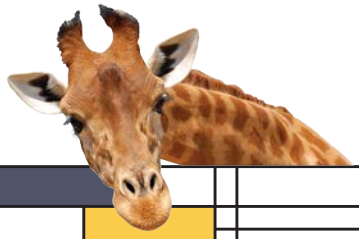


head and shoulders above the rest...

Verific® Design Automation

Highlights

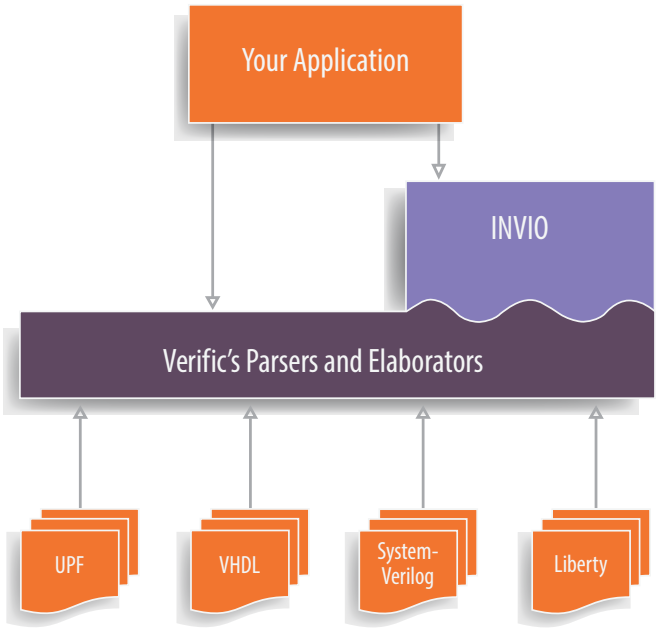
- INVIO lies on top of Verific's parsers for SystemVerilog, VHDL, UPF, and Liberty
- Extends Verific with a platform focus on usability and rapid development
- Language agnostic by using the same APIs for VHDL and SystemVerilog
- Easy to use Python and C++ APIs
- Interoperable with existing Verific applications
- Custom GUI builder
- Application packager

Verific with INVIO

INVIO is a layer on top of Verific's existing SystemVerilog, VHDL, UPF, and Liberty parsers. It extends Verific with a concise yet productive set of platform APIs focused on usability and rapid development.

INVIO objects can be directly referenced by connectivity to other objects at the HDL level, allowing very quick and natural navigation of designs and test benches. All INVIO objects (instances, signals, processes, etc.) can be referenced by their full hierarchical names and all instance names are fully resolved, enabling a more natural design view. An intuitive 'regex' search on all attributes is included.

All references (signals, classes, interfaces, etc.) are fully resolved by INVIO, enabling simple automatic support for complex SystemVerilog including hierarchical references, classes, virtual functions, and UVM. INVIO supports user attributes and automatically enables powerful search on new user attributes.



The INVIO in-place modification API enables development of tools that modify existing RTL designs while preserving the look and feel of the original text file (indentation, comments, whitespace, etc.).

Verific with INVIO advantages

- Complete object connectivity layer

```
local_reset = get_signals("top_i/ent_i/sw_rstb")[0]
drivers     = local_reset.inputs
sinks       = local_reset.outputs
```

- Instance-based design hierarchy

```
# find all signals across the entire hierarchy
get_signals(flags=HIER)
# find all signals across a local hierarchy
set_current_instance("top_i/enet_i/fifo_i")
get_signals(flags=HIER)
```

- Full regex search on HDL and attributes

```
for cls in get_classes(filter='base_name=~"b[aeo]t"', flags=HIER|REGEXP):
    if cls.base_class: # If the class has a base class
       print "found class"
```

- Automatic scope resolution

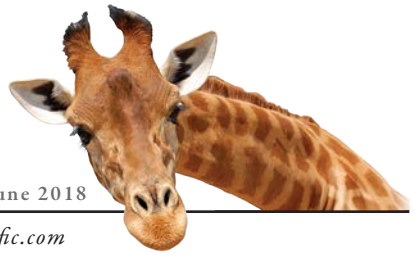
```
# Example RTL: "assign U0.U1.x = 1;"
t_assign = get_assigns("top_i/usb_i/*")[0]
hier_ref = t_assign.lhs
resolved_signal = hier_ref.referent
```

- User attributes with full native search

```
inst = get_instances("top_i/pcie_i")[0]
# Set a user-attribute on the INVIO instance object
inst.clock_domain = "clk250"
# In this case, find all instances on a given clock domain
get_instances(filter='clock_domain == "clk250"')
```

- RTL modification with minimal impact

```
# Get all instantiations of "DFF"
for inst in get_instances(filter='module_name="DFF"', flags=HIER):
    replace_module_of_instance(inst, new_module)
# Commit modifications to file
write_modifications()
```



Verific
Design Automation

June 2018