Course 15

Assertions

Rev. Feb 20, 2012

Applies to Active-HDL 9.1 and later

Table of Contents

Assertion Support in GUI 3

Editing Source Files 3

Compilation 3

- Compiling Single File 3
- Compiling Unbound Files 3
- Compiling Bound Files 4
- Compilation Options 4

Initializing Simulation 7

Debugging 8

- Assertion messages in Console 8
- Assertions in Hierarchy Viewer 8
- Assertions in Assertion Viewer 9

Assertion Support in GUI

Active-HDL supports three assertion languages: OVA, PSL ant the assertion subset od SystemVerilog.

Editing Source Files

The HDL Editor supports auto-complete for OVA, PSL and SystemVerilog assertions.

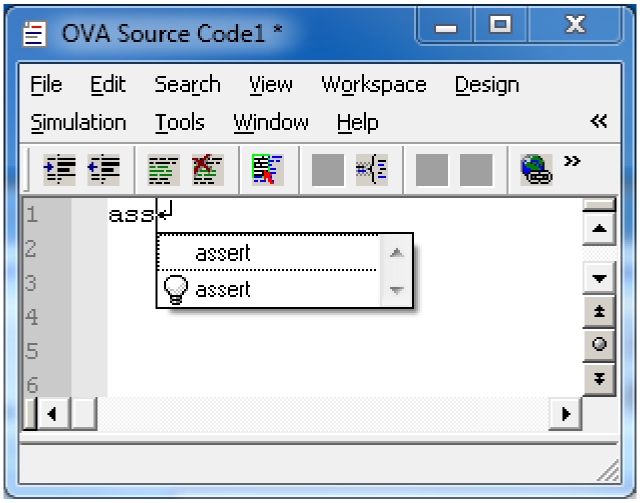


Figure 1 Auto-complete in HDL Editor.

Compilation

External files with assertion units are treated as regular HDL files and must be compiled to the design library. If an assertion unit is bound to a Verilog module or a VHDL entity, it must be compiled in one run, together with the module or the entity. Otherwise it can be compiled separately.

Compiling Single File

Selecting an OVA file in the Design Browser and choosing the Compile option from the context menu invokes a command in the Console (several arguments were omitted for clarity):

alog -work {library_name} -ova {file.ova}

The same functionality is available for PSL and SystemVerilog files.

Compiling Unbound Files

You can compile a few unbound files together (OVA, PSL, SystemVerilog, Verilog, VHDL, etc). To do so, select files in the Design Browser, open the context menu and select the **Compile** option. This will run compilation with different commands (several arguments were omitted for clarity):

```
alog -work -dbg -ova {receiver.ova}
acom -work -dbg {receiver.vhd}
```

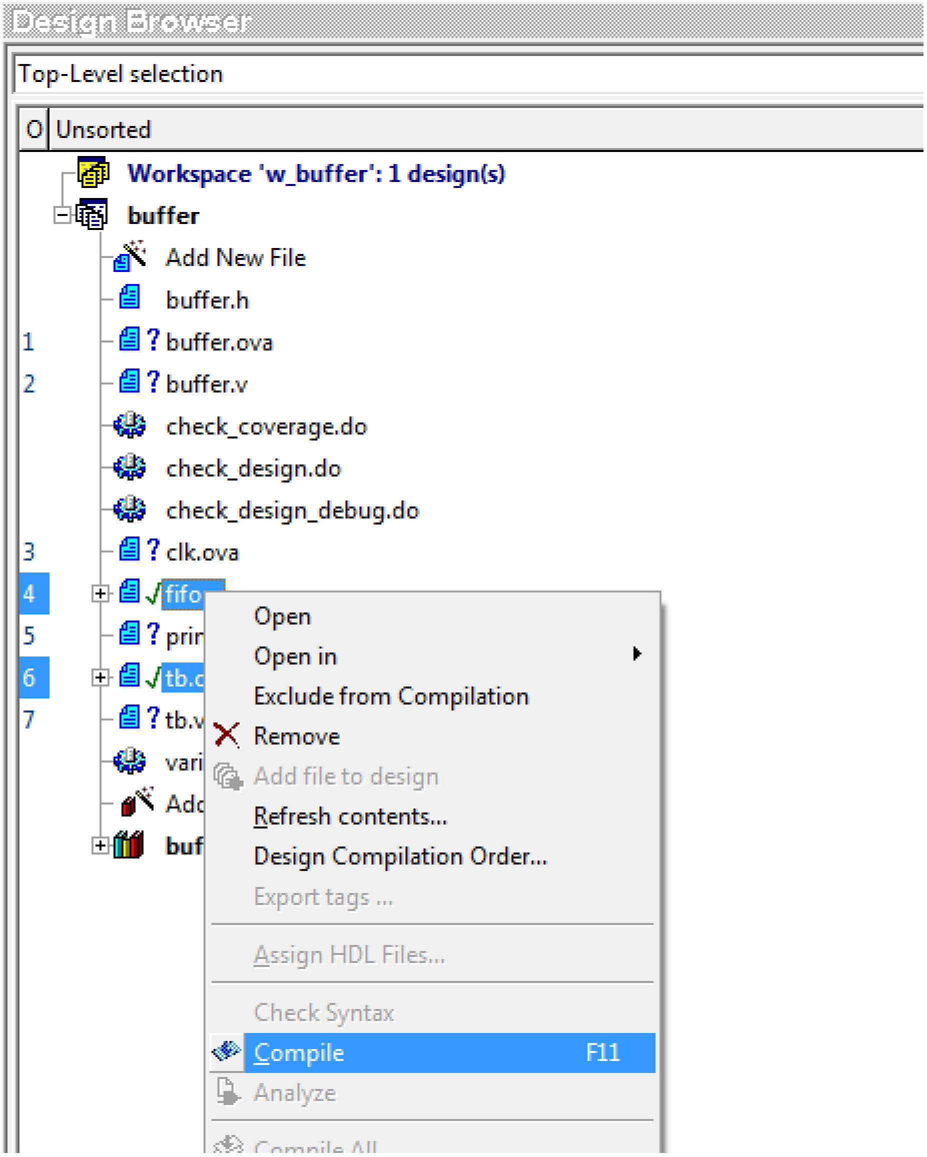


Figure 2 Compiling unbounded files in the Design Browser.

Compiling Bound Files

To compile files that need to be compiled in one invocation of the compiler, please invoke compilation command in the Console:

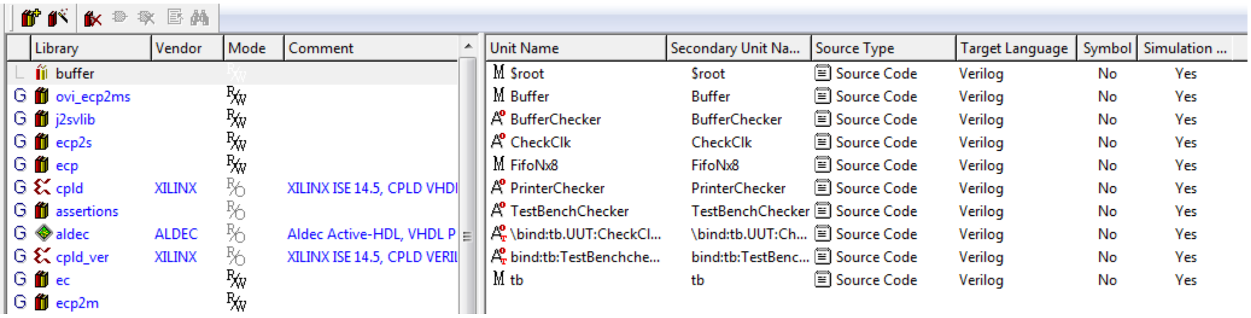
```
alog -work buffer -dbg {tb.v} -ova {buffer.ova}  {buffer.v}  -ova {printer.ova}
```

Compilation Options

Compilation options related to assertions can be set on **Design Settings** dialog box | **Compilation** | **VHDL** | **Assertions** category or the **Design Settings** dialog box | **Compilation** | **Verilog** | **Assertions** category

Compiled assertions are visible in the Library Manager and in the Design Browser when you view contents of the design libraries.

Icons in the Library Manager indicate compiled assertion units (see User Guide | Active-HDL Tools | Library Manager | Library Contents for more details).

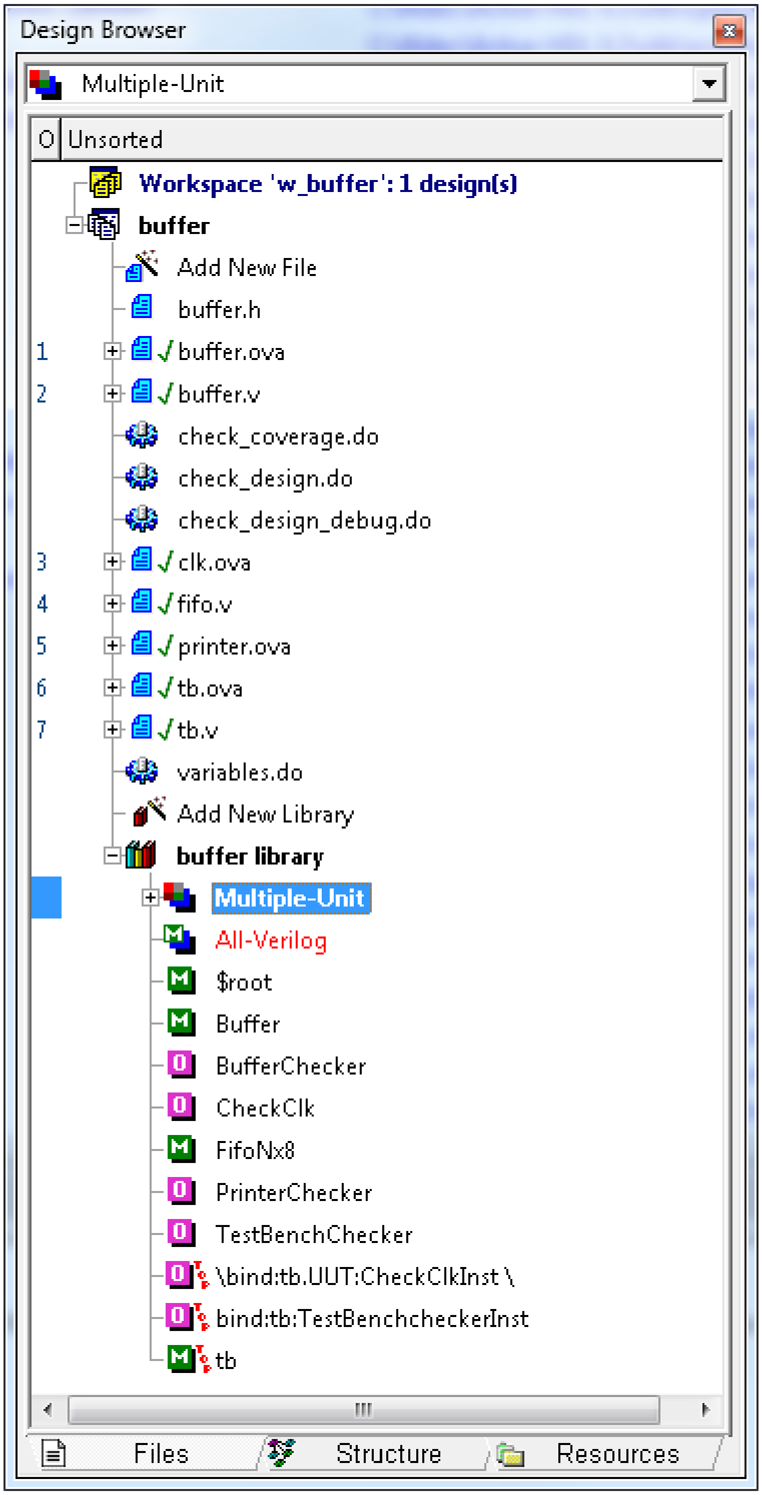


	Library	Vendor	Mode	Comment
	buffer		RW	
G	ovi_ecp2ms		RW	
G	j2svlib		RW	
G	ecp2s		RW	
G	ecp		RW	
G	cpld	XILINX	RO	XILINX ISE 14.5, CPLD VHDI
G	assertions		RO	
G	aldec	ALDEC	RO	Aldec Active-HDL, VHDL P
G	cpld_ver	XILINX	RO	XILINX ISE 14.5, CPLD VERIl
G	ec		RW	
G	ecp2m		RW	

Unit Name	Secondary Unit Na...	Source Type	Target Language	Symbol	Simulation ...
$root	$root	Source Code	Verilog	No	Yes
Buffer	Buffer	Source Code	Verilog	No	Yes
BufferChecker	BufferChecker	Source Code	Verilog	No	Yes
CheckClk	CheckClk	Source Code	Verilog	No	Yes
FifoNx8	FifoNx8	Source Code	Verilog	No	Yes
PrinterChecker	PrinterChecker	Source Code	Verilog	No	Yes
TestBenchChecker	TestBenchChecker	Source Code	Verilog	No	Yes
\bind:tb.UUT:CheckCl...	\bind:tb.UUT:Ch...	Source Code	Verilog	No	Yes
bind:tb:TestBenchche...	bind:tb:TestBenc...	Source Code	Verilog	No	Yes
tb	tb	Source Code	Verilog	No	Yes

Figure 3 Compiled assertions in the Library Manager.

Icons in the Design Browser indicate compiled assertion units (see User Guide | Active-HDL Tools | Design Browser | Design Libraries for more details).



Design Browser

Multiple-Unit

O | Unsorted

```
Workspace 'w_buffer': 1 design(s)
  buffer
      Add New File
      buffer.h
1     buffer.ova
2     buffer.v
      check_coverage.do
      check_design.do
      check_design_debug.do
3     clk.ova
4     fifo.v
5     printer.ova
6     tb.ova
7     tb.v
      variables.do
      Add New Library
      buffer library
          Multiple-Unit
          All-Verilog
          $root
          Buffer
          BufferChecker
          CheckClk
          FifoNx8
          PrinterChecker
          TestBenchChecker
          \bind:tb.UUT:CheckClkInst \
          bind:tb:TestBenchcheckerInst
          tb
```

Figure 4 Compiled assertions in the Design Browser.

Initializing Simulation

You can initialize simulations with the assertion top-level module using the **Initialize Simulation** option from the context menu in the Design Browser. See User Guide | Active-HDL Tools | Simulation | Initialization of Simulation for details. After simulation initialization assertions are visible in the Hierarchy Viewer and in the Assertion Viewer.

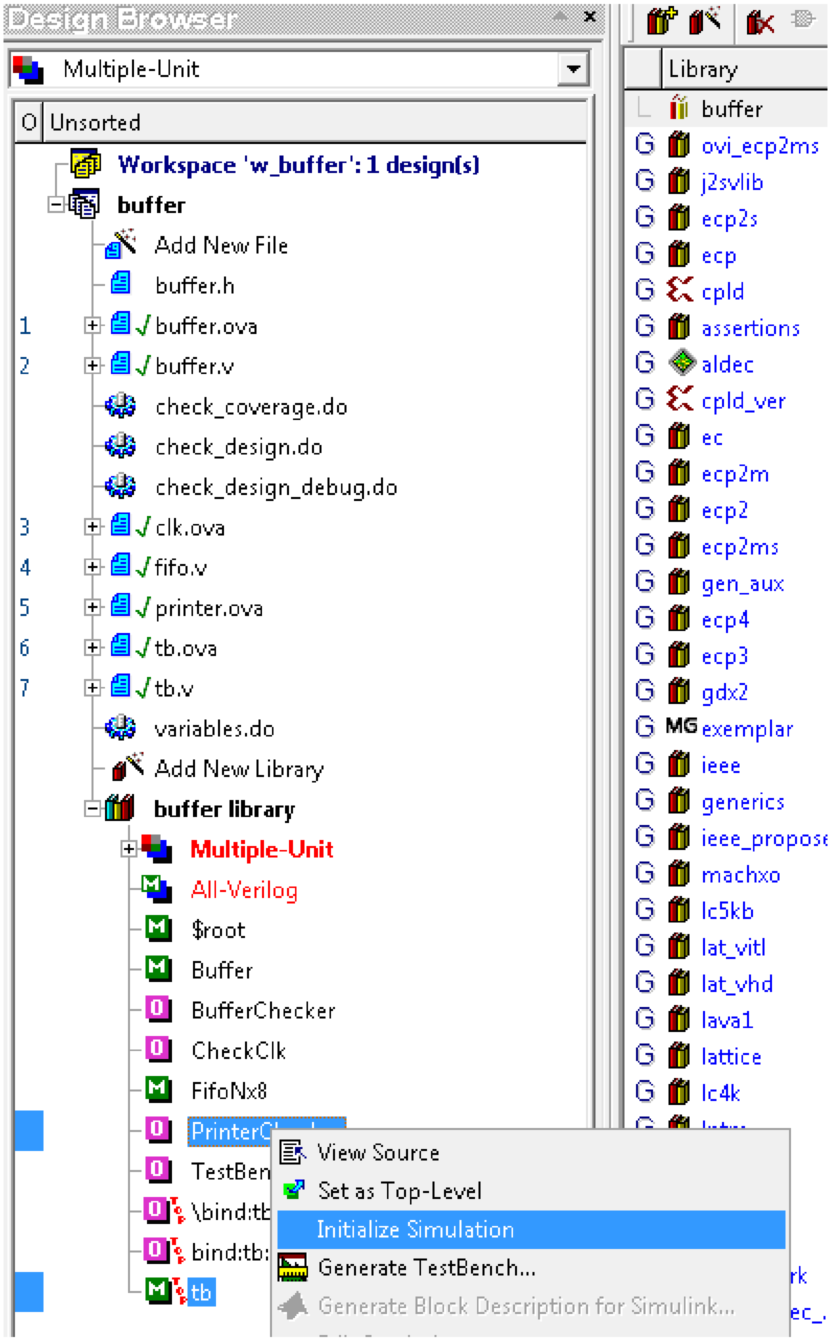


Figure 5 Initialize simulation with the assertion top-level module.

This will initialize simulation with the following command (several arguments were omitted for clarity):

```
asim PrinterChecker tb
```

If an assertion unit is bound to a module or an entity it is enough to initialize the design top level $root only. The $root module is created and updated automatically after each invocation of the Verilog compile. The $root module contains all Verilog top-level modules and all assertion top-level modules. Initializing simulation will load the simulation top-level HDL unit and all assertion top-level units (i.e. units bound to instances).

This will initialize simulation with the following command (several arguments were omitted for clarity):

```
asim +access +r {$root}
```

Simulation options related to assertions can be set on the Design Settings dialog box | Simulation | Assertions category.

Debugging

Assertion messages in Console

```
# KERNEL: Error: Assertion 'BufferChecker.CheckSaveToPrinterScheme' FAILED at
time: 2,870ps (144 clk), ./buffer.ova(100), scope: tb.UUT.BufferCheckerInst,
start-time: 2,570ps (129 clk)
```

A double-click on a message about VHDL, OVA, PSL or SystemVerilog assertion opens the HDL Editor and shows the assertion statement.

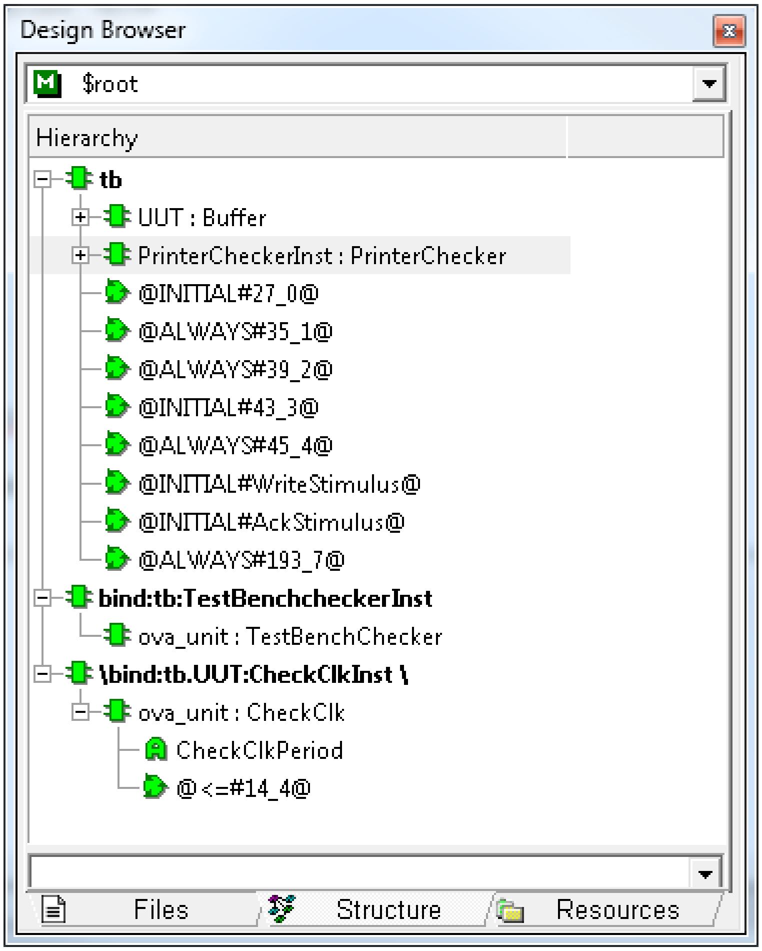
You can enable or disable printing messages to the Console. Debug options related to assertions can be set on the Design Settings dialog box | Simulation | Assertions category.

Assertions in Hierarchy Viewer

After initializing simulation, assertions are displayed in the design hierarchy.

The OVA, SystemVerilog, and PSL assert construct is marked with the capital A icon (A or A).

The OVA and PSL units are displayed as the HDL blocks 🟩 🟦.



- tb
  - UUT : Buffer
  - PrinterCheckerInst : PrinterChecker
  - @INITIAL#27_0@
  - @ALWAYS#35_1@
  - @ALWAYS#39_2@
  - @INITIAL#43_3@
  - @ALWAYS#45_4@
  - @INITIAL#WriteStimulus@
  - @INITIAL#AckStimulus@
  - @ALWAYS#193_7@
- bind:tb:TestBenchcheckerInst
  - ova_unit : TestBenchChecker
- \bind:tb.UUT:CheckClkInst \
  - ova_unit : CheckClk
    - CheckClkPeriod
    - @<=#14_4@

Figure 6 Assertions in Design Hierarchy.

See User Guide | Using Active-HDL | Assertions in Design Hierarchy for details.

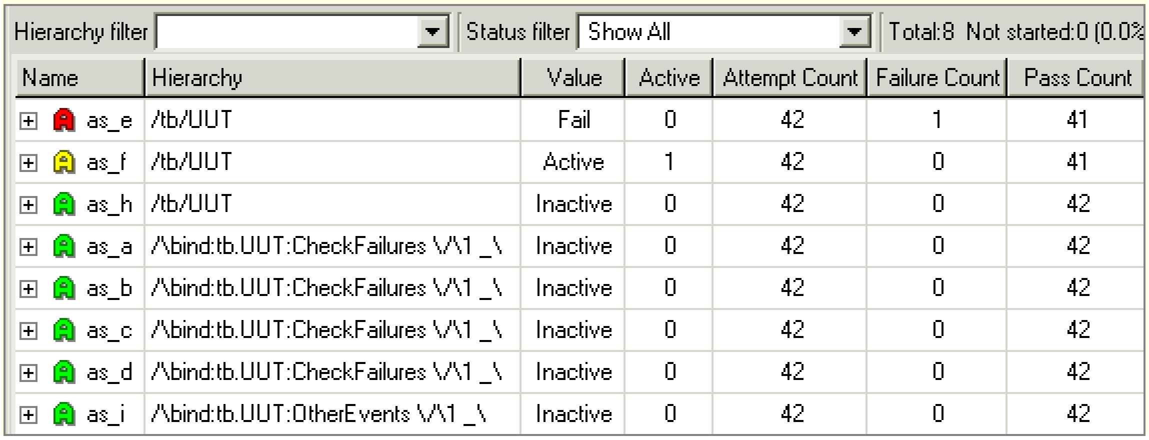
Assertions in Assertion Viewer

The **Assertion Viewer** window shows two types of OVA, PSL, and SystemVerilog objects:

- assertions
- cover statements (covers for short)

Statistics gathered for these objects during simulation are presented on separate tabs. The **Assertions** tab displays statistics for assertions while **Covers** for the cover statements used in a design. The information provided by the viewer includes names, signals used in each assertion/cover, values, execution counts etc.

The Assertion Viewer can be started after the initialization of simulation by using the **Assertion Viewer** option from the **View** menu or the `view assert` macro command. The **Assertion Viewer** window is shown below.



Hierarchy filter: (empty) | Status filter: Show All | Total:8 Not started:0 (0.0%

Name	Hierarchy	Value	Active	Attempt Count	Failure Count	Pass Count
⊞ 🟥 as_e	/tb/UUT	Fail	0	42	1	41
⊞ 🟨 as_f	/tb/UUT	Active	1	42	0	41
⊞ 🟩 as_h	/tb/UUT	Inactive	0	42	0	42
⊞ 🟩 as_a	/\bind:tb.UUT:CheckFailures \/\1 _\	Inactive	0	42	0	42
⊞ 🟩 as_b	/\bind:tb.UUT:CheckFailures \/\1 _\	Inactive	0	42	0	42
⊞ 🟩 as_c	/\bind:tb.UUT:CheckFailures \/\1 _\	Inactive	0	42	0	42
⊞ 🟩 as_d	/\bind:tb.UUT:CheckFailures \/\1 _\	Inactive	0	42	0	42
⊞ 🟩 as_i	/\bind:tb.UUT:OtherEvents \/\1 _\	Inactive	0	42	0	42

Figure 7 Assertions in the Assertion Viewer.