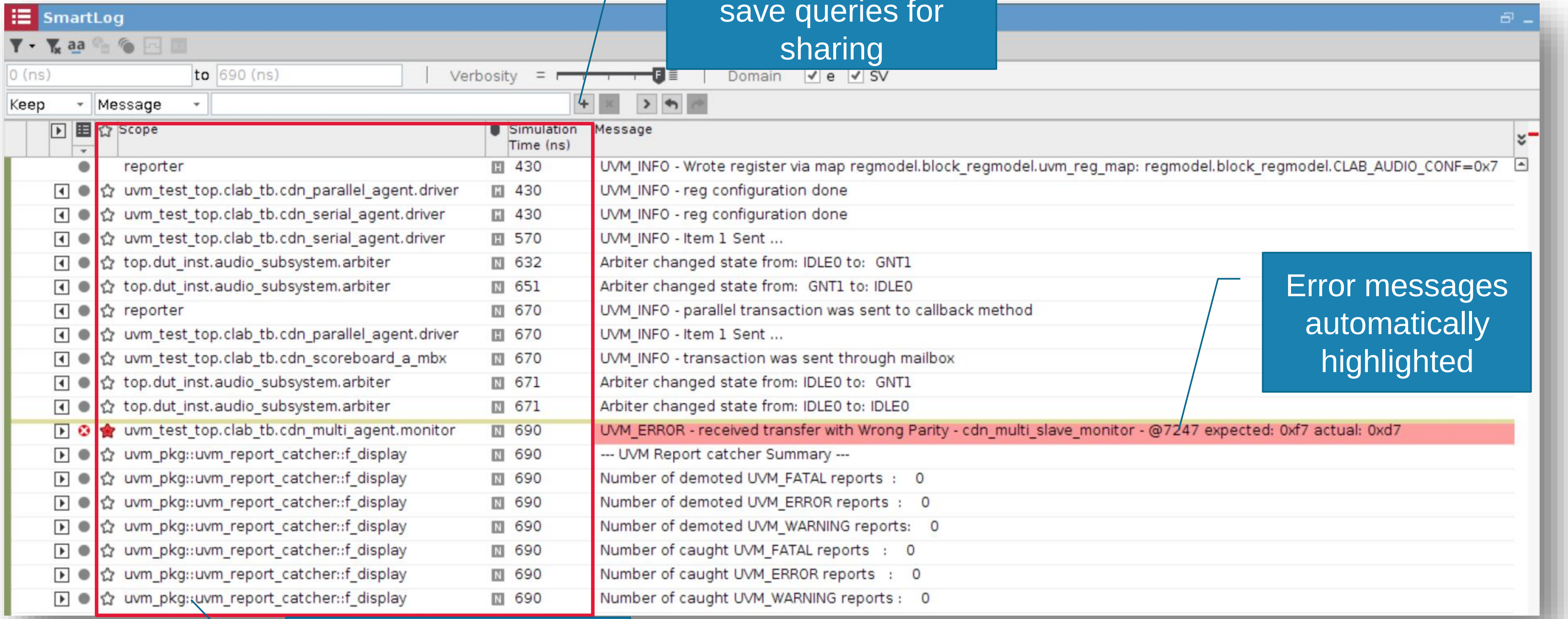
Set breakpoints

Direct Access buttons indicate which lines have been covered in recording window

Indago™ Big Data Analysis

SmartLog

- Write messages to SmartLog on the fly while debugging
- Messages from all languages (tool messages as well)
- Powerful filtering/querying
- Direct Access
- Colourized messages by type
- Debug Location indication
- Error message indication
- Message waveform visualization
- Verbosity slide control



0 (ns) to 690 (ns)

Keep | Message

Scope	Simulation Time (ns)	Message
reporter	430	UVM_INFO - Wrote register via map regmodel.block_regmodel.uvm_reg_map: regmodel.block_regmodel.CLAB_AUDIO_CONF=0x7
uvm_test_top.clab_tb.cdn_parallel_agent.driver	430	UVM_INFO - reg configuration done
uvm_test_top.clab_tb.cdn_serial_agent.driver	430	UVM_INFO - reg configuration done
uvm_test_top.clab_tb.cdn_serial_agent.driver	570	UVM_INFO - Item 1 Sent ...
top.dut_inst.audio_subsystem.arbiter	632	Arbiter changed state from: IDLE0 to: GNT1
top.dut_inst.audio_subsystem.arbiter	651	Arbiter changed state from: GNT1 to: IDLE0
reporter	670	UVM_INFO - parallel transaction was sent to callback method
uvm_test_top.clab_tb.cdn_parallel_agent.driver	670	UVM_INFO - Item 1 Sent ...
uvm_test_top.clab_tb.cdn_scoreboard_a_mbx	670	UVM_INFO - transaction was sent through mailbox
top.dut_inst.audio_subsystem.arbiter	671	Arbiter changed state from: IDLE0 to: GNT1
top.dut_inst.audio_subsystem.arbiter	671	Arbiter changed state from: IDLE0 to: IDLE0
uvm_test_top.clab_tb.cdn_multi_agent.monitor	690	UVM_ERROR - received transfer with Wrong Parity - cdn_multi_slave_monitor - @7247 expected: 0xf7 actual: 0xd7
uvm_pkg::uvm_report_catcher::f_display	690	--- UVM Report catcher Summary ---
uvm_pkg::uvm_report_catcher::f_display	690	Number of demoted UVM_FATAL reports : 0
uvm_pkg::uvm_report_catcher::f_display	690	Number of demoted UVM_ERROR reports : 0
uvm_pkg::uvm_report_catcher::f_display	690	Number of demoted UVM_WARNING reports: 0
uvm_pkg::uvm_report_catcher::f_display	690	Number of caught UVM_FATAL reports : 0
uvm_pkg::uvm_report_catcher::f_display	690	Number of caught UVM_ERROR reports : 0
uvm_pkg::uvm_report_catcher::f_display	690	Number of caught UVM_WARNING reports : 0

Can layer filters and save queries for sharing

Error messages automatically highlighted

Configurable columns allow complete control over messaging output

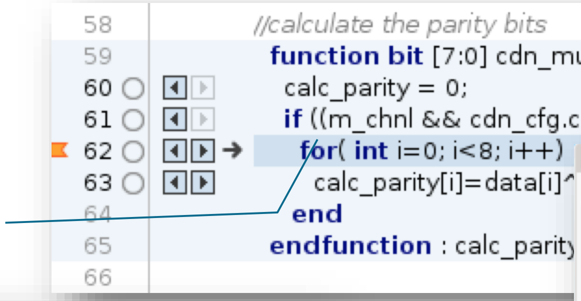
cādence®

*Available in ESW and IDA Apps

Indago™ Big Data Analysis

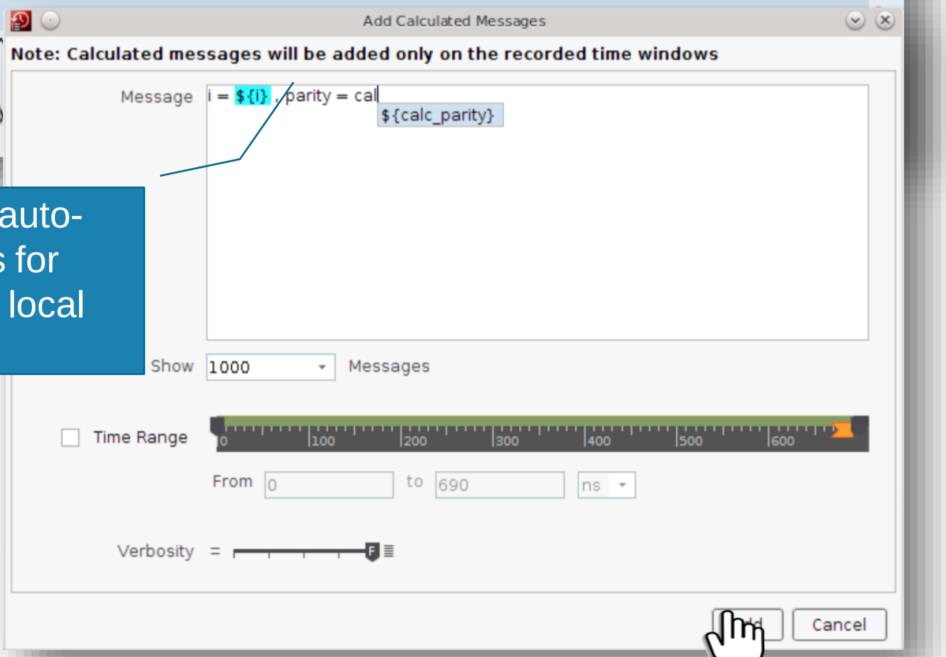
SmartPrint*

- Indago allows users to add a print statement to your log without re-running
- Saves lengthy debug iterations
  - Traditional debug flow requires adding print statement, recompile, re-elaborate, re-run, remove print statements



```
//calculate the parity bits
function bit [7:0] cdn_multi_transfer::calc_parity(cdn_regs_cfg cdn_cfg);
  calc_parity = 0;
  if ((m_chnl && cdn_cfg.chnl_b.parity_en) || (!m_chnl && cdn_cfg.chnl_a.parity_en)) begin
    for( int i=0; i<8; i++)
      calc_parity[i]=data[i]^
  end
endfunction : calc_parity
```

Activate SmartPrint for any line with Direct Access through RMB



Add Calculated Messages

Note: Calculated messages will be added only on the recorded time windows

Message: i = ${i} , parity = cal
${calc_parity}

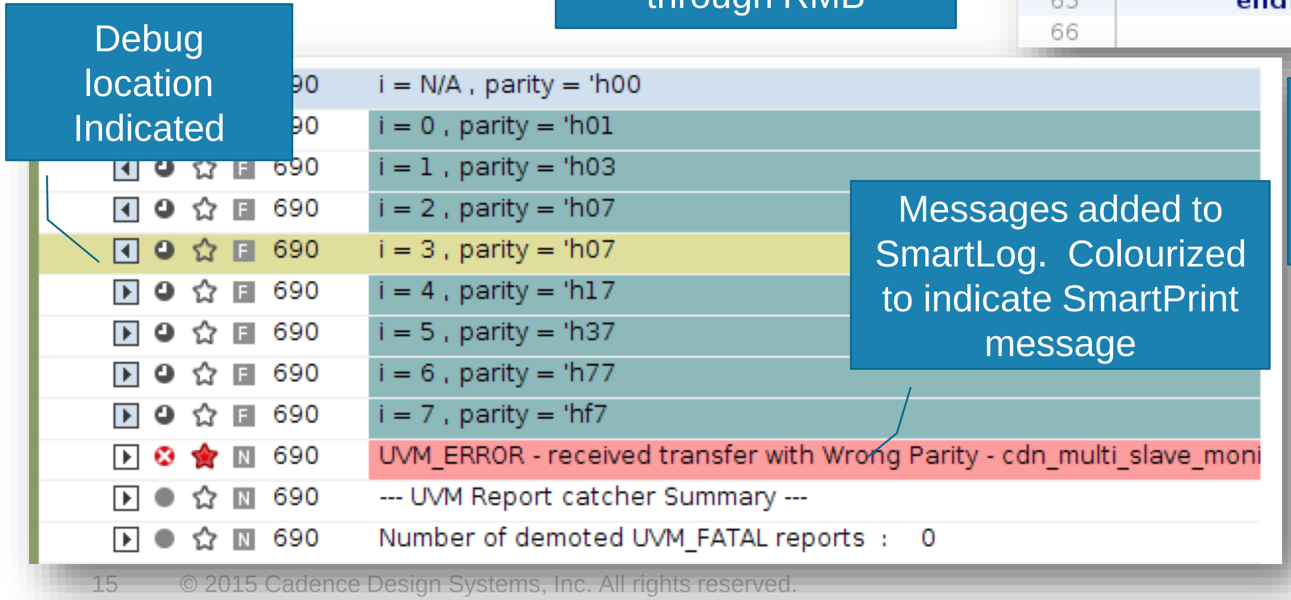
Show 1000 Messages

☐ Time Range

From 0 to 690 ns

Verbosity =

SmartPrint auto-completes for variables in local scope



Debug location Indicated

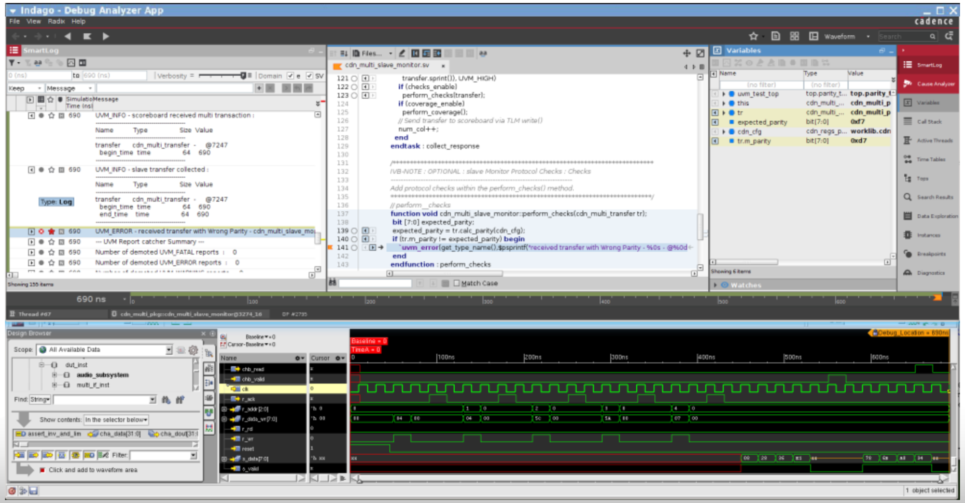
```
690  i = N/A , parity = 'h00
690  i = 0 , parity = 'h01
690  i = 1 , parity = 'h03
690  i = 2 , parity = 'h07
690  i = 3 , parity = 'h07
690  i = 4 , parity = 'h17
690  i = 5 , parity = 'h37
690  i = 6 , parity = 'h77
690  i = 7 , parity = 'hf7
690  UVM_ERROR - received transfer with Wrong Parity - cdn_multi_slave_moni
690  --- UVM Report catcher Summary ---
690  Number of demoted UVM_FATAL reports  :    0
```

Messages added to SmartLog. Colourized to indicate SmartPrint message

Indago™ Debug Platform

Unified Analysis GUI

- Debug data from all sources visualized in the same GUI
  - Eliminates GUI context switching
  - Consistent debug experience
  - Quick ramp up
  - Unified RCA across debug data sources
  - Complete synchronization
  - App specific customization



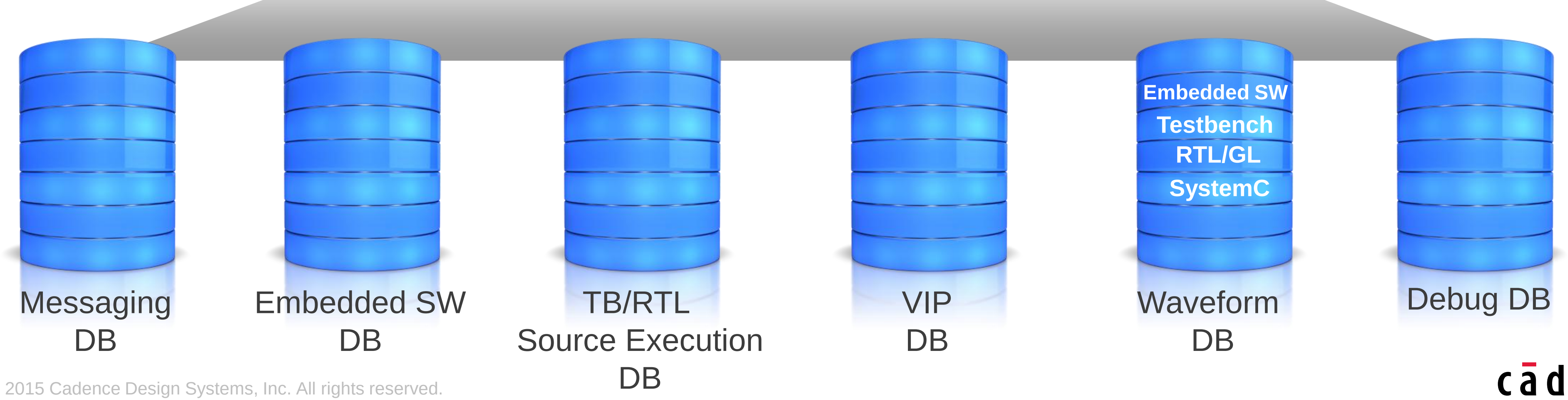
```
690   UVM_INFO - scoreboard received multi transaction :
      ---------------------------------------------------
      Name          Type                Size  Value
      ---------------------------------------------------
      transfer      cdn_multi_transfer  -     @7247
        begin_time  time                64    690
      ---------------------------------------------------
690   UVM_INFO - slave transfer collected :
      ---------------------------------------------------
      Name          Type                Size  Value
      ---------------------------------------------------
      transfer      cdn_multi_transfer  -     @7247
        begin_time  time                64    690
        end_time    time                64    690
      ---------------------------------------------------
690   UVM_ERROR - received transfer with Wrong Parity - cdn_multi_slave_mo
690   --- UVM Report catcher Summary ---
690   Number of demoted UVM_FATAL reports  :   0
690   Number of demoted UVM_ERROR reports  :   0
```

```
      transfer.sprint()), UVM_HIGH)
    if (checks_enable)
      perform_checks(transfer);
    if (coverage_enable)
      perform_coverage();
    // Send transfer to scoreboard via TLM write()
    num_col++;
  end
endtask : collect_response

/*****************************************************************
IVB-NOTE : OPTIONAL : slave Monitor Protocol Checks : Checks
-----------------------------------------------------------------
Add protocol checks within the perform_checks() method.
*****************************************************************/
// perform__checks
function void cdn_multi_slave_monitor::perform_checks(cdn_multi_transfer tr);
 bit [7:0] expected_parity;
 expected_parity = tr.calc_parity(cdn_cfg);
 if (tr.m_parity != expected_parity) begin
   `uvm_error(get_type_name(),$psprintf("received transfer with Wrong Parity - %0s - @%0d
 end
endfunction : perform_checks
```

Name	Type	Value
uvm_test_top	top.parity_t...	top.parity_t
this	cdn_multi_...	cdn_multi_p
tr	cdn_multi_...	cdn_multi_p
expected_parity	bit[7:0]	0xf7
cdn_cfg	cdn_regs_p...	worklib.cdn
tr.m_parity	bit[7:0]	0xd7

Showing 6 items



Indago™ Unified Analysis GUI

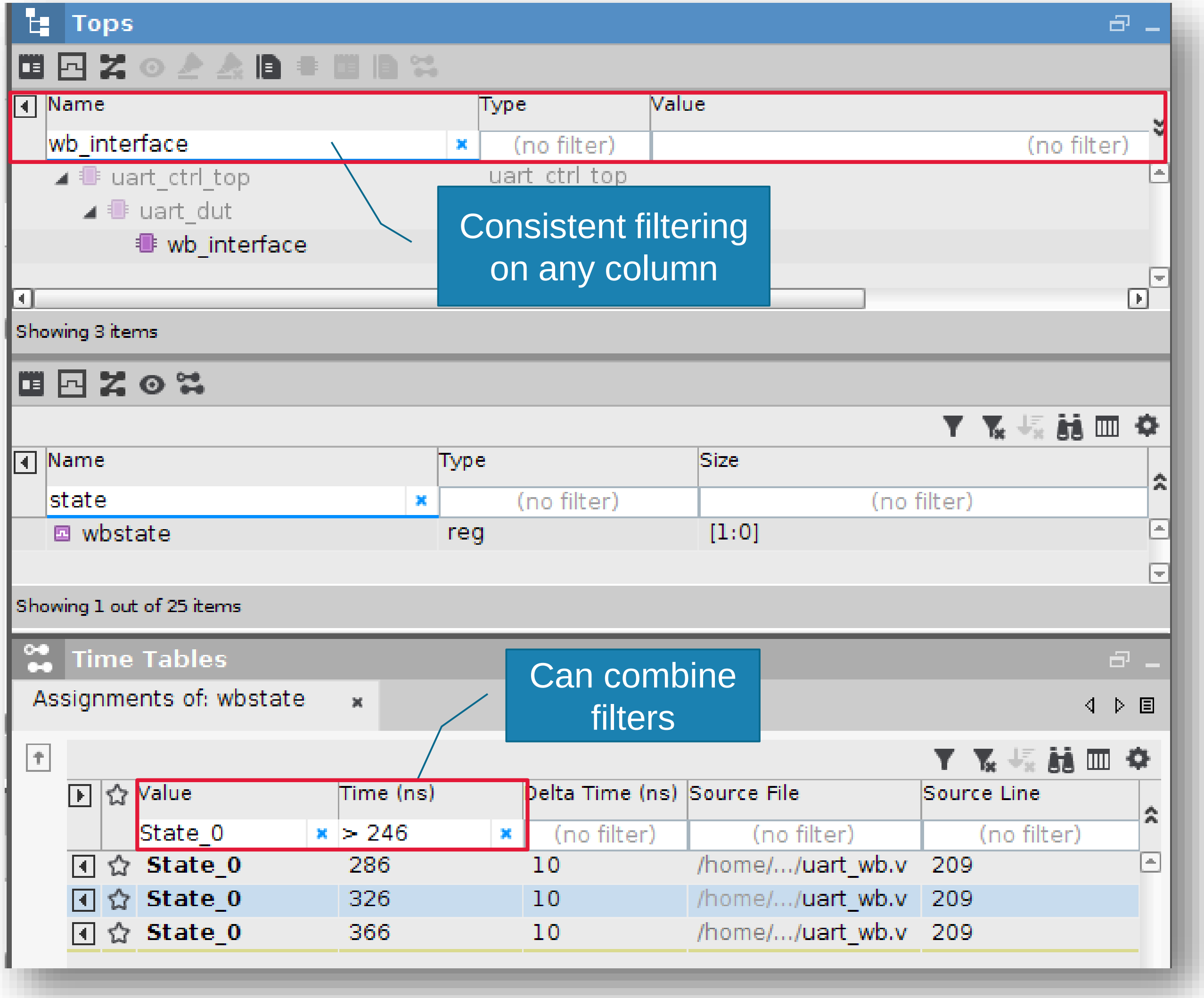
What makes Indago unique

- The GUI itself has many unique features, including:
  - Quick Launch (easy to access supporting debug components)
  - Filtering and sorting in most components (allows users to find information quickly)
  - Debug Stars (set bookmarks for quick return to any debug location)
  - Value Highlighting (visual comparisons/pattern identification through colourization)
  - Debug Location Indication (clear marker in all components)
  - Debug Notes (capture debug information for quick handoff)
  - Debug Handoff (save the state of the entire GUI for quick handoff)
- Let’s take a closer look at some of these features now …

Indago™ Unified Analysis GUI

Filtering and Sorting

- Fast search and filtering is key to finding the right information quickly
- Most all windows in Indago have a consistent filtering solution
  - Can filter criteria for each column
  - Can combine filters to narrow results



Tops

Name	Type	Value
wb_interface	(no filter)	(no filter)
uart_ctrl_top		uart_ctrl_top
uart_dut		
wb_interface		

Showing 3 items

Name	Type	Size
state	(no filter)	(no filter)
wbstate	reg	[1:0]

Showing 1 out of 25 items

Time Tables

Assignments of: wbstate

Value	Time (ns)	Delta Time (ns)	Source File	Source Line
State_0	> 246	(no filter)	(no filter)	(no filter)
State_0	286	10	/home/.../uart_wb.v	209
State_0	326	10	/home/.../uart_wb.v	209
State_0	366	10	/home/.../uart_wb.v	209

Consistent filtering on any column

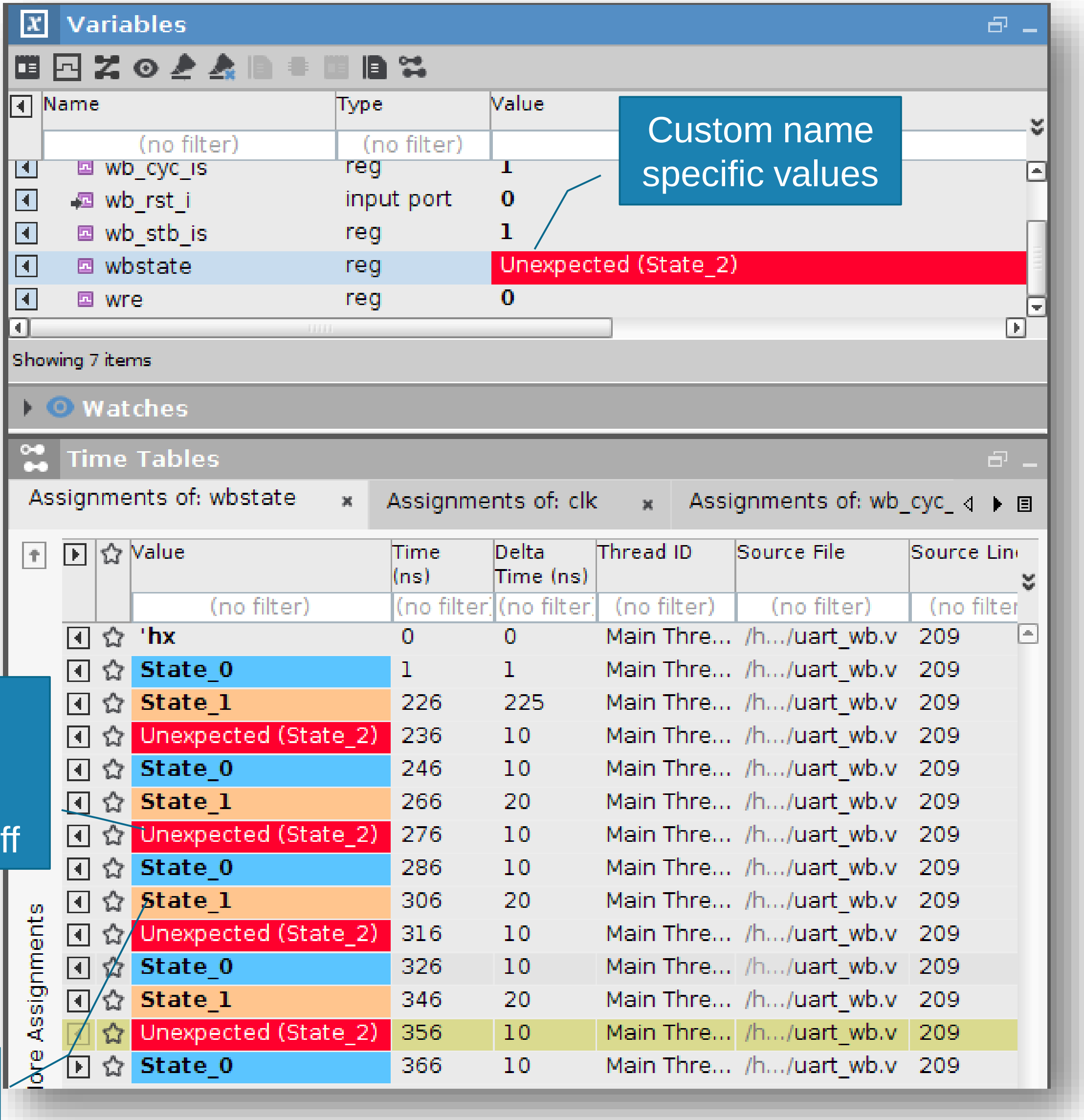
Can combine filters

cadence®

Indago™ Unified Analysis GUI

Value Highlighting

- Tag values with a colour or custom name
- All matching GUI values are highlighted
- Quick compare of values
  - No need to write value down
- Recognize patterns within data sets
- Colouring also applied to waveforms
- Very useful for debug handoff



Variables

Name	Type	Value
(no filter)	(no filter)	
wb_cyc_is	reg	1
wb_rst_i	input port	0
wb_stb_is	reg	1
wbstate	reg	Unexpected (State_2)
wre	reg	0

Showing 7 items

Watches

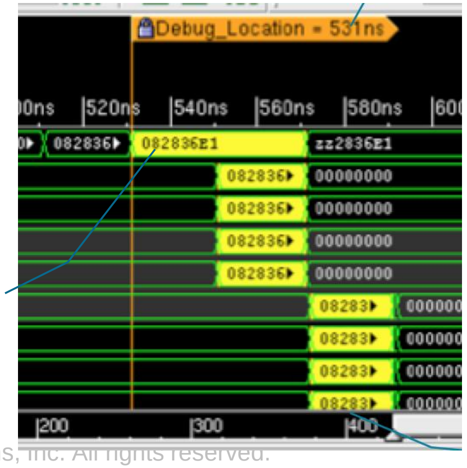
Time Tables

Assignments of: wbstate × | Assignments of: clk × | Assignments of: wb_cyc_

Value	Time (ns)	Delta Time (ns)	Thread ID	Source File	Source Lin
(no filter)	(no filter	(no filter	(no filter)	(no filter)	(no filter
'hx	0	0	Main Thre...	/h.../uart_wb.v	209
State_0	1	1	Main Thre...	/h.../uart_wb.v	209
State_1	226	225	Main Thre...	/h.../uart_wb.v	209
Unexpected (State_2)	236	10	Main Thre...	/h.../uart_wb.v	209
State_0	246	10	Main Thre...	/h.../uart_wb.v	209
State_1	266	20	Main Thre...	/h.../uart_wb.v	209
Unexpected (State_2)	276	10	Main Thre...	/h.../uart_wb.v	209
State_0	286	10	Main Thre...	/h.../uart_wb.v	209
State_1	306	20	Main Thre...	/h.../uart_wb.v	209
Unexpected (State_2)	316	10	Main Thre...	/h.../uart_wb.v	209
State_0	326	10	Main Thre...	/h.../uart_wb.v	209
State_1	346	20	Main Thre...	/h.../uart_wb.v	209
Unexpected (State_2)	356	10	Main Thre...	/h.../uart_wb.v	209
State_0	366	10	Main Thre...	/h.../uart_wb.v	209

Custom name specific values

Highlight unexpected values for debug handoff

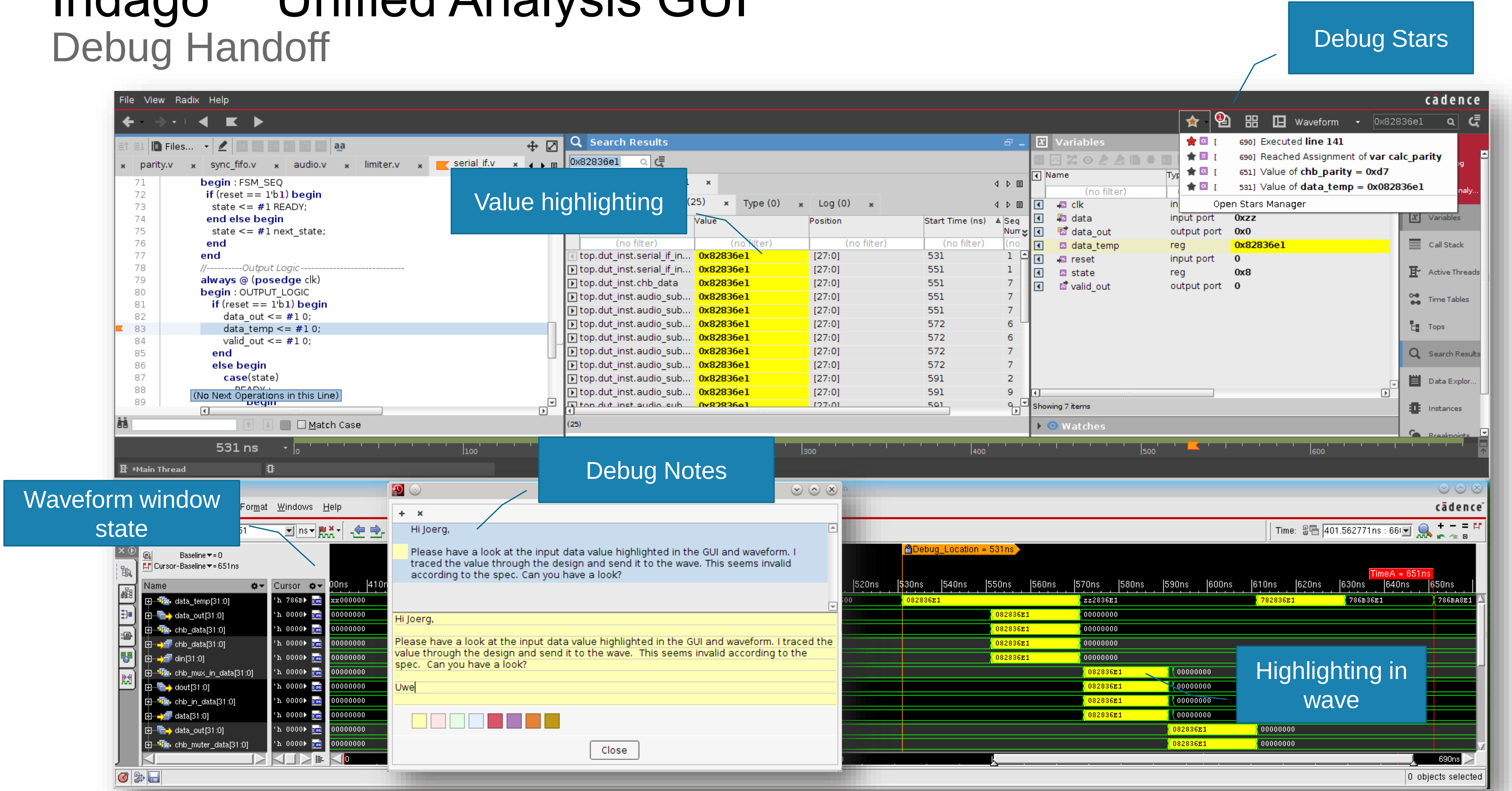


Highlighting also appears in waveform

Colours help identify patterns within the data

Indago™ Unified Analysis GUI

Debug Handoff



```
begin : FSM_SEQ
 if (reset == 1'b1) begin
  state <= #1 READY;
 end else begin
  state <= #1 next_state;
 end
end
//---------Output Logic--------------------------
always @ (posedge clk)
begin : OUTPUT_LOGIC
  if (reset == 1'b1) begin
    data_out <= #1 0;
    data_temp <= #1 0;
    valid_out <= #1 0;
  end
  else begin
    case(state)
```

(No Next Operations in this Line)

(25) Type (0) Log (0)

	Value	Position	Start Time (ns)	Seq Num
top.dut_inst.serial_if_in...	0x82836e1	[27:0]	531	1
top.dut_inst.serial_if_in...	0x82836e1	[27:0]	551	1
top.dut_inst.chb_data	0x82836e1	[27:0]	551	7
top.dut_inst.audio_sub...	0x82836e1	[27:0]	551	7
top.dut_inst.audio_sub...	0x82836e1	[27:0]	551	7
top.dut_inst.audio_sub...	0x82836e1	[27:0]	572	6
top.dut_inst.audio_sub...	0x82836e1	[27:0]	572	6
top.dut_inst.audio_sub...	0x82836e1	[27:0]	572	7
top.dut_inst.audio_sub...	0x82836e1	[27:0]	572	7
top.dut_inst.audio_sub...	0x82836e1	[27:0]	591	2
top.dut_inst.audio_sub...	0x82836e1	[27:0]	591	9

Name	Type	
clk	in	
data	input port	0xzz
data_out	output port	0x0
data_temp	reg	0x82836e1
reset	input port	0
state	reg	0x8
valid_out	output port	0

- ★ [690] Executed **line 141**
- ★ [690] Reached Assignment of **var calc_parity**
- ★ [651] Value of **chb_parity = 0xd7**
- ★ [531] Value of **data_temp = 0x082836e1**

Open Stars Manager

Hi Joerg,

Please have a look at the input data value highlighted in the GUI and waveform. I traced the value through the design and send it to the wave. This seems invalid according to the spec. Can you have a look?

Hi Joerg,

Please have a look at the input data value highlighted in the GUI and waveform. I traced the value through the design and send it to the wave. This seems invalid according to the spec. Can you have a look?

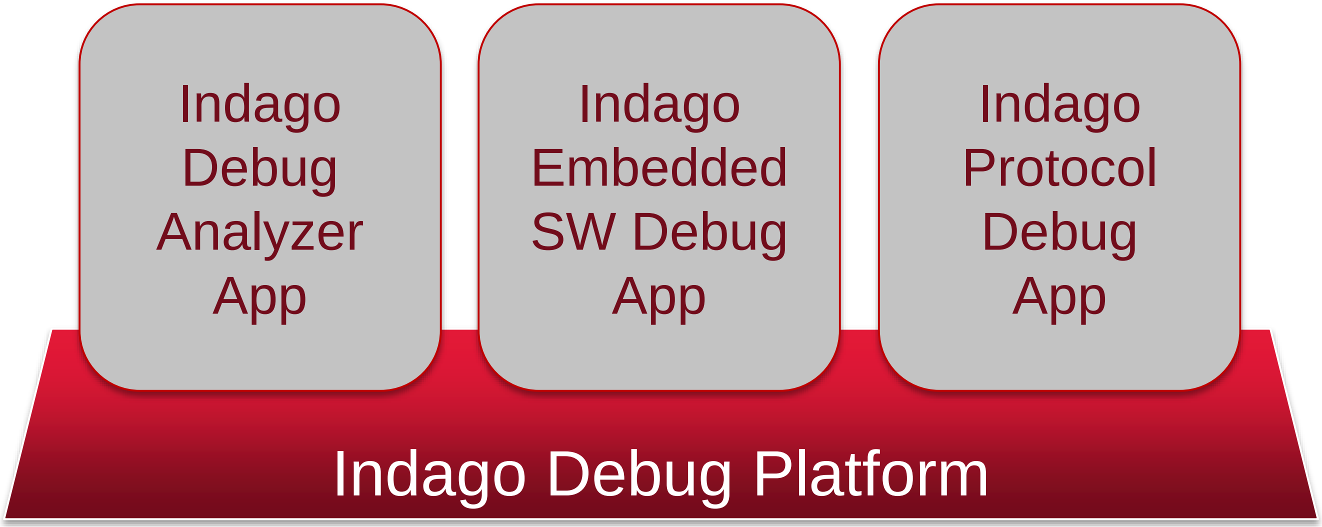
Uwe

Close

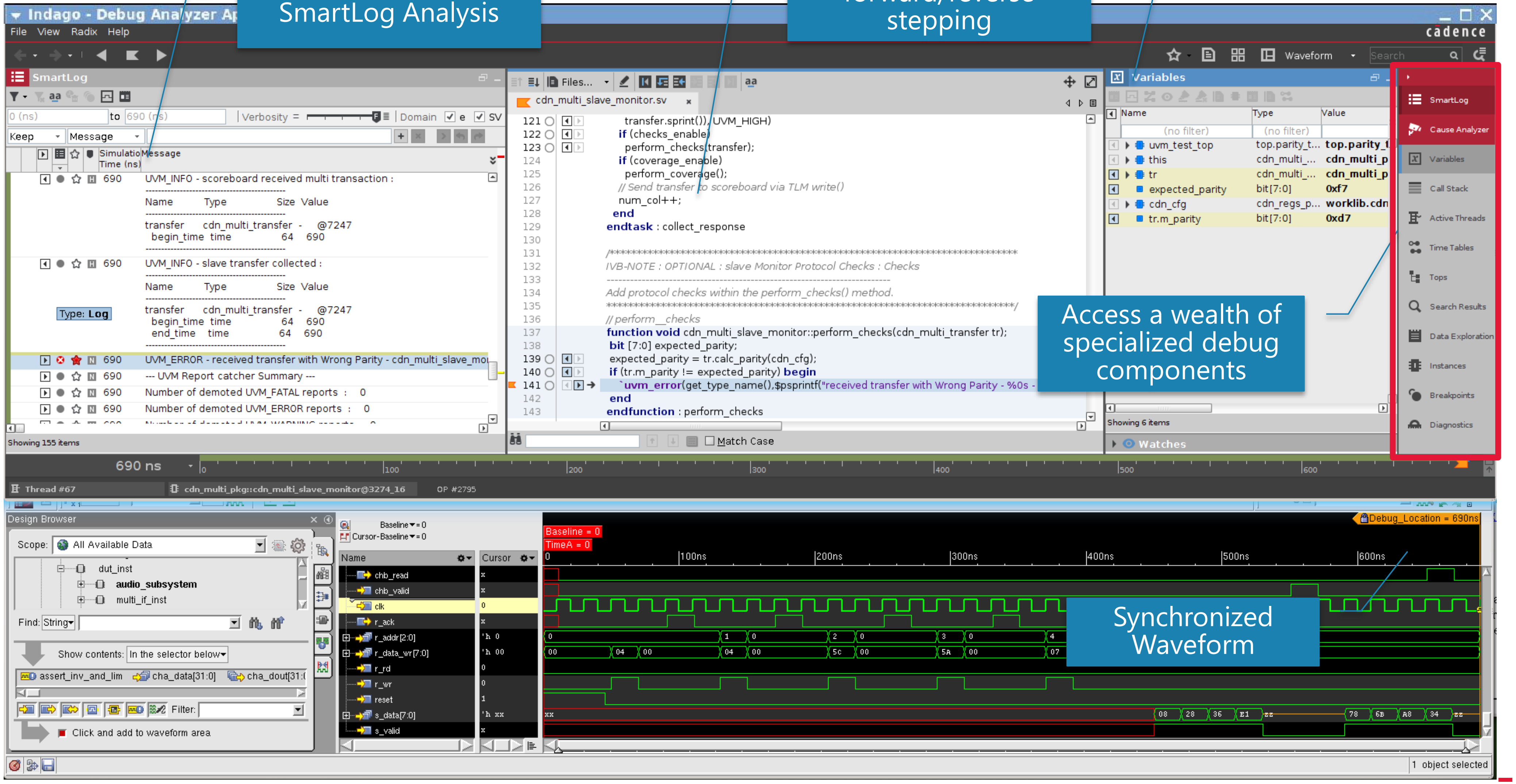
Name	Cursor
data_temp[31:0]	'h 786E
data_out[31:0]	'h 0000
chb_data[31:0]	'h 0000
chb_data[31:0]	'h 0000
din[31:0]	'h 0000
chb_mux_in_data[31:0]	'h 0000
dout[31:0]	'h 0000
chb_in_data[31:0]	'h 0000
data[31:0]	'h 0000
data_out[31:0]	'h 0000
chb_muter_data[31:0]	'h 0000

Indago™ Apps

- Apps are individual products targeted at a specific debug task
  - Indago Debug Analyzer App: RTL/GL/TB debug
  - Indago Embedded SW Debug App: Embedded SW/HW Debug
  - Indago Protocol Debug App: Debug of Verification IP Protocol Traffic


Indago Debug Analyzer App
Indago Embedded SW Debug App
Indago Protocol Debug App
Indago Debug Platform

Indago™ Debug Analyzer App



SmartLog Analysis

Source Viewer with forward/reverse stepping

Variables Table

Access a wealth of specialized debug components

Synchronized Waveform

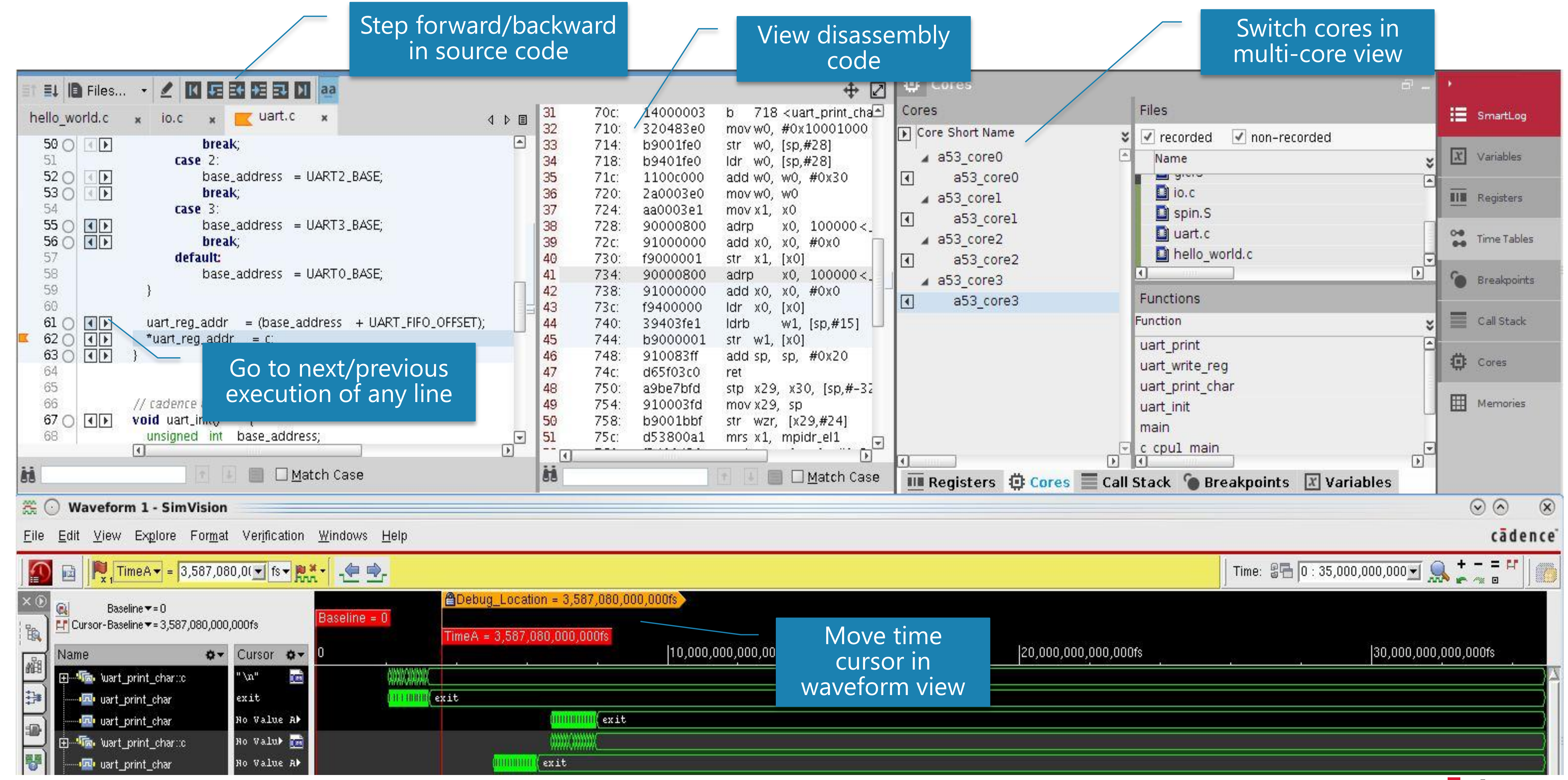
```
690  UVM_INFO - scoreboard received multi transaction :
     ------------------------------------------
     Name          Type               Size  Value
     ------------------------------------------
     transfer      cdn_multi_transfer  -     @7247
       begin_time  time                64    690
     ------------------------------------------
690  UVM_INFO - slave transfer collected :
     ------------------------------------------
     Name          Type               Size  Value
     ------------------------------------------
     transfer      cdn_multi_transfer  -     @7247
       begin_time  time                64    690
       end_time    time                64    690
     ------------------------------------------
690  UVM_ERROR - received transfer with Wrong Parity - cdn_multi_slave_mo
690  --- UVM Report catcher Summary ---
690  Number of demoted UVM_FATAL reports  :   0
690  Number of demoted UVM_ERROR reports  :   0
```

```
      transfer.sprint()), UVM_HIGH)
    if (checks_enable)
      perform_checks(transfer);
    if (coverage_enable)
      perform_coverage();
    // Send transfer to scoreboard via TLM write()
    num_col++;
  end
endtask : collect_response

/*****************************************************************************
IVB-NOTE : OPTIONAL : slave Monitor Protocol Checks : Checks
-----------------------------------------------------------------------------
Add protocol checks within the perform_checks() method.
*****************************************************************************/
// perform__checks
function void cdn_multi_slave_monitor::perform_checks(cdn_multi_transfer tr);
 bit [7:0] expected_parity;
 expected_parity = tr.calc_parity(cdn_cfg);
 if (tr.m_parity != expected_parity) begin
   `uvm_error(get_type_name(),$psprintf("received transfer with Wrong Parity - %0s -
 end
endfunction : perform_checks
```

Name	Type	Value
uvm_test_top	top.parity_t...	top.parity_t
this	cdn_multi_...	cdn_multi_p
tr	cdn_multi_...	cdn_multi_p
expected_parity	bit[7:0]	0xf7
cdn_cfg	cdn_regs_p...	worklib.cdn
tr.m_parity	bit[7:0]	0xd7

Indago™ Embedded Software Debug App



Step forward/backward in source code

View disassembly code

Switch cores in multi-core view

Go to next/previous execution of any line

Move time cursor in waveform view

```
            break;
        case 2:
            base_address  = UART2_BASE;
            break;
        case 3:
            base_address  = UART3_BASE;
            break;
        default:
            base_address  = UART0_BASE;
    }

    uart_reg_addr   = (base_address  + UART_FIFO_OFFSET);
    *uart_reg_addr    = c;
}

// cadence
void uart_in
    unsigned  int   base_address;
```

```
70c:  14000003  b    718 <uart_print_cha
710:  320483e0  mov w0,  #0x10001000
714:  b9001fe0  str  w0,  [sp,#28]
718:  b9401fe0  ldr  w0,  [sp,#28]
71c:  1100c000  add w0,  w0,  #0x30
720:  2a0003e0  mov w0,  w0
724:  aa0003e1  mov x1,  x0
728:  90000800  adrp     x0,  100000 <_
72c:  91000000  add x0,  x0,  #0x0
730:  f9000001  str  x1,  [x0]
734:  90000800  adrp     x0,  100000 <_
738:  91000000  add x0,  x0,  #0x0
73c:  f9400000  ldr  x0,  [x0]
740:  39403fe1  ldrb     w1,  [sp,#15]
744:  b9000001  str  w1,  [x0]
748:  910083ff  add sp,  sp,  #0x20
74c:  d65f03c0  ret
750:  a9be7bfd  stp  x29,  x30,  [sp,#-32
754:  910003fd  mov x29,  sp
758:  b9001bbf  str  wzr,  [x29,#24]
75c:  d53800a1  mrs  x1,  mpidr_el1
```

- a53_core0
  - a53_core0
- a53_core1
  - a53_core1
- a53_core2
  - a53_core2
- a53_core3
  - a53_core3

Files: io.c, spin.S, uart.c, hello_world.c

Functions: uart_print, uart_write_reg, uart_print_char, uart_init, main, c_cpu1_main

TimeA = 3,587,080,0(

Time: 0 : 35,000,000,000

Baseline = 0
Cursor-Baseline = 3,587,080,000,000fs

Debug_Location = 3,587,080,000,000fs
Baseline = 0
TimeA = 3,587,080,000,000fs
0 | 10,000,000,000,00 | 20,000,000,000,000fs | 30,000,000,000,000fs

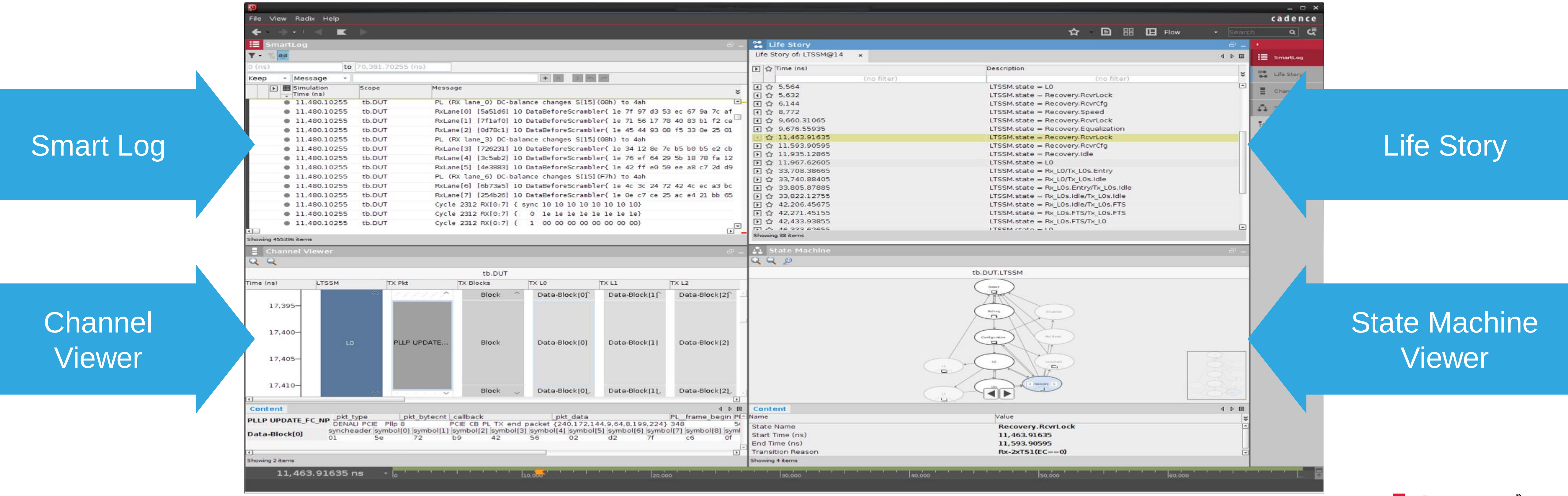
Name	Cursor
\uart_print_char::c	"\n"
uart_print_char	exit
uart_print_char	No Value A
\uart_print_char::c	No Valu
uart_print_char	No Value A

exit
exit
exit

Indago™ Protocol Debug App

Next-generation protocol debug aid

- Simplifies debug by illuminating design and VIP behavior
- Support for 12 popular protocols in 2015, others to follow
- Seamless integration with all major simulators



Smart Log

Life Story

Channel Viewer

State Machine Viewer

SmartLog

0 (ns) to 70,381.70255 (ns)

Keep · Message

```
Simulation Time (ns)  Scope    Message
11,480.10255          tb.DUT   PL (RX lane_0) DC-balance changes S[15](08h) to 4ah
11,480.10255          tb.DUT   RxLane[0] [5a51d6] 10 DataBeforeScrambler{ 1e 7f 97 d3 53 ec 67 9a 7c af
11,480.10255          tb.DUT   RxLane[1] [7f1af0] 10 DataBeforeScrambler{ 1e 71 56 17 78 40 83 b1 f2 ca
11,480.10255          tb.DUT   RxLane[2] [0d78c1] 10 DataBeforeScrambler{ 1e 45 44 93 08 f5 33 0e 25 01
11,480.10255          tb.DUT   PL (RX lane_3) DC-balance changes S[15](08h) to 4ah
11,480.10255          tb.DUT   RxLane[3] [726231] 10 DataBeforeScrambler{ 1e 34 12 8e 7e b5 b0 b5 e2 cb
11,480.10255          tb.DUT   RxLane[4] [3c5ab2] 10 DataBeforeScrambler{ 1e 76 ef 64 29 5b 18 78 fa 12
11,480.10255          tb.DUT   RxLane[5] [4e3883] 10 DataBeforeScrambler{ 1e 42 ff e0 59 ee a8 c7 2d d9
11,480.10255          tb.DUT   PL (RX lane_6) DC-balance changes S[15](F7h) to 4ah
11,480.10255          tb.DUT   RxLane[6] [6b73a5] 10 DataBeforeScrambler{ 1e 4c 3c 24 72 42 4c ec a3 bc
11,480.10255          tb.DUT   RxLane[7] [254b26] 10 DataBeforeScrambler{ 1e 0e c7 ce 25 ac e4 21 bb 65
11,480.10255          tb.DUT   Cycle 2312 RX[0:7] { sync 10 10 10 10 10 10 10 10}
11,480.10255          tb.DUT   Cycle 2312 RX[0:7] {   0  1e 1e 1e 1e 1e 1e 1e 1e}
11,480.10255          tb.DUT   Cycle 2312 RX[0:7] {   1  00 00 00 00 00 00 00 00}
Showing 455396 items
```

Life Story of: LTSSM@14

Time (ns)	Description
(no filter)	(no filter)
5,564	LTSSM.state = L0
5,632	LTSSM.state = Recovery.RcvrLock
6,144	LTSSM.state = Recovery.RcvrCfg
8,772	LTSSM.state = Recovery.Speed
9,660.31065	LTSSM.state = Recovery.RcvrLock
9,676.55935	LTSSM.state = Recovery.Equalization
11,463.91635	LTSSM.state = Recovery.RcvrLock
11,593.90595	LTSSM.state = Recovery.RcvrCfg
11,935.12865	LTSSM.state = Recovery.Idle
11,967.62605	LTSSM.state = L0
33,708.38665	LTSSM.state = Rx_L0/Tx_L0s.Entry
33,740.88405	LTSSM.state = Rx_L0/Tx_L0s.Idle
33,805.87885	LTSSM.state = Rx_L0s.Entry/Tx_L0s.Idle
33,822.12755	LTSSM.state = Rx_L0s.Idle/Tx_L0s.Idle
42,206.45675	LTSSM.state = Rx_L0s.Idle/Tx_L0s.FTS
42,271.45155	LTSSM.state = Rx_L0s.FTS/Tx_L0s.FTS
42,433.93855	LTSSM.state = Rx_L0s.FTS/Tx_L0

Showing 38 items

Content

```
PLLP UPDATE_FC_NP  _pkt_type    _pkt_bytecnt  _callback            _pkt_data                                PL_frame_begin
                   DENALI PCIE  Pllp 8        PCIE CB PL TX end packet {240,172,144,9,64,8,199,224}  348
Data-Block[0]      syncheader symbol[0] symbol[1] symbol[2] symbol[3] symbol[4] symbol[5] symbol[6] symbol[7] symbol[8]
                   01         5e        72        b9        42        56        02        d2        7f        c6        0f
Showing 2 items
```

Content

Name	Value
State Name	Recovery.RcvrLock
Start Time (ns)	11,463.91635
End Time (ns)	11,593.90595
Transition Reason	Rx-2xTS1(EC==0)

Showing 4 items

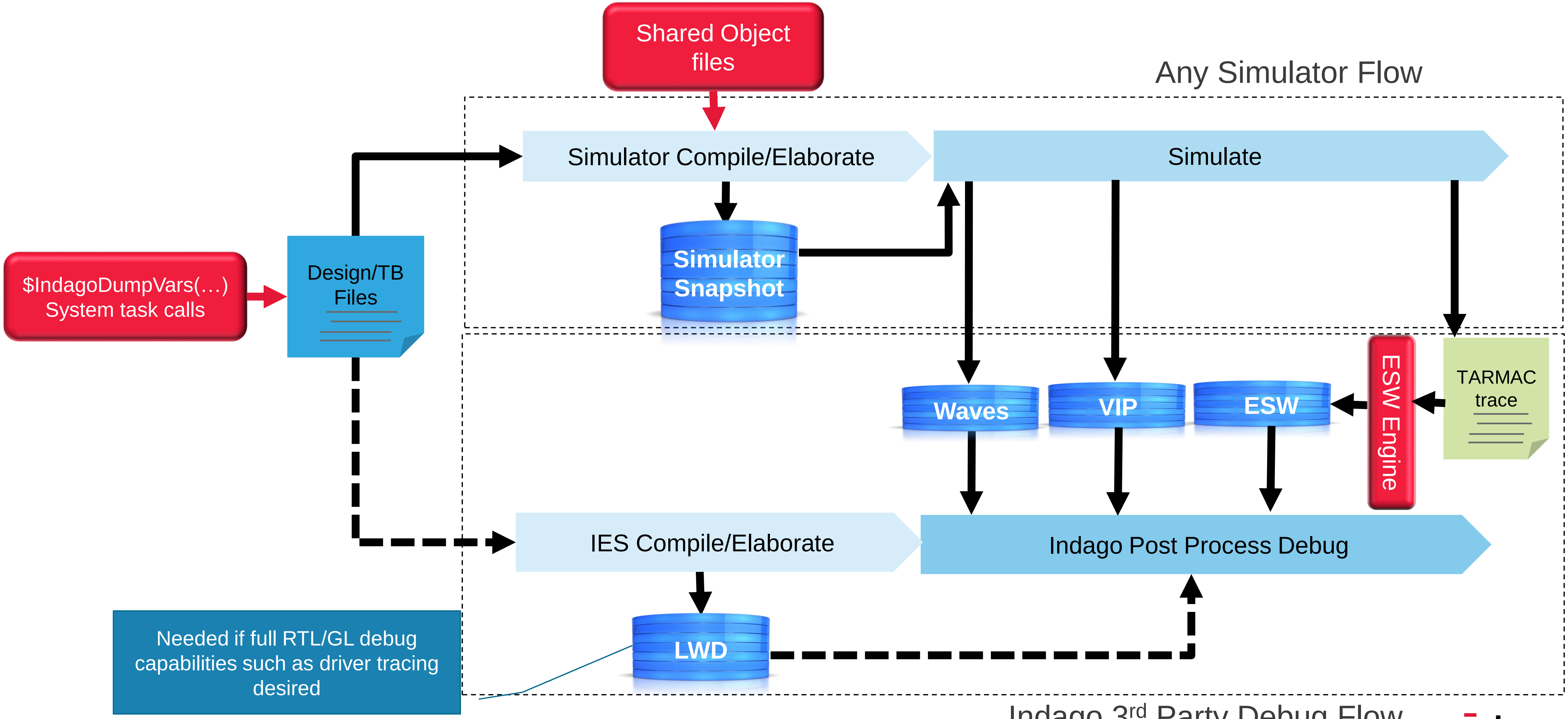
Indago™ 3rd Party Support

Initial feature set for 15.1

- Indago:
  - – Enables synchronized debug of Verilog RTL/GL signals with VIP or Embedded SW on any simulator
- Indago Apps:
  - – Debug Analyzer App
    - – RTL/GL: Verilog Basic types (No MDA's, No Assertions, No transactions, No SV, No VHDL)
  - – Protocol Debug App:
    - – Full support for dumping of Indago™ debug DB's from any simulator
    - – Synchronized debug of supported RTL/GL signals together with VIP
  - – Embedded SW Debug App:
    - – Embedded SW debug DB creation from unmodified ARM TARMAC trace files generated by any simulator
      - – Only for currently supported cores
    - – Synchronized debug of supported RTL/GL signals together with Embedded SW
- More RTL/GL/TB features will come in phases over the next few releases

Indago™ 3rd Party Debug Flow

All Apps Combined



Needed if full RTL/GL debug capabilities such as driver tracing desired

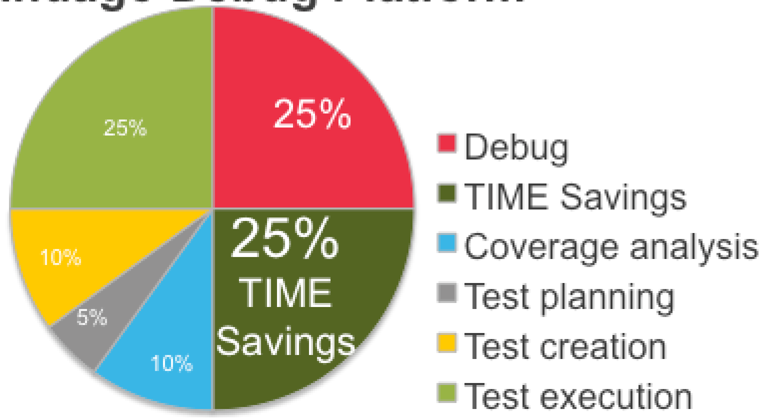
Summary: New Cadence® Indago™ Debug Platform

- ✓ **2X** debug productivity improvement
  - – Indago Debug Platform CUTS Your DEBUG TIME in HALF
  - – Gain more TIME back in your LIFE
  - – Customers* are seeing these benefits today!

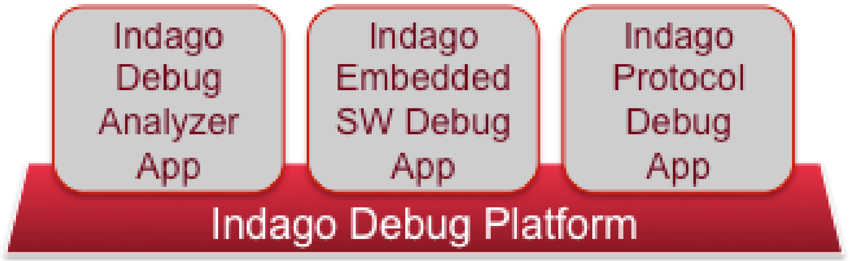
- ✓ **3 Apps** addressing specific debug tasks
  - – RTL/GL and Testbench
  - – Synchronized ESW/HW
  - – Interface protocol functional validation
  - – *More Apps to come*

- ✓ **Available Today!**

Today's Verification Effort using
Indago Debug Platform



Source: Verification engineer survey by Cadence



Indago Debug Analyzer App

Indago Embedded SW Debug App

Indago Protocol Debug App

Indago Debug Platform

*Customers like Renesas, Siemens, and TI presented about their success at CDNLive. ST has a success story published on Cadence.com.

cadence®