2.1i FPGA Editor

XAPP401 (Version 1.0) October 13, 1999 Application Note

Summary/ Introduction

This application note presents the new, easier to use FPGA Editor and how it differs from the previous version of EPIC. For general FPGA Editor usage, refer to the FPGA Editor Guide. This application note will also cover how to return to EPIC type actions for zoom and pan actions.

FPGA Editor vs. EPIC

FPGA Editor Overview

The new FPGA Editor is written using Microsoft® MFCs allowing for Windows look and feel along with new functionality that EPIC did not support. The new functions allow for many ease of use features. Important new functionality was also added to enhance the editor. In addition to the following features listed, MFC controls like shift select for selecting contiguous list items and control select for selecting non-contiguous items in the list have been implemented.

Probing Internal Signals

FPGA Editor has taken the "`probe`" script and added it as a feature. The probe capability allows the user to tap into an internal signal and route it to an IOB for external analysis. FPGA Editor has automated the probe feature, allowing the user to specify the signal and whether or not the user wants to specify an IOB or let the tool pick the best IOB. The tool configures the IOBs and routes the signal out to the IOB automatically. A bitstream can be generated from the probe dialog. Downloading through the Hardware Debugger from the dialog box is also possible. This prevents the user from having to invoke other tools just to see the signal.

Probe can be accessed by selecting either the "**Probes**" button on the right side of the editor or from the **Tools** pulldown menu. To bring out an internal signal to an IOB simply enter a name for the package pin to be probed to, i.e., "DPO<3>", choose the internal signal you wish to probe from the list window, and enter the package pin number you wish to probe to. In just a few seconds Probe will route the signal to the selected pin (see Figure 1).

Perhaps one of the best uses of Probe is to give it a list of possible package pins. In a large, congested design, this can improve the odds of routing out internal signals. A very effective method of accomplishing this is to enter one or two package pins per side of the FPGA. Probe will try to route to each one of these, and pick the one with the shortest routing delay.

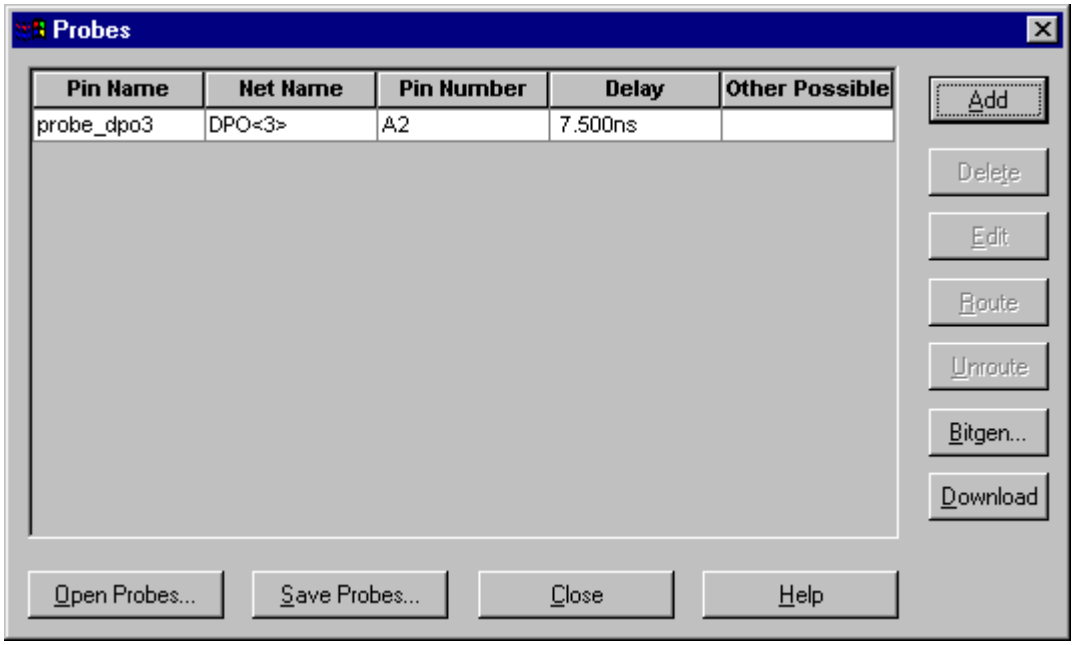

Figure 1: Probes Dialog

All New List Window Functions

The list windows have taken advantage of new graphics libraries to supply more design information to the user. Information is presented in a spreadsheet that allows the user to sort on a variety of columns. For instance with "**All Nets**" displayed, the user can sort the list of nets by fanout. This allows the user to see all of the high fanout nets together. See "Determining Net Skew" for evaluating skew on one of these nets. For example to replace the "**?**" in the "**Max Pin Delay**" column with a delay value, select the net(s) that you want the delay for and press the "**Delay**" button on the buttons down the right side. One may also create a button to do this for any selected net or nets, type the following in the command line:

```
button “net delays” “select net * ; delay”
```

To have this alias available each time FPGA Editor is invoked, add to your `fpga_editor_user.ini` as described in section "Editing the fpga_editor_user.ini Files". The "**Selected**" column allows the user to bring together a list of items that may not be contiguous when sorted by other columns. If the user has selected several items throughout the list, they can then sort the items by the "**Selected**" column to bring then all to the top of the list.

Opening Multiple Windows

There are several advantages for allowing multiple windows to be open at one time. Several window types allow this functionality including: Array, Block, List, and World. Having two List windows open can be useful in being able to see both nets and comps (components used by the design such as CLBs or IOBs) at the same time for following a path through its nets and components. Having two Array windows open allows the user to zoom into two different areas of interest while looking at path end points or performing manual routing.

Displaying Resources

EPIC used a Layer Visibility window to toggle on and off different resources. The FPGA Editor has changed to buttons for turning on and off resources. These buttons have two modes controlled by the "**Apply**" button (far right button) and are either dynamic or applied. While the "**Apply**" button is in the depressed state, the resource buttons are dynamic and toggle each type of resource as the resource button is pushed. For large designs where the redraw time is longer, for quicker redraws, make sure the "**Apply**" button is in the up position. Multiple layer buttons can be toggled before the "**Apply**" button is depressed, implementing all changes at once.

Analyzing Timing Paths

FPGA Editor can be used to analyze and highlight the same timing paths that the PAR and Timing tools report. To highlight a timing path, select **Tools-> Trace-> Setup and Run…**, OK the Trace dialog, select a constraint from the Trace Summary dialog, and get the Details, finally selecting a path and the **hi-lite** button from the Trace Errors dialog. The full path should be highlighted in the display window.

Determining Net Skew

With one or more nets selected, press the "**Attrib**" button on the right side column of buttons of the editor. Once the net dialog is open, select the"**Pins**" tab, then click in the "**Delay**" column to sort by delay. Look at the delays for the first and last loads and subtract the fastest delay from the slowest, this will give you the skew on the signal (see Figure 2).

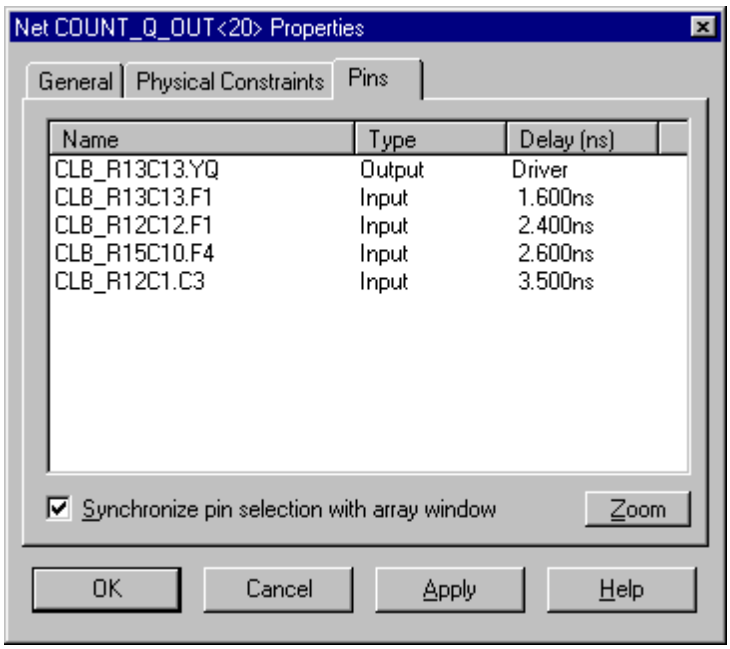

Figure 2: Determining Net Skew

Using the Logical Block Editor (LBE)

Components can be doubled clicked to enter the LBE. EPIC required a "SHIFT+Select". To edit the configuration of a component, the "**Begin Editing**" button near the left side of the buttons on the top of the LBE must be pressed. When exiting, changes must be saved. Exiting the LBE will not prompt to save changes. Press the "**Save Changes and Closes Window**" or "**Apply**" button in the LBE to properly save the changes.

Printing the Array Window

There are several Array window printing options for FPGA Editor. Printing was not available from EPIC. When print is selected from the File menu the design is broken into as many sections as needed to print the whole design out. Just the section being shown can be selected along with “**All**” or a specified set of the pages. Print Preview allows the user to see what is represented in each page.

Creating Key Assignments

Hotkeys allow the user to perform common actions that require a menu pulldown or button. Hotkeys are assigned a little differently than EPIC. In FPGA Editor, simply place the desired hotkey letter(s) within square brackets [...]. For example to assign the letter "**y**" to attribute a selected item, type the following in the command line:

```
alias [y] “post attr”
```

In this example, the letter "**y**" could then be pressed after selecting an object in the array window to get the attributes dialog for that object.

To have this alias available each time FPGA Editor is invoked, add to your `fpga_editor_user.ini` as described in section "Editing the fpga_editor_user.ini Files".

Configuring FPGA Editor to Zoom and Pan like EPIC

This will allow the user to set the FPGA Editor Zoom and/or Pan control back to the actions that EPIC used for these functions. Note that when editing the file, look at how other actions, aliases and buttons are set up and can be changed.

Editing the *fpga_editor_user.ini* Files

The `fpga_editor.ini` file is located in the `$XILINX/data` directory. If this is a server installation changes made here will affect all users. To make local changes "*cut and paste*" the EPIC section into a file named `fpga_editor_user.ini` in either the user’s home directory or in the directory where the design (revision) exists. The section of the `fpga_editor.ini` file is called "# Define some hot-keys (accelerators) … ". This section defines how the zoom and pan actions are done. The following section allows these actions to be redefined to be EPIC like. Comment out the action in the first section and uncomment out the action in the EPIC section. If adding to a file, then just add the EPIC sections uncommented to the `fpga_editor_user.ini` file.

Revision History

Date	Version	Revision
10.13.99	1.0	Initial Xilinx release.