

Foundation

Q Besides the CD-ROM supplied with the product, are there any sources of additional information about using the Xilinx Foundation Series software?

The Foundation Documentation Update Pack contains application notes about the

Foundation Logic Simulator, Foundation XVHDL compiler, and a new simulation tutorial. Get FND_DOC1.EXE (self-extracting ZIP file) from the Xilinx FTP site ([ftp.xilinx.com/pub/swhelp/foundation](ftp://ftp.xilinx.com/pub/swhelp/foundation)) or the Xilinx BBS (Software Help ➤ Aldec).

Mentor Graphics

Q Can XACTstep 5.2.1 be used with Mentor Graphics' B.x release?

A: Under Mentor B.x, Gen_sch8 and XBLXGS may fail with either "call to undefined procedure" (SunOS) or "unresolved propagate symbol" (HP-UX) error messages due to problems with dynamically linking to Mentor's Design Data Port (DDP). A patch is available to fix these problems and make XACTstep 5.2.1 interface smoothly with Mentor B.x. The patch may be downloaded from the Xilinx FTP site at:

```
ftp://ftp.xilinx.com/pub/
  swhelp/mentor/bl_521s.tar.Z
  (SunOS)
ftp://ftp.xilinx.com/pub/
  swhelp/mentor/bl_521h.tar.Z
  (HP-UX)
```

Q A schematic originally targeted for the XC4000 family is now targeting an XC4000E device. During Men2XNF8, the following error occurs when running EDIF2XNF:

```
Error: 6 EDIF data "ofd.eds"
not found in directory
"/usr/xact/data/unified/
edif4000e"
```

In this case, the schematic was created using the XC4000 library, but the design is now targeted to an XC4000E device. (Other family combinations may also cause this error.) Certain symbols are primitives in the schematic library with which the design was built (XC4000), but are macros in the library associated with the device being targeted (XC4000E). In this example, OFD is a primitive in the XC4000 library but is a macro in the XC4000E library (since it has, underneath it, the clock-enabled OFDX symbol).

ENWrite (Mentor's EDIF netlister) looks at the XC4000 version of OFD in the schematic. This component has attached to it a COMP=OFD property, indicating that this component is a primitive and should be written as such in the EDIF file. EDIF2XNF then takes this primitive description and looks for the corresponding ofd.eds file in the \$LCA/data/unified/edif4000e directory, that contains EDIF descriptions for XC4000E primitives. Since OFD is not a primitive in the XC4000E family, the ofd.eds file does not exist in the directory, resulting in the error.

The correct OFD from the XC4000E library has no COMP=OFD property attached to it, since the COMP property is reserved for primitives. Therefore, if the correct XC4000E component had been used, ENWrite would have written out the hierarchy below OFD. Then, EDIF2XNF would have never seen the OFD "primitive" and would have never tried to look for a non-existent ofd.eds file in the XC4000E data directory.

The proper way to retarget the design to a new device family is to use the Convert Design utility in PLD_DA before running the implementation flow. Convert Design replaces the library components in a schematic or set of schematics so that they come from the proper library. For instructions on how to use Convert Design, see Solution 798, "Retargeting a design in Mentor Design Architect (Convert Design)," from the Xilinx Solutions Database:

<http://www.xilinx.com/techdocs/798.htm>

Q What is happening when Quicksim issues this warning on every primitive in a design:

```
// Warning: Instance
'/GIVE_ME_AN_E'
// Could not find a
registered simulation model
with label: 'xc4000'
// NULL model will be
inserted. (from: Analysis/
Digital/Simulation
Utilities/Dsim 85)
```

This is caused by an incorrectly-written simulation viewpoint for the design, which can result when a design is retargeted to a new device family. The solution is to delete

that default viewpoint, then run PLD_DVE_SIM on the design, specifying the correct part family. If done from the command line, an XC4000 design, for example, might use:

```
delete_object
blanking_design/default
pld_dve_sim blanking_design
xc4000
```

Note: If Timsim8 -o was used to create the simulation model, be sure to run Timsim8 instead of PLD_DVE_SIM. Timsim8 -o runs PLD_DVE_SIM, then adds links to timing information into the viewpoint afterwards.

**Mentor
Graphics**
(continued)

Q How are Boundary Scan components for the XC4000 family instantiated in Verilog-based designs?

To use Boundary Scan components in XC4000 Series devices, instantiate the boundary scan symbol (BSCAN) and the associated dedicated I/O. Use the “dont_touch” attribute; otherwise, BSCAN

may be deleted by the Synopsys compiler.

The Verilog code for instantiating BSCAN in the XC4000 is shown below. VHDL and Verilog examples for both the XC4000 and XC5000 can be found on the Xilinx Web site (www.xilinx.com), or can be emailed to users via the Xdocs mail server (email xdocs@xilinx.com).

Synopsys

XC4000/XC4000E example of instantiating the BSCAN symbol:

```
module example (a,b,c);

input a, b;
output c;
reg c;

wire tck_net;
wire tdi_net;
wire tms_net;
wire tdo_net;

BSCAN u1 (.TDI(tdi_net), .TMS(tms_net),
.TCK(tck_net), .TDO(tdo_net));

TDI u2 (.I(tdi_net));
TMS u3 (.I(tms_net));
TCK u4 (.I(tck_net));

TDO u5 (.O(tdo_net));

always@(posedge b)
    c<=a;

endmodule
```

Runscript for compiling the XC4000/XC4000E BSCAN Verilog Example:

```
PART = 4025ehq240-3
TOP = example

read -format verilog "bscan4k.v"

set_port_is_pad ""
insert_pads

set_dont_touch u1
set_dont_touch u2
set_dont_touch u3
set_dont_touch u4
set_dont_touch u5

compile

replace_fpga

set_attribute TOP "part" -type string
PART

write -f xnf -h -o "bscan4k.sxnf"
```