EFFICIENT DEBUGGING Using PROBE

You can easily view internal signals by routing them to device pins.

by Davin Lim, Software Technical Marketing Engineer, Xilinx, Davin.Lim@xilinx.com

The PROBE capability within the FPGA_Editor application (included in all Xilinx Foundation Series and Xilinx Alliance Series development systems), allows you to quickly identify and route any internal signal to an unused I/O pin. Once routed to the pin, you can then view that signal's real-time activity using normal lab test equipment such as logic/state analyzers and oscilloscopes. PROBE is very easy to use and it makes your lab debug cycles as quick and efficient as possible.

PROBE gives you a simple, natural way to observe any internal signals in your design and analyze their switching behavior with essentially no impact on design performance. The signals are viewed in real-time, so you are sure to have the most accurate picture possible on how your design is functioning in it's actual operating form and environment.

Your HDL design effort only needs to focus on the device's primary intended function because all of PROBE's power can be accessed automatically during the lab-debug phase of your project design cycle—there is no need to guess before lab debugging which signals you wish to analyze. PROBE easily guides you through the debug phase, so you can pick which signals need to be observed at any time during the process, and changes to the signal list can be made instantaneously.

Using PROBE

The PROBE function can be used either in manual or automatic mode.

Manual Mode

As shown in Figure 1, using PROBE is as simple as loading the design into FPGA_Editor and clicking on the "PROBES" button on the User Toolbar.

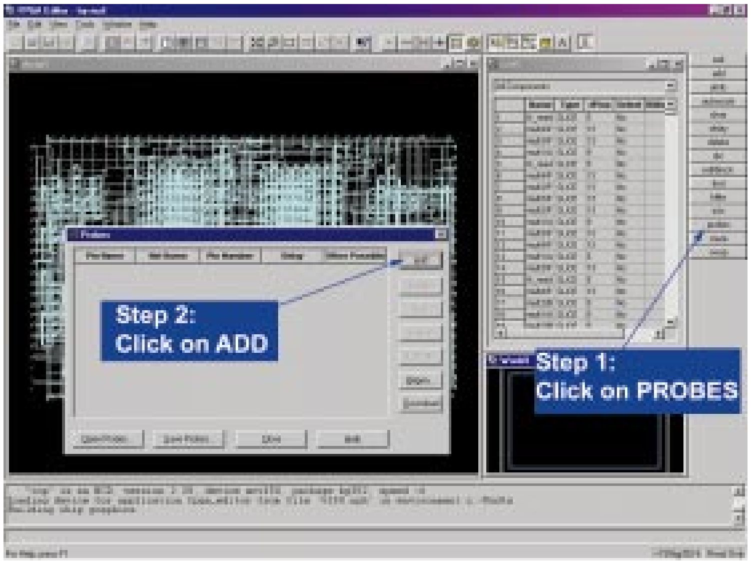


Figure 1 - How to access PROBE.

Once the PROBES dialog box appears, all you need to do is pick the signal to be analyzed and assign it to an unused I/O pin, as shown below in Figure 2.

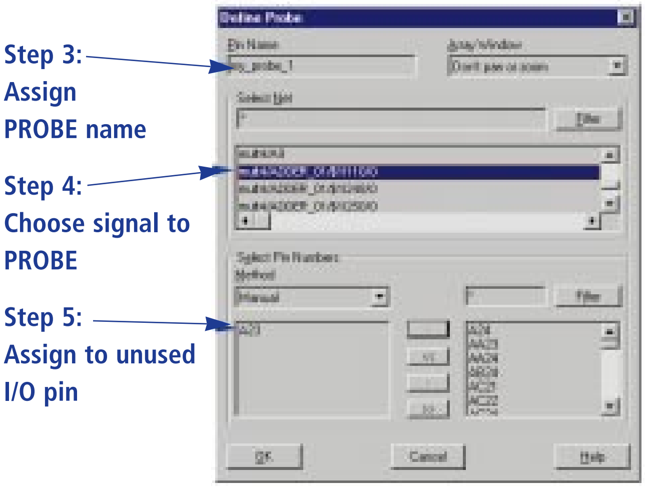


Figure 2 - Assigning signals to probes.

This is the most commonly used mode because most of the time you will know which I/O pins are best to use for connection to external logic analysis equipment. Usually, and especially in cases where ball-grid array (BGA) packages are involved, you will connect test points on your PC board to specific FPGA device pins. In such cases, you'll know exactly which unused I/O pins are most convenient to use for PROBE connections, and the manual mode is the best choice.

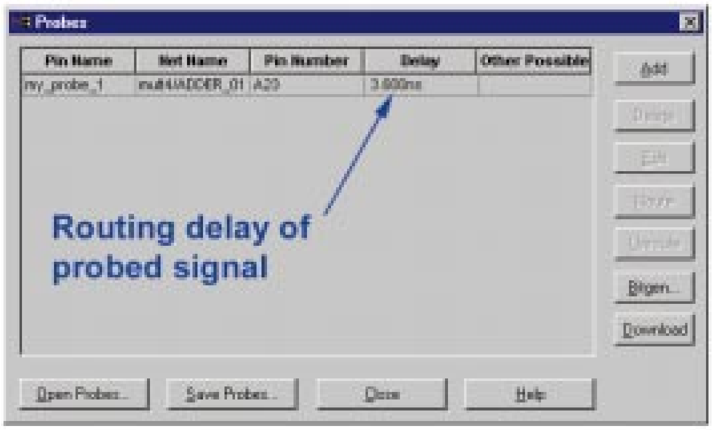
When reserving the I/O pins you wish to use as test points, you can increase the effectiveness of PROBE by selecting at least one I/O pin on each side of the FPGA (as viewed in the Floorplanner or FPGA Editor). This way, no matter where on the FPGA the internal signal to be probed exists, there will be a test I/O pin located nearby.

Automatic Mode

In automatic mode, PROBE automatically chooses an unused I/O pin that will incur the least amount of routing delay when connected to the signal you want to monitor. To use the automatic mode, simply select "Automatic" under the "Method" selection box. Once the signal, pin list, and pin selection method have been chosen, pressing the OK button will complete the PROBE operation.

Displaying PROBE Results

The Probe list window lists all signals that have been assigned to PROBE I/O pins. This list can be sorted by any of the headings simply by clicking on the desired column's heading. It will often be important for you to know how much routing delay is incurred to route the signal to the I/O pin. This delay amount is listed in the PROBE list



Pin Name	Net Name	Pin Number	Delay	Other Possible
my_probe_1	mult4/ADDER_01	A20	3.600ns	

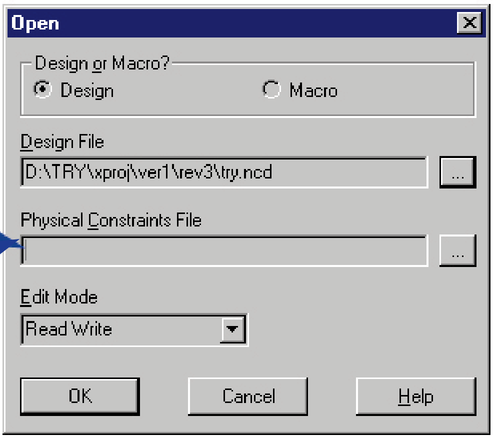
Figure 3 - Routing delay of probed signal.

under the “Slack” heading. In the example shown in Figure 3, there is 3.6ns of routing delay to route the signal “mult4/ADDER_01” to the external I/O pin.

Controlling Probe Execution Time and Routing Delay

If your design has timing constraints which specify timing requirements for items such as clock periods and I/O timing, the default routing algorithms used by PROBE will account for those timing specifications and work to ensure that all timing constraints are still met after the probe signal has been inserted. FPGA_Editor accomplishes this by reading timing constraints from the design.pcf physical constraints file.

The amount of route delay on the actual probed signals is often not critical to the debug task at hand. As long as you know what the route delay is, you can account for that amount in the analysis process, and therefore it may not be necessary to always optimize the route delay to achieve minimum delay. If this is the case for the signals you are probing in your design, you can significantly quicken the PROBE execution process by simply ensuring that the .pcf file is not read in when the design is originally loaded into FPGA_Editor. This is accomplished by making sure the Physical Constraints File field is left blank, when completing the File->Open dialog box as shown in Figure 4.



Speed up PROBE routing time by loading design without Physical Constraints File (.pcf).

Make sure this field is blank.

Design or Macro?
● Design ○ Macro

Design File
D:\TRY\xproj\ver1\rev3\try.ncd

Physical Constraints File

Edit Mode
Read Write

Figure 4 - Omit the PCF file to speed routing times.

Conclusion

To quickly and effectively get through the lab-debug phase of your FPGA design, you will need tools that are fast, easy to use, and work with a minimum of fuss. The PROBE capability within the FPGA_Editor application directly addresses this need. PROBE provides you with direct access to the internal signals of your design, with the absolute minimum impact on the design’s performance and resource utilization.