

Click the “Cancel” button in the report window

Step 8. Write a gate-level netlist

We have so far synthesized a sample design and analyzed the results using powerful capabilities of NaviGates. Now let’s write out the netlist for this design.

Right-click on “cpu” in hierarchy viewer and select “Write Module” from the pop-up menu.

Select the format of netlist as Verilog or VHDL.

By default, the whole hierarchy of the current design will be written.

Let’s name the netlist in “cpu.gates”.

Type in the name in the field under “Files”.

Click the OK button

You can now see the Verilog or VHDL netlist by reading it into HDL editor of NaviGates.

Click on the “Hdl” tab of schematic viewer. Click “Open” to pop up “Edit HDL file” window.

Under “Files” change the filter to “*.gates”.

This will show the netlist we just wrote.

Double-click on “cpu.gates”.

The netlist file can be viewed in the Hdl editor.

Summary

This demonstration walked you through a simple synthesis flow using NaviGates. We looked at features such as

Ease of editing HDL files

Correlation between HDL and gate structures

Easy to use tracing of worst paths, worst endpoints and fan-in/ fan-out cones

Various reporting capabilities

BuildGates also has key features such as Time Budgeting, built-in Fast Timing Analysis and atomic synthesis transforms. Please look at the other demonstrations for these features.

Click the Cancel button to get rid of the pop up panel.

Step 6. Highlight the critical path

Put the cursor in the schematic window.

Use the right mouse button to bring up the menu. Select “Worst Path”.

The worst slack path in the design will be highlighted.

Double-click on one of the hierarchical instances in the worst path.

Zoom in on the highlighted gates.

You see the arrival times on the pins of the instances. With the timing numbers on the path itself, it is very easy to understand the cause of delays and fix any issues.

Use the right mouse button to bring up the menu. Select “Worst Endpoint / 5”.

This will highlight the worst five endpoints in the current design with the corresponding timing numbers.

Select “Clear” from menus at the top of schematic window to clear the highlighting.

Click on one of the gates, use the right mouse button to bring up the menu. Select “Fan-in cone”.

This will highlight the fan-in cone to that particular pin.

Zoom in on the highlighted gates in the cone.

You can see the arrival times from various paths.

NaviGates allows user to easily track worst paths, worst endpoints in the design. You can also trace fan-in and fanout cones of logic. The timing numbers for all the arrivals are right there on the instances making it easier to locate problem spots in the design.

Step 7. Write out the various reports

Right-click on the “cpu” in module hierarchy and select “Set current and top”.

This will make “cpu” the current module.

Select the “Reports/Report Timing” menu from the main menu.

Click “Report” button at the bottom of the Report window.

Now the generation of timing report is initiated.

The timing report describes in detail the worst path in the design. It identifies the begin and endpoint, instance by instance delays, logic arcs and the cumulative delays.

Click the “Cancel” button in the report window to get rid of the window.

Select the “Reports/Report Area” menu item from the main menu.

Click the “Report” button to initiate area report.

The area report details the module by module area and the wireload model used for each of the modules.

address[4:0] output bus to the right hand signal list.

Enter “5” as required time for these signals and click “Set”.

You have now declared that the content of address[4:0] bus output is required to be stable by 5ns with respect to the leading edge of ideal clock.

Apart from applying constraints through pull down menu, one can do the same by writing a script. Let's look at an example of such a script.

Click on the “Tcl” tab of schematic viewer.

Click on “Open” button from the menu bar that shows up. Double-click on the “constraints.tcl” file.

The constraints.tcl appears in the window. The script removes all of the old constraints on the design, defines the clock, applies constraints to inputs as well as outputs and finally declares all the paths from “reset” input pin as false paths.

Click on “Save+Source” button from the menu bar.

This will execute this script and now the design has the constraints defined in this script.

Click on the “Schematic” tab of schematic viewer.

Click on one of the ports, say “mem_rd” Then right-click the mouse button for pop-up menu.

Select “Show Constraints” in the popup menu.

A window showing all the constraints on the port pops up.

This is an easy-to-use interactive way of looking at constraints on ports of the design.

Step 4. Optimize the design

Now that the constraints for the design are defined, let's synthesize the design.

Select the “Tools/Optimize” menu item from the main menu bar.

The “Optimize Options” window that pops up and shows the default optimization settings.

Click “Optimize” button at the bottom to start the synthesis with these constraints.

BuildGates performs various optimization steps during this phase. It goes through the steps of propagating constants, removing redundancies, structuring, mapping and then optimization. During optimization, BuildGates reports information such as design area, the worst slack and the path endpoint it is working on.

Step 5. Navigating through the optimized schematic

Double-Click into a hierarchical module.

This takes you deeper into the hierarchy.

Go up the hierarchy by dragging the mouse upward and to the left.

Zoom in by dragging the mouse down-right.

Zoom out by holding the left mouse button down and dragging the mouse up-right.

Positioning the cursor over any cell and holding down the right mouse button.

Select “Show Properties” from the menu that pops up.

This allows you to view the properties of any cell in the design.

Step 2. Convert the RTL into a technology independent netlist

From the main menu bar, click-select the “Tools/Build Generic” menu item.

This step will first link the whole design, will report the case statement statistics and register inferences found in the RTL. The design is now synthesized into a technology independent netlist form. The hierarchy tree of the design will pop-up in the Module Hierarchy Viewer. The hierarchy viewer shows the instance name, the module name and the form of the design (g= generic, o= structured, m = mapped).

Double-click on the “cpu” item at the root of the hierarchy tree.

This shows the schematic of the design in the schematic viewer.

Drag the left mouse button (move cursor down/right) over the upper right quadrant of the schematic.

You're now zoomed in on the MUX gates.

Right mouse-clicking on one of the selected MUX gates, select “Show HDL/New Window”

The line in the RTL code that generated the MUX gate is highlighted. This way you can view the gates and associated HDL constructs.

Click on the “Close” button in the HDL window that had popped up.

NaviGates allows the user to trace the constructs in the generic netlist, back to RTL. The designer can now experiment new constructs in the HDL and easily correlate the effects in the netlist.

Step 3. Apply constraints to the design

Click-select the “Constraints/Ideal Clocks” from the main menu bar.

The “Set Ideal Clocks” panel will pop up.

Enter “clock” in the Name field of the panel, 0 in the “Lead Edge” field, 5 in the “Trail Edge” field and 10 in the Period field.

Click “ADD” button at the bottom of the panel.

You have now created an ideal clock of period 10 ns with leading edge at 0 ns and trailing edge at 5ns. All the data arrival and required times will be constrained using this ideal clock.

Click “Cancel” button to get out of the panel.

Click-select “Constraints/Data Arrival Constraints” from the main menu bar.

The “Set Data Arrival Time” panel will pop up.

Click the “All” button.

This will transfer all the input signals to the right hand signal list.

Enter “3” as arrival time for these signals and click “Set”.

You have now declared that all the inputs are arriving at 3ns with respect to the leading edge of ideal clock.

Click-select “Constraints/Data Required Constraints” from the menu bar.

The “Set Data Required Time” panel will pop up.

Drag select from “address[0]” to “address[4]”. Click the “Add” button to transfer the

Standard Flow Demonstration

This step by step demonstration will take a user through a standard synthesis flow and will introduce the user to certain commonly used menus and analysis tools in NaviGates.

The steps followed in the demonstration are

- 1) Read in the libraries and the RTL of a design.
- 2) Convert the RTL into a technology independent netlist
- 3) Apply constraints to the design
- 4) Optimize the design
- 5) Analyze the results through reports and schematic viewer
- 6) Write out the design netlist

Step 1. Read in the libraries and the RTL of a design.

Click on the “TECH” icon on the icon menu bar.

The “Read ALF file” menu will pop up.

Select the lca300k.alf library and click on “OK” button

This will read in the LSI lca300k technology library and will set it to be the target technology.

Click on the “HDL” icon on the menu bar.

A “Read HDL files” menu will pop up.

Select either Verilog or VHDL button at the bottom of the panel.

The HDL files for that language will appear in the “Files” list.

**Use the left mouse button to drag over the entire list of HDL files in the “Files” list.
Keeping the pointer in the selected files pane, click the middle mouse button.**

This transfers the files to the lower file list.

Click on the “OK” button at the bottom of the “Read HDL files” panel.

This step will read in all the RTL files into BuildGates. A syntax error will be detected and marked in red color in the ac_shell command window.

Triple-click on the first “ERROR: ...“ error line in the ac_shell command window.

The faulty HDL will pop up in the HDL browser.

Verilog: Edit the ‘@@’ and make it a single ‘@’ in the HDL browser.

VHDL: Edit the ‘entity entity’ and make it a single ‘entity’ in the HDL browser.

Click on the “Save+Parse” button.

This will save the corrected HDL and read it back into BuildGates.

NaviGates allows user to interactively change the HDL without having to quit the synthesis process. Further, syntax errors are flagged and fault in the corresponding HDL is pinpointed. This improves user PRODUCTIVITY. The user does not have to chase the problem through multiple files, just to fix a simple problem.