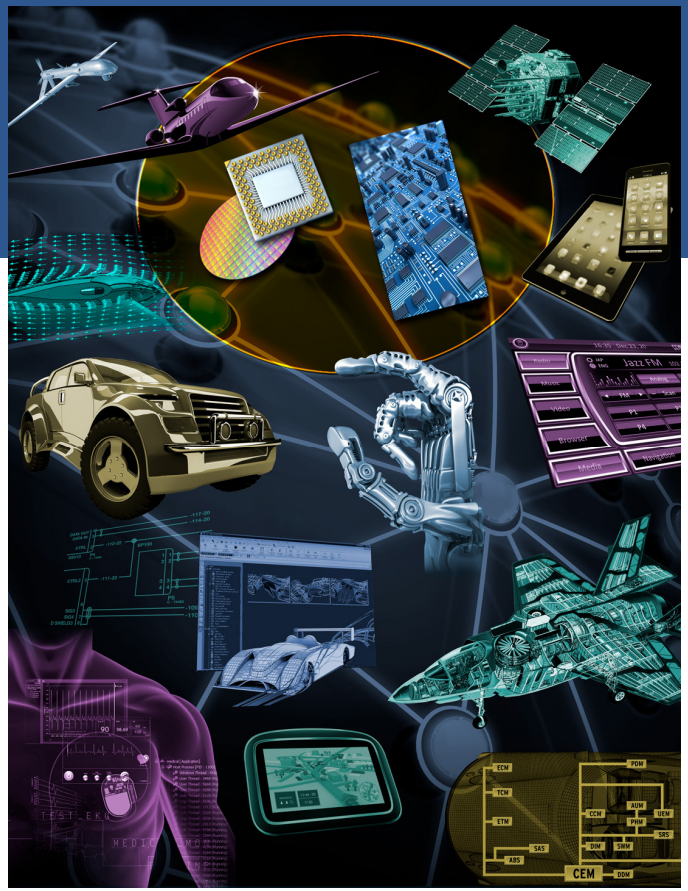


"I Didn't Know Questa Could Do That!" & More

Rick White

Applications Engineer Consultant

ICVS

October 2019

Mentor®
A Siemens Business

Agenda – I Didn’t Know Questa Could Do That!

- Help/Resources
- GUI
- Optimizations and Performance
- Debug Tools
  - Wave Window
  - Source Window
  - FSM and Memory Windows
- Tips & Tricks
- Coverage

© Mentor Graphics Corp.

Agenda – I Didn’t Know Questa Could Do That!

- Help/Resources
- GUI
- Optimizations and Performance
- Debug Tools
  - Wave Window
  - Source Window
  - FSM and Memory Windows
- Tips & Tricks
- Coverage

© Mentor Graphics Corp.

PDF Docs

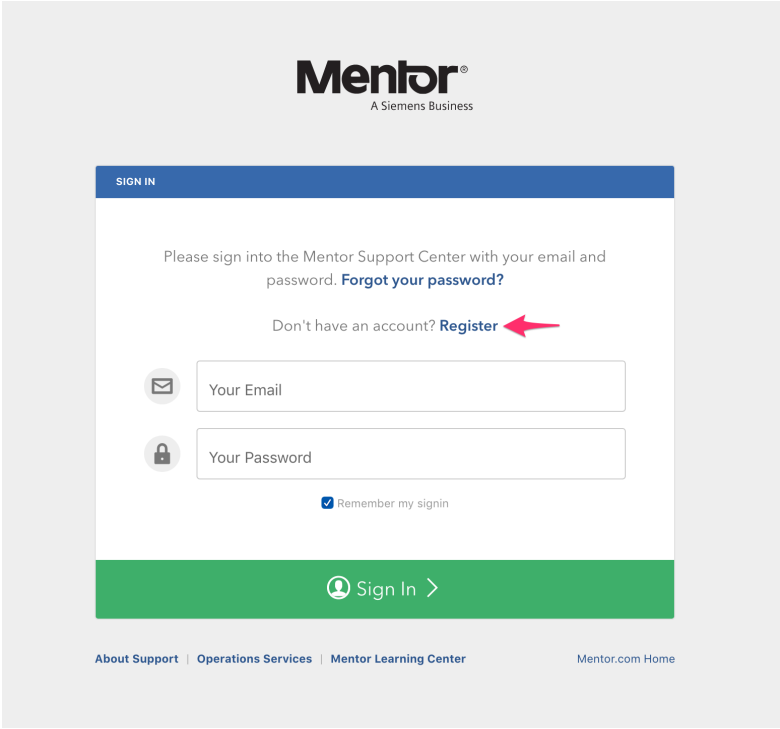
- $QUESTA_HOME/questasim/docs/pdfdocs

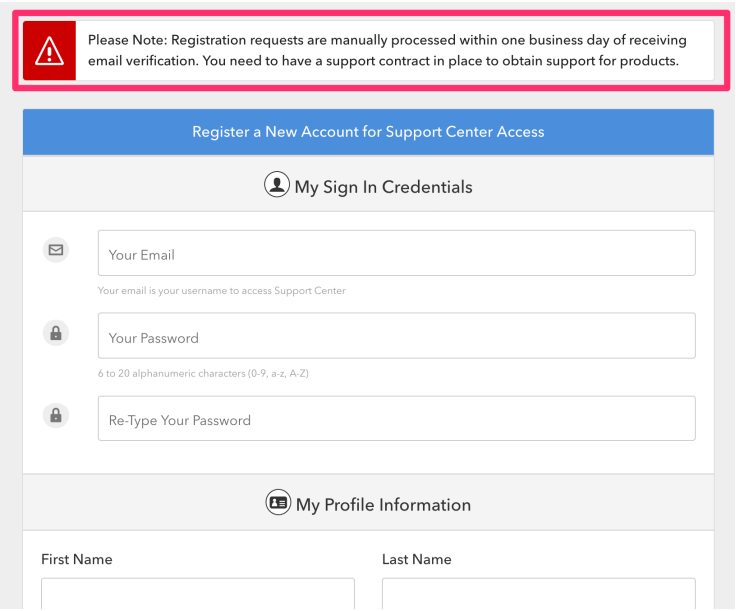
```
[vgill@localhost pdfdocs]$ ls
_bk_questa_sim.pdf     mgc_licen.pdf    q_qk_guide.pdf          questa_sim_ref.pdf    third_party_ver.pdf
flexnet_lic_admin.pdf  mgc_lic_rn.pdf   questa_sim_gui_ref.pdf  questa_sim_tut.pdf    ucdbapi_ref.pdf
fli.pdf                ovm_ref.pdf      questa_sim_install.pdf  questa_sim_user.pdf   vovl_user.pdf
mgc_html_help.pdf      pa_user.pdf      questa_sim_multicore.pdf questa_sim_vm.pdf    vrm_user.pdf
[vgill@localhost pdfdocs]$
```

© Mentor Graphics Corp.

SupportNet

- Register Now! @ support.mentor.com





© Mentor Graphics Corp.

SR – Service Request

Mentor

Products

Service Requests

Open a New Service Request

Describe your issue | Service Request Details

What can we help you with?

Questa

Questa CDC/Formal 10.7c

Please describe your issue

Compile error on encrypted code.

We can be more effective if you provide us important details, such as a specific error code.

Find Solution

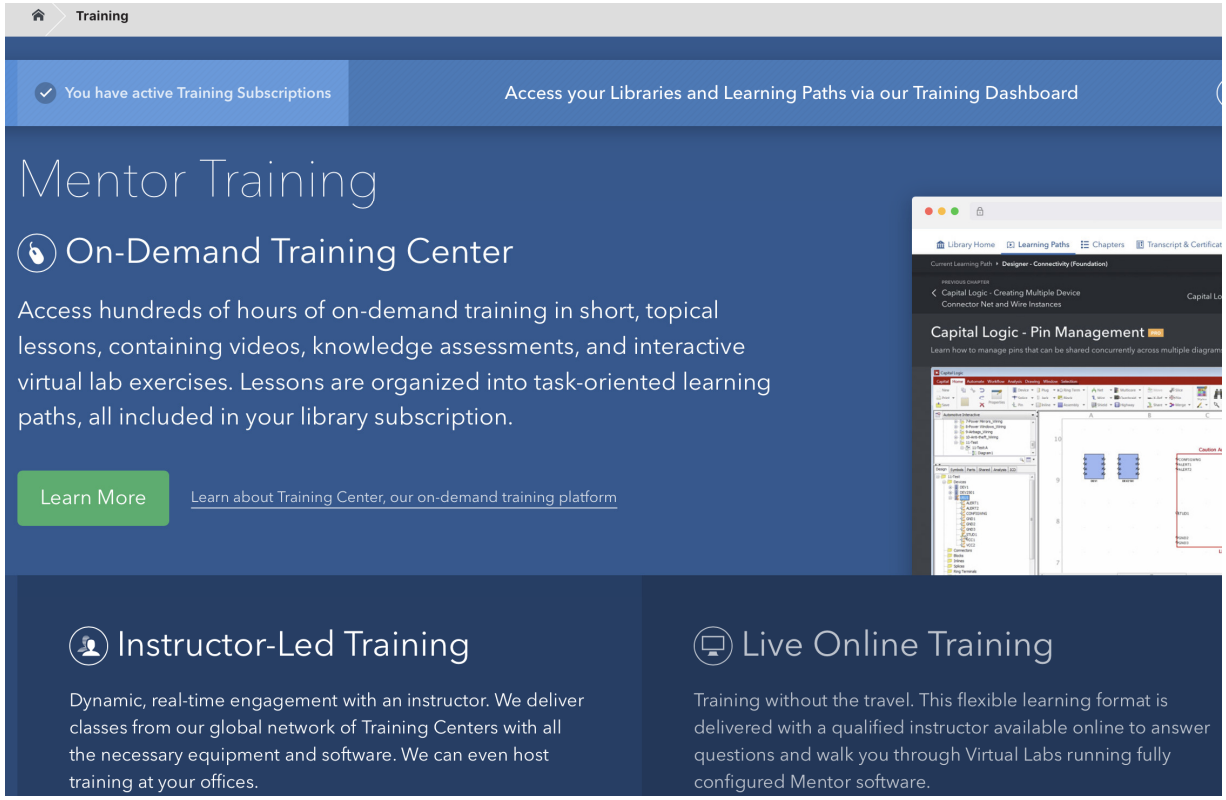
Open a New Service Request

© Mentor Graphics Corp.

Mentor A Siemens Business

On Demand Training

- www.mentor.com/training



© Mentor Graphics Corp.

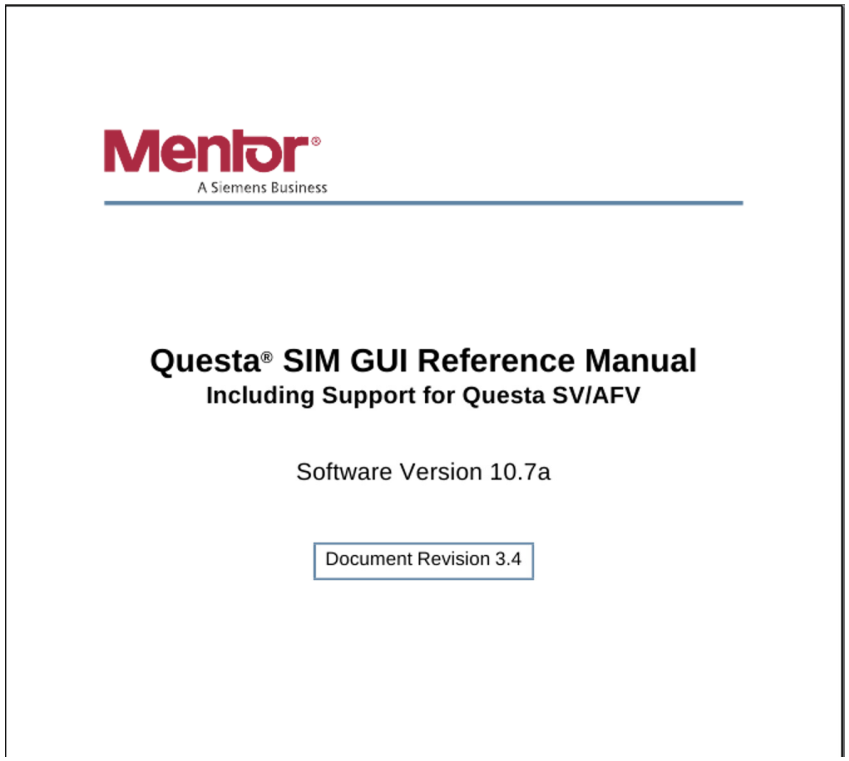
Agenda – I Didn’t Know Questa Could Do That!

- Help/Resources
- GUI
- Optimizations and Performance
- Debug Tools
  - Wave Window
  - Source Window
  - FSM and Memory Windows
- Tips & Tricks
- Coverage

© Mentor Graphics Corp.

Questa SIM GUI Reference Manual

- questa_sim_gui_ref.pdf



Mentor®
A Siemens Business

Questa® SIM GUI Reference Manual
Including Support for Questa SV/AFV

Software Version 10.7a

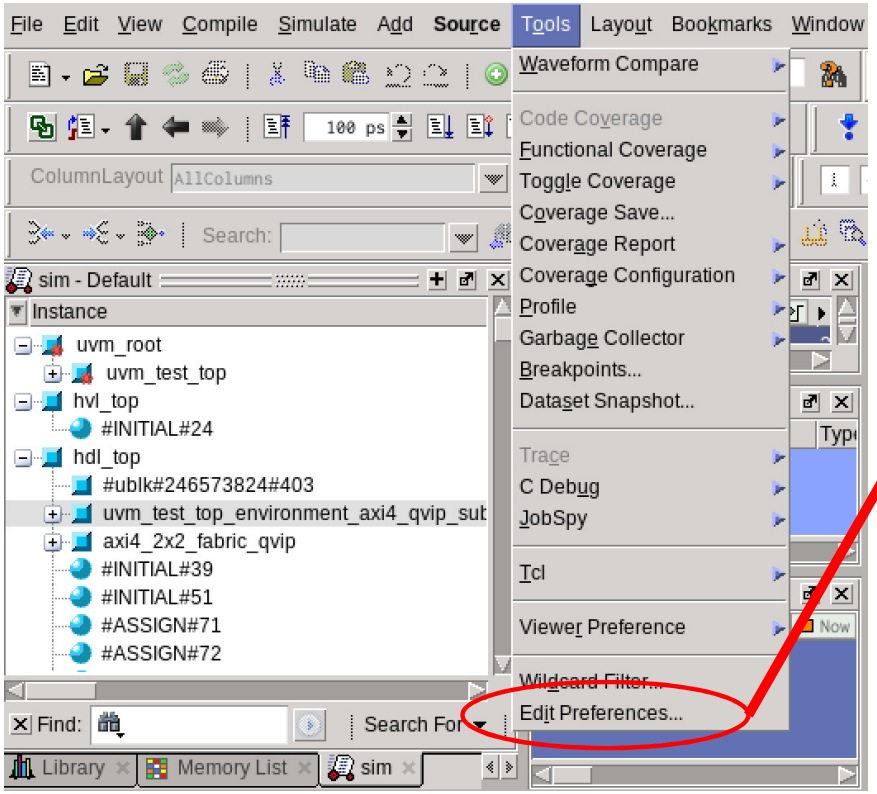
Document Revision 3.4

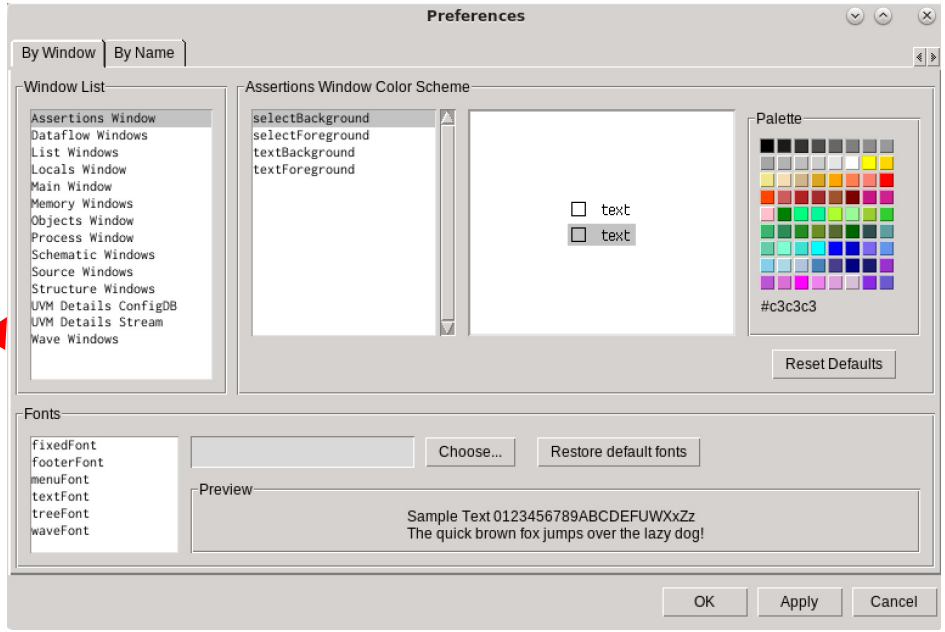
InfoHub	
Bookcase	1
Revision History	3
Table of Contents	5
List of Figures	11
List of Tables	17
+ Chapter 1 Overview	21
+ Chapter 2 Menus	47
+ Chapter 3 Toolbars	59
+ Chapter 4 Window Reference	109
+ Chapter 5 Keyboard Shortcuts and Mouse Actions	319
+ Chapter 6 GUI Customization	331
+ Chapter 7 GUI Preferences	345
Index	353
End-User License Agreement	357
Documentation Feedback	

© Mentor Graphics Corp.

Edit GUI Preferences

- Preferences are applied immediately and saved upon exit. Either in the '.modelsim' file in Linux home directory or registry of Windows machine.

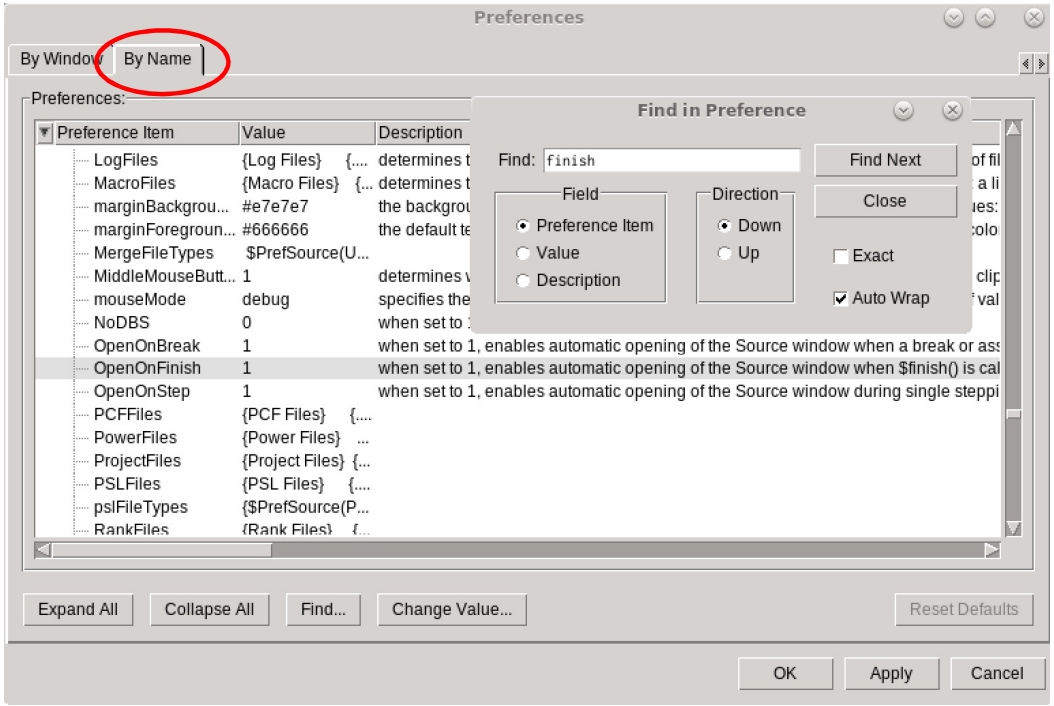




© Mentor Graphics Corp.

Edit GUI Preferences → By Name → Find

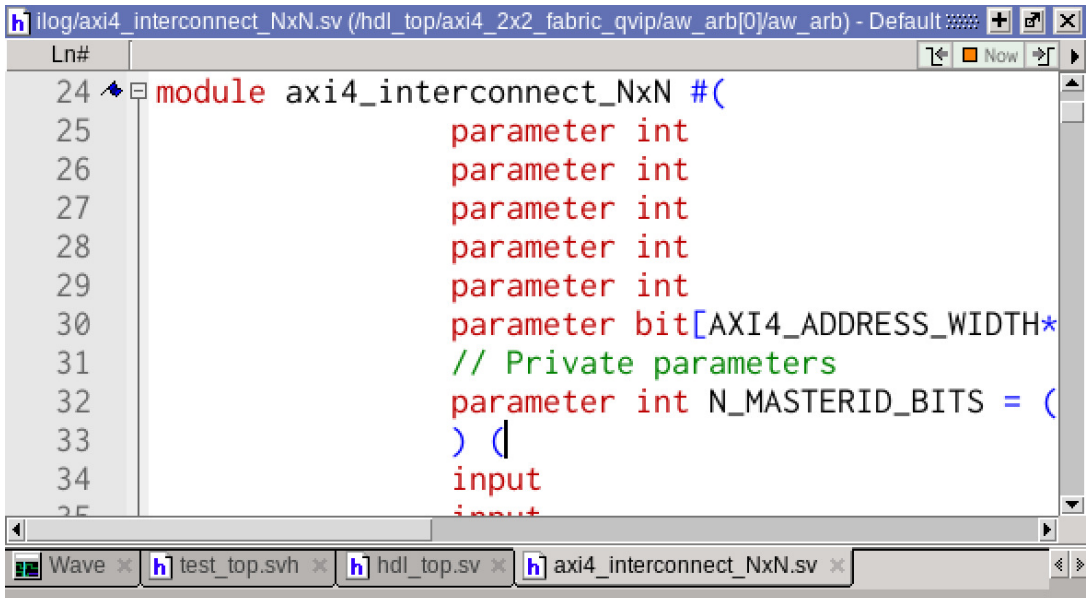
- OpenOnFinish
  - Controls behavior when 'finish' is called



© Mentor Graphics Corp.

Change text size in Source Window

- Smaller
  - Hold Ctrl and tap –
- Larger
  - Hold Ctrl and tap +



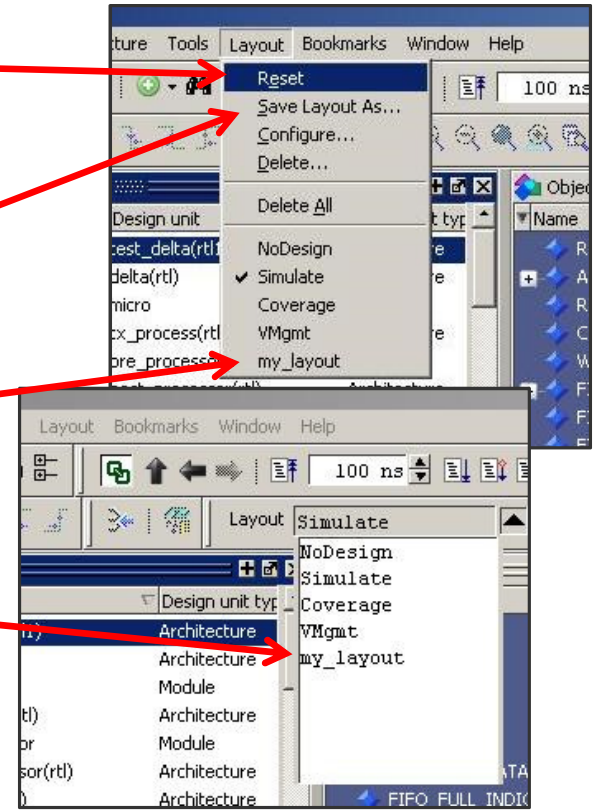
© Mentor Graphics Corp.

Working with Window Layouts

- The *Layout* Menu Offers the Ability to:
  - *Reset* the tool to the default

    Reset can be very helpful when your windows get "messy"

  - *Save* any layouts that you like
- Predefined Layouts
- User Saved Layouts

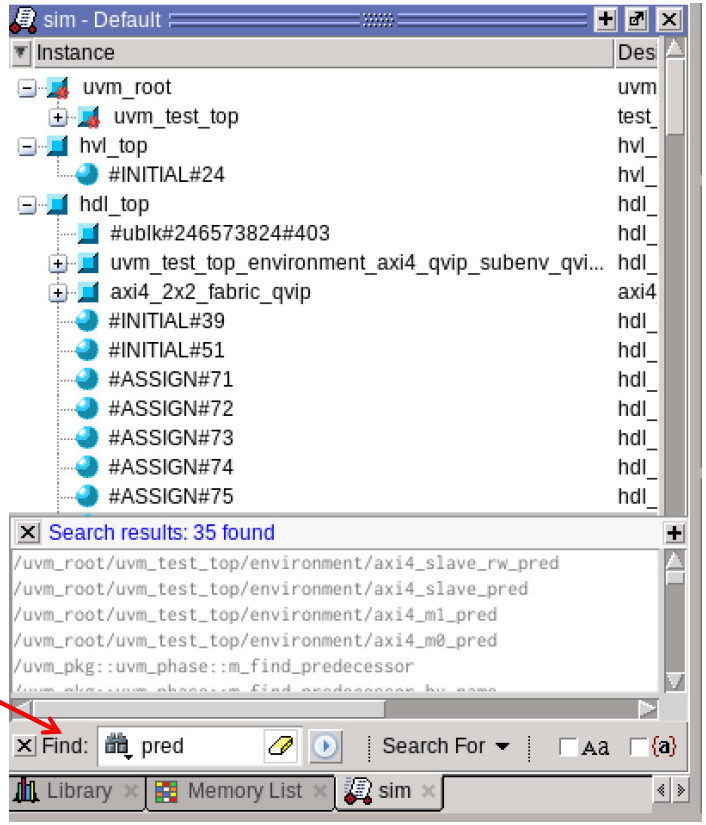


© Mentor Graphics Corp.

Search - “find” and “contains”

- Ctrl+F
- Edit -> Find
- Toolbar (Standard)
- Windows that support both Find and Filter modes only
  - transcript, objects, library, memory, sim, files, wave, schematic, etc

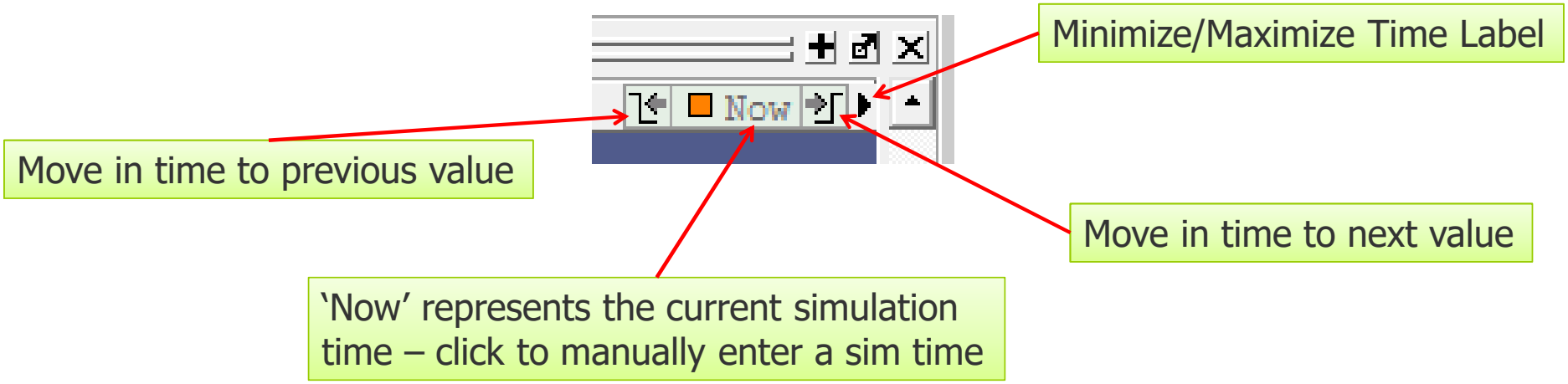
Click on Find to toggle between Find & Contains



© Mentor Graphics Corp.

Current Time Label

- Allows for Displaying Values at a Selected Time
- Works for Several Windows; windows are dynamically linked
  - Source, Schematic, Dataflow, FSM, Watch, Objects
  - Select a signal to move in time



© Mentor Graphics Corp.

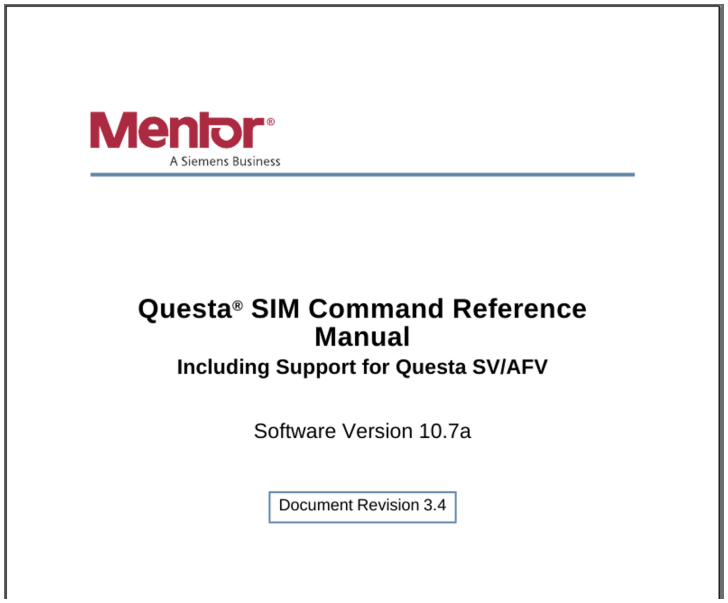
Agenda – I Didn’t Know Questa Could Do That!

- Help/Resources
- GUI
- Optimizations and Performance
- Debug Tools
  - Wave Window
  - Source Window
  - FSM and Memory Windows
- Tips & Tricks
- Coverage

© Mentor Graphics Corp.

Questa SIM Command Reference Manual

- questa_sim_ref.pdf



InfoHub	
Bookcase	1
Revision History	3
Table of Contents	5
List of Figures	15
List of Tables	17
+ Chapter 1 Syntax and Conventions	19
+ Chapter 2 Commands (A - M)	55
− Chapter 3 Commands (N - Z)	441
next	463
noforce	464
nolog	465
notepad	467
noview	468
nowhen	469
onbreak	470
onElabError	472
onerror	473
onfinish	475
pa autotestplan	476
pa msg	477
pa report	483
pause	484
pop	485
power add	486
power off	488
power on	491
power report	494
power reset	497
precision	499
printenv	500
process report	501
profile clear	502
profile configure	503
profile interval	505
profile off	506
profile on	508
profile open	510
profile option	511
profile reload	513

© Mentor Graphics Corp.

Optimization

- 2-step flow for easing migration to performance mode
  - `vopt` enabled by default
- 3-step method
  - Full control of `vopt` process

© Mentor Graphics Corp.

2-Step Flow

1. Compile
2. Simulate (full optimization done implicitly before simulation)

```
vlog verilog.v
vcom vhdl.vhd
sccom tb.cpp

vsim tb -voptargs="+acc+my_module"
```

- Use `-voptargs` to pass arguments and control optimization
- To preserve visibility (recommended method)
  - Use `+acc` switch as argument: Ex: `-voptargs=+acc`

© Mentor Graphics Corp.

3-Step Flow (Recommended)

1. Compile
2. Optimize
3. Simulate

```
vlog verilog.v
vcom vhdl.vhd
sccom tb.cpp

vopt top +acc=nprl+my_module -o top_opt

vsim top_opt
```

- Can create many different version of optimized design
  - Full/Partial debug visibility
  - Fully optimized for regressions
  - Somewhere in between
- Recommended flow for fine grained control of optimization

© Mentor Graphics Corp.

Preserving Visibility in vopt

- To preserve design objects for debugging purposes
  - Use ***+acc[=<spec>][+<module>[.]|+<entity>[.]]***

```
a: SVA/PSL objects, enables vsim -assertdebug
b: net bits
c: cells
l: line debugging
m: module, interface, & program instances
n: nets
p: ports
r: Verilog variables & parameters
s: enable override of built-in system tasks
t: tasks & function scopes
u: primitive instances
v: VHDL variables
```

  - Example
    - ***+acc=npr+top - net, port, reg on top***
    - ***+acc=npr+top. - net, port, reg from top recursively***
    - ***+acc=npr3+top - net, port, reg from top 3 levels***
    - ***+acc - optimize with full visibility***

© Mentor Graphics Corp.

Optimizations For Newer Versions of Questa

- With later versions of Questa, the use of the +acc switches for vopt are discouraged due to simulation speed improvements.

Table 3-12. Recommended vopt Arguments for Access Visibility

Previously Supported Argument[1]	Recommended Argument
+acc=a	-assertdebug
+acc=b	-bitscalar
+acc=c	-cellaccess
+acc=f	-fsmdebug
+floatparameters	-floatparameters[2]
+floatgenerics	-floatgenerics
+acc=l	-linedebug
+nosparse	-nosparse
+acc=s	-systfoverride
+acc=u	-primitiveaccess

© Mentor Graphics Corp.

Log Command – WLF Dumping

- The `log` command allows you to log waves from a TCL file or at the VSIM prompt; does not require changing and recompiling your source code
  - `log -depth <level> -recursive <scope|var> …`
- To log the entire design
  `log -r /*` (Note: this can affect performance)
- Use the `nolog` command to turn off logging

```
VSIM 1> log -depth 2 /top/dut/*
VSIM 2> run 1000ns
VSIM 3> nolog -all
VSIM 4> run 1000ns
VSIM 5> log -ports -r /top/dut/inst1/*
```

© Mentor Graphics Corp.

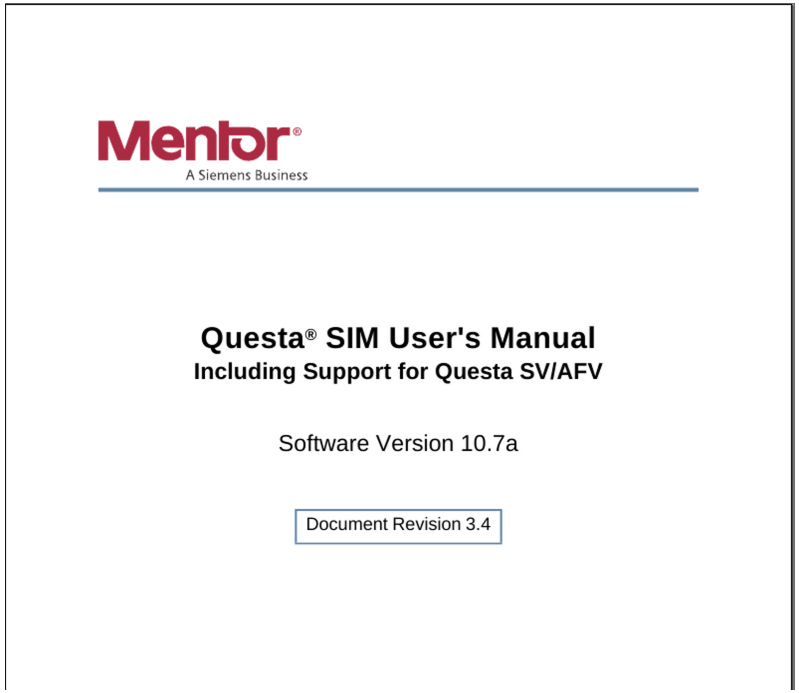
Agenda – I Didn’t Know Questa Could Do That!

- Help/Resources
- GUI
- Optimizations and Performance
- Debug Tools
  - Wave Window
  - Source Window
  - FSM and Memory Windows
- Tips & Tricks
- Coverage

© Mentor Graphics Corp.

Questa SIM User's Manual

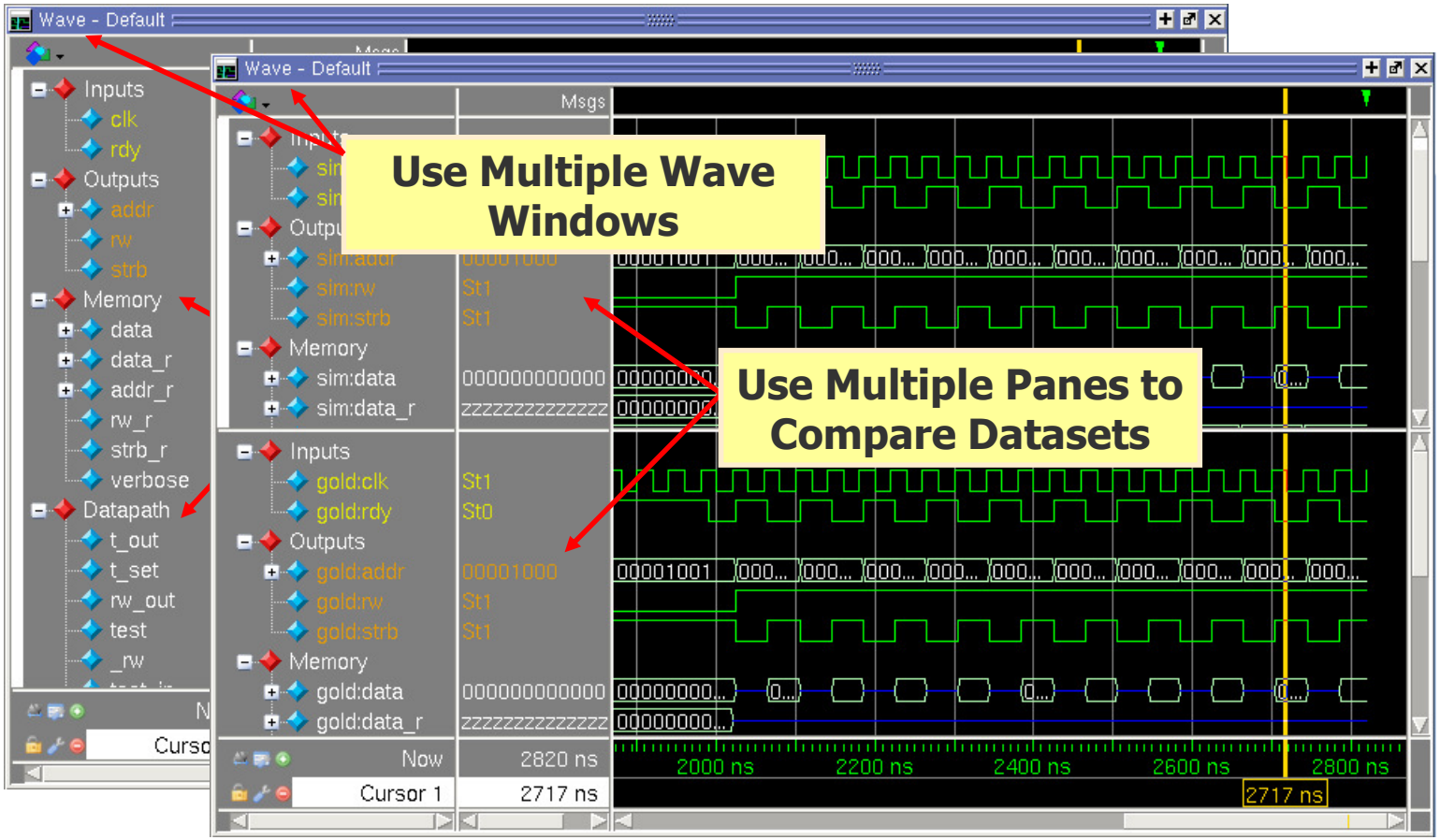
- questa_sim_user.pdf



InfoHub	
Bookcase	1
Revision History	3
Table of Contents	5
List of Figures	45
List of Tables	55
+ Chapter 1 Introduction	61
+ Chapter 2 Protecting Your Source Code	85
+ Chapter 3 Optimizing Designs with vopt	127
+ Chapter 4 Projects	157
+ Chapter 5 Design Libraries	179
+ Chapter 6 VHDL Simulation	197
+ Chapter 7 Verilog and SystemVerilog Simulation	247
+ Chapter 8 SystemC Simulation	415
+ Chapter 9 Mixed-Language Simulation	491
+ Chapter 10 Advanced Simulation Techniques	591
+ Chapter 11 Recording and Viewing Transactions	617
+ Chapter 12 Verifying Designs with Questa Verification IP Library Com...	673
+ Chapter 13 Recording Simulation Results With Datasets	697
+ Chapter 14 Waveform Analysis	715
+ Chapter 15 Schematic Window	811
+ Chapter 16 Debugging with the Dataflow Window	855
+ Chapter 17 Source Window	885
+ Chapter 18 Using Causality Traceback	913
+ Chapter 19 Code Coverage	941
+ Chapter 20 Finite State Machines	1045
+ Chapter 21 Verification with Assertions and Cover Directives	1065
+ Chapter 22 Verification with Functional Coverage	1147
+ Chapter 23 Verification with Constrained Random Stimulus	1187
+ Chapter 24 Coverage and Verification Management in the UCDB	1201
+ Chapter 25 C Debug	1275
+ Chapter 26 Profiling Performance and Memory Use	1293
+ Chapter 27 Signal Spy	1323
+ Chapter 28 Monitoring Simulations with JobSpy	1347
+ Chapter 29 Generating Stimulus with Waveform Editor	1361
+ Chapter 30 Standard Delay Format (SDF) Timing Annotation	1375
+ Chapter 31 Value Change Dump (VCD) Files	1401
+ Chapter 32 Tcl and DO Files	1421
+ Appendix A modelsim.ini Variables	1453
+ Appendix B Location Mapping	1811
+ Appendix C Error and Warning Messages	1815
+ Appendix D Questa Verification IP Errors	1831
+ Appendix E Verilog Interfaces to C	1867
+ Appendix F System Initialization	1917
+ Appendix G Third-Party Model Support	1931

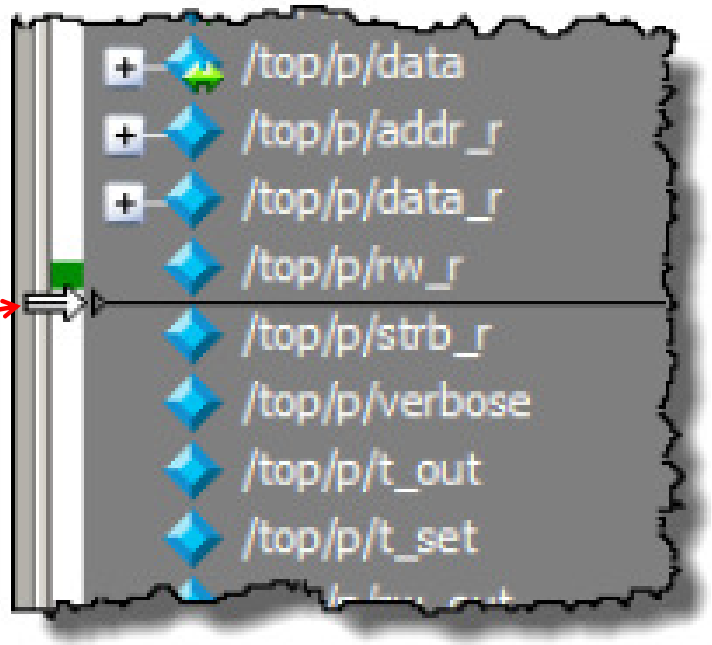
© Mentor Graphics Corp.

Wave Window - Managing Signals



© Mentor Graphics Corp.

Setting the Insertion Point for adding signals



Click in the white column to set the insertion point

© Mentor Graphics Corp.

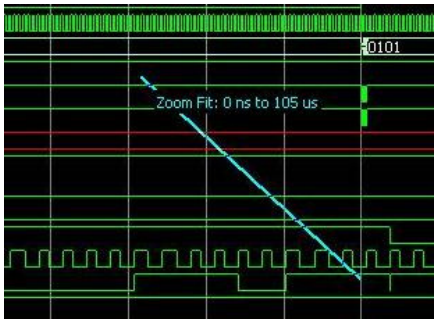
Adding Objects to the Wave Window

- Drag and drop objects (signals, nets, registers, transactions) into the Wave window
  - Structure, Objects, Source, Locals, Processes, Memory, List
- Objects or Locals
  - Place the cursor over an individual object or selected objects
    - MMB
- RMB→ Add to wave
- Highlight Selection → `Ctrl+W`

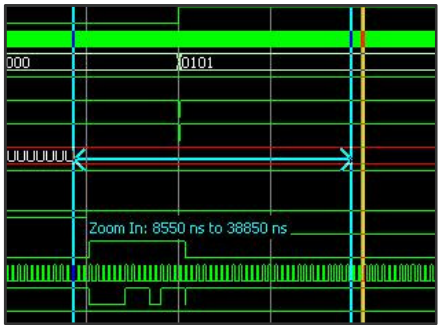
© Mentor Graphics Corp.

MMB Gestures

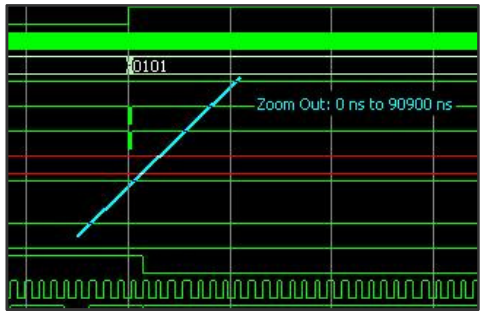
- Mouse gestures are a quick way to execute zooming functions in a window
- Just hold down MMB and drag in any direction



Zoom Full:
drag up & left



Zoom Area:
drag down & left or
drag down & right

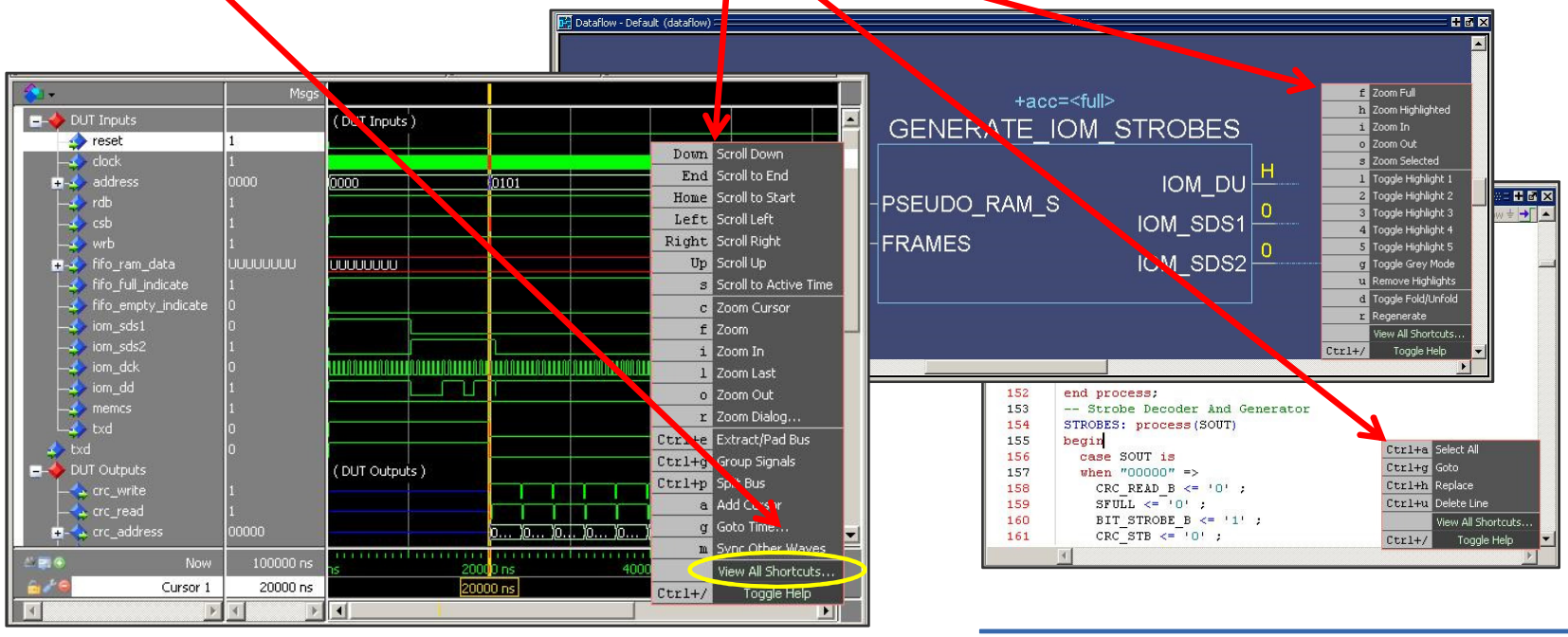


Zoom Out:
drag up & right

© Mentor Graphics Corp.

Keyboard Shortcuts

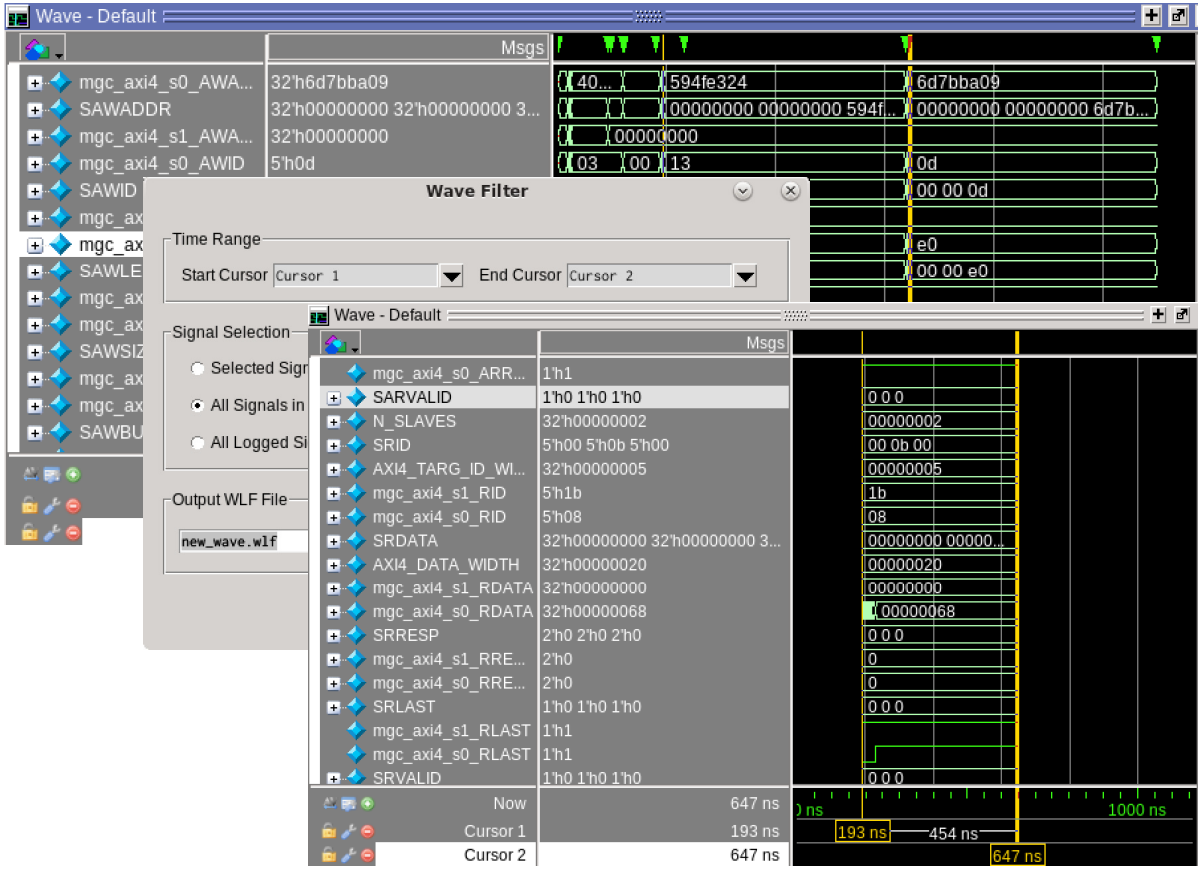
- Each window has a variety of keyboard shortcuts
  - Toggle menu ON/OFF using "`Ctrl+/`"
  - 'View All Shortcuts' to modify



© Mentor Graphics Corp.

Filter Waveform

- Wave > Filter Waveform
- Quickly generate smaller subset of wave database
- Eliminate uninteresting data such as during initialization/reset



© Mentor Graphics Corp.

Dynamically Read wlf File

- Opening a live wlf file with the **-view** option results in a dynamically scrolling, read-only view of the wlf file
  - New data is available for viewing as the simulation progresses

© Mentor Graphics Corp.

Toolbar – Wave Add Contributing

Wave - Default

	Msgs
/hdl_top/axi4_2x2_fabric_qvip/R_AWID	5'h15 5'h03
/hdl_top/axi4_2x2_fabric_qvip/R_AWLEN	8'h22 8'h2f
/hdl_top/axi4_2x2_fabric_qvip/R_AWPROT	3'h3 3'h7
/hdl_top/axi4_2x2_fabric_qvip/R_AWQOS	4'h0 4'hd
/hdl_top/axi4_2x2_fabric_qvip/R_AWREGION	4'hc 4'h8
/hdl_top/axi4_2x2_fabric_qvip/R_AWSIZE	3'h0 3'h0
Contributors: R_AWSIZE	
/hdl_top/axi4_2x2_fabric_qvip/AWADDR	-No Data-
/hdl_top/axi4_2x2_fabric_qvip/AWBURST	-No Data-
/hdl_top/axi4_2x2_fabric_qvip/AWCACHE	-No Data-
/hdl_top/axi4_2x2_fabric_qvip/AWID	-No Data-
/hdl_top/axi4_2x2_fabric_qvip/AWLEN	-No Data-
/hdl_top/axi4_2x2_fabric_qvip/AWLOCK	-No Data-
/hdl_top/axi4_2x2_fabric_qvip/AWPROT	-No Data-
/hdl_top/axi4_2x2_fabric_qvip/AWQOS	-No Data-
/hdl_top/axi4_2x2_fabric_qvip/AWREADY	-No Data-
Now	1000 ns
Cursor 1	90.234 ns

Transcript | test_top.svh | hdl_top.sv | axi4_interconnect_NxN.sv | Wave | Objects | Processes | Locals | Library | Memory List | sim

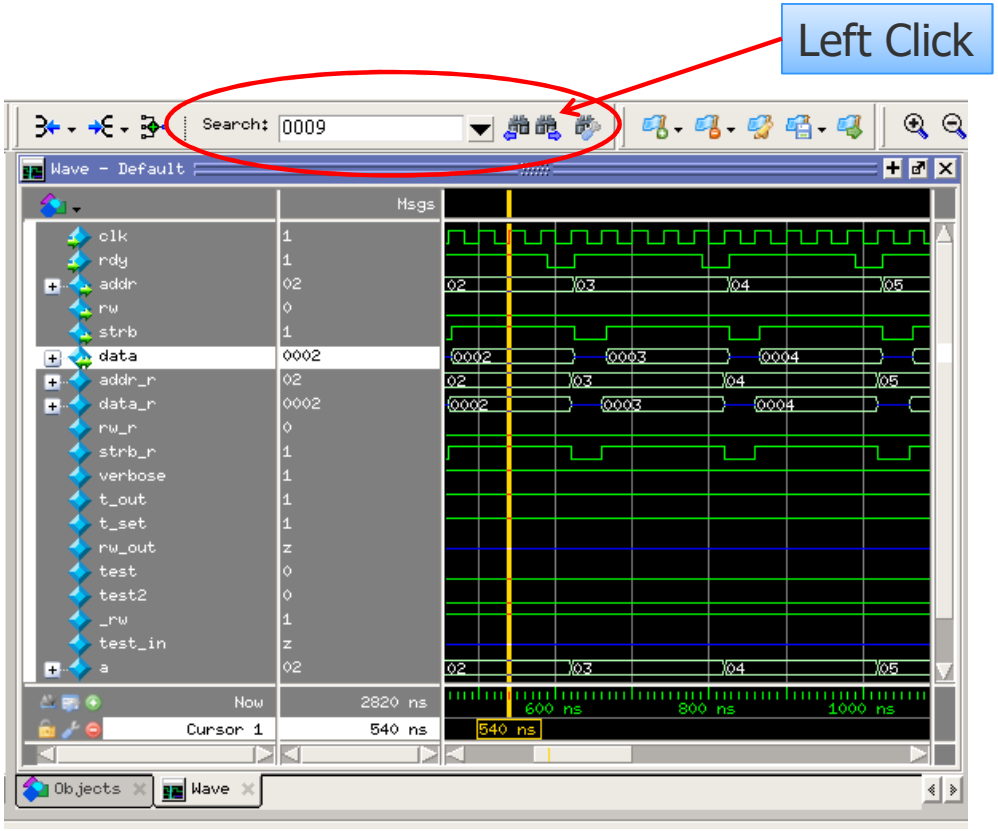
© Mentor Graphics Corp.

Toolbar – Wave Search

Allows for 'Quick' Search

1. Add a cursor
2. Select Signal
3. Specify Value
4. Cursor moves to that location if found

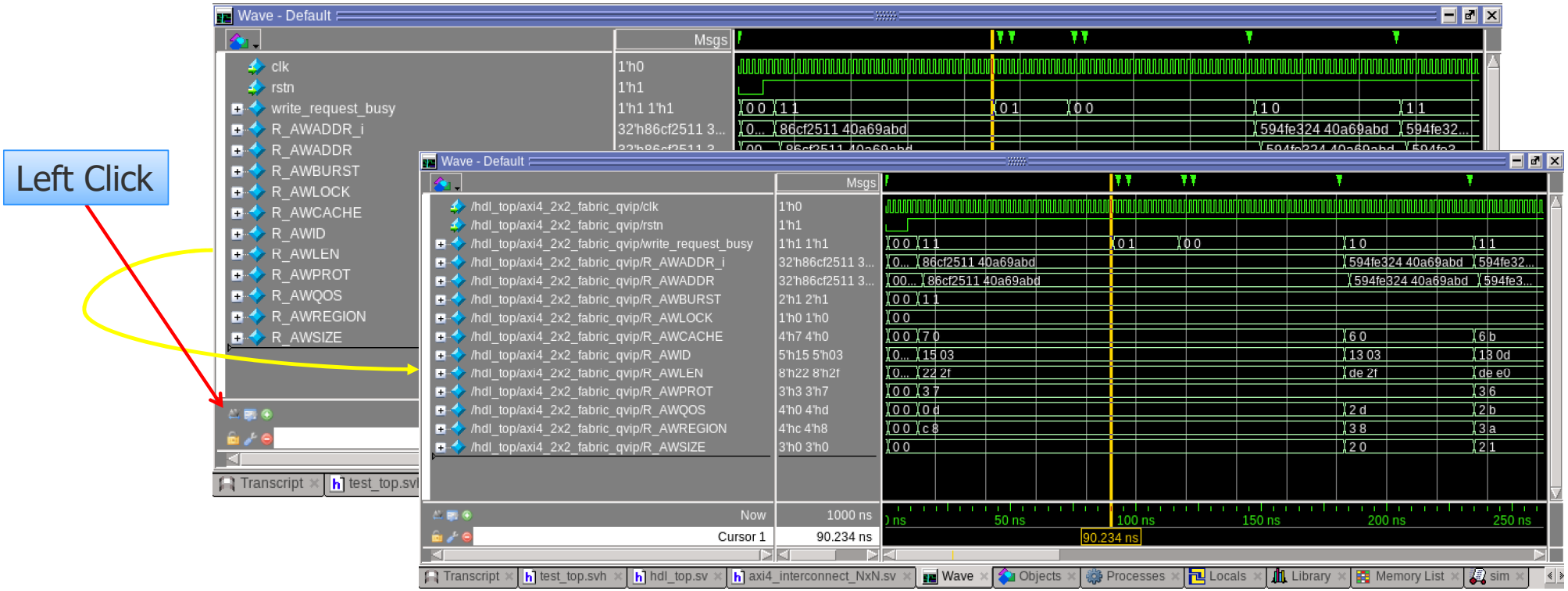
Note: Signal values only! Use CTRL-F to search signal names



© Mentor Graphics Corp.

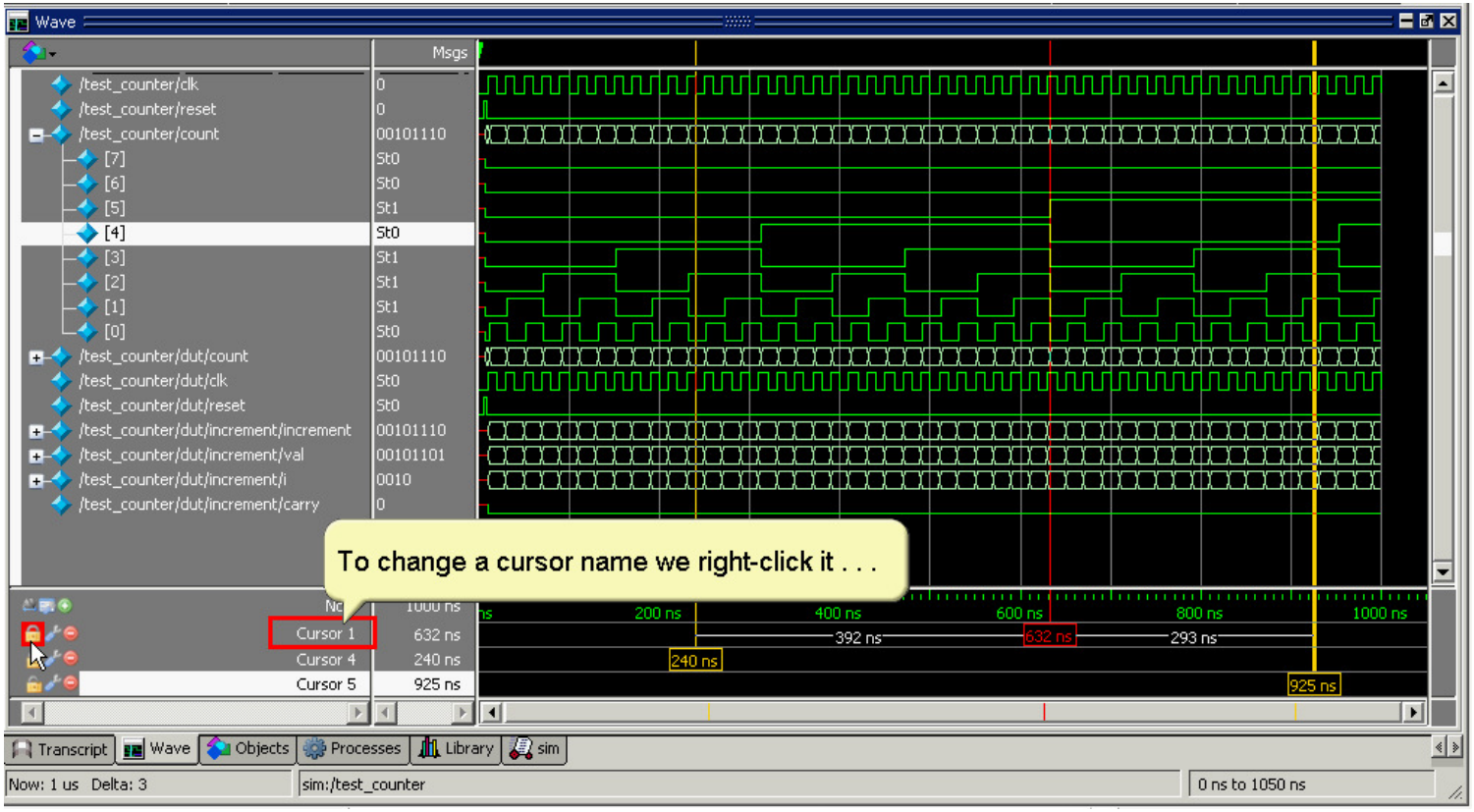
Toggle Signal Leaf Names

- See signal hierarchy by toggling signal leaf names



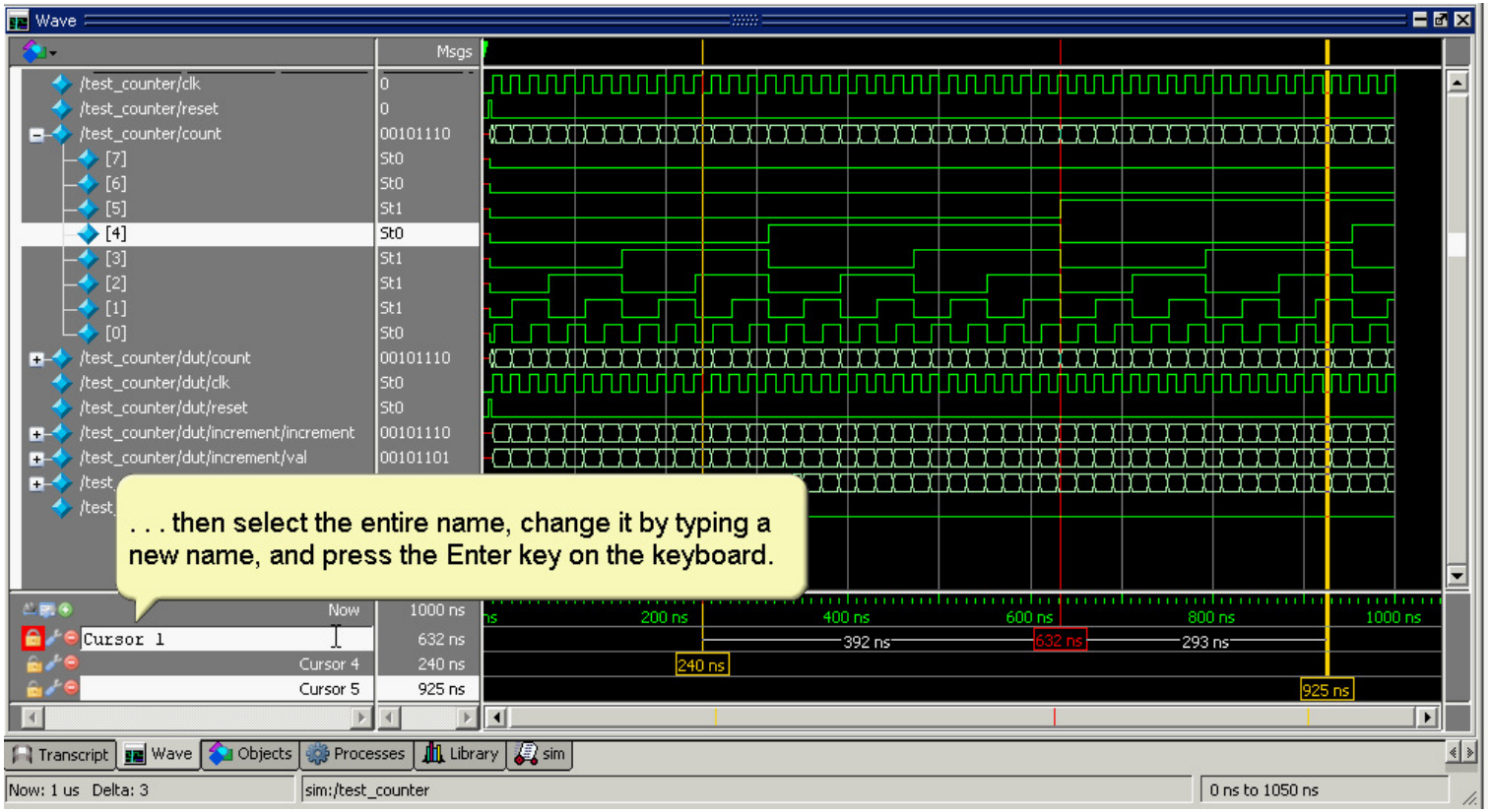
© Mentor Graphics Corp.

Cursor Naming



© Mentor Graphics Corp.

Cursor Naming Continued



© Mentor Graphics Corp.

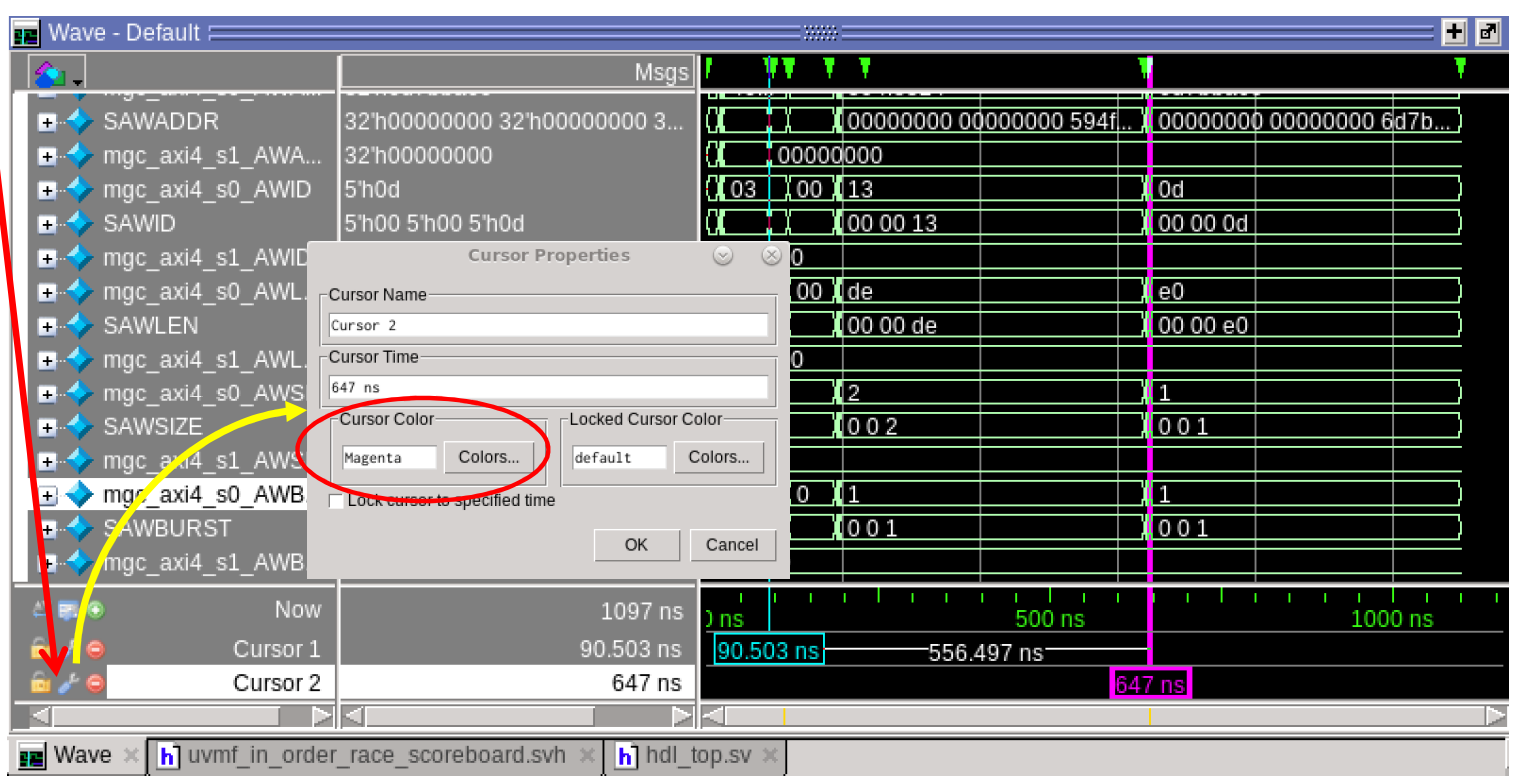
Cursor Shortcuts

- Jump to a cursor by clicking the cursor name or value
- Very fine moves of the cursor
  - Select a cursor
  - Use left and right arrows while within the cursor 'tracks'
- Manual enter a time with RMB in the time field

© Mentor Graphics Corp.

Change Cursor Color

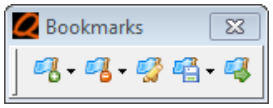
- Open Cursor Properties

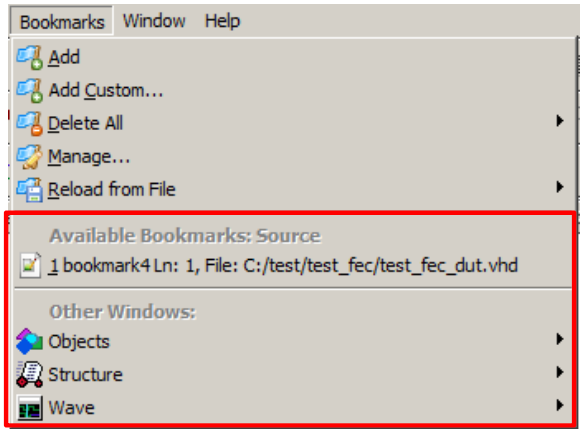


© Mentor Graphics Corp.

Bookmark Menus

- Can be added "as-is" or customized
- Can be reloaded from previous session
- Shortcuts are automatically added to the pull-down menu



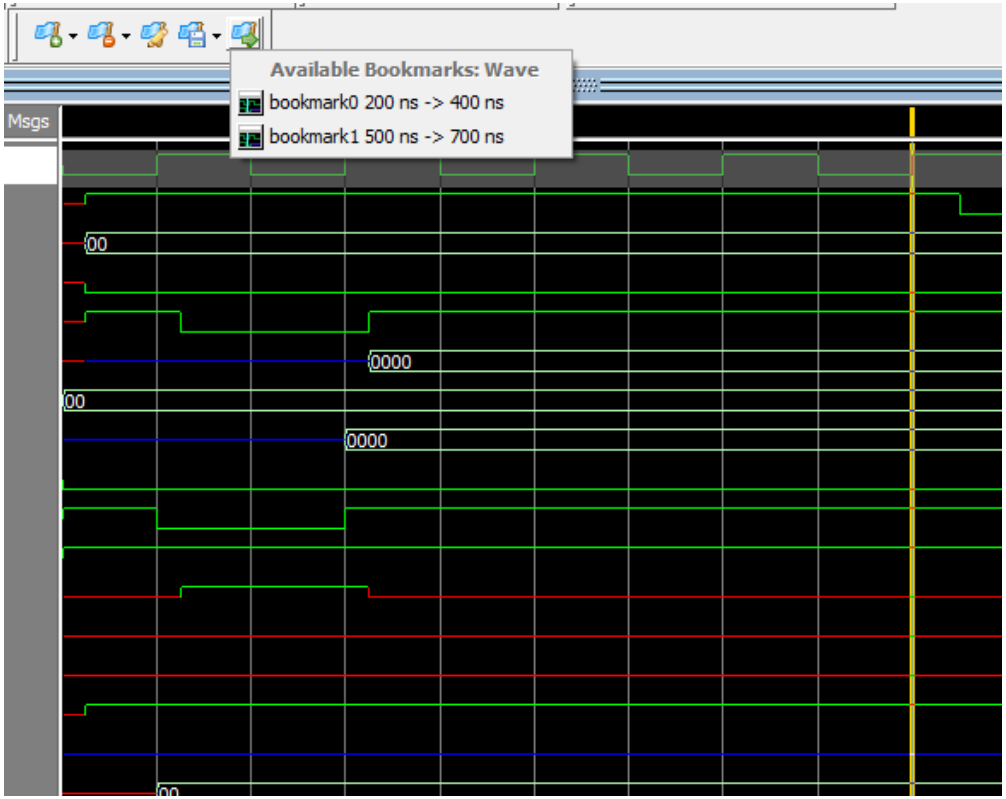


Supported Windows:

Source	Wave
Objects	Process
Files	Structure

© Mentor Graphics Corp.

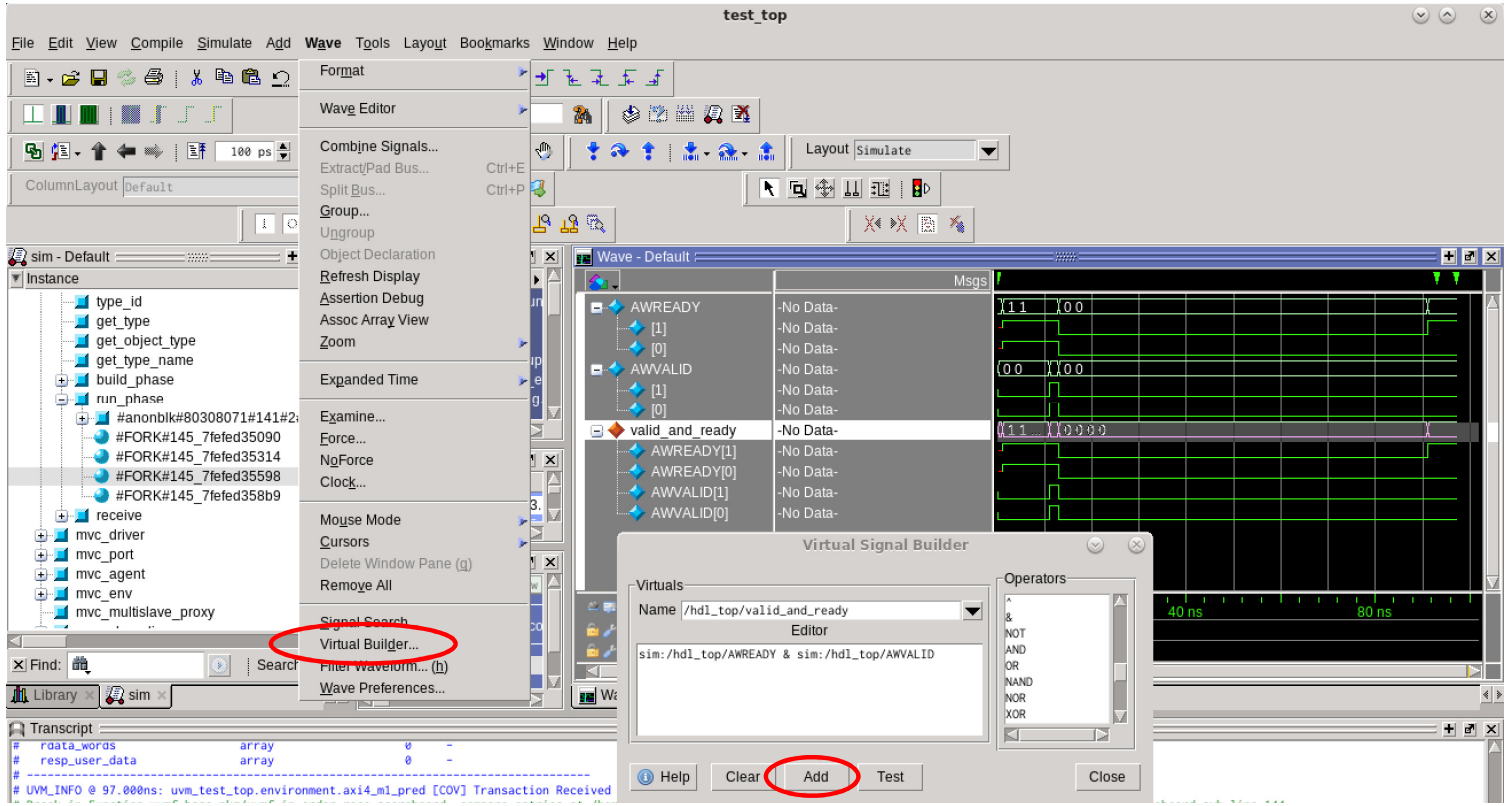
Jump to Bookmark



© Mentor Graphics Corp.

Virtual Signals

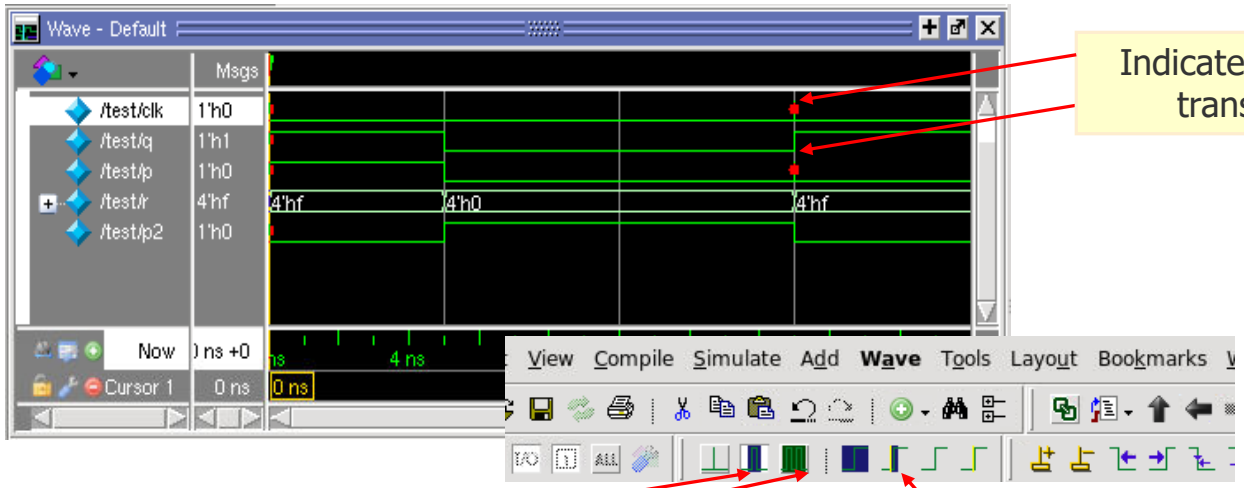
- Wave > Virtual Builder



© Mentor Graphics Corp.

Expanded Time

- Expand the Wave Window time to view all transitions during a single time step



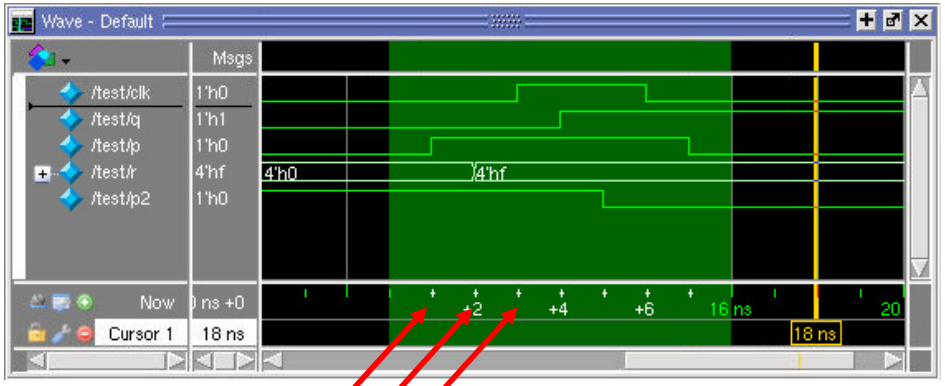
Indicates multiple transitions

Expand on active cursor

- Place the cursor at the transition, select either event or delta view, then expand on active cursor
- Use **`vsim -nowlfcollapse`** when creating WLF file

© Mentor Graphics Corp.

Delta vs. Event View



p and r change at the same delta-time step, but p changes before r in the actual event ordering

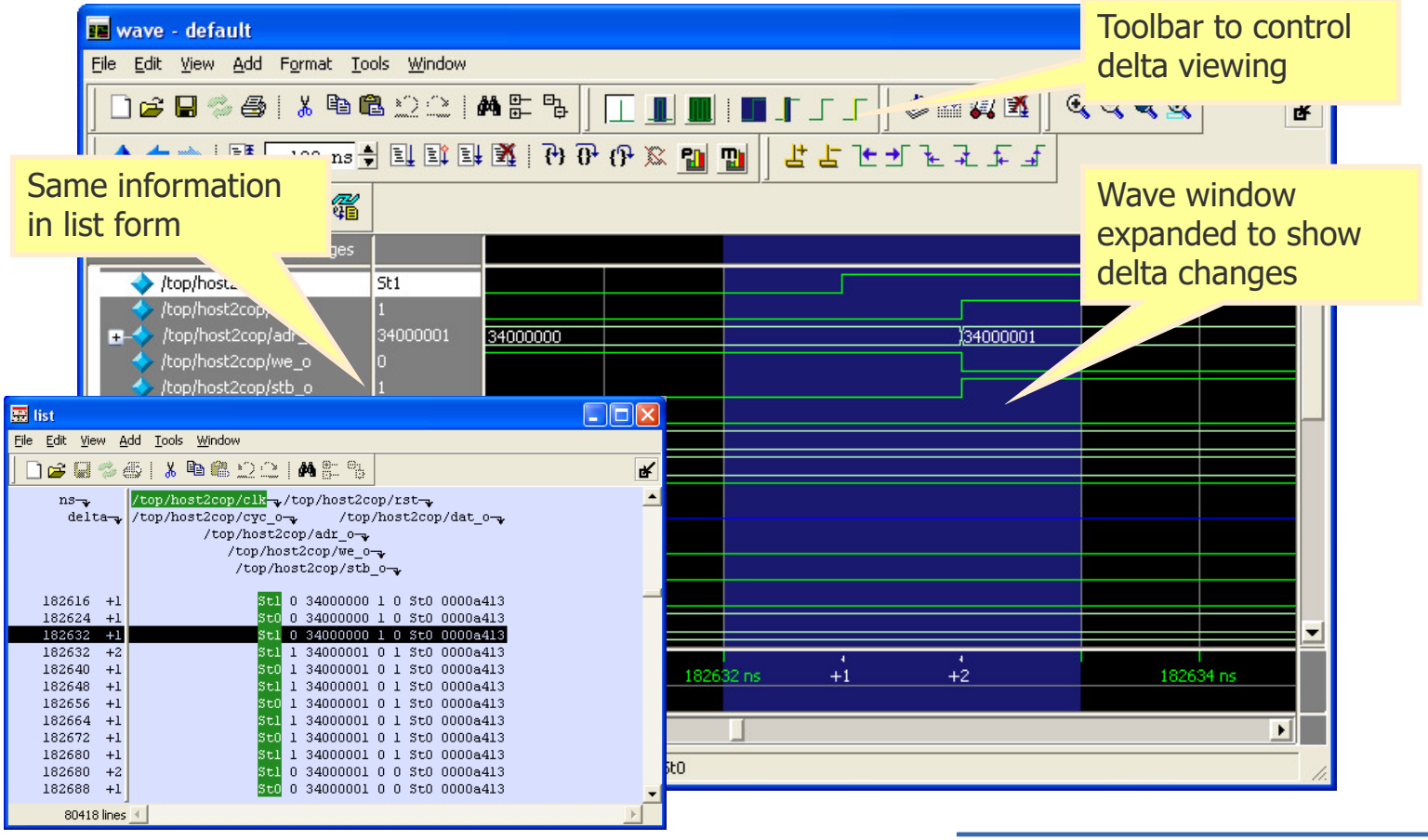
Event-time steps

Use these views to debug glitches, race conditions and event ordering issues

Delta-time steps

© Mentor Graphics Corp.

Wave Window Delta Viewing



© Mentor Graphics Corp.

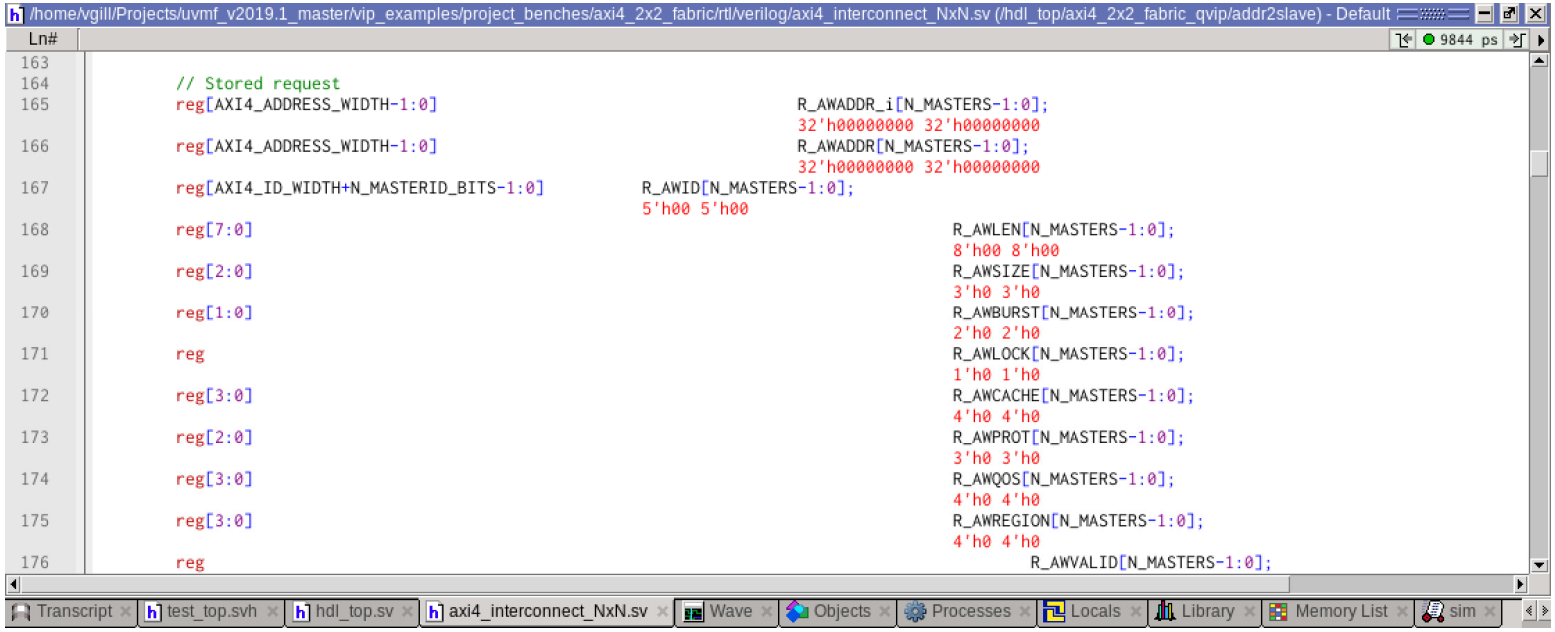
Agenda – I Didn’t Know Questa Could Do That!

- Help Resources
- GUI
- Optimizations and Performance
- Debug Tools
  - Wave Window
  - Source Window
  - FSM and Memory Windows
- Tips & Tricks
- Coverage

© Mentor Graphics Corp.

Source Annotation

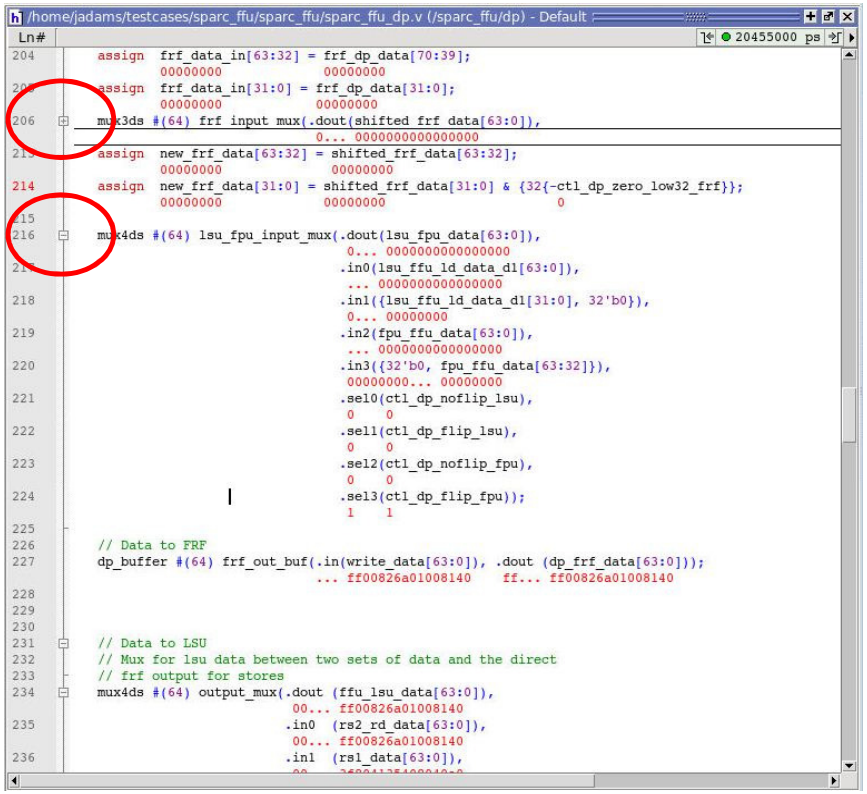
- Source annotation displays simulation values, including transitions, for each signal in your source file
  - Select Source > Show Source Annotation



© Mentor Graphics Corp.

Source Window - Code Folding

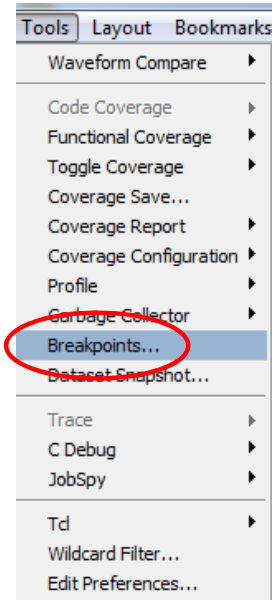
- Enable efficient viewing of source

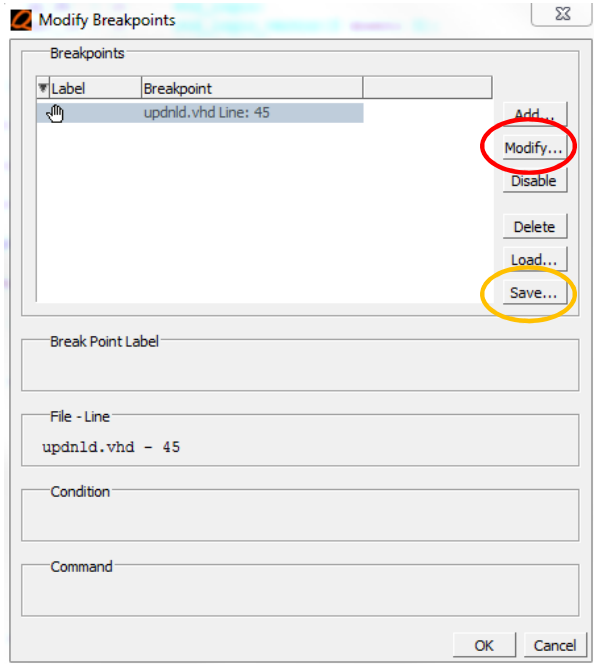


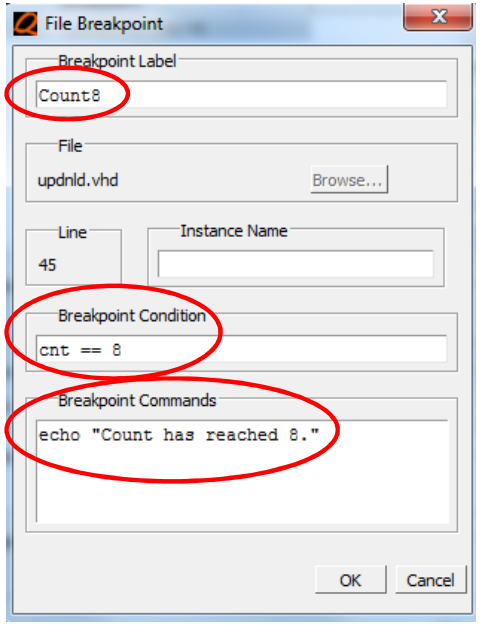
© Mentor Graphics Corp.

Controlling Breakpoints

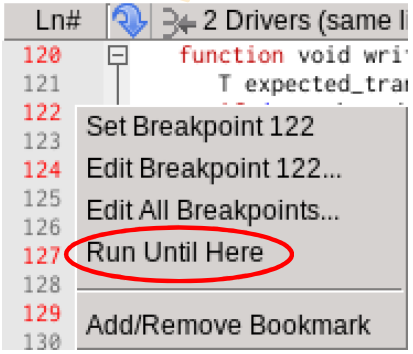
- Can specify an Instance and/or a Condition
- Can save to a *.do file





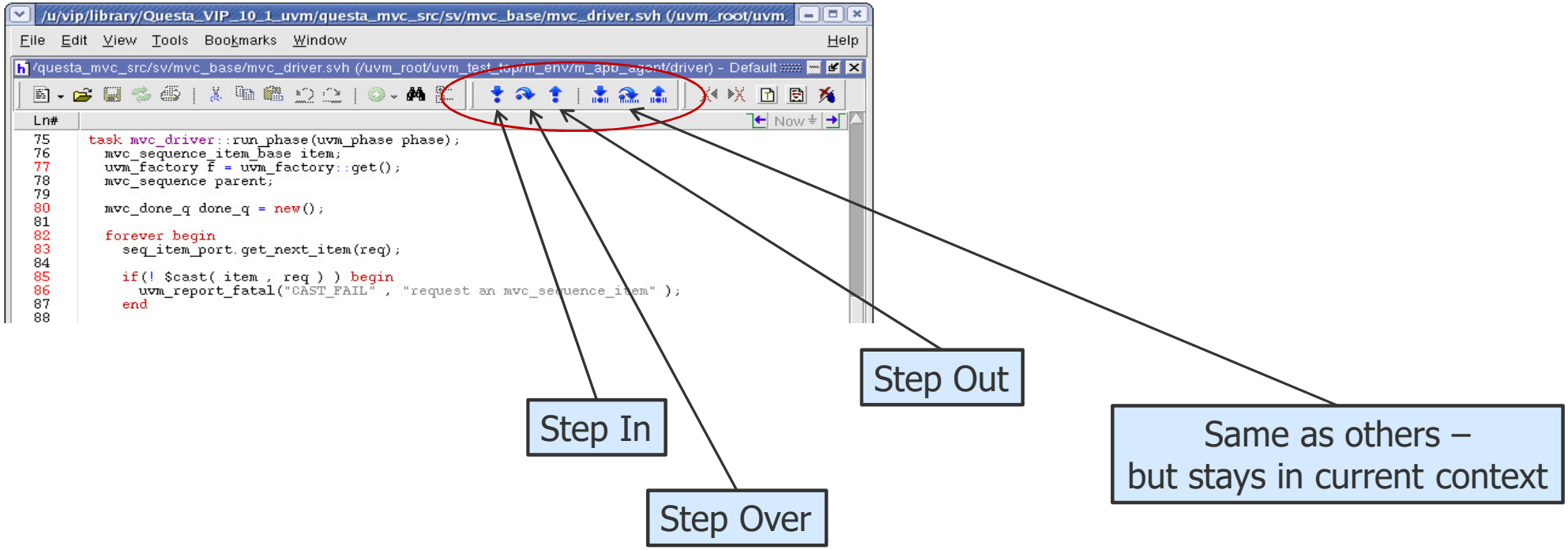


Run the simulation from the current simulation time up to the specified line of code.



© Mentor Graphics Corp.

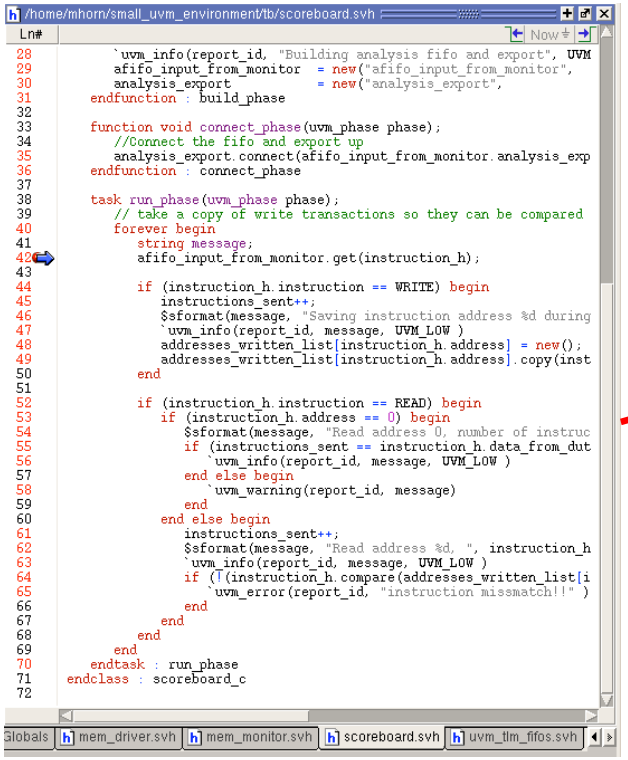
Step Toolbar

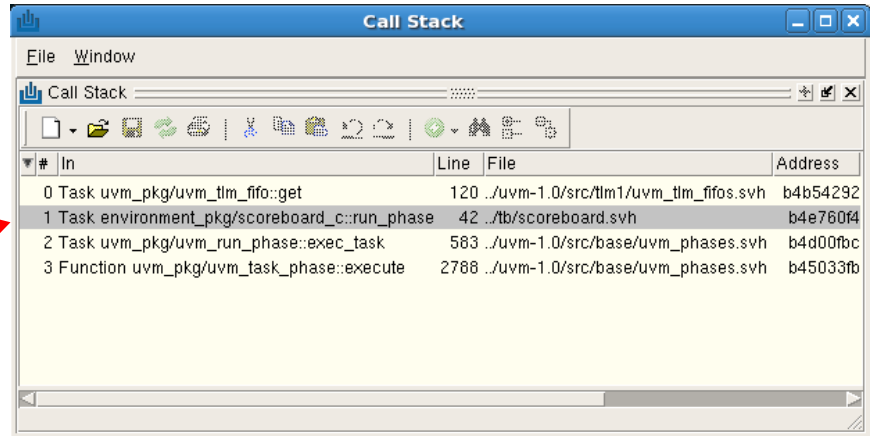


© Mentor Graphics Corp.

Call Stack

- How did I get here?
  - View > Call Stack





© Mentor Graphics Corp.

Locals Window

- View > Locals

Locals

Name	Value
axi4_2x2_fabric_env_pkg	
axi4_master_predictor	
write_axi4_master_ae	
this	@axi4_master_predictor__1@2
t	@axi4_master_rw_transaction__1@1
super	{} 32'h000006d9 @uvm_object_string_pool__1@156 @uvm_event@27 @uvm_event@28 32'hffffffff 64'h0000000000002af8 64'h0000000000053bd8 64'hffffffffffffffff null 32'h00000000 32'h00000000 1
read_or_write	AXI4_TRANS_WRITE
addr	32'h4bf2e8d2
prot	AXI4_NORM_NONSEC_DATA
region	4'hd
size	AXI4_BYTES_2
burst	AXI4_INCR
lock	AXI4_NORMAL
cache	AXI4_CACHE_14
qos	4'h5
id	4'h3
burst_length	8'h9d
addr_user_data	4'hc
wdata_words	32'hbcef0e13 32'h6af538ce 32'h2052e5d4 32'h35a0e81a 32'h969db67e 32'h243cc958 32'h96f75847 32'h3847fe87 32'h0f63d5ae 32'h4c8db883 32'h9a8530a4 32'hf33d46fe 32'h401aab03 32'h6a3e
write_strobes	4'hc 4'h1 4'hc 4'h1 4'hc 4'h0 4'h0 4'h1 4'h8 4'h1 4'hc 4'h3 4'h8 4'h3 4'h4 4'h3 4'h0 4'h2 4'hc 4'h0 4'h4 4'h3 4'h0 4'h1 4'h8 4'h1 4'hc 4'h1 4'hc 4'h1 4'h8 4'h1 4'hc 4'h1 4'h4 4'h2 4'h8 4'h1 4'h4 4'h1 4'h8 4'h0
wdata_user_data	4'h6 4'h6 4'hb 4'h1 4'h5 4'h8 4'h1 4'h9 4'h4 4'h1 4'h6 4'h7 4'h7 4'h2 4'hd 4'h4 4'he 4'h3 4'h0 4'h8 4'he 4'h8 4'h6 4'h9 4'h1 4'hf 4'h6 4'h0 4'h4 4'hc 4'h4 4'h2 4'he 4'h5 4'h8 4'h7 4'h2 4'h1 4'h0 4'hf 4'h0 4'ha
resp	AXI4_OKAY
rdata_words	
resp_user_data	4'h0
address_valid_delay	32'h00000005
data_valid_delay	32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'
write_response_valid_delay	32'h00000003
address_ready_delay	32'h00000000
data_ready_delay	32'h00000005 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'h00000000 32'
write_response_ready_delay	32'hffffffff
write_data_with_address	1'h1

Transcript | Wave | axi4_2x2_fabric_qvip_environment.svh | axi4_agent.svh | axi4_master_predictor.svh | Objects | Processes | Locals | Library | Memory List | sim

© Mentor Graphics Corp.

Watch Window

- View > Watch
  - Can add from structure, objects, locals window, and source windows

Watch

sim:/uvm_root/uvm_test_top/environment/axi4_m1_sb/transaction_count (0x7fefed96d548)
00000008

sim:/uvm_root/uvm_test_top/environment/axi4_m1_rw_sb/match_count (0x7fefed96dc08)
00000008

sim:/uvm_root/uvm_test_top/environment/axi4_m1_rw_sb/mismatch_count (0x7fefed96dc18)
00000000

sim:/uvm_root/uvm_test_top/environment/axi4_m0_sb/mismatch_count (0x7fefed96d0d8)
00000000

sim:/uvm_root/uvm_test_top/environment/axi4_m0_sb/match_count (0x7fefed96d0c8)
00000007

sim:/uvm_root/uvm_test_top/environment/axi4_m1_pred/axi4_ap_output_transaction (@axi4_master_rw_transaction__1@8)

Now

.super.super.super.super.super.super (/uvm_pkg::uvm_void)
.super.super.super.super.super (/uvm_pkg::uvm_object)
.super.super.super.super (/uvm_pkg::uvm_transaction)
.super.super.super (/uvm_pkg::uvm_sequence_item)
.super.super (/mvc_pkg::mvc_sequence_item_base)
.super (/mvc_pkg::mvc_sequence_item::mvc_sequence_item__2)
.read_or_write = AXI4_TRANS_READ
.addr = eca32e9f
.prot = AXI4_PRIV_SEC_INST
.region = 8
.size = AXI4_BYTES_2
.burst = AXI4_INCR
.lock = AXI4_NORMAL
.cache = AXI4_NONMODIFIABLE_NONBUF
.qos = 5
.id = f
.burst_length = 29
.addr_user_data = 0
.wdata_words =
.write_strobes =
.wdata_user_data =
.resp = {...}
.rdata_words = {...}
.resp_user_data =
.address_valid_delay = 00000000
.data_valid_delay =

Transcript | Wave | Watch | uvmf_in_order_race_scoreboard.svh | Objects | Processes | Locals | Library | sim

© Mentor Graphics Corp.

Agenda – I Didn’t Know Questa Could Do That!

- Help/Resources
- GUI
- Optimizations and Performance
- Debug Tools
  - Wave Window
  - Source Window
  - FSM and Memory Windows
- Tips & Tricks
- Coverage

© Mentor Graphics Corp.

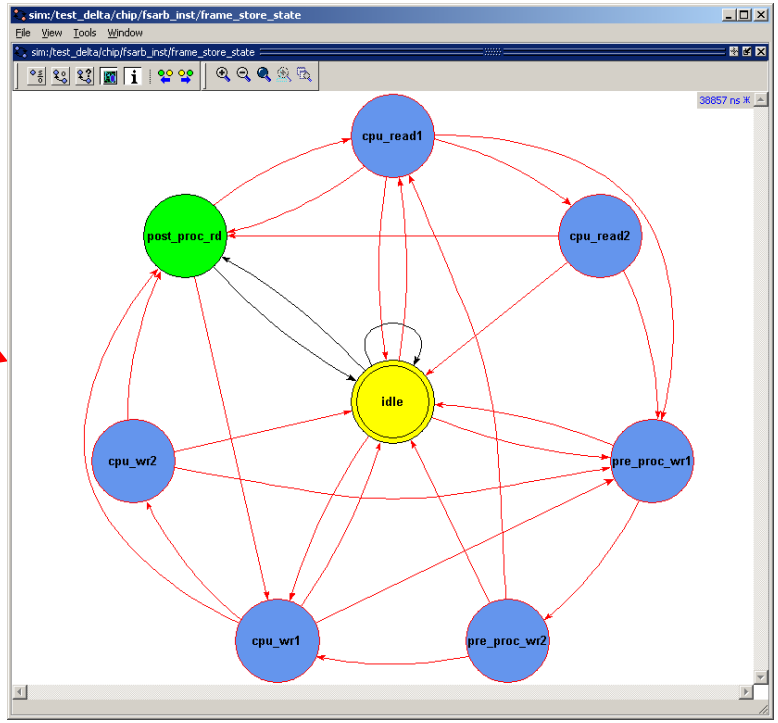
FSM List Window

- Need to add '`-fsmdebug`' to `vsim` command line
  - Auto extracted
- View > FSM List

FSM List

Instance	States	Transitions
sim:/rice_tb/calvin/I0/present_state	10	21
sim:/rice_tb/calvin/I1/present_state	7	17
sim:/rice_tb/hobbs/present_state	3	7
sim:/rice_tb/hobbs/I0/present_state	10	21
sim:/rice_tb/hobbs/I1/present_state	14	30

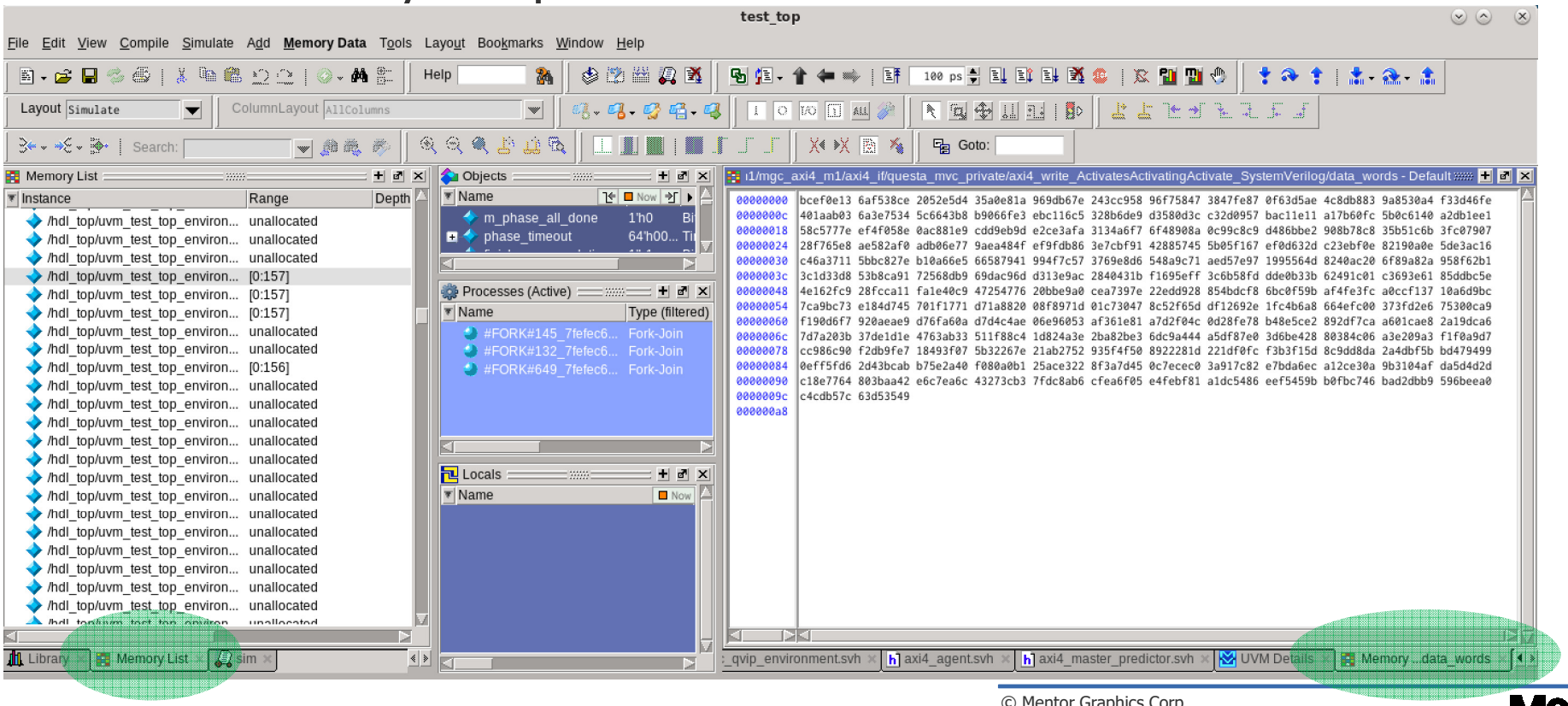
Graphical Element	Description	Definition
CTRL	Blue state bubble	Default appearance for non-reset states.
RD_WD_2	Green state bubble	Indicates the FSMs current state, or the state of the wave cursor location (when tracking the wave cursor).
RD_WD_1	Yellow state bubble	Indicates the FSMs previous state, or the state of the wave cursor location (when tracking the wave cursor).
IDLE	Tan state bubble with double outline.	Indicates a reset state.



© Mentor Graphics Corp.

Memory List Window

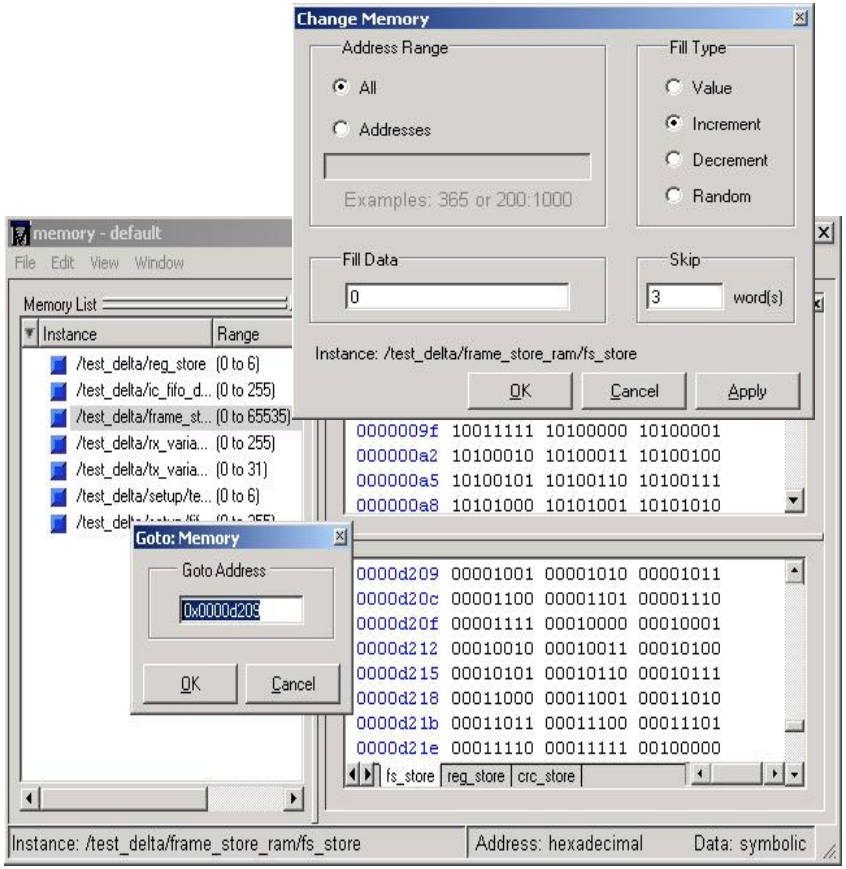
- Double click entry to open



© Mentor Graphics Corp.

Memory Data Window

- Auto extraction
- View complete memory contents
  - User-defined address range
  - Binary, decimal, hexadecimal, octal
  - Split Screens
- Find, replace, fill
- Save & load patterns
  - Easy memory initialization



© Mentor Graphics Corp.

Memory Capacity Window

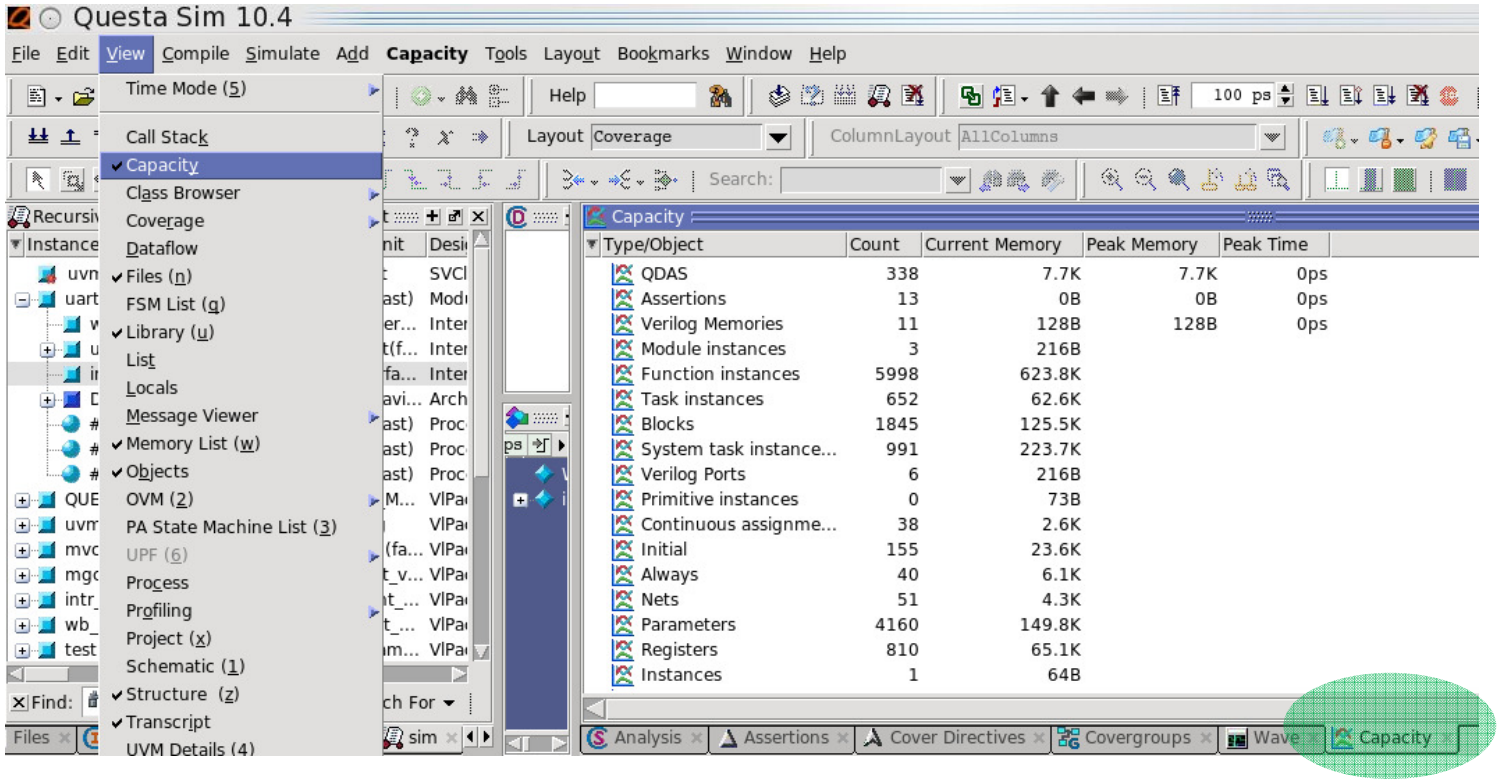
- Collects memory usage of SV constructs in the design
  - Classes
  - Queues, dynamic arrays, and associative arrays (QDAS)
  - Assertions & cover directives
  - Covergroups
  - Solver (calls to randomize())
- Usage
  - Course grain analysis enabled (by default)
  - Use "vsim -capacity" for fine grain analysis
- Shows memory usage at current timestep as well as peak usage for dynamic types
- Capacity Reports – Generate at end of simulation

```
write report -capacity [-l | -s] [-classes | -qdas |
             -assertions | -cvg | -solver]
```

© Mentor Graphics Corp.

Open Capacity Window

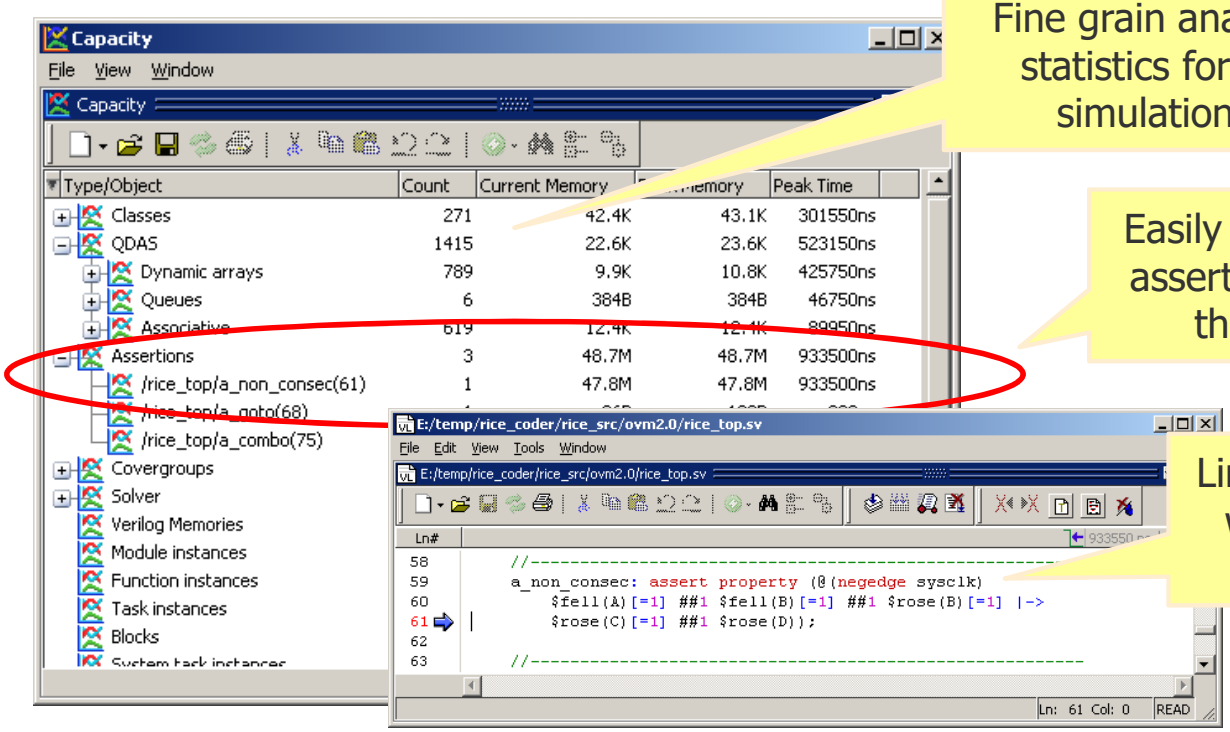
- View > Capacity



© Mentor Graphics Corp.

Enabling Fine Grain Analysis

- To find a specific problem object, enable fine grain analysis using **`vsim -capacity`**



Fine grain analysis shows statistics for individual simulation objects

Easily identify which assertion is causing the problem

Link from the Capacity window to problem assertion's source

© Mentor Graphics Corp.

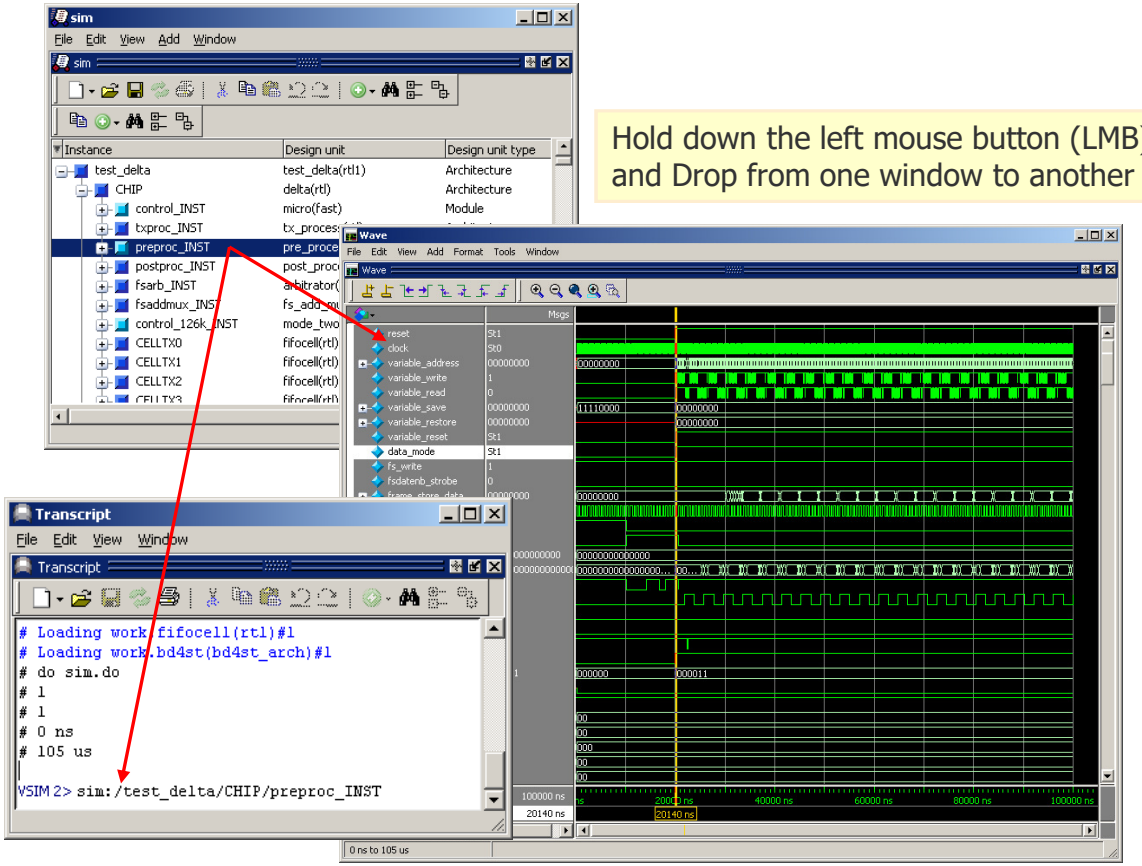
Agenda – I Didn’t Know Questa Could Do That!

- Help/Resources
- GUI
- Optimizations and Performance
- Debug Tools
  - Wave Window
  - Source Window
  - FSM and Memory Windows
- Tips & Tricks
- Coverage

© Mentor Graphics Corp.

Drag and Drop Feature

- Select items in one window, drag and drop into another window
- Drag items from the Structure to the transcript prints the object's complete path
- Use Drag & Drop to rearrange items in the Wave and List windows too



Hold down the left mouse button (LMB) to Drag and Drop from one window to another

© Mentor Graphics Corp.

Debugging Macros

- Use -Epretty switch
- Compile input files with –Epretty <output_file>
- Then, compile the output file with vlog
- Example:
  - vlog myfile.sv –lib work –Epretty myfile_pretty.sv
  - vlog myfile_pretty.sv

© Mentor Graphics Corp.

Debugging Macros

```
1.    class ethernet_txn  extends uvm_sequence_item;
2.       rand bit [47:0] dest_addr;
3.       rand bit [47:0] srce_addr;
4.       rand bit [15:0] payload_size;
5.       rand logic [7:0] payload [];
6.       bit[31:0]
7.
8.       `uvm_obje
9.         `uvm_fi
10.        `uvm_fi
11.        `uvm_fi
12.        `uvm_fi
13.        `uvm_fi
14.      `uvm_obje
```

```
1.    class ethernet_txn  extends uvm_sequence_item;
2.       rand bit [47:0] dest_addr;
3.       rand bit [47:0] srce_addr;
4.       rand bit [15:0] payload_size;
5.       rand logic [7:0] payload [];
6.       bit[31:0] crc;
7.       static bit m_registered_converter__ =
      m_uvm_resource_sprint_converter#(ethernet_txn)::register
      ();
8.       typedef
      uvm_object_registry#(ethernet_txn,"ethernet_txn")
      type_id;
9.       static function type_id get_type();
10.        return type_id::get();
11.      endfunction
12.      virtual function uvm_object_wrapper
      get_object_type();
13.        return type_id::get();
14.      endfunction
15.      function uvm_object create (string name="");
16.        ethernet_txn tmp;
17.        tmp = new();
18.        if (name!="")
19.        tmp.set_name(name);
20.        return tmp;
21.      endfunction
22.      const static string type_name = "ethernet_txn";
23.      virtual function string get_type_name ();
24.        return type_name;
25.      endfunction
```

© Mentor Graphics Corp.

Mentor A Siemens Business

Questa UVM Debug

- Controlled via -uvmcontrol switch

all	Enables all UVM-Aware functionality and debug options except disable and verbose You must specify verbose separately
disable	Prevents the UVM-Aware debug package from being loaded Changes the results of randomized values
msglog	Enables messages logged in UVM to be integrated into the Message Viewer window
none	Turns off all UVM-Aware debug features
struct	(**default**) Enables UVM component instances to appear in the Structure window under /uvm_top
trlog	Enables UVM transaction logging Logs UVM transactions for viewing in the wave window
verbose	Sends UVM debug package information to the transcript Does not affect functionality

© Mentor Graphics Corp.

SystemVerilog Class Debugging

- Default value of seed is 0
- `vsim -sv_seed random` prints seed in transcript
  - # Sv_Seed = 2389389234
- To check seed value at the VSIM prompt or in your do file
  - echo $Sv_Seed

© Mentor Graphics Corp.

Questa Object Naming Nomenclature

- Every class instance is assigned a unique ID in the following format:

  `@<type_name>@<#>`

  — `type_name` is the name of the class type

  — `#` is the number assigned to each unique instance of the class type

  i.e. `@mem_item@5` is the 5th instance of the class type `mem_item`

- This naming system is used throughout Questa

  — Can be returned in SV code by using `$get_id_from_handle()`

  e.g. `$get_id_from_handle(trans_h)`

© Mentor Graphics Corp.

Questa Parameterized Class Naming Nomenclature

- Every parameterized class specialization has its own type name
  - Each specialization is considered a different type and are not equivalent

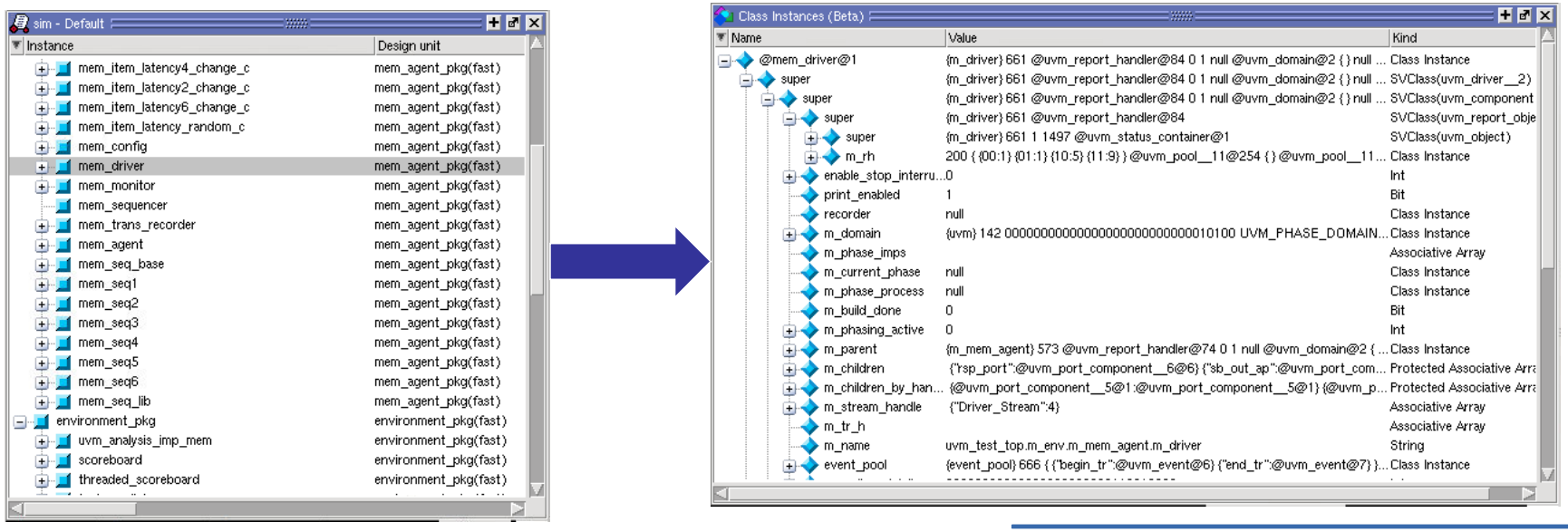
```
class myclass #(type T = bit, int size = 1);
…
child #(logic, 3) child_inst1 = new();
child #(byte, 4) child_inst2 = new();
```

- Questa will show
  - `/child/child__1` for the first type defined
  - `/child/child__2` for the second type defined

© Mentor Graphics Corp.

The -classdebug Switch

- Enables enhanced visibility into class instances
- Allows logging of class objects and class types
- Enables usage of Class Instances window



© Mentor Graphics Corp.

Tcl 'classinfo' Command

- Use to find information about class usage
- classinfo report: statistics on a per class basis

Full Path	Class Name	Total	Peak	Current
/mem_agent_pkg::mem_agent	mem_agent	1	1	1
/mem_agent_pkg::mem_config	mem_config	1	1	1
/mem_agent_pkg::mem_driver	mem_driver	1	1	1
/mem_agent_pkg::mem_item	mem_item	140	23	21
/mem_agent_pkg::mem_item_latency2_change_c	mem_item_latency2_change_c	0	0	0
/mem_agent_pkg::mem_item_latency4_change_c	mem_item_latency4_change_c	0	0	0
/mem_agent_pkg::mem_item_latency6_change_c	mem_item_latency6_change_c	0	0	0
/mem_agent_pkg::mem_item_latency_random_c	mem_item_latency_random_c	0	0	0
/mem_agent_pkg::mem_monitor	mem_monitor	1	1	1
/mem_agent_pkg::mem_seq1	mem_seq1	0	0	0
/mem_agent_pkg::mem_seq2	mem_seq2	0	0	0
/mem_agent_pkg::mem_seq3	mem_seq3	0	0	0
/mem_agent_pkg::mem_seq4	mem_seq4	0	0	0
/mem_agent_pkg::mem_seq5	mem_seq5	1	1	1
/mem_agent_pkg::mem_seq6	mem_seq6	0	0	0
/mem_agent_pkg::mem_seq_base	mem_seq_base	0	0	0
/mem_agent_pkg::mem_seq_lib	mem_seq_lib	0	0	0
/mem_agent_pkg::mem_trans_recorder	mem_trans_recorder	0	0	0

- classinfo stats: statistics across all classes

```
class type count                 778
class instance count (total)    1910
class instance count (peak)     1215
class instance count (current)  1207
```

Requires vsim –classdebug

© Mentor Graphics Corp.

Tcl 'describe' Command

- describe <package> - package contents

```
VSIM 37> describe sim:/mem_agent_pkg
# Design Unit
#   Package: mem_agent_pkg(fast)
#   Library: /small_uvm_environment/questa/work
#   Source files:
#     ../sv/mem_agent/mem_agent_pkg.sv
#     ../sv/mem_agent/mem_item.svh
#     ../sv/mem_agent/mem_config.svh
#     ../sv/mem_agent/mem_driver.svh
#     ../sv/mem_agent/mem_monitor.svh
#     ../sv/mem_agent/mem_trans_recorder.svh
#     ../sv/mem_agent/mem_agent.svh
#     ../sv/mem_agent/mem_seq_base.svh
#     ../sv/mem_agent/mem_seq1.svh
#     ../sv/mem_agent/mem_seq2.svh
#     ../sv/mem_agent/mem_seq3.svh
#     ../sv/mem_agent/mem_seq4.svh
#     ../sv/mem_agent/mem_seq5.svh
#     ../sv/mem_agent/mem_seq6.svh
#     ../sv/mem_agent/mem_seq_lib.svh
```

© Mentor Graphics Corp.

Tcl 'call' Command

- Call function - call SV functions or zero time tasks

```
VSIM 19> examine -handle {sim:/uvm_root/top_levels[0].super.m_env.m_mem_agent.m_monitor}
# @mem_monitor@1
VSIM 20> call @mem_monitor@1.print_config
# UVM_INFO @ 101.5 ns: uvm_test_top.m_env.m_mem_agent.m_monitor [CFGPRT] visible resource
s:
# recording_detail [/^.*$/] : (uvm_bitstream_t) 400
# -
# mem_sequencer [/^.*$/] : (class uvm_pkg::uvm_sequencer #(class mem_agent_pkg::mem_item,class
mem_agent_pkg::mem_item ))
# ----------------------------------------------------------
# Name                Type                Size  Value
# ----------------------------------------------------------
# m_sequencer         uvm_sequencer       -     @692
#   rsp_export        uvm_analysis_export -     @699
#     recording_detail uvm_verbosity      32    UVM_FULL
#   seq_item_export   uvm_seq_item_pull_imp -   @793
#     recording_detail uvm_verbosity      32    UVM_FULL
#   recording_detail  uvm_verbosity       32    UVM_FULL
#   arbitration_queue array               0     -
#   lock_queue        array               0     -
#   num_last_reqs     integral            32    'd1
#   num_last_rsps     integral            32    'd1
# ----------------------------------------------------------
#
# -
# mem_config [/^uvm_test_top\..*mem_agent.*$/] : (class mem_agent_pkg::mem_config)
# ----------------------------------------------------------
# Name                Type                Size  Value
# ----------------------------------------------------------
# m_mem_cfg           mem_config          -     @555
# ----------------------------------------------------------
```

© Mentor Graphics Corp.

UVM Convenience Functions

- Built-in UVM functions callable from Tcl command-line

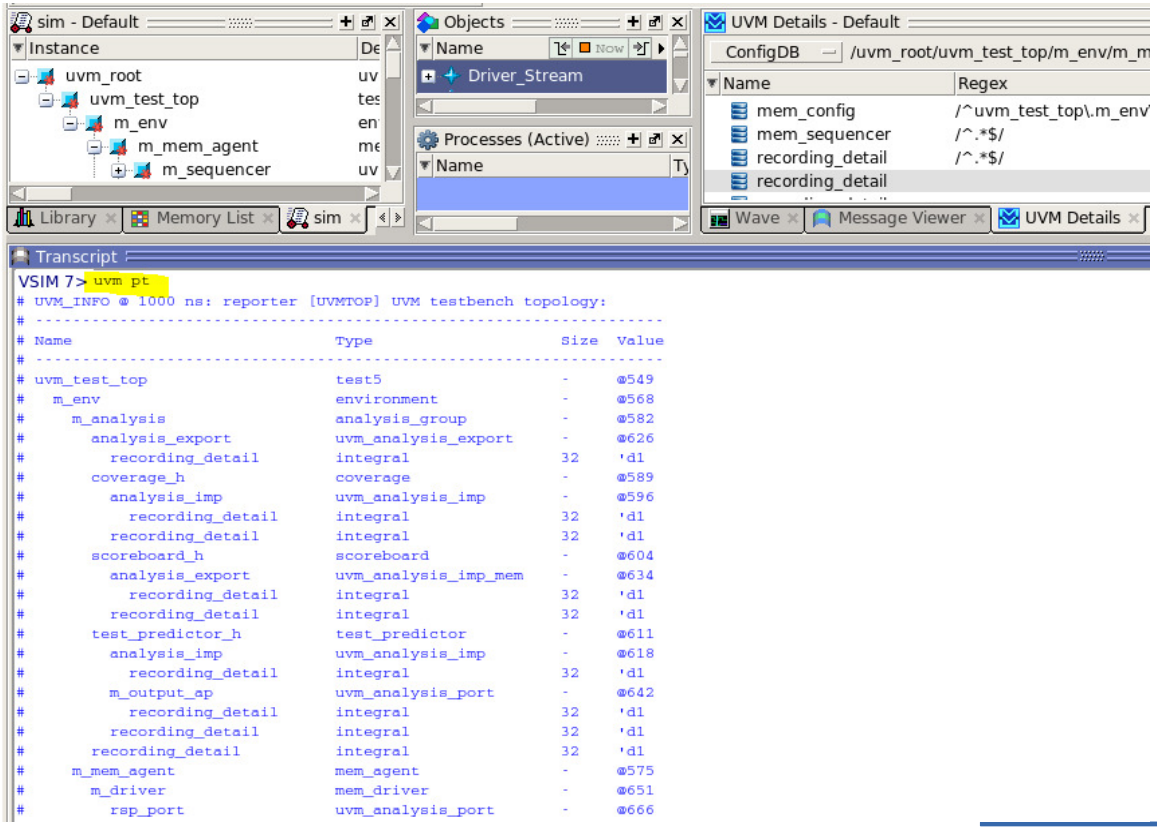
call	call a UVM SV function - default subcommand
configtracing, ct	enables or disables config tracing
displayobjections, do	displays current objections
handle	return simulation handle of object
findregisters, fr	finds the HDL registers for a UVM register model
findsequences, fs	finds the currently active sequences
help	prints this message
mapmode, mm	sets the uvm command path mapping mode (default is uvm-style)
printconfig, pc	prints the global component-level config data
printfactory, pf	prints the global factory information
printstreams, ps	prints a list of transaction streams for a given hierarchy
printtopology, pt	prints the global topology structure
setverbosity, sv	set the uvm reporter verbosity level
simpath, sp	returns a simulator /uvm_root path given a UVM component path
traceobjections, to	enable objection tracing
uvmpath, up	returns a UVM component path given a simulator /uvm_root path

© Mentor Graphics Corp.

UVM Convenience Functions

Example 1

- uvm pt - prints the global topology structure



© Mentor Graphics Corp.

UVM Convenience Functions

Example 2

- uvm fs - finds the currently active sequences

```
Transcript
VSIM 6> run 200 ns
VSIM 7> run 1000ns
VSIM 8> uvm fs
# Sequencer: uvm_test_top.env_h.cpu_agent_h.seqr_h
#    ../sv/cpu_agent/cpu_rand_seq.svh:21:  uvm_test_top.env_h.cpu_agent_h.seqr_h.rand_seq_h  handle = @cpu_rand_seq@1
# Sequencer: uvm_test_top.env_h.mem_agent_h.seqr_h
#    ../sv/mem_agent/mem_sequence.svh:21:  uvm_test_top.env_h.mem_agent_h.seqr_h.mem_seq_h  handle = @mem_sequence@1

VSIM 9>
Now: 1,300 ns   Delta: 2        sim:/tb
```

- uvm fr - finds the HDL registers for a UVM register model

```
Transcript
# ***********************************************************************
# UVM_INFO ../sv/bus_agent/bus_monitor.svh(44) @ 0 ns: uvm_test_top.m_env.m_bus_agent.m_monitor [m_monitor] running m
onitor
# UVM_INFO @ 0 ns: uvm_test_top.m_env.m_bus_agent.m_driver [DRIVER] running bus_driver
VSIM 5> uvm fr -acc sim:/uvm_root/uvm_test_top/m_env/dut_rm
# +acc=rn+/testbench/i_dut/apply_holding_reg +acc=rn+/testbench/i_dut/back_reg +acc=rn+/testbench/i_dut/data_reg +acc
=rn+/testbench/i_dut/front_reg +acc=rn+/testbench/i_dut/holding_reg +acc=rn+/testbench/i_dut/i_normal_regs/regA +acc=
rn+/testbench/i_dut/i_normal_regs/regB +acc=rn+/testbench/i_dut/i_normal_regs/status_reg +acc=rn+/testbench/i_dut/i_n
ormal_regs/thirtytwobit +acc=rn+/testbench/i_dut/i_quirky_regs/apply_holding_reg +acc=rn+/testbench/i_dut/i_quirky_re
gs/back_reg +acc=rn+/testbench/i_dut/i_quirky_regs/data_reg +acc=rn+/testbench/i_dut/i_quirky_regs/front_reg +acc=rn+
/testbench/i_dut/i_quirky_regs/holding_reg +acc=rn+/testbench/i_dut/regA +acc=rn+/testbench/i_dut/regB +acc=rn+/testb
ench/i_dut/status_reg +acc=rn+/testbench/i_dut/thirtytwobit

VSIM 6>
Now: 0 ns   Delta: 159        sim:/uvm_root/uvm_test_top/m_env
```

© Mentor Graphics Corp.

Mentor A Siemens Business

UVM Convenience Functions

Example 3

- uvm do - displays current objections

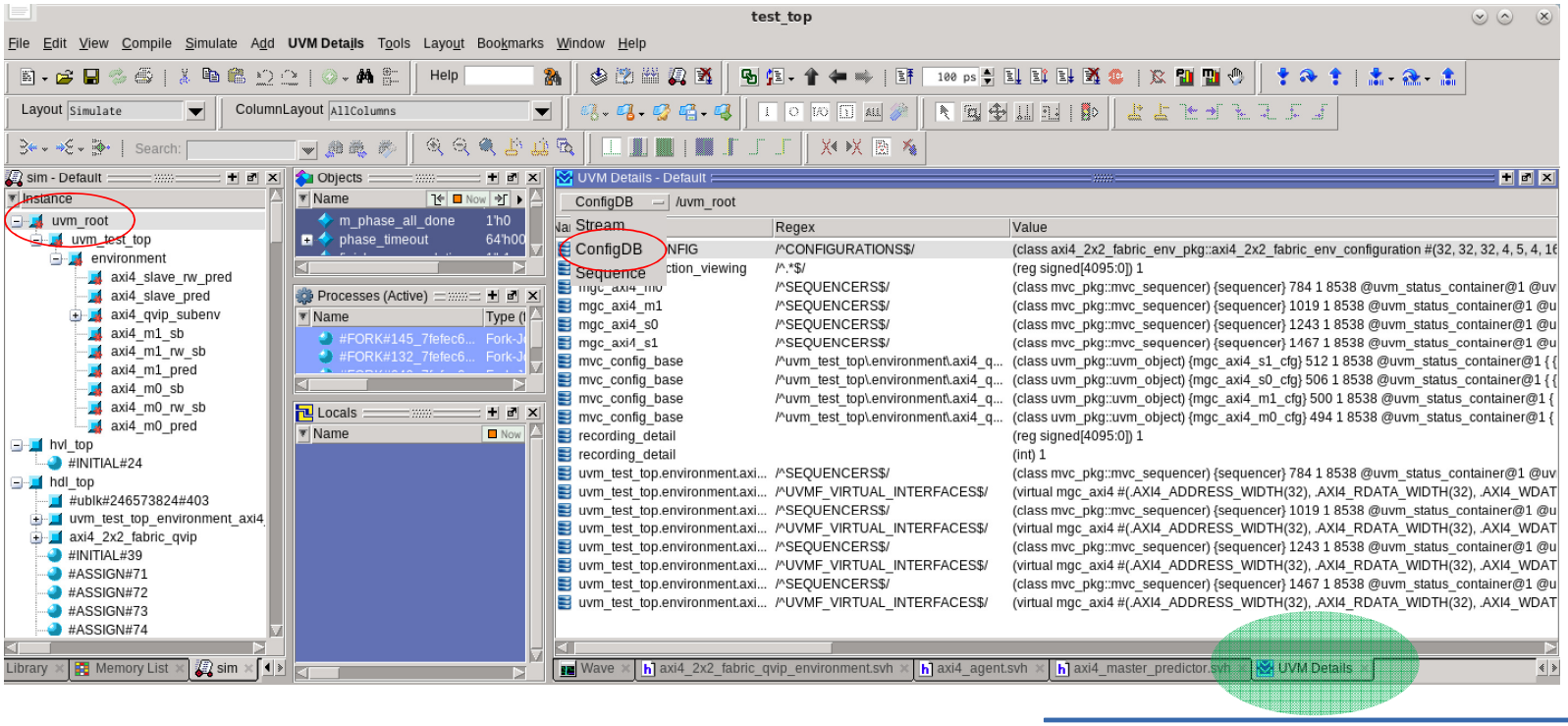
Transcript

```
#
VSIM 7> uvm do
#     Domain: common
#  Component: uvm_root
#      Phase: uvm.uvm_sched.main
#  Objection: main_objection
# The total objection count is 1
# ---------------------------------------------------------
# Source  Total
# Count   Count   Object
# ---------------------------------------------------------
# 0       1       uvm_top
# 1       1         uvm_test_top
# ---------------------------------------------------------
#
#     Domain: uvm
#  Component: uvm_root
#      Phase: uvm.uvm_sched.main
#  Objection: main_objection
# The total objection count is 1
# ---------------------------------------------------------
# Source  Total
# Count   Count   Object
# ---------------------------------------------------------
# 0       1       uvm_top
# 1       1         uvm_test_top
# ---------------------------------------------------------
#
```

© Mentor Graphics Corp.

UVM Details

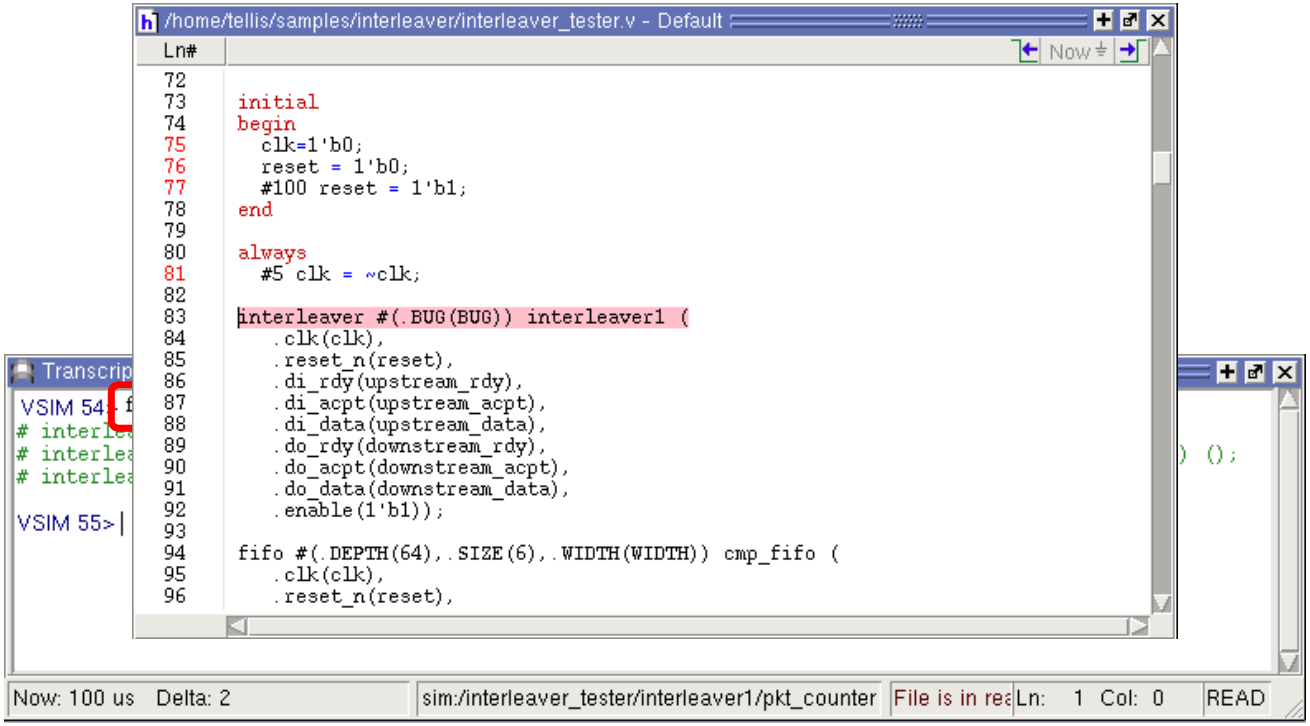
- View > UVM Details
  - Results are context/scope sensitive to structure window



© Mentor Graphics Corp.

Find insource

- ***find insource***
  - Searches for a string in the source files of current design



© Mentor Graphics Corp.

Other Helpful Commands

- ***help <command>***
  - Displays brief description and syntax for the specified command
  - Example:

```
> help drivers
# The drivers command displays the names of all drivers of the specified object.
# Usage: drivers <object_name>
```

- ***verror <msg_number>***
  - Displays detailed description about message
  - Example:

```
# ** Error: (vsim-3462) Value/time pairs must be listed in increasing time order.

> verror 3462
#
# vsim Message # 3462:
# The value/time pairs specified to the force command must be listed in
increasing order of time. For example, 'force busa 1 10, 0 20, 1 30'.
```

© Mentor Graphics Corp.

Agenda – I Didn’t Know Questa Could Do That!

- Help/Resources
- GUI
- Optimizations and Performance
- Debug Tools
  - Wave Window
  - Source Window
  - FSM and Memory Windows
- Tips & Tricks
- Coverage

© Mentor Graphics Corp.

Types of Code Coverage

- Coverage controlled using `+cover=<option>`
- Coverage options:
  - s: statement
  - b: branch (if/then/else & case statements)
  - c: condition (if & ternary decisions)
  - e: expression (RHS of assignments)
  - f: FSM (both states and transitions)
  - t: toggle (2-state)
  - x: extended toggle (4-state)
- All options (except x) applied when `+cover` used without options

© Mentor Graphics Corp.

Enabling Code Coverage

- Code coverage enabled in the compilers and/or optimizer

```
vlog +cover=f -f files_vlog.f
vcom -f files_vhdl.f
vopt +cover=sbc top -o opt_cov
vopt top -o opt_no_cov -nocover
```

Cover metrics added in compiler OR'd with those added in vopt

Can also tell vopt to remove all instrumentation added by compiler

- Code coverage turned on at simulator invocation

```
vsim -coverage opt_cov
vsim opt_no_cov
```

Run with or without coverage enabled in the simulator

© Mentor Graphics Corp.

Controlling Code Coverage Focus

- Specific source files in the design

```
vlog top.v proc.v cache.v +cover=bce
```

Branch
Condition
Expression

- Specific modules or instances

```
vlog top.v proc.v cache.v

vopt -o top_opt +cover=cf+moduleA \
      +cover=st+/top/proc/cache
```

Condition
FSM

Statement
Toggle

- Entire design (globally)

```
vlog top.v proc.v cache.v

vopt -o top_opt +cover=bces
```

Branch
Condition
Expression
Statement

© Mentor Graphics Corp.

Controlling Code Coverage Focus Cont.

- Excluding modules or instances

recursive

```
vopt -o top_opt +cover=bces+/top. \
      +nocover+/top/altera_core.
```

Excludes Altera instance

- Specifying recursion level

```
vopt -o top_opt
+cover=st+/top/proc/cache+3
```

Includes 3 levels below (integer between 0 and 128, where a value of 0 applies full recursion)

- Please refer to the Questa User's Guide for more information on priorities, conflicts, and examples
  - *Priorities for Resolving Conflicting Control Arguments*
  - *Conflicting Control Rules Examples with Description*

© Mentor Graphics Corp.

Adaptive Exclusions

- Typical exclusion behavior

```
1   module top(sel);
2
3   input sel;
4   reg a,b,c,d,f;
5
6   initial begin
7     d = b & a;
8     c = a & b;
9   end
10
11  always @(sel) begin
12    if (sel == 1)
13      f = a;
14    else
E 15    f = b;
16  end
17  endmodule
```

```
1   module top(sel);
2
3   input sel;
4   reg a,b,c,d,f;
5
6   initial begin
7     d = b & a;
8     c = a & b;
9   end
10
11  always @(sel) begin
12    //I added a comment
13    if (sel == 1)
14      f = a;
X 15  else
E 16    f = b;
17  end
18  endmodule
```

```
1   module top(sel);
2
3   input sel;
4   reg a,b,c,d,f;
5
6   initial begin
7     d = b & a;
8     c = a & b;
9   end
10
11  always @(sel) begin
12    //I added a comment
13    //that is two lines
14    if (sel == 1)
E 15    f = a;
16    else
E 17    f = b;
18  end
19  endmodule
```

coverage exclude -src top.v -code s -line 15

** Warning: (vsim-4036) The 'coverage exclude' command had no effect on some/all objects because no matching coverage data was found.

© Mentor Graphics Corp.

Enabling Adaptive Exclusions

1. `vopt +cover -adaptive ...`

2. `vsim -coverage ...`

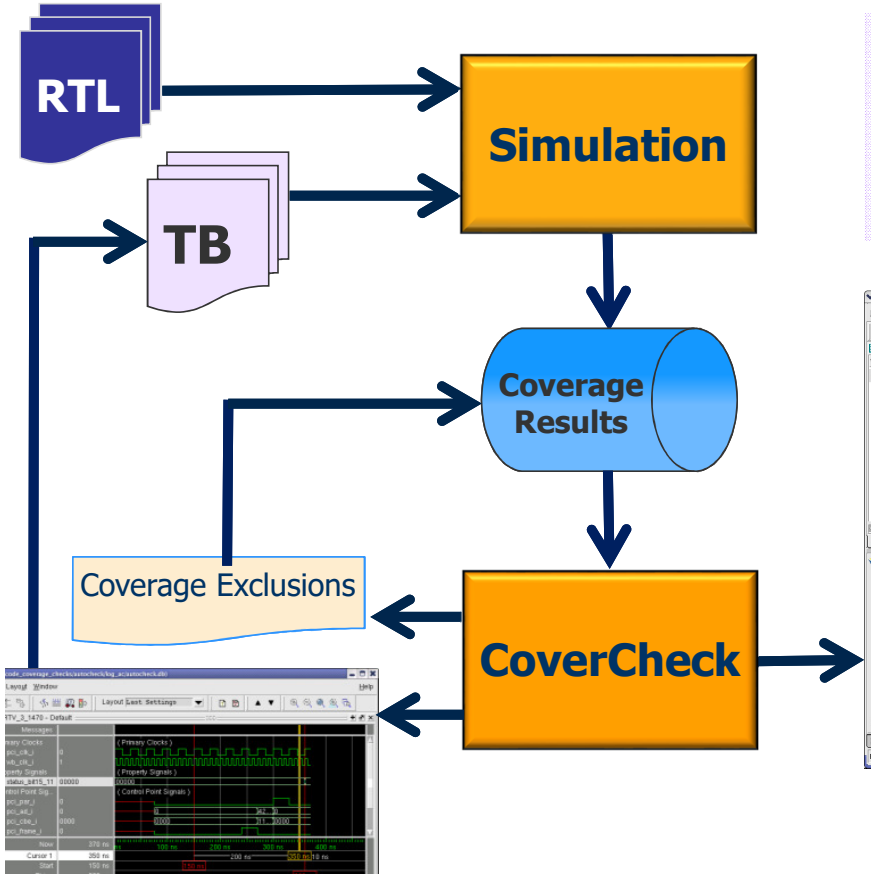
3. `coverage report -excluded -adaptive -file adapt.do`

   or

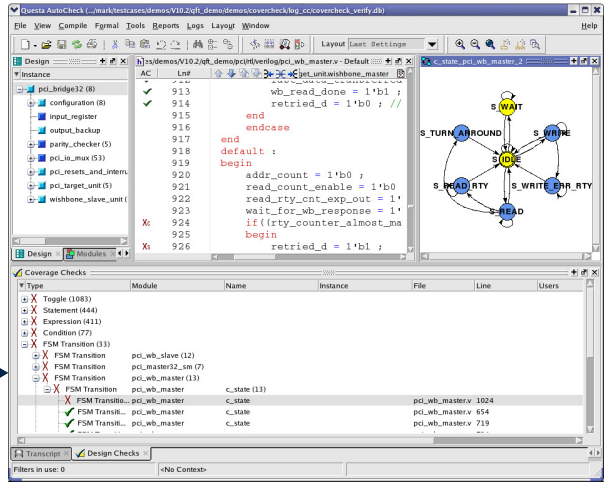
   `vcover report -excluded -adaptive -file adapt.do`

© Mentor Graphics Corp.

Questa CoverCheck Verification Flow



This flow is optimal because only the coverage items that **were not hit in simulation** are targeted by CoverCheck



© Mentor Graphics Corp.

How to run CoverCheck

- Makefile

Batch mode

Specify Output Directory

Run TCL cmds (can be file)

```
covercheck:
   qverify -c -od Output_Results -do " \
   do qs_files/directives.tcl; \
   covercheck compile -d top; \
   covercheck load ucdb merged.ucdb; \
   covercheck verify -effort low -witness_waveforms; \
   covercheck generate exclude exclusions.do; exit"
```

Add code cov directives

Specify top level

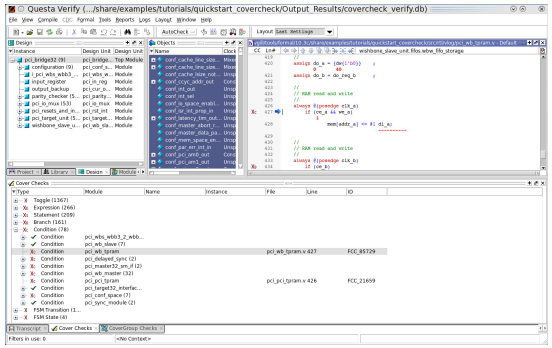
Loads merged UCDB

Common options to 'covercheck verify'	
-effort {low|medium|high}	Specifies effort level
-witness_waveforms	Enables waveform view of covered elements
-work <lib_name>	Specifies a work library (*work* is default)
-modelsimini <ini_file>	Points to a modelsim.ini file
-init <filename>	Specifies an initialization file
-timeout	Specifies how long to run

© Mentor Graphics Corp.

View/Analyze Results

- Covercheck_verify.db [%> qverify covercheck_verify.db]
  - View & analyze results in GUI



- covercheck_verify.log
  - Provides summary & run stats

```
CoverCheck Summary
-----------------------------------------------------------------------------
Coverage Type                  Active   Unreachable     Reachable Inconclusives
-----------------------------------------------------------------------------
Branch                            183            41           120            22
Condition                          88            11            67            10
Expression                        290            66           201            23
FSM                                34             5            17            12
  States                            8             1             3             4
  Transitions                      26             4            14             8
Statement                         279            51           158            70
Toggle                           1430           498           869            63
Coverbin                           15             6             3             6
-----------------------------------------------------------------------------
Total                            2319           678          1435           206
-----------------------------------------------------------------------------


---------------------------------------
Property Summary                  Count
---------------------------------------
Assumed                               0
Inconclusive                          0
---------------------------------------
Total                                 0
---------------------------------------


--------- Process Statistics ----------
Elapsed Time (s):                   462
--------------- vgill:0 ---------------
Total CPU Time (s):                 461
Memory Used (MB):                   449
---------------------------------------
```

- covercheck_design.rpt
  - # of clks, BB's, FSMs, RAMs, etc.
- covercheck_compile.log
  - compile errors/warnings

© Mentor Graphics Corp.

CoverCheck GUI Analysis View

Questa Verify (.../mark/testcases/demos/V10.3/qft_demo/demos/covercheck/log_cc/covercheck_verify.db)

File View Compile CDC Formal Tools Reports Logs Layout Window Help

AutoCheck | Layout Last Settings

Design

Instance
- pci_bridge32 (8)
  - configuration (8)
  - input_register
  - output_backup
  - parity_checker (5)
  - pci_io_mux (53)
  - pci_resets_and_inter
  - pci_target_unit (5)
  - wishbone_slave_unit

Project | Library

Details

```
Type        : Condition
Result      : Unreachable
Module      : pci_wb_slave
Instance    : wishbone_slave_unit.wishbone_slave
Location    : /zin/testcases/mark/testcases/demos/V10.3/qft_
Item Number: 1
Row Numbers : 1,2
File        /zin/testcases/mark/testcases/demos/V10.3/qft_d
Line        886 Item    1          (burst_transfer || ( 1
     Rows:  FEC Target            Matching input patterns
 ---------  --------------------  -------------------------
  Row   1:  burst_transfer_0      { 01 }
  Row   2:  burst_transfer_1      { 1- }
  Row   3:  alligned_address_0    { 00 }
  Row   4:  alligned_address_1    { 01 }

NOTE:
  * Order of matching input pattern values: {burst_transfe:
```

qft_demo/pci/rtl/verilog/pci_wb_slave.v - Default

```
CC  Ln#
    880        del_done          = do_ccyc_comp &&
               0
    881        del_in_progress   = do_ccyc_comp &&
               0
    882    `endif
    883
Xs  884    n_state          = S_IDLE ; // next stat
               0
    885
X   886    if ( burst_transfer || -alligned_addres
    887    begin
Xs  888        err = 1'b1 ;
    889    end
XB  890    else
    891    begin
    892        `ifdef HOST
    893        if ( do_ccyc_req || (ccyc_hit && -
               0                 0
    894        begin
    895            rty = 1'b1 ;
```

Cover Checks

Type	Module	Name	Instance	File	Line	ID
Condition...	pci_wb_slave		wishbone_slave_unit....	pci_wb_slave.v	864	FCC_28692
Condition...	pci_wb_slave		wishbone_slave_unit....	pci_wb_slave.v	864	FCC_36884
Condition...	pci_wb_slave		wishbone_slave_unit....	pci_wb_slave.v	864	FCC_45076
Xc Condition...	pci_wb_slave		wishbone_slave_unit....	pci_wb_slave.v	886	FCC_19796
Xc Condition...	pci_wb_slave		wishbone_slave_unit....	pci_wb_slave.v	886	FCC_27988
Xc Condition...	pci_wb_slave		wishbone_slave_unit....	pci_wb_slave.v	926	FCC_48260
Xc Condition...	pci_wb_slave		wishbone_slave_unit....	pci_wb_slave.v	926	FCC_56452
Condition...	pci_wb_slave		wishbone_slave_unit....	pci_wb_slave.v	1014	FCC_63332
Condition...	pci_wb_slave		wishbone_slave_unit....	pci_wb_slave.v	1014	FCC_38756
Xc Condition	pci_wb_tpram (12)					
Xc Condition	pci_delayed_sync (16)					

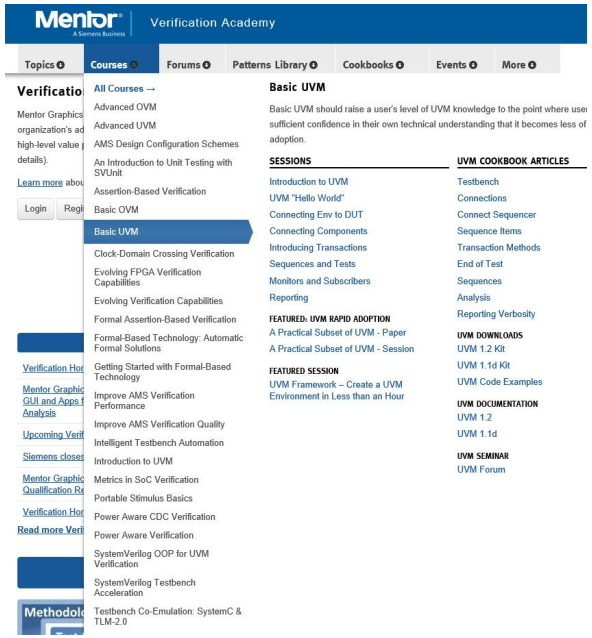
Transcript | Cover Checks | CoverGroup Checks

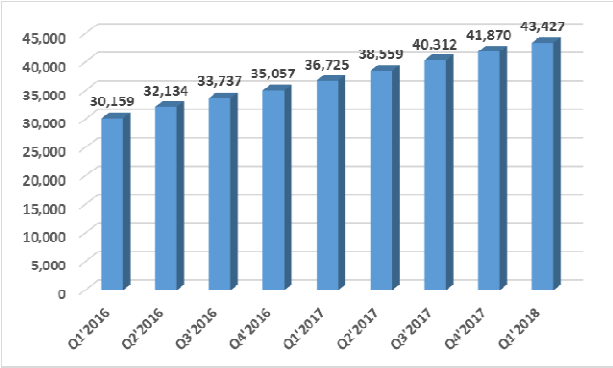
Filters in use: 0 | <No Context>

© Mentor Graphics Corp.

Verification Academy
The most comprehensive online resource

- 27 Online Courses
  - FPGA Verification
  - SystemVerilog
  - Basic & Advanced UVM
  - CDC & Formal
  - Many more..
- Verification Cookbooks
  - UVM
  - Coverage
- Discussion Forums
  - More than 43,000 Members
- Patterns Library





www.verificationacademy.com

© Mentor Graphics Corp.

Mentor®

A Siemens Business

www.mentor.com