Xilinx Project Navigator XST - XPLA Professional Design Flow for CoolRunner™ CPLDs

XAPP316 (Version 1.0) September 28, 1999

Application Note

Summary

This document provides an overview of the design flow for WebPACK Verilog/VHDL users targeting the Xilinx CoolRunner™ CPLDs.

Introduction

Xilinx provides XPLA Professional for use with WebPACK's Project Navigator (PN) at no charge. Project Navigator includes Xilinx Synthesis Technology (XST) which allows designers who use VHDL or Verilog to target CoolRunner CPLDs as large as 960 macrocells. This application note discusses the use of XPLA Professional with Project Navigator's XST.

The CoolRunner series is a family of advanced 3V and 5V complex programmable logic devices (CPLDs). The XPLA1 series, designated the XCR5000 (5V) and XCR3000 (3V) series devices, ranges from 32-128 macrocells. The XPLA2 series consists of the 320 macrocell XCR3320 and the 960 macrocell XCR3960. The CoolRunner family also includes the XCR22V10. The principle advantage of the CoolRunner series over alternative CPLDs is that static power consumption is zero.

Design Flow

Project Navigator is the graphical user interface (GUI) provided in Xilinx's WebPACK software. WebPACK supports ABEL, Verilog, and VHDL design entry targeting Xilinx's 9500, 9500XL, 9500XV, and CoolRunner devices. Project Navigator provides a seamless entry and fitting interface for all 9500 series CPLDs. For ABEL-based designs, WebPACK provides a single interface for designs targeting CoolRunner CPLDs. Figure 1 shows the flow to use XPLA Professional with Project Navigator's XST for Verilog and VHDL designs targeting CoolRunner CPLDs.

The design flow involves the use of Project Navigator and XPLA Professional as stand-alone tools. Project Navigator is invoked first, and an `edif` file is created. Then XPLA Professional uses the `edif` input file to create the jedec, report, and simulation files. XPLA Professional contains a gate level simulator, and can optionally generate a delay-annotated VHDL or Verilog timing model for use in a third party simulator.

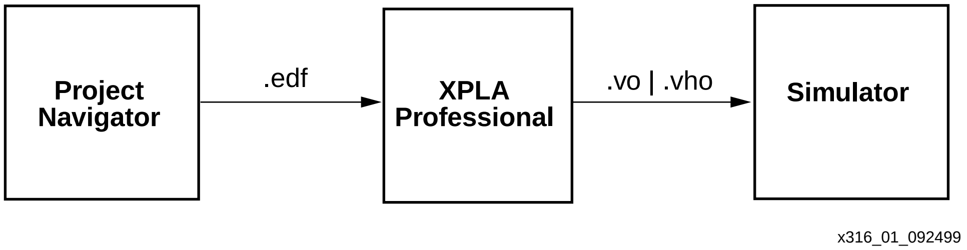


Figure 1: Design Flow

Using Project Navigator

The procedure for using Project Navigator is as follows.

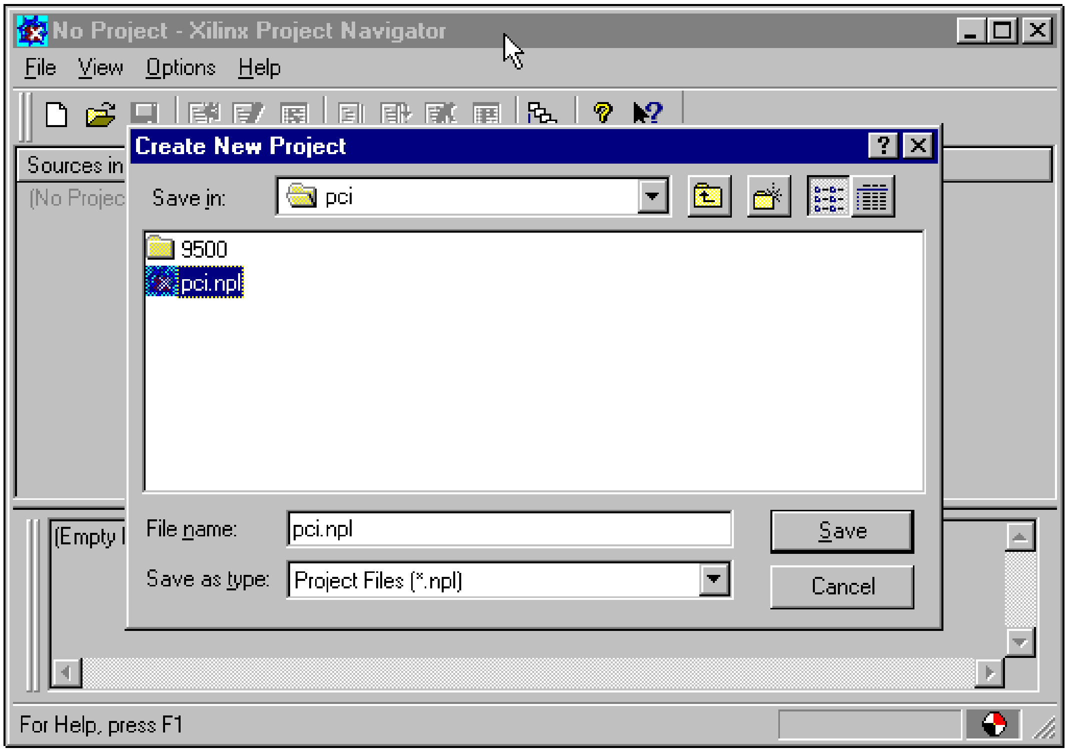
Invoke WebPACK’s Project Navigator

Create a project by entering **File > New** from the pull down menu (Figure 2). Name the project in the dialog box.

Note that the most important architectural considerations when writing verilog or VHDL code to target CoolRunner CPLDs are:

1. For XPLA1 and XPLA2 Series, the flip-flop can have an asynchronous reset or an asynchronous preset. Both cannot be asynchronous. If both are needed, one must be synchronous.
2. Latches are implemented in combinatorial logic in the XPLA1 and XPLA2 series.

The first generation device in the XPLA1 series had a limited number of clocks. In general, select the newer devices which have CS or AS in the part number after the macrocell count, e.g., XCR3128AS7BE.



No Project - Xilinx Project Navigator

Create New Project

Save in: pci

9500

pci.npl

File name: pci.npl

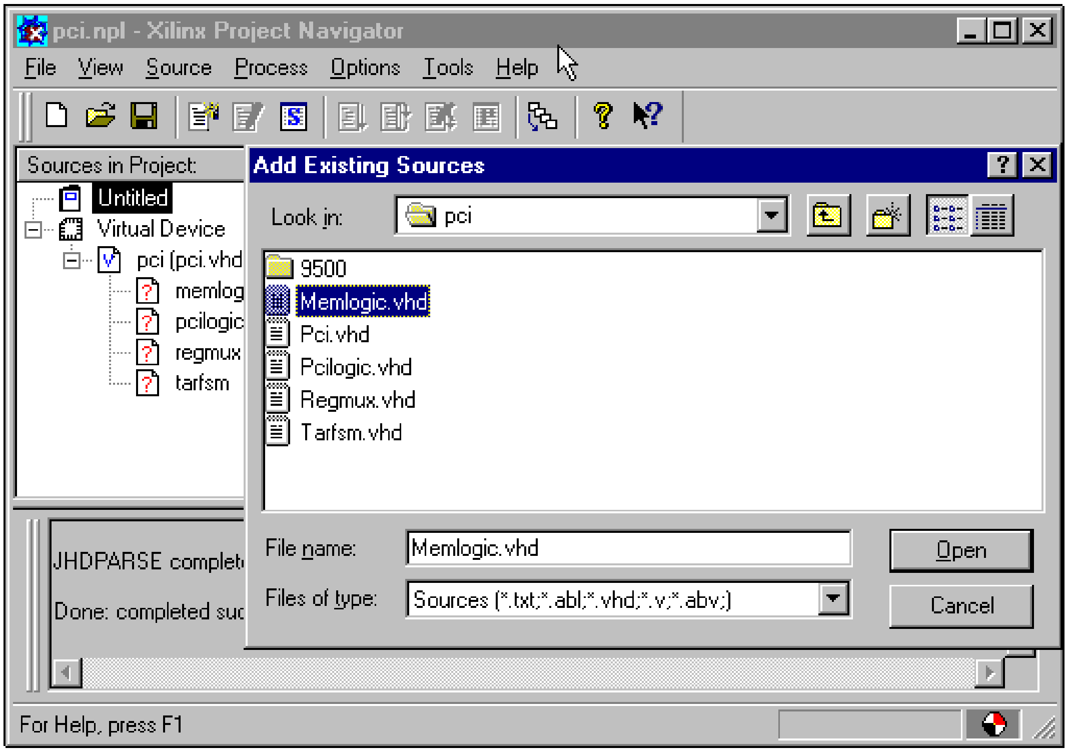
Save as type: Project Files (*.npl)

Figure 2: Starting WebPACK Project Navigator

Add Existing Sources

Select **Source > Add Source** and use the dialog box to add VHDL or Verilog file(s) (Figure 3). Hierarchy can be handled using either single or multiple files. The top level file name should be the same as the top level module name. If a single file containing multiple modules is used, the top-level module should be the last module in the file. For hierarchy in Verilog designs, the `‘include` construct is supported.

For VHDL designs, a dialog box requires the user to define the file(s) as a module, testbench, or package. Modules and packages must be contained in separate files.



Look in: pci

File name: Memlogic.vhd

Files of type: Sources (*.txt;*.abl;*.vhd;*.v;*.abv;)

```
JHDPARSE complet

Done: completed suc
```

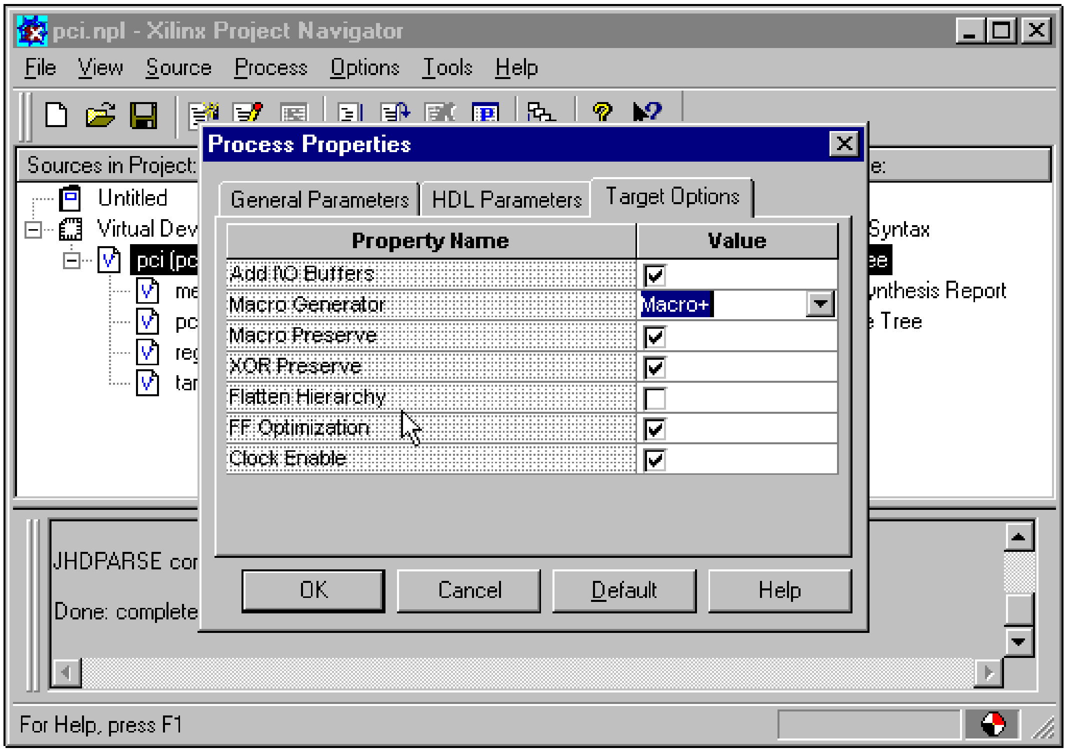
Figure 3: Adding Existing Sources

Process Properties

After adding the source files, highlight the top level design file in the **Source** window. Then highlight the **Synthesize Tree** entry in the **Process** window. Right click the mouse, and select **Target Options** in the pop-up menu. Change the MacroGenerator entry to Macro+. (Figure 4)

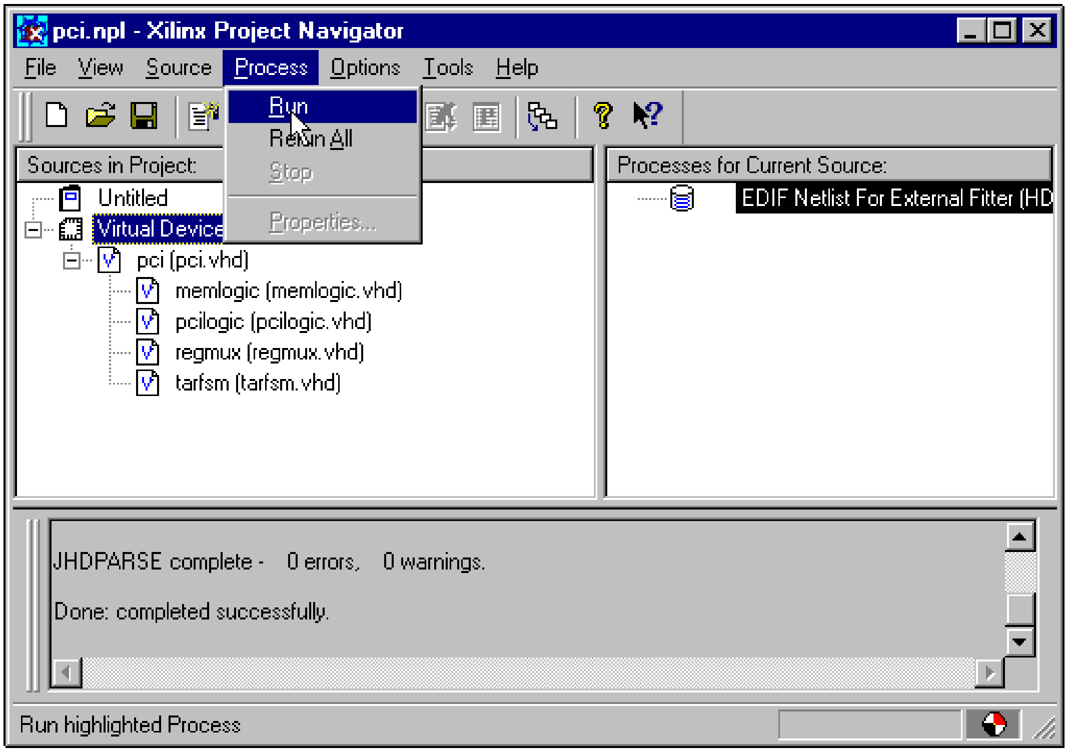
Generate an `edif` File

Highlight **Virtual Device**, then from the pull down menu, select **Process > Run** to generate an `edif` file. (Figure 5)



Property Name	Value
Add I/O Buffers	☑
Macro Generator	Macro+
Macro Preserve	☑
XOR Preserve	☑
Flatten Hierarchy	☐
FF Optimization	☑
Clock Enable	☑

Figure 4: Process Properties



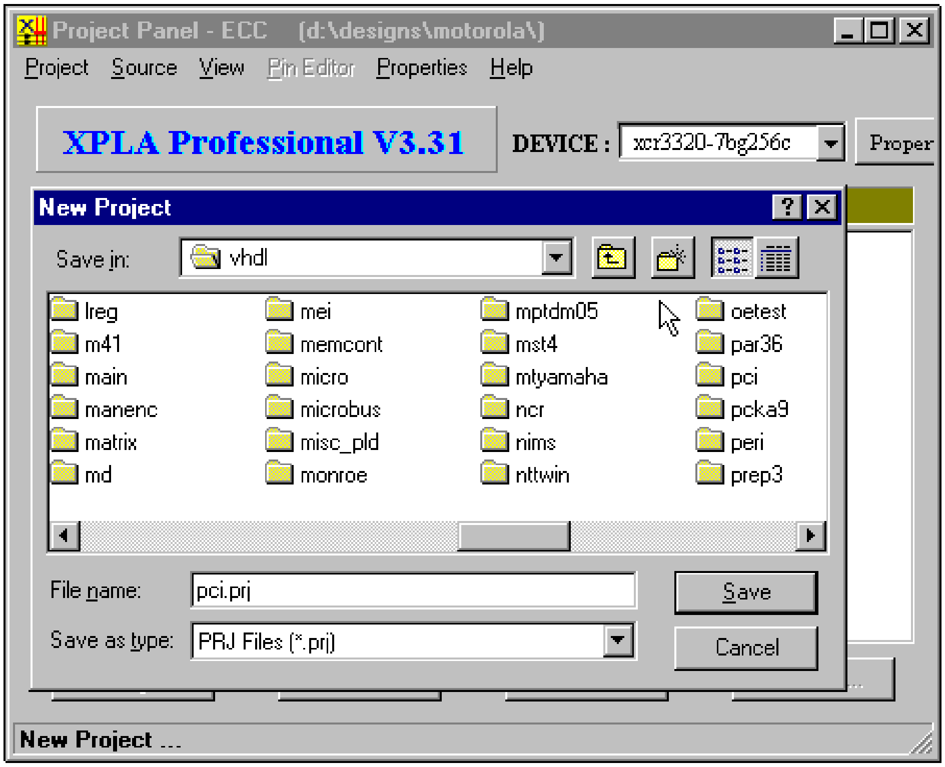
```
JHDPARSE complete -   0 errors,   0 warnings.

Done: completed successfully.
```

Run highlighted Process

Figure 5: Generating an `edif` File

Naming the Project

Invoke XPLA Professional, select **Project > New**, and name the project in the dialog box. (Figure 6.)



DEVICE : xcr3320-7bg256c

Save in: vhdl

File name: pci.prj

Save as type: PRJ Files (*.prj)

Figure 6: Naming the Project

Importing Files to the Project

Select **Source > Import File to Project**, enable **Files of type EDIF Files (`*.edf`)** in the dialog box, select the `edif` file, and click on **Open**. (Figure 7.)

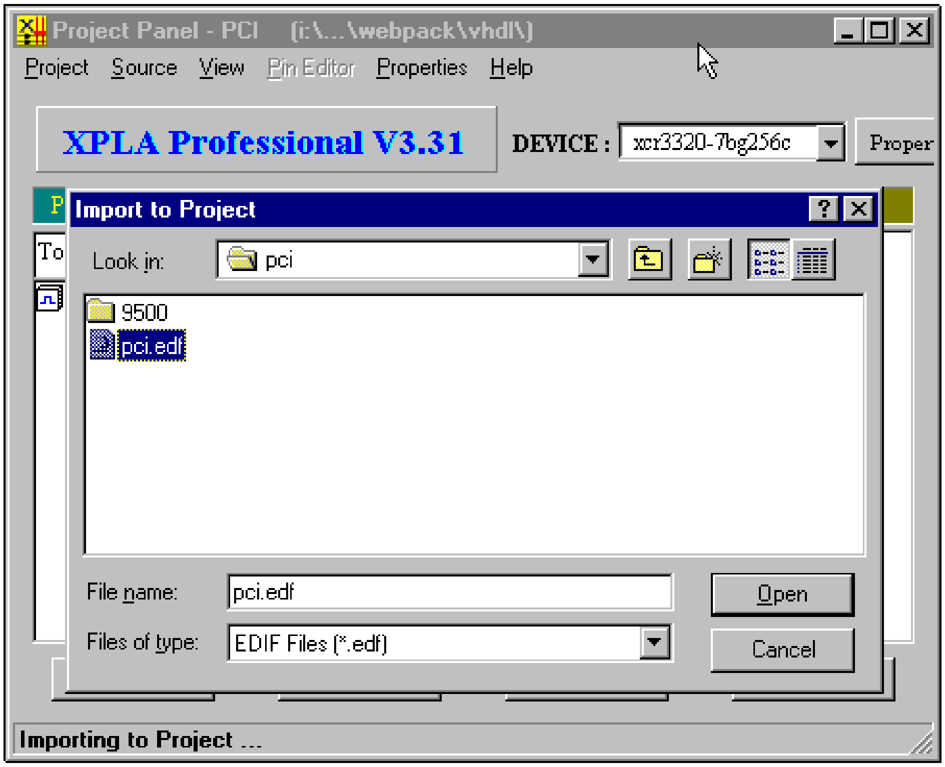
Selecting a Device

Click **Device** and use the cascading menus to select a device. (Figure 8)

The Properties Menu

Most PC users provide compile options in XPLA Professional using the **Properties** menu in the GUI (Figure 9). Alternatively, a control file, generally named `<design>.ctl`, can be used to assign pins and provide compile options. The `<design>.ctl` file is created with a text editor. To enable the control file, position the cursor over the control file section of the **Properties** menu and click the right mouse button. Select the `<design>.ctl` in the dialog box.

The control file is used exclusively in the workstation flow and is discussed in detail later.



DEVICE : xcr3320-7bg256c

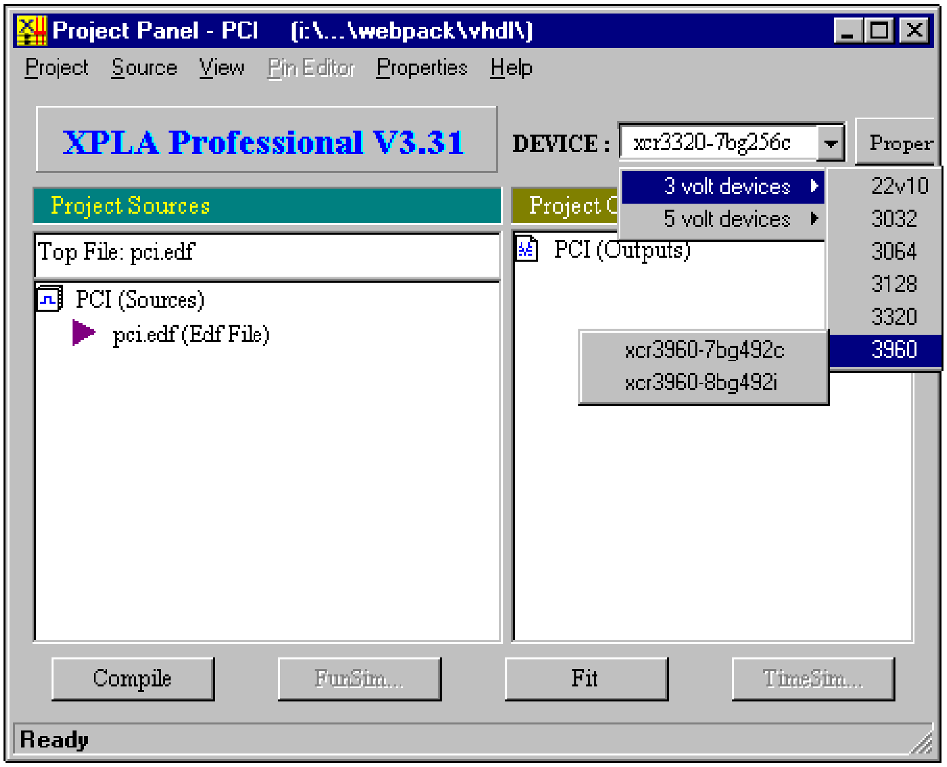
Look in: pci

File name: pci.edf

Files of type: EDIF Files (*.edf)

Importing to Project ...

Figure 7: Importing a File to a Project



DEVICE : xcr3320-7bg256c

Top File: pci.edf

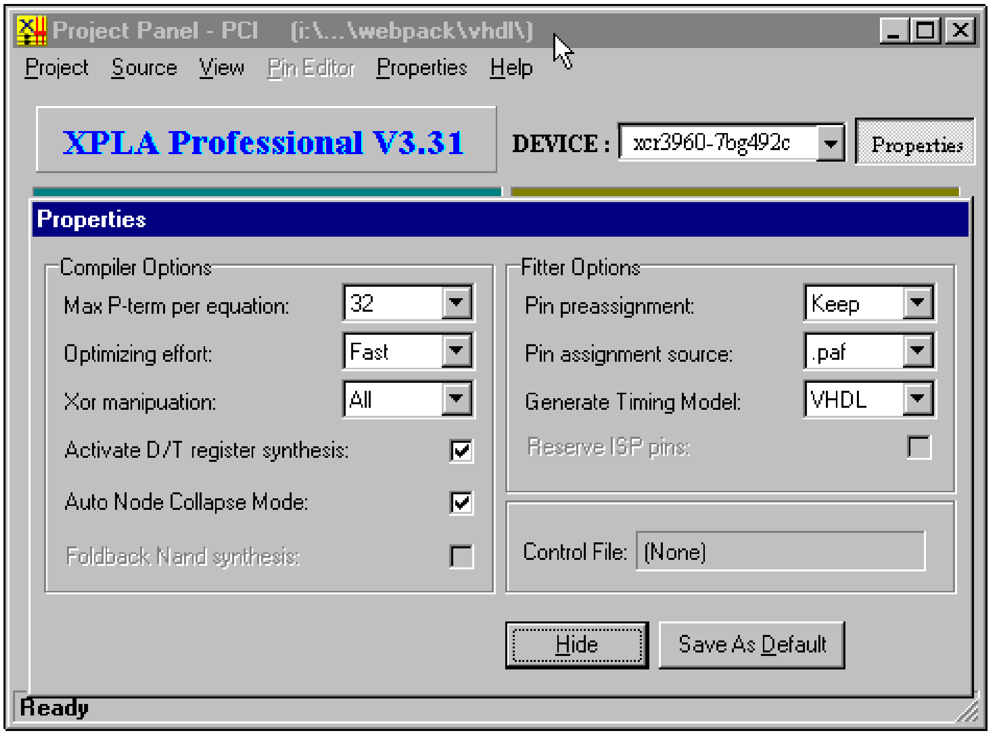
PCI (Sources)

pci.edf (Edf File)

PCI (Outputs)

Ready

Figure 8: Selecting a Device


Project Panel - PCI (i:\...\webpack\vhdl\)
Project Source View Pin Editor Properties Help
XPLA Professional V3.31
Properties
Hide
Save As Default
Ready

DEVICE : xcr3960-7bg492c

Properties

Compiler Options

Max P-term per equation: 32

Optimizing effort: Fast

Xor manipuation: All

Activate D/T register synthesis: ☑

Auto Node Collapse Mode: ☑

Foldback Nand synthesis: □

Fitter Options

Pin preassignment: Keep

Pin assignment source: .paf

Generate Timing Model: VHDL

Reserve ISP pins: □

Control File: (None)

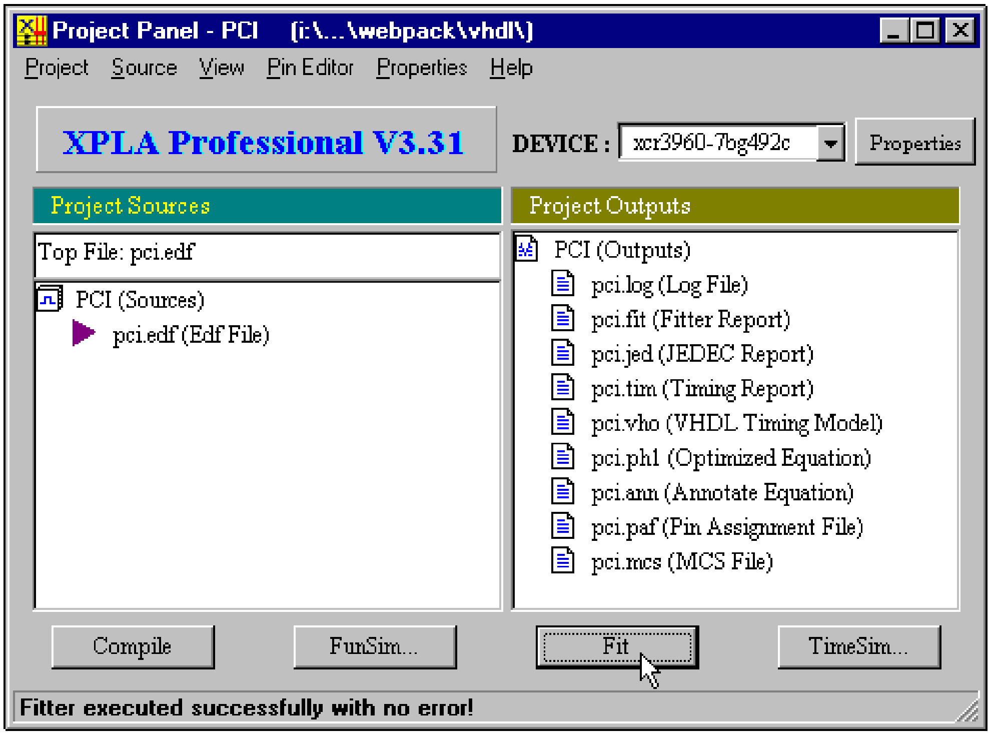
Figure 9: Using the Properties Menu

Project Output

Click **Fit**. The jedec (`.jed`), timing (`.tim`), utilization (`.fit`), pinout (`.spf`), and optionally a delay-annotated Verilog and/or VHDL file for timing simulation are displayed in the Project Outputs window. (Figure 10)

Pin Assignment Editor

Pin assignments can be done in the pin assignment editor or in a `<design>.ctl` file (Figure 11). If using the pin assignment editor, select the pin assignment file (`<design>.paf`) as the **Pin Assignment source** and **Keep** as the **Pin Preassignment** option in the **Properties** menu.



XPLA Professional V3.31

DEVICE : xcr3960-7bg492c

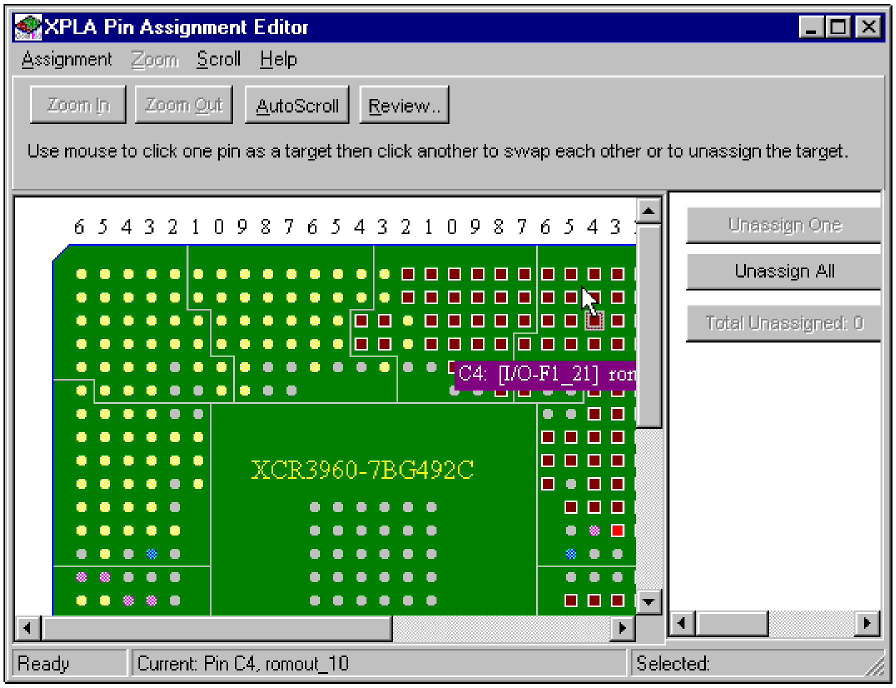
Top File: pci.edf

PCI (Sources)
pci.edf (Edf File)

PCI (Outputs)
pci.log (Log File)
pci.fit (Fitter Report)
pci.jed (JEDEC Report)
pci.tim (Timing Report)
pci.vho (VHDL Timing Model)
pci.phl (Optimized Equation)
pci.ann (Annotate Equation)
pci.paf (Pin Assignment File)
pci.mcs (MCS File)

Fitter executed successfully with no error!

Figure 10: Project Outputs



Use mouse to click one pin as a target then click another to swap each other or to unassign the target.

C4: [I/O-F1_21] rom

Ready | Current: Pin C4, romout_10 | Selected:

Figure 11: Pin Assignment Editor

Simulation

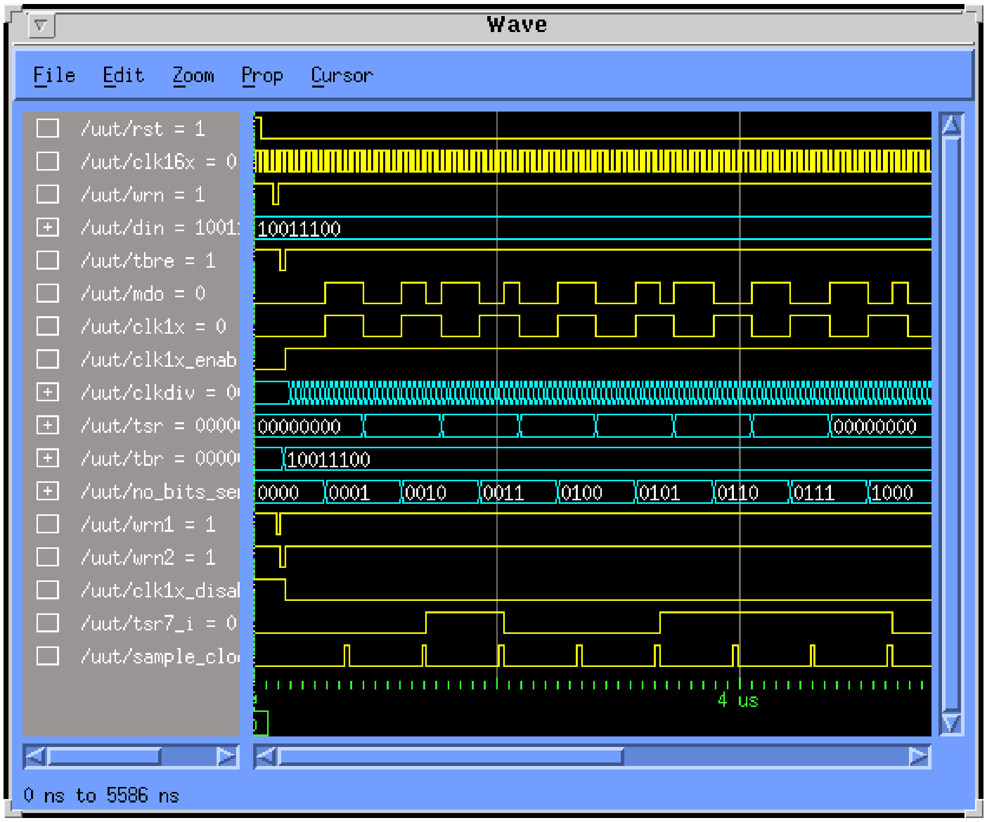
XPLA Professional generates delay-annotated Verilog and VHDL models intended to be used in timing simulation (Figure 12). In these simulation models, busses are broken into discrete signals, so the test bench for the behavioral code may require revision.

Using the Manchester encoder in $XPLA_PATH/examples as an example, run a timing simulation in Model Technology's ModelSim by invoking ModelSim and entering:

```
vlib work
vmap work “./work”
vcom me.vho
vcom me_tb.vhd
vsim testbench v1
```

To run a Verilog timing simulation in Verilog-XL, enter:

```
verilog me.vo mv_tf.v
```



```
☐ /uut/rst = 1
☐ /uut/clk16x = 0
☐ /uut/wrn = 1
⊞ /uut/din = 1001
☐ /uut/tbre = 1
☐ /uut/mdo = 0
☐ /uut/clk1x = 0
☐ /uut/clk1x_enab
⊞ /uut/clkdiv = 0
⊞ /uut/tsr = 0000
⊞ /uut/tbr = 0000
⊞ /uut/no_bits_se
☐ /uut/wrn1 = 1
☐ /uut/wrn2 = 1
☐ /uut/clk1x_disal
☐ /uut/tsr7_i = 0
☐ /uut/sample_clo
```

0 ns to 5586 ns

Figure 12: XPLA Professional Timing Simulation

Format of the Control File

A control file must be named the same as its design name, with the extension of "`.ctl`" The "#" character at the beginning of a line is used to indicate a comment. The control file contains up to three sections: command, property, and pin_assignment. The only required lines to edit in a control file template are `-i <design.edf>`, and `-dev <device>` in the [command] section. The `th` parameter is commonly varied, and the `pre` parameter, and the [pin_assignment] section are generally used. For hard to fit designs, increase `bfi`. The signal grouping properties in the [property] section are seldom needed to meet fit requirements. Refer to Table 1and Table 2 for Section and Property Commands and Table 3 for Pin Assignments.

Table 1: Command Section [command]

Command	Description
-i <design>.[edf | v | phd]	Required field used to specify input filename.
-it <edif | verilog | phdl>	Required field used to specify input type.
-th <number>	Specify max `pterm` for each equation (default is 11, range is 5-37).
-fi <number>	Specify max fan-in for each equation (default for 22v10 is 22; default for larger devices is 36, range is 5-37).
-bfi <number>	Specify max fan-in for each logic block (default is 36, range is 36-40).
-vho	Directs fitter to generate a delay-annotated VHDL simulation model (default is to not generate VHDL model).
-vo	Directs fitter to generate a delay-annotated Verilog model (default is to not generate Verilog model).
-reg	Apply register synthesis to the design.
-co <best | none>	Specify collapsing method (default is **best**).
-effort <exhaust | fast>	Specify synthesis effort (default is **fast**).
-xor <all | exp | none >	Specify **xor** synthesis type (all, exp, or none, default is **none**) - XPLA2 devices only.
-dev <device>	Required field used to specify device to target
-pre <keep | try | ignore>	Specify pin assignment effort by fitter (default is **try**). If keep is specified, the fitter will not complete a fit if the pinout assignments cannot be met. **Note**: To keep pin assignments, enter `-pre keep` in the [command] section. If the fitter report produces the following message(s), "*XPLA Designer-XL has changed the signal names slightly. > WARNING 3271: 'UVI2' found in pin file but not in design file, ignored*". This message will occur when busses which are originally named `din[4]` are renamed renamed by XPLA Designer-XL to `din_4_`. In some cases, the case of the signal name may change. To work around this, run the design using `-pre ignore`, look at the signal names in the fit (or spf) report, and revise the signal names in the `.ctl` file to match those in the `.fit` file. To reserve a pin, enter in the [command] section `3-state : reserved1:<pin_location>` Do not use the reserved pin name anywhere else.

Table 2: Property Section [property]

Command	Description
- maxpt <signal>	Specify maximum `pterm` for specified pin/node.
- keep <signal>	Directs optimizer to keep (do not optimize out) the specified signal.
-retain <signal>	Directs optimizer to retain (do not optimize out) the specified signal.
-dut on	Activates the global 3-state pin GTS(N). (Default is disabled, allowing the pin to be used as user input.)
-isp off	Disables the four JTAG pins for use in in-system programming, allowing them to be used for user I/O. When off, the device can be programmed through the JTAG pins once.
-tri-state all -fm_group<signals>	Directs fitter to group specified signals within a fast module. (XPLA2 only.)
-lb_group <signals>	Directs fitter to group specified signals within a logic block.
-slow_slew_rate <outputs>	Enables SLOW slew rate output buffer on specified outputs. (Default is fast slew rate.) (XPLA2 only.)

Table 3: Pin Assignment Section [pin_assignment]

Pin Assignment	Signal
[pin_assignment]	<signal1>:4 <signal2>:5

Revision History

Date	Version #	Revision
09.28.99	1.0	Initial release.