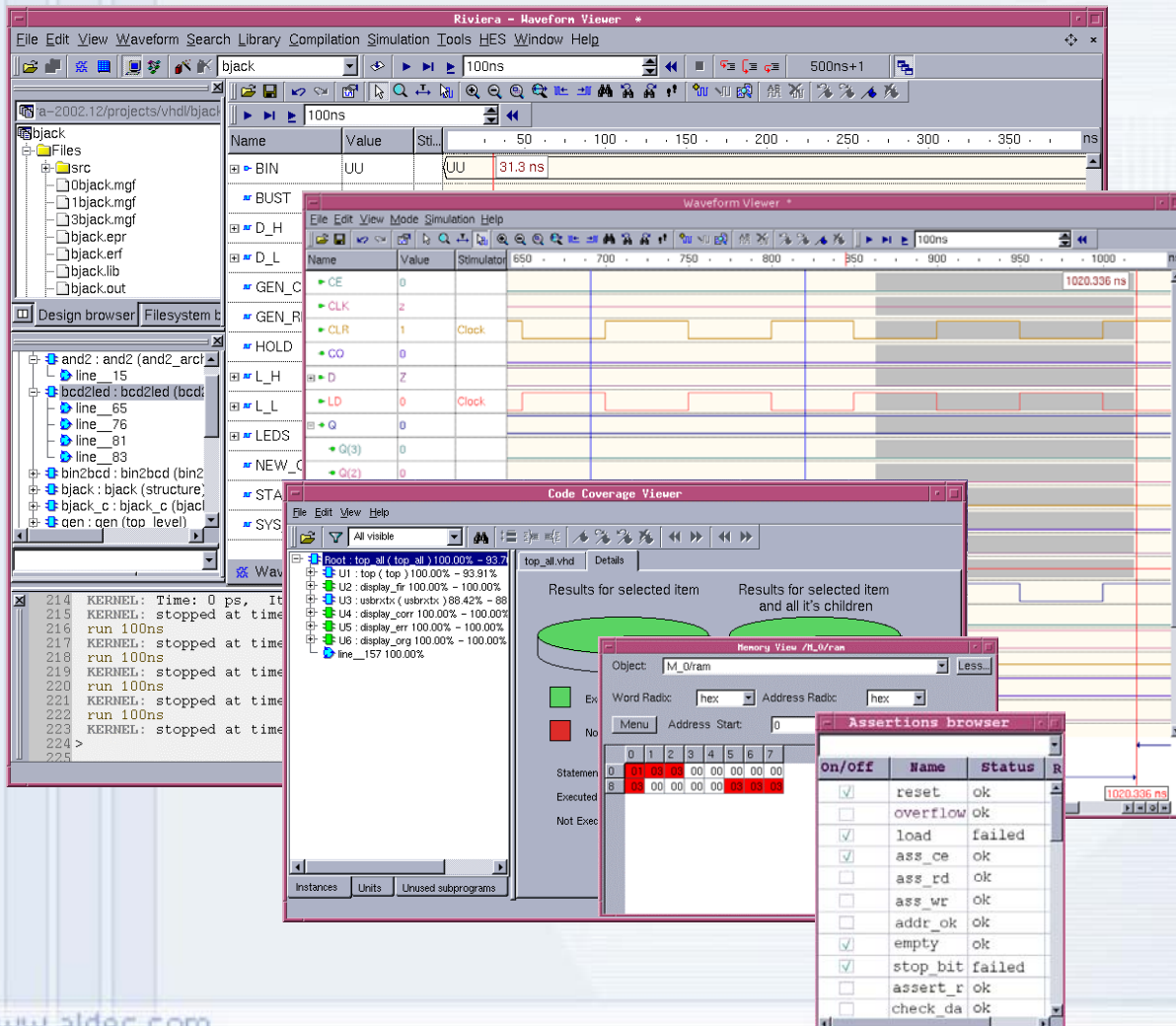




Implementing High-Performance OpenVera Assertions in Mixed-HDL Software Simulation

The *Riviera*[™] software series

Riviera Overview



Features

- IEEE Compliant VHDL and Verilog Mixed Language Simulator
- Design Browser
- HDL Editor
- Waveform Viewer and Editor
- Code Coverage
- Memory Viewer
- Design Profiler
- SystemC Support
- Assertion Support
- Hardware Acceleration Ready

Riviera Overview

HDL Entry

- **VHDL:** IEEE Std. 1076-87 & 93
- **Verilog:** IEEE Std. 1364-1995,
Verilog 2001, System Verilog 3.0/3.1
- **EDIF:** version 2.0.0

Third party tools interfaces

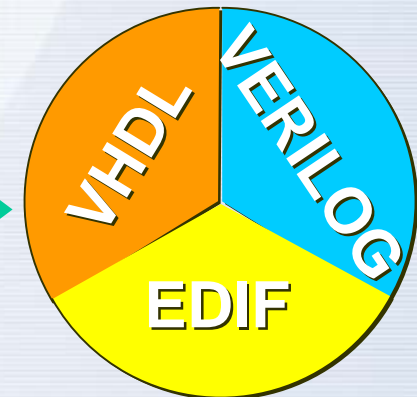
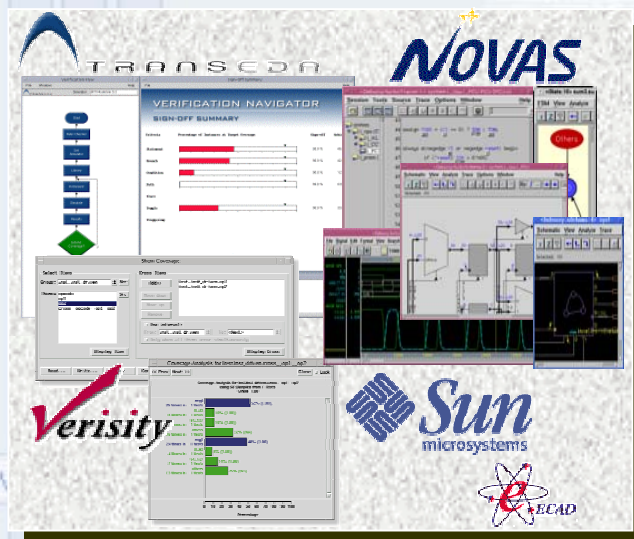
- **Debussy** (Novas Software)
- **Memory Models** (Denali)
- **Specman Elite** (Verisity)
- **Specman Elite** (Verisity)
- **TestBuilder** (Cadence)
- **Visual Elite, Virtual CPU** (Summit)

Simulator Interfaces

- **PLI**
- **VHPI**
- **SWIFT** (Synopsys)
- **SystemC**

Mixed
VHDL/Verilog/EDIF

Common Kernel



Assertions in Riviera

Assertions Entry

OVA (available)

- SystemVerilog (planned)
- PSL (planned)

HDL interfaces

- Verilog
- VHDL

Tools interfaces

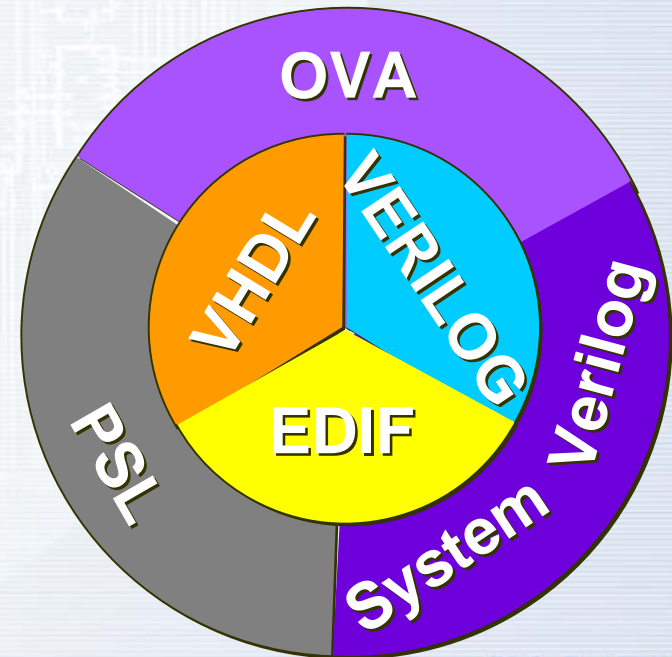
- Code Debug
- Waveform
- Coverage
- Feedback signals

Outputs

- Simulation
- Synthesis

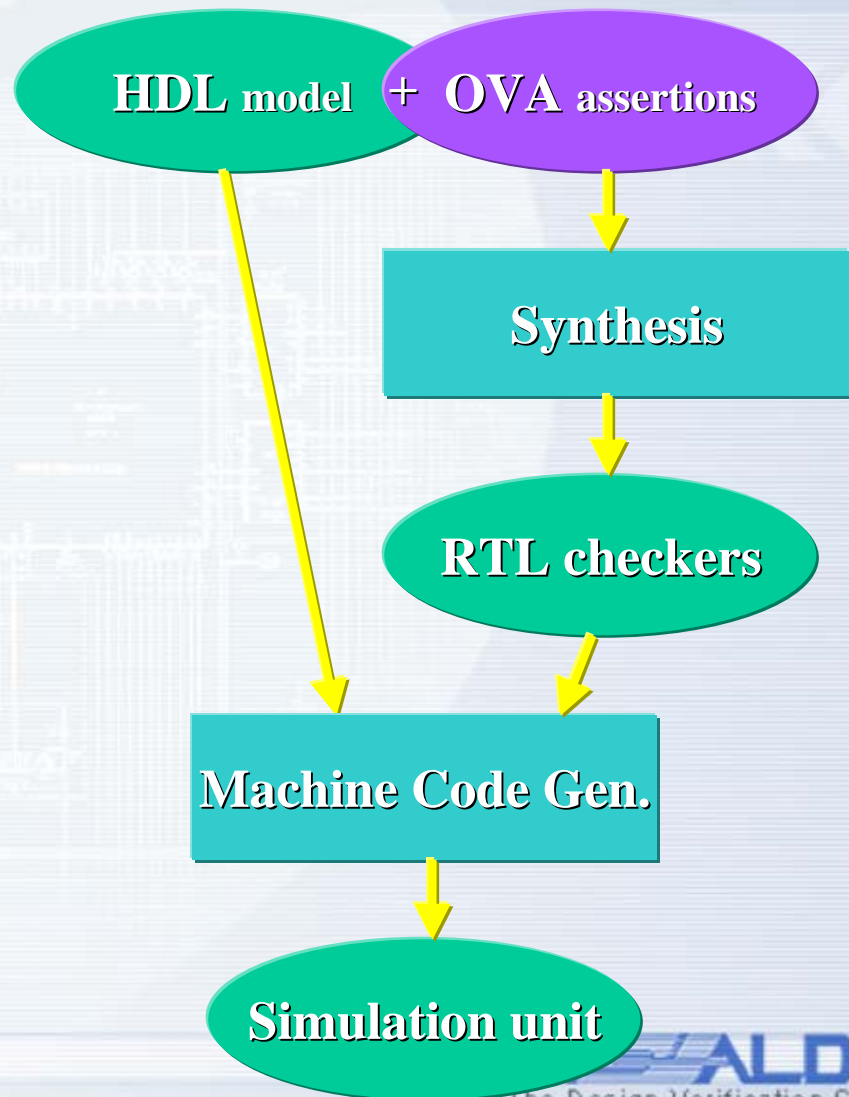
Mixed
OVA/SV/PSL

Generic Assertions Entry

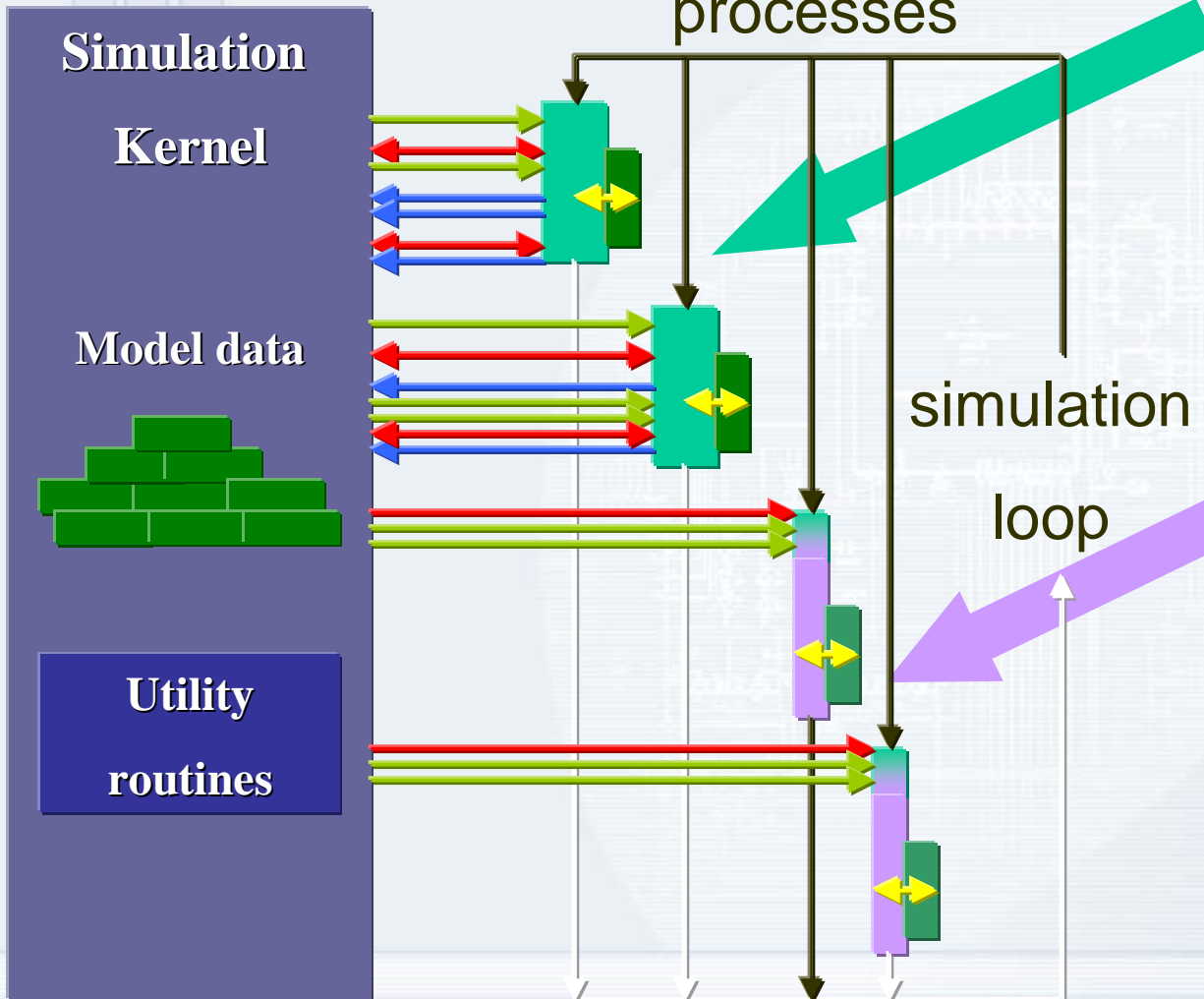


Direct Compilation Technology

- Model and assertions checkers compiled together into a **single simulation unit**
- Fully-optimized machine code generated for integrated unit
- Compiled unit is compatible with generic mixed HDL simulation environment
- Execution benefits from all simulation speedup technologies in **Riviera**

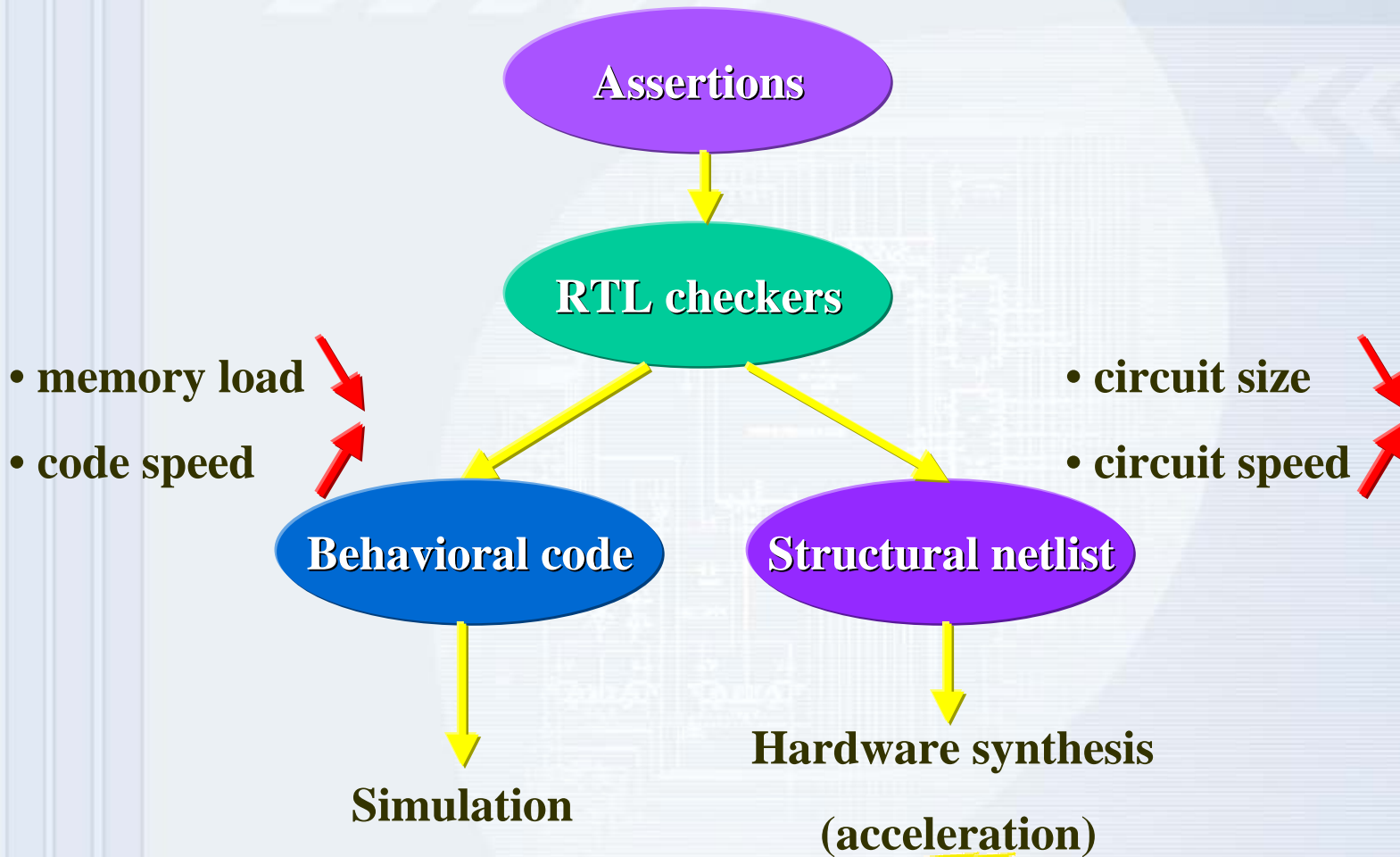


Optimized Assertions Checkers



- HDL processes:
n-value logic,
kernel interrupts,
extensive
communication
- Assertions
processes:
2-state logic for
direct translation
into machine
instructions, no-
interrupted process
code, minimal
kernel
communication &
synchronization

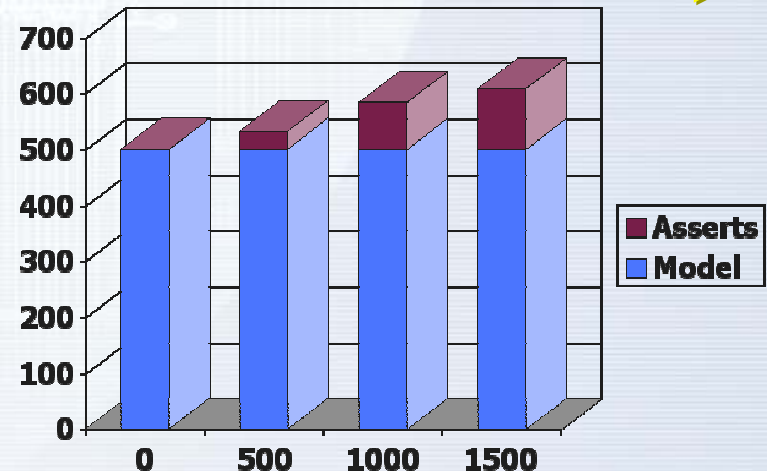
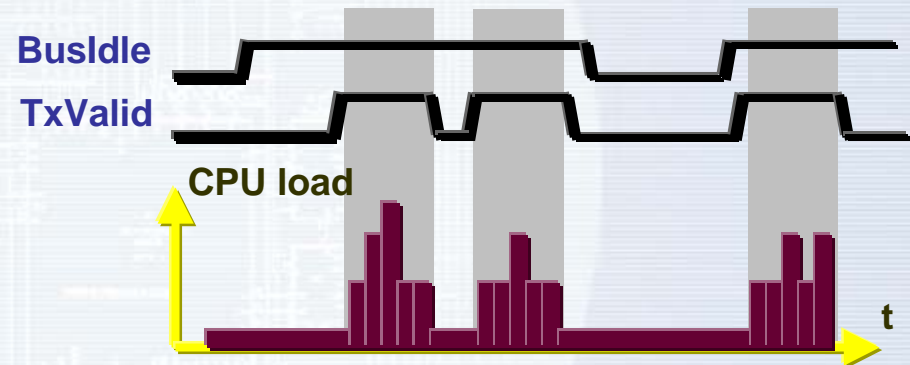
Target Specialized Code Generation



Simulation Performance Control

- Use the **if-then** clause to activate assertions in specified time-windows only
- Assertion process will be suspended when an idle-state is detected
- Embedded assertions in **Riviera** have nominal impact on simulation performance

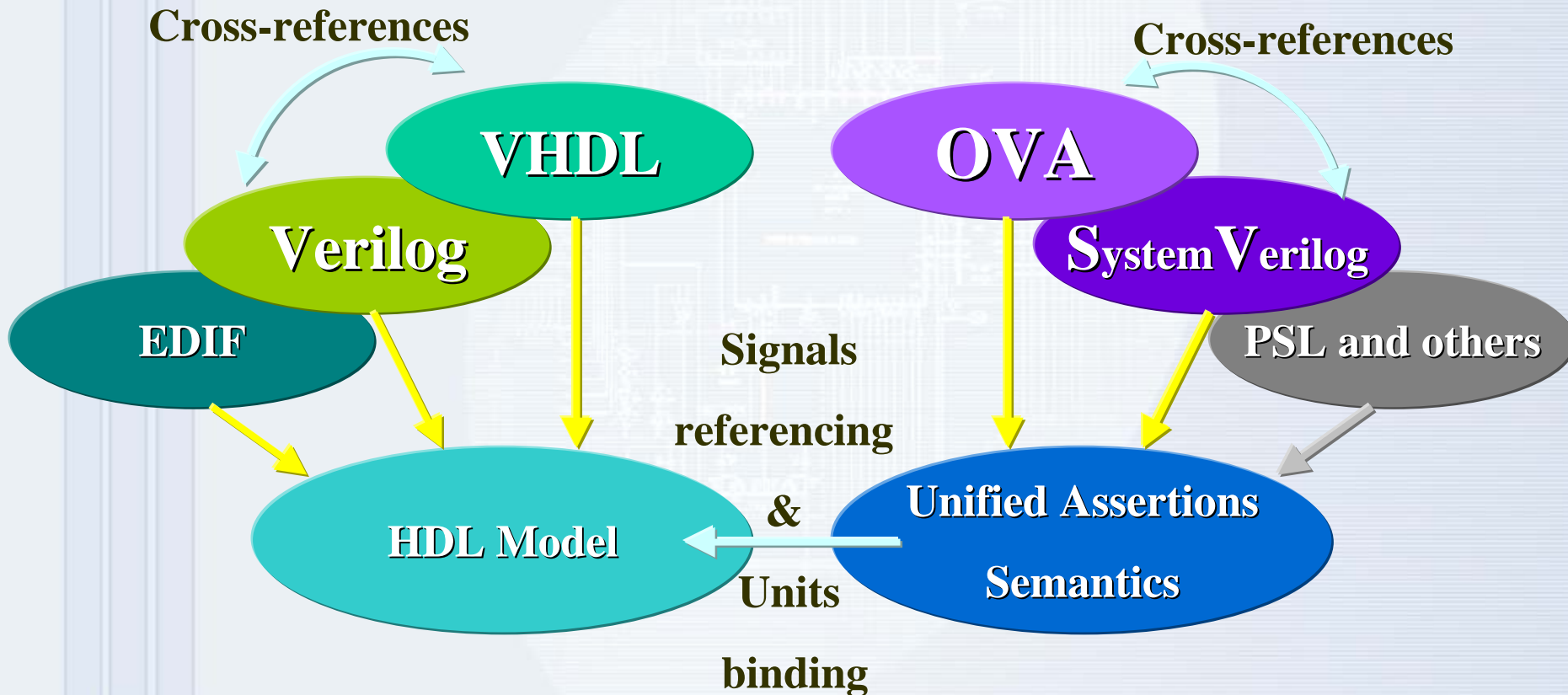
```
assert : check ( if (TxValid & BusIdle) then  
                (XcvrSelect && #[1..2] SYNC) ||  
                (XcvrSelect && #[8..16] SYNC));
```



Universal Source Entry

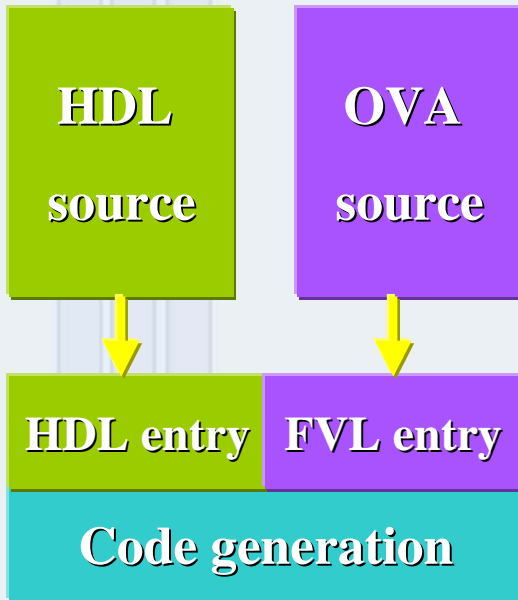
Mixed-HDL entry

Mixed-FVL entry



Universal Source Entry

Separate HDL/FVL sources



The screenshot shows two windows of a Verilog IDE. The top window, titled 'c:\Data\Verilog\tests\dag\test\asserts\8051\AL8051SC.v', displays Verilog code for a module with parameters .ACC_BUS (ACC_BUS), .BLOCK_JMP (BLOCK_JMP) and assignments for SFRDATA, MEMADDR, MEMRD, STATE, and an always block for RST. The bottom window, titled 'c:\Data\Verilog\tests\dag\test\asserts\8051\ASSERT_INTERRUPT.OVA', shows a Verilog scope definition for U1.U2.U2.U_a18051, including a clock posedge CLK, several event declarations for interrupt flags and addresses, and an assert statement for blocked interrupts.

Universal Source Entry

Ideal choice: Assertions ready HDL

System Verilog
source

SV entry

HDL entry FVL entry

Code generation

```
c:\Data\Verilog\tests\dag\test\asserts\8051\usb_monitor_sv.v
File Edit Search Compilation Simulation Options Help
1 module USB_Monitor(CLK,Reset,LineState,RXActive,RXValid,RXValidH,RXData,TXValid,TXValidH,TXReady,TXData);
2 input CLK;
3 input Reset;
4 input [1:0] LineState;
5 input RXActive;
6 input RXValid,RXValidH;
7 input [15:0] RXData;
8 input TXValid,TXValidH;
9 input TXReady;
10 input [15:0] TXData;
11
12
13 sequence @(posedge CLK) SendToken_to_USB = ($rose(RXActive)) =>
14     (
15         [1] : $rose(RXValid) and RXValidH ;
16         $rose(RXValidH) ;
17         $fell(RXValid) and $fell(RXValidH)
18     );
19
20 sequence @(posedge CLK) Start_Data_Transm_to_USB = (
21     $rose(RXActive) ;
22     $rose(RXValid) and $rose(RXValidH)
23 );
24
25 sequence @(posedge CLK) End_Data_Transm_to_USB = (
26     ($fell(RXValid) and $fell(RXValidH)
27     !RXValidH & !RXValid & !RXActive)
28 );
29
30 sequence @(posedge CLK) USB_Data_ACK = ( [1:10] : $rose(TXValid) );
31
32 sequence @(posedge CLK) USB_Data_ACK_End = (
33     (TXValid & TXReady) *[1:10] ;
34     ($fell(TXValid) and $fell(TXReady))
35 );
36
37
38 property Host_RST_HS_Req = Host_RST_HS;
39 property LineState_Reset = LineState_RST_cycle1;
40 property LineStateForbidSeq = LineState_RST_cycle2;
41
```

Debug Features

- Use assertion as conditional breakpoint:

The screenshot displays the Riviera-PRO simulation environment. The main window shows a Verilog source file with several assertions. Two assertions are highlighted with red boxes:

```
--@ova event ReqHS : if (negedge RESET) then #2 RXVa...
--@ova.assert_HostRstHS : check(ReqHS);
...
--@ova event HSSet : if (negedge RESET) then #4 !Xc...
--@ova.assert_USBSetHS : check(HSSet);
```

The waveform viewer shows a signal named `LineState` with a value of 3. A red vertical line indicates a breakpoint at 177.9 ns. The console window shows the following error messages:

```
85 EXECUTION:: NOTE : ResUSB_Pause
86 EXECUTION:: Time: 176 ns, Iteration: 0, Instance: /U3.
87 EXECUTION:: NOTE : ResUSB_Pause
88 EXECUTION:: Time: 208 ns, Iteration: 0, Instance: /U3.
89 KERNEL: Error: Assertion 'ResetHS_FSMstate' FAILED at time: 240000 (8 clk), ./src/usb/...
90 KERNEL: stopped at time: 240 ns
91 VSIM: Stopped at breakpoint in file 'c:\Data\Verilog\tech\ovaforum2003\test\breakpnts...
93 KERNEL: Error: Assertion 'USBSetHS' FAILED at time: 240000 (8 clk), ./src/Auxiliary/top...
94 KERNEL: stopped at time: 240 ns
95 VSIM: Stopped at breakpoint in file 'c:\Data\Verilog\tech\ovaforum2003\test\breakpnts...
97
```

The Structure Browser shows the following variables:

Name	Type	Value
CLK	std_logic	1
CLKEN	std_logic	1
RESET	std_logic	0
RXActive	std_logic	0
RXError	std_logic	0
RXValid	std_logic	1
DMVst4U	std_logic	0

The ALDEC logo and tagline "The Design Verification Company" are visible in the bottom right corner.

Finding bugs with Assertions

- Find potential problems in your design that wouldn't be detected without assertions:

Bus scan test run:

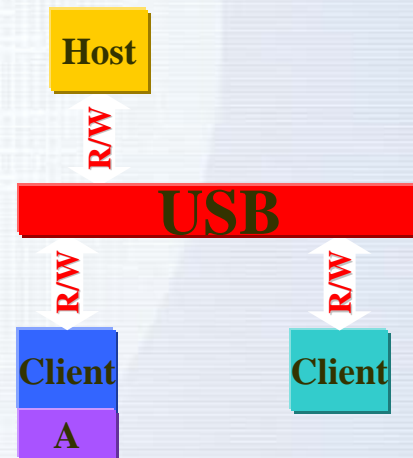


Test passed



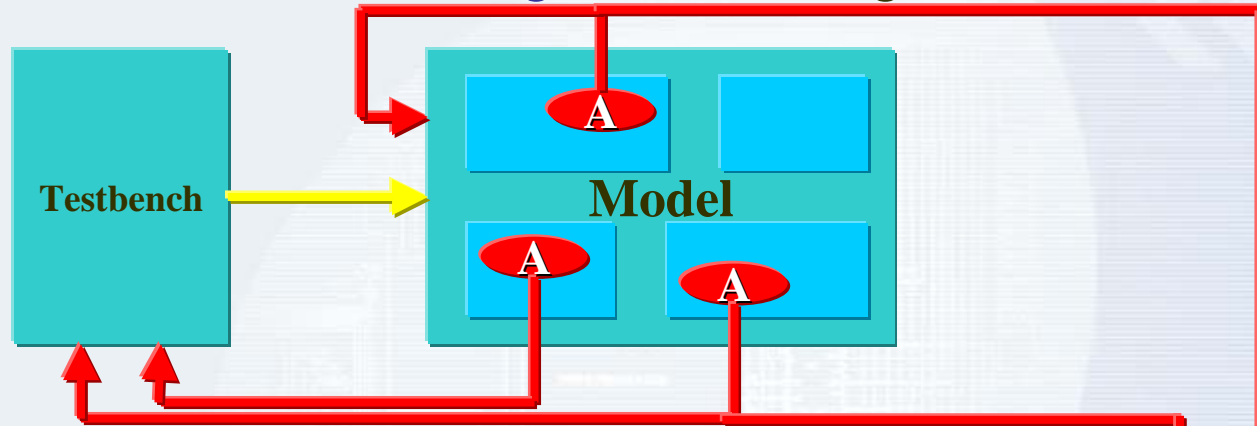
Test failed due to
assertions violations

In real application ...

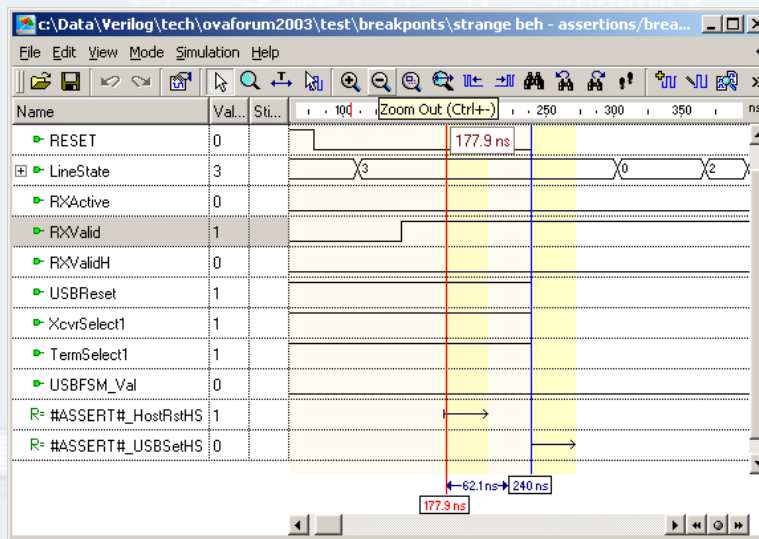


Advanced Features

- Use assertion feedback signal for design interaction:



- Use assertion feedback signal for visualization:

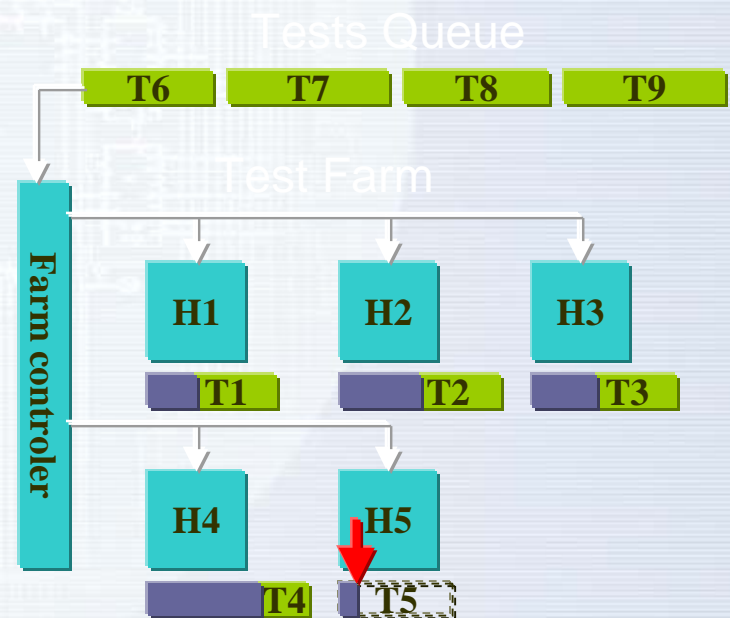


Assertion Feedback Signal

Example application:

- Quit test-session on assertion violation
(Don't waste resources on failed tests runs)

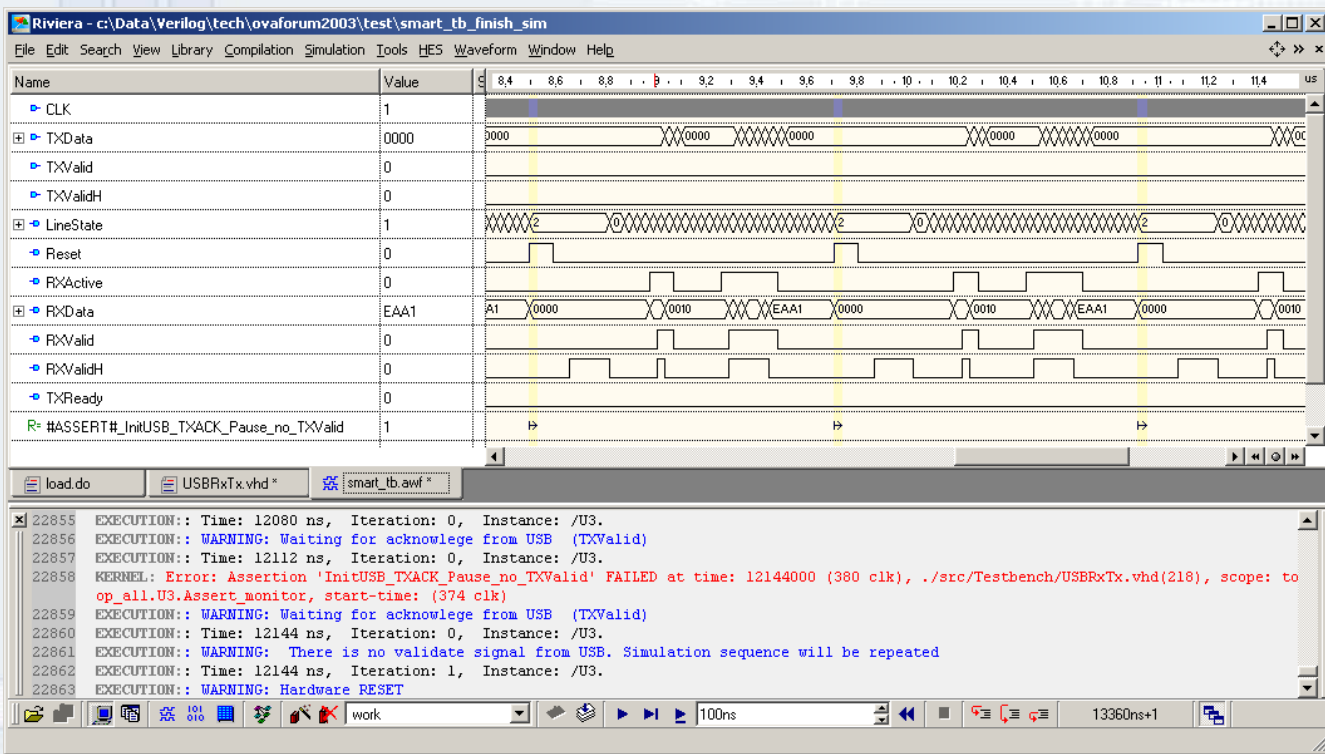
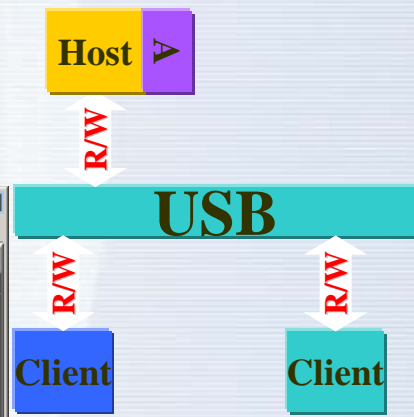
```
TB_Smart_Monitor : process(InitUSB_TXACK_Pause_no_TXValid_fdbk,InitUSB_StatDat
begin
  --@clk posedge CLK
  -- *** Simulation termination when no TXValid in Pause states : ***
  -- InitUSB_TXACK_Pause_val = 36
  --@ova event TXACK_Pause : if (Sreg0_val == 36) then #[1..5] TXValid;
  --@ova assert InitUSB_TXACK_Pause_no_TXValid : check(TXACK_Pause);
  if InitUSB_TXACK_Pause_no_TXValid_fdbk = '1' then
    report "There is no TXValid response signal from USB in InitUSB_TXACK
    severity Failure;
  end if;
  -- InitUSB_StatDat_Pause_val = 43
  --@ova event StatDat_Pause : if (Sreg0_val == 43) then #[1..5] TXValid;
  --@ova assert InitUSB_StatDat_Pause_no_TXValid : check(StatDat_Pause);
  if InitUSB_StatDat_Pause_no_TXValid_fdbk = '1' then
    report "There is no TXValid response signal from USB in InitUSB_StatD
    severity Failure;
  end if;
```



Assertion Feedback Signal

Example USB application:

- Retry initialization step if there is no acknowledge from client side:
(Give up after 10 failures)



end if;

Advanced Applications

USB protocol monitor unit

Riviera - c:\Data\Verilog\tech\ovaforum2003\test\usb_monitor

```
1 module USB_Monitor(CLK, Reset, LineState, RXActive, R
2 input CLK;
3 input Reset;
4 input [1:0] LineState;
5 input RXActive;
6 input RXValid, RXValidH;
7 input [15:0] RXData;
8 input TXValid, TXValidH;
9 input TXReady;
10 input [15:0] TXData;
11 wire a;
12
13 //@clk posedge CLK
14
15 // Reset and initialization
16
17 // LineState = SE1 pulse
18 //@ova event LineState_SE1 : #1 (LineState==3)*[7
19
20 // Reset cause LineState is in SE1
21 //@ova event RST_SE1 : if (negedge Reset) then Li
22
23 // Assertion : UTMI should set LineState control
24 //@ova assert Check_RST_SE1 : check(RST_SE1);
25
26 // HS mode request : RXValidH=1 for 5 cycles
27 //@ova event HS_Req : #2 RXValidH * [6] #1 negedge
28
29 // USB Reset in HS mode after hardware reset
30 //@ova event RST_HS_Req : if (negedge Reset) then HS_Req
31
```

c:\Data\Verilog\tech\ovaforum2003\test\usb_monitor\usb_monitor.awf

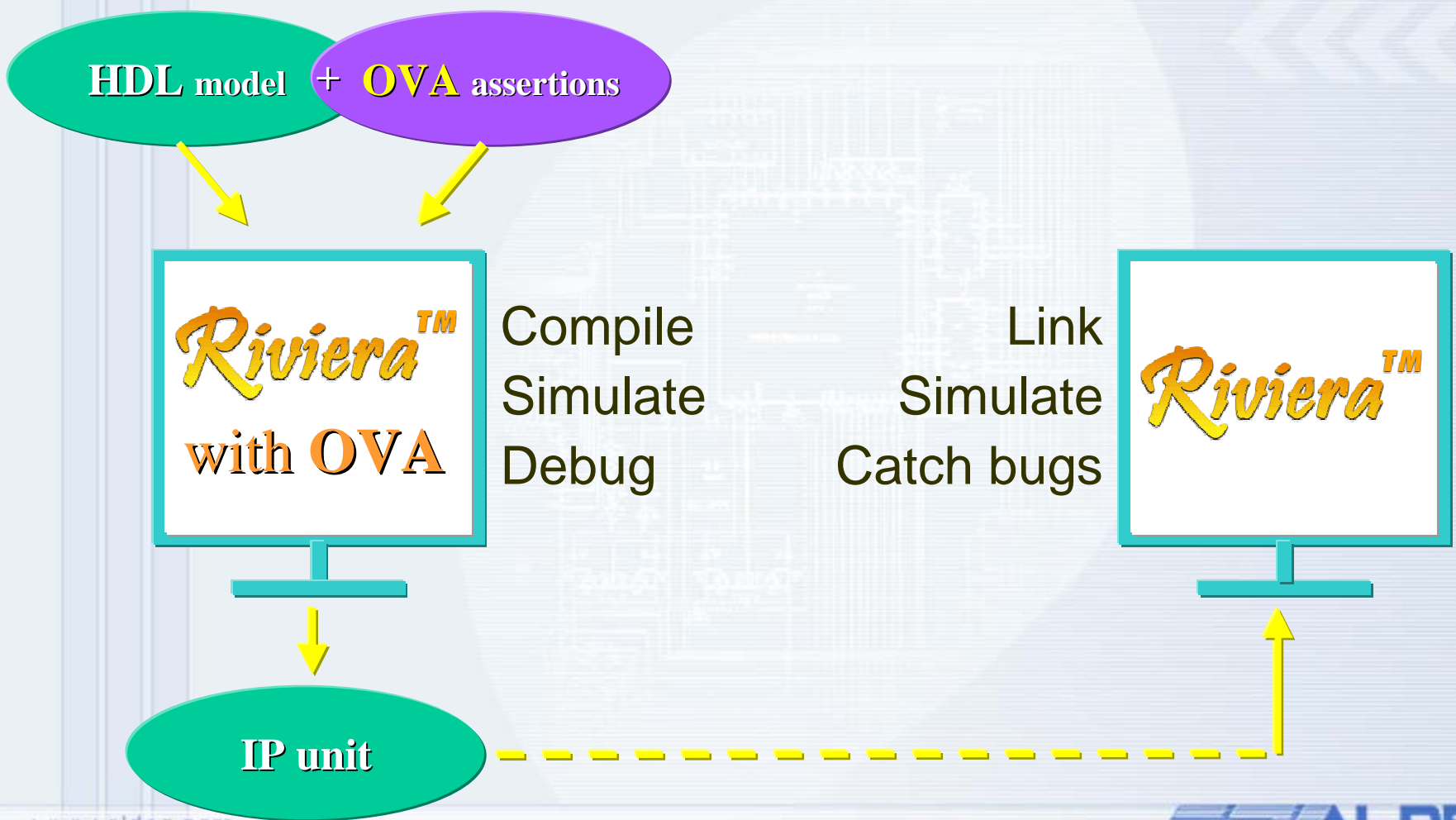
Name	Value	Signal
CLK	1	Periodic square wave
Reset	0	Low
LineState	2	Signal with multiple pulses
TXData	0000	Hex data: 00D2, 0000, 0000, 0000, 0000, 0000
TXValid	0	Pulses corresponding to TXData
TXValidH	0	Pulses corresponding to TXValid
TXReady	0	Pulses
RXData	863E	Hex data: 0010, 00D2, 05D0
RXActive	0	Pulses
RXValid	0	Pulses
RXValidH	0	Pulses
R# #ASSERT#_Host_RS...	0	Assertion failure markers
R# #ASSERT#_Check_Tr...	0	Assertion failure markers
R# #ASSERT#_Check_R...	0	Assertion failure markers
R# #ASSERT#_Check_R...	0	Assertion failure markers

usb_display.v | **usb_monitor.v** | **load.do** | **compile.do** | **auxiliary.do**

```
4210 KERNEL: 29904000Incorrect Transmit Timing Pattern
4211 KERNEL: Error: Assertion 'Check_Transmit_Seq_Data' FAILED at time: 29936000 (936 clk), c:/Data/Verilog/tech/ovaforum2003/test/usb_monitor/src/USB_Monitor/usb_monitor.v(61), scope: top_all.U1.Assert_Monitor, start-time: (933 clk)
4212 KERNEL: 29936000Incorrect Transmit Timing Pattern
4213 KERNEL: 29968000Incorrect Transmit Timing Pattern
4214 EXECUTION:: NOTE : End of transmission
4215 EXECUTION:: Time: 30 us, Iteration: 0, Instance: /U3.
4216 KERNEL: stopped at time: 30 us
4217 >
4218
```

Ln 61, Col 1

OVA enabled IP Libraries distribution



OVA enabled IP Libraries distribution

