MicroSim FPGA

FPGA Design Software

User's Guide

MicroSim Corporation
20 Fairbanks
Irvine, California 92618
(714) 770-3022

Version 8.0, June, 1997.

TradeMarks

Copyright Notice

Technical Support

Internet	Tech.Support@microsim.com
Phone	(714) 837-0790
FAX	(714) 455-0554
WWW	http://www.microsim.com

Customer Service

Internet	Sales@MicroSim.com
Phone	(714) 770-3022

Contents

Before You Begin

Welcome to MicroSim DesignLab xi
MicroSim FPGA Overview xii
How to Use this Guide xiii
    Typographical Conventions xiii
Related Documentation xiv
    Online Help xv
What's New xvi

Chapter 1 Preparing to Use MicroSim FPGA

Overview 1-1
Installation 1-2
Configuration 1-3
Design Flow 1-4

Chapter 2 Design Entry of FPGA Devices in Schematics

Overview 2-1
Naming Conventions 2-2
Xilinx Universal Library 2-3
Entering an FPGA Design 2-4
    Entering an FPGA-Only Design 2-4
    Entering an FPGA Block in a Mixed Design 2-5
        How Schematics Sets up the FPGA Block 2-5
    Incorporating an Existing Xilinx Schematic in a Design 2-6
    Using an Existing XNF File in a Schematic 2-7
Defining Inputs and Outputs 2-8
Specifying Xilinx Parameters 2-9
Changing the Device Family for an FPGA 2-10
Using X-BLOX 2-11
    The X-BLOX Library 2-11
    Using Buses 2-11

Using X-BLOX Attributes 2-12
X-BLOX Example 2-12
Using MEMGEN 2-13
Where to Put MEMGEN XNF Files 2-13
To Attach a MEMGEN XNF file to a Block 2-14
Using the TIMESPEC and TIMEGRP Symbols 2-15
Creating Your Own Symbol Library 2-17
Adding Symbols to Your Library 2-17
Directory Structure for Custom Libraries 2-18
Configuring a Custom Library 2-19

Chapter 3 Running XACTstep

Overview 3-1
Setting Up an FPGA in XACTstep 3-2
Starting XACTstep 3-2
Setting Up and Translating an FPGA in XACTstep 3-2
Regenerating All Netlists without Running XACTstep 3-4

Chapter 4 Simulation

Overview 4-1
Specifying Stimulus 4-2
Special Signals 4-3
Specifying Global Signals 4-3
Specifying Global Reset on XC2000 and XC3000 4-3
Specifying Global/Set Reset on XC4000 and XC4000E Devices 4-4
Specifying Global Reset on XC5200 Devices 4-4
Specifying Global Tristate on XC4000, XC4000E, and XC5200 Devices 4-4
Accessing Oscillators 4-5
Using the OSC (XC2000/XC3000) 4-5
Using the OSC4 (XC4000/XC4000E) 4-5
Using the OSC5 (XC5200) 4-6
Running Simulations 4-7
Using Probe Markers 4-8
Xilinx Net Names in Probe 4-9
Xilinx Net Names for FPGA-Only Designs 4-9
Net Names for Mixed Designs 4-10

Glossary

Index

Figures

Figure 1-1 Directory Structure for Symbol Libraries (Device Families) 1-2
Figure 1-2 FPGA Design Flow 1-5
Figure 2-1 Defining LOC Equal to B in Attribute Definition Dialog Box 2-9
Figure 2-2 Block Set Up for MEMGEN XNF File 2-14
Figure 2-3 TIMESPEC Symbol with TS01 Attribute 2-15
Figure 2-4 TIMEGROUP Symbol with FLOPS Attribute 2-16
Figure 2-5 Directory Structure for Custom Libraries 2-18
Figure 4-1 Xilinx Net Names in an FPGA-Only Design 4-9
Figure 4-2 Xilinx Net Names in a Mixed Design 4-11

Tables

Table 4-1 OSC1 Frequencies for DIVIDE1_BY Values 4-6
Table 4-2 OSC2 Frequencies for DIVIDE2_BY Values 4-6

Before You Begin

Welcome to MicroSim DesignLab

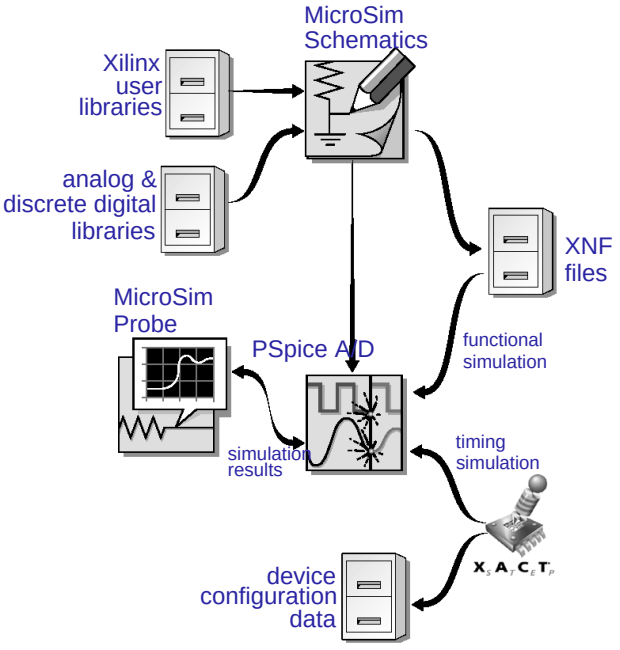
Welcome to DesignLab, the new standard for EDA. With this single-system solution, you can design circuits with mixed analog, discrete digital, PLDs, and FPGAs. With MicroSim DesignLab you can enter your schematic, simulate your design, view all simulation results within a single window and lay out your printed circuit board—going back and forth between design stages, improving your design from start to finish.

MicroSim FPGA Overview

MicroSim FPGA is a DesignLab component that allows you to use MicroSim Schematics and MicroSim PSpice A/D to enter designs for Xilinx FPGA (field programmable gate array) devices. The FPGA devices are implemented with Xilinx XACTstep; MicroSim FPGA provides the interface between Schematics and XACTstep.

The design you enter with Schematics can be an FPGA-only device, or it can be a mixed design which incorporates one or more FPGA devices with other components entered using Schematics.

Schematics with MicroSim FPGA supports XACTstep 5.2/ 6.0. XACTstep must be purchased separately from Xilinx.



How to Use this Guide

This guide is designed so you can quickly find the information you need to use MicroSim FPGA.

This guide assumes you are familiar with Xilinx XACTstep programs and Xilinx FPGA architectures and terminology.

This guide assumes that you are familiar with Microsoft Windows (NT or 95), including how to use icons, menus and dialog boxes. It also assumes you have a basic understanding about how Windows manages applications and files to perform routine tasks, such as starting applications and opening and saving your work. If you are new to Windows, please review your *Microsoft Windows User's Guide.*

Typographical Conventions

Before using MicroSim FPGA, it is important to understand the terms and typographical conventions used in this documentation.

This guide generally follows the conventions used in the *Microsoft Windows User's Guide*. Procedures for performing an operation are generally numbered with the following typographical conventions.

Notation	Examples	Description
Ctrl+R	Press Ctrl+R	A specific key or key stroke on the keyboard.
`monospace font`	Type VAC...	Commands/text entered from the keyboard.

Related Documentation

Documentation for DesignLab is available in both hard copy and online. To access an online manual instantly, you can select it from the Help menu in its respective program (for example, access the Schematics User's Guide from the Help menu in Schematics).

Note *The documentation you receive depends on the software configuration you have purchased.*

The following table provides a brief description of those manuals available in both hard copy and online.

This manual...	Provides information about how to use...
MicroSim Schematics User's Guide	MicroSim Schematics, which is a schematic capture front-end program with a direct interface to other MicroSim programs and options.
MicroSim PCBoards User's Guide	MicroSim PCBoards, which is a PCB layout editor that lets you specify printed circuit board structure, as well as the components, metal, and graphics required for fabrication.
MicroSim PSpice A/D & Basics+ User's Guide	PSpice A/D, Probe, the Stimulus Editor, and the Parts utility, which are circuit analysis programs that let you create, simulate, and test analog and digital circuit designs. It provides examples on how to specify simulation parameters, analyze simulation results, edit input signals, and create models.
MicroSim PSpice & Basics User's Guide	MicroSim PSpice & MicroSim PSpice Basics, which are circuit analysis programs that let you create, simulate, and test analog-only circuit designs.
MicroSim PSpice Optimizer User's Guide	MicroSim PSpice Optimizer, which is an analog performance optimization program that lets you fine tune your analog circuit designs.
MicroSim PLSyn User's Guide	MicroSim PLSyn, which is a programmable logic synthesis program that lets you synthesize PLDs and CPLDs from a schematic or hardware description language.

The following table provides a brief description of those manuals available online *only*.

This online manual...	Provides this...
MicroSim PSpice A/D Online Reference Manual	Reference material for PSpice A/D. Also included: detailed descriptions of the simulation controls and analysis specifications, start-up option definitions, and a list of device types in the analog and digital model libraries. User interface commands are provided to instruct you on each of the screen commands.
MicroSim Application Notes Online Manual	A variety of articles that show you how a particular task can be accomplished using MicroSim‘s products, and examples that demonstrate a new or different approach to solving an engineering problem.
Online Library List	A complete list of the analog and digital parts in the model and symbol libraries.
MicroSim PCBoards Online Reference Manual	Reference information for MicroSim PCBoards, such as: file name extensions, padstack naming conventions and standards, footprint naming conventions, the netlist file format, the layout file format, and library expansion and compression utilities.
MicroSim PCBoards Autorouter Online User’s Guide	Information on the integrated interface to Cooper & Chyan Technology’s (CCT) SPECCTRA autorouter in MicroSim PCBoards.

Online Help

Selecting Search for Help On from the Help menu brings up an extensive online help system.

The online help includes:

- Step-by-step instructions on how to use MicroSim FPGA features.
- Reference information about MicroSim FPGA.
- Technical Support information.

If you are not familiar with Windows (NT or 95) Help System, select How to Use Help from the Help menu.

What's New

To find out more, see Using an Existing XNF File in a Schematic on page 2-7.

Attach XNF files to a block in a schematic You can now create a file of type XNF from a description of an FPGA and attach the file to a block in your schematic. Schematics automatically adds pins corresponding to the interface ports for the FPGA. Then, when you simulate the schematic, the block exhibits the behavior of the described FPGA.

Use multiple FPGAs in a schematic You can now have multiple FPGAs in a single schematic.

Preparing to Use MicroSim FPGA

1

Overview

This chapter describes the installation and configuration needed before you can begin using MicroSim FPGA and provides an overview of how you enter and simulate FPGA designs.

Installation on page 1-2 describes the MicroSim and Xilinx programs that must be installed before you can use MicroSim FPGA.

Configuration on page 1-3 describes how to configure Xilinx and MicroSim files to use MicroSim FPGA.

Design Flow on page 1-4 provides an overview of entering and simulating FPGA designs.

Installation

To use MicroSim FPGA, you need to install the following programs:

- MicroSim DesignLab
- Xilinx XACTstep

For the MicroSim products, refer to the installation instructions that came with the products.

XACTstep must be purchased separately from Xilinx. Schematics with MicroSim FPGA supports XACTstep 5.2/6.0. Refer to the Xilinx documentation for a description of how to install XACTstep.

During installation, you can select the device families to install. The installation program only creates the directories for the device families you have selected.

When you install MicroSim FPGA, the library files for the device families are organized in directories as shown in Figure 1-1.

Each library directory includes the following files:

family.slb	symbol library
family.lib	simulation model library
filename.sch	macro schematic files

where *family* is the device family name (such as xc4000) and *filename* is the file name assigned to each schematic file.

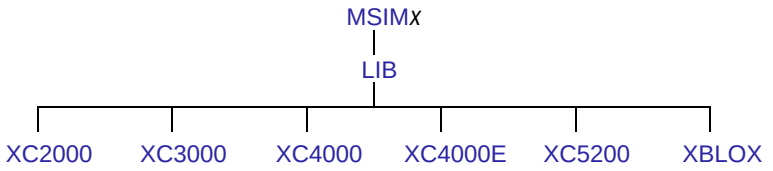


Figure 1-1 *Directory Structure for Symbol Libraries (Device Families)*

Configuration

After all programs are installed, you need to modify the `xactinit.dat` file for XACTstep to run MicroSim FPGA.

Using the MicroSim text editor, add the following line to the `xactinit.dat` file in the XACT\DATA directory:

```
orcad_names=false
```

Note *The default installation assumes that the XACT installation directory is on your path. If it is not, MicroSim FPGA cannot start the programs needed to run XACTstep.*

Design Flow

The process of entering and simulating FPGA designs with MicroSim FPGA follows this general flow:

1. **Design entry in Schematics.** The schematic is entered using symbols from the Xilinx Universal Library which comes with MicroSim FPGA.

   For FPGA-only designs, you select the Xilinx device family and enter the schematic. For mixed designs, the FPGA design is contained in hierarchical block, and non-FPGA devices are connected to the block.

2. **Creation of the XNF netlist file.** The Xilinx design is translated into a hierarchical XNF netlist file. The netlist is in the input used by XACTstep.

3. **Functional simulation.** The design is simulated using unit delays, identifying logic design errors prior to the place-and-route of the design. In a mixed design, the FPGA is simulated along with other discrete analog and/or digital components.

4. **Place-and-route.** The design is implemented in the FPGA using the Xilinx place-and-route program.

5. **Timing simulation.** The design is simulated using the timing as determined by the actual place-and-route.

6. **Device configuration.** The device configuration data is generated using XACTstep.

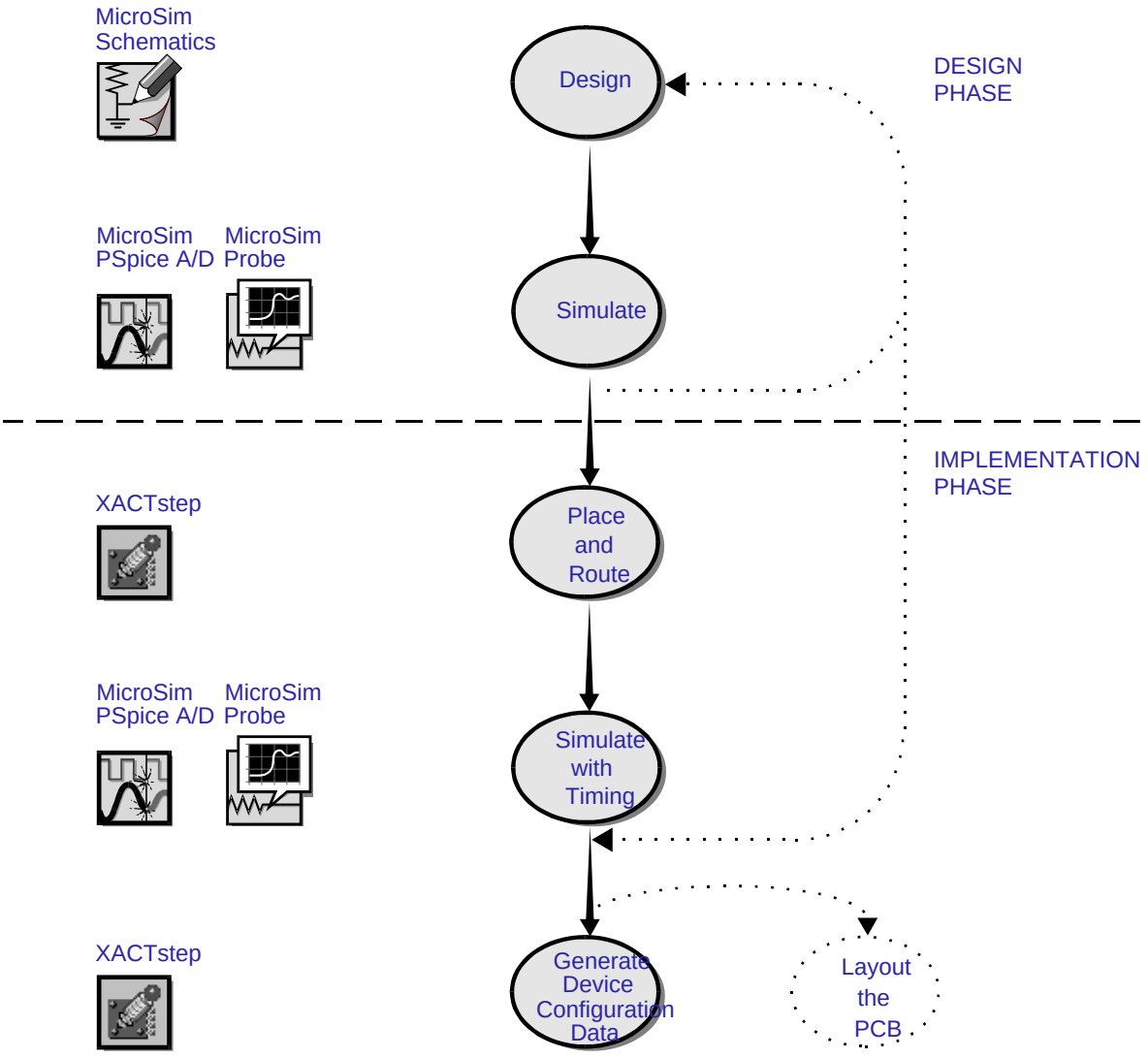


Figure 1-2 *FPGA Design Flow*

Design Entry of FPGA Devices in Schematics

2

Overview

This chapter discusses special considerations and other topics related to design entry of FPGA devices in Schematics. This chapter includes the following:

Naming Conventions on page 2-2

Xilinx Universal Library on page 2-3

Entering an FPGA Design on page 2-4

Specifying Xilinx Parameters on page 2-9

Changing the Device Family for an FPGA on page 2-10

Using X-BLOX on page 2-11

Using MEMGEN on page 2-13

Using the TIMESPEC and TIMEGRP Symbols on page 2-15

Creating Your Own Symbol Library on page 2-17

Refer to your *MicroSim Schematics User's Guide* for basic instructions about using Schematics.

Naming Conventions

This section describes limitations to the names you can use for signals or reference designators.

For signal names and reference designators, observe the following limitations:

- Names may contain only the following characters:

  A-Z, a-z (case is ignored)
  0-9
  $ _ - [ ]

- Names must contain at least one non-numeric character.
- Names can start with any legal character.

Xilinx Universal Library

MicroSim FPGA includes the Xilinx Universal Library for the following device families:

- XC2000
- XC3000
- XC4000
- XC4000E
- XC5200
- X-BLOX

The Universal Library contains primitive symbols and macro symbols. The macro symbols contain child schematics which can be accessed by double-clicking on the symbol.

MicroSim libraries and the Xilinx Universal Library have the following differences:

- The IPAD, OPAD, and IOPAD symbols (and multi-bit variants) are not included. In MicroSim FPGA, interface ports are used instead of pad symbols. See Defining Inputs and Outputs on page 2-8 for more information.
- The INPUTS and OUTPUTS symbols are not included in the X-BLOX library. See Using X-BLOX on page 2-11 for more information.
- The GXTL symbol in the XC2000 library and XC3000 library is not supported. Use the OSC symbol driving an ACLK symbol instead.
- The OSC5 and CK_DIV symbols in the XC5200 library are implemented as primitives instead of macros. This does not change the behavior or use of the symbol in any way.

Refer to your *Xilinx Libraries Guide* for information about the Universal Library.

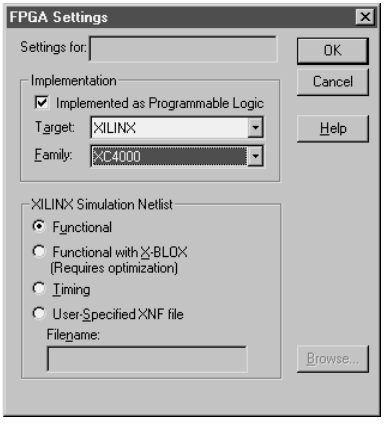
Entering an FPGA Design

This section includes the following:

- Entering an FPGA-Only Design, below
- Entering an FPGA Block in a Mixed Design on page 2-5
- Incorporating an Existing Xilinx Schematic in a Design on page 2-6
- Using an Existing XNF File in a Schematic on page 2-7

Entering an FPGA-Only Design

To enter an FPGA-only design



Xilinx macro symbols have underlying schematics. To display the underlying schematic for a symbol, double-click the symbol.

1. In Schematics, from the File menu, select New.
2. From the Tools menu, select FPGA Settings.
3. In the FPGA Settings dialog box, do the following:
   a. Select the Implemented as Programmable Logic check box.
   b. From the Target list, select XILINX.
   c. From the Family list, select the appropriate Xilinx device family.

   Click OK when finished. The appropriate symbol libraries are configured automatically as local libraries on the current schematic.
4. Place symbols from the selected Xilinx device family as needed for the FPGA.

Note *XACTstep 5.2/6.0 does not support long file or directory names. When saving Xilinx schematics, do not use more than eight characters or use spaces in a file or directory name.*

Entering an FPGA Block in a Mixed Design

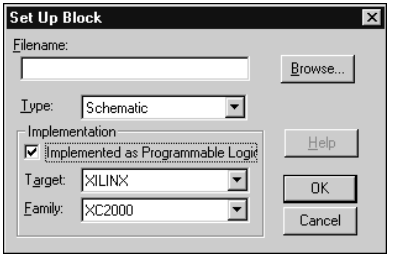
You can create one or more FPGA blocks within a mixed design. Each block's child schematic defines the contents of the FPGA.

To enter an FPGA block in a mixed design

1. From the Draw menu, select Block and place a new block to represent the FPGA.
2. Double-click the block.
3. In the Set Up Block dialog box, do the following:
   a. In the Filename text box, type a file name for the FPGA schematic design.

      Note *XACTstep 5.2/6.0 does not support long file or directory names. When saving Xilinx schematics, do not use more than eight characters or use spaces in a file or directory name.*

   b. From the Type list, select Schematic.
   c. Click OK.



4. Place symbols from the selected Xilinx device family as needed for the FPGA.

Many Xilinx symbols have underlying schematics. To display the underlying schematic for a symbol, double-click the symbol.

How Schematics Sets up the FPGA Block

When you create an FPGA block, Schematics does the following:

- Adds a view named XILINX to the block, which references the file name you specified. The XILINX view is used by the XNF netlister.
- Adds the attribute IMPL=XILINX. This attribute informs the simulation netlister that the block defines the interface of the FPGA.
- Configures the chosen family on the Xilinx child schematic.

- Configures the appropriate symbol libraries on the Xilinx child schematic.

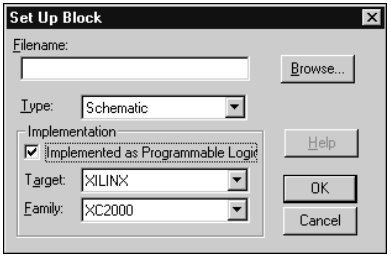
Incorporating an Existing Xilinx Schematic in a Design

There are two ways to incorporate an existing Xilinx schematic into another design:

- Set up the schematic in a block.
- Symbolize the schematic (convert the schematic into a symbol) and then place the symbol.

To create an FPGA block for an existing Xilinx schematic

1. Place a block.
2. Double-click the block.
3. In the Set Up Block dialog box, do the following:
   a. In the Filename text box, type the file name of the existing schematic.
   b. From the Type list, select Schematic.
   c. Click OK. Schematics automatically generates pins for the interface ports in the subdesign.



To symbolize an existing Xilinx schematic

1. Open the Xilinx schematic in Schematics.
2. From the File menu, select Symbolize to create the symbol and save it to a library.
3. Load the new symbol in the Symbol Editor.
4. From the Part menu, select Attributes and add the following attribute:

   IMPL=XILINX

If you put the symbol in a new library, you need to configure the library. See Configuring a Custom Library on page 2-19.

5 From the Edit menu, select Set Schematic and change the view named DEFAULT to XILINX as follows:

   a In the View Name text box, type XILINX.

   b Click Save View. This creates a XILINX view with the same file name as the DEFAULT view.

   c Select DEFAULT from the list.

   d Click Delete, then click OK.

Using an Existing XNF File in a Schematic

You can create a file of type XNF from a description of an FPGA and attach the file to a block in your schematic. Schematics automatically adds pins corresponding to the interface ports for the FPGA. Then, when you simulate the schematic, the block exhibits the behavior of the described FPGA.

To create an FPGA block for an existing XNF file

1 Place a block.

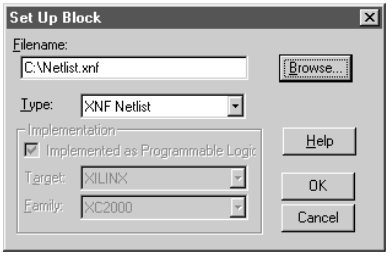
2 Double-click the block.

3 In the Set Up Block dialog box, do the following:

   a In the Filename text box, type the name of the XNF file.

   b From the Type list, select XNF Netlist.

   c Click OK. Schematics creates pins corresponding to the interface ports for the FPGA.



Defining Inputs and Outputs

Interface ports are in the `port.slb` symbol library.

MicroSim FPGA does not include the Xilinx Universal Library pad symbols IPAD, OPAD, and IOPAD. To define pads, use the interface ports IF_IN, IF_OUT, and INTERFACE in the top-most Xilinx schematic. By using the interface ports in a mixed design, the pins on the FPGA block correspond to the pads on the actual device.

Unlike the Xilinx pad symbols, interface ports can be connected to either signals or buses of any width. You can add pad attributes (such as LOC) directly to interface ports.

Note *It is not possible to create I/O pads within child schematics. However, the UPAD (unbonded pad) symbol can be used anywhere within the Xilinx schematic.*

Specifying Xilinx Parameters

The Attribute function in Schematics is used to define the equivalent of Xilinx parameters for an FPGA. You can specify attributes for the following:

- interface ports at the top level
- wires
- symbols
- blocks

Define attributes according to the same usage defined for Xilinx parameters.

Note *Within Xilinx schematics, attribute values cannot be passed from a symbol or block to its child schematic (using @VALUE, for example). In addition, instance attributes are not supported.*

For a description of Xilinx parameters, refer to the *Attributes, Constraints, and Carry Logic* chapter in your XACTstep *Libraries Guide*.

Example: defining the LOC parameter

1. Select the desired symbol.
2. From the Edit menu, select Attributes.
3. In the Name text box for the attribute definition dialog box, type `LOC`.
4. In the Value text box, type the desired value.

   For example, to define LOC as equal to B, you would type `B`. To define LOC as not equal to B, you would type `<>B`. (The <> brackets signify “not equal to.”)

   When entering “not equal” values, be sure to put the <> brackets at the beginning of the Value text box.
5. Click Save Attr.
6. Click OK.

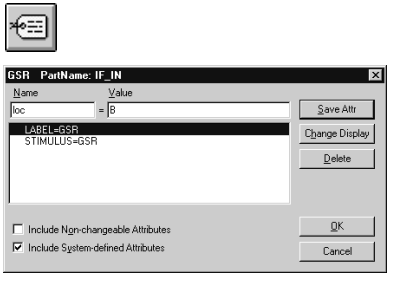


Figure 2-1 *Defining LOC Equal to B in Attribute Definition Dialog Box*

Changing the Device Family for an FPGA

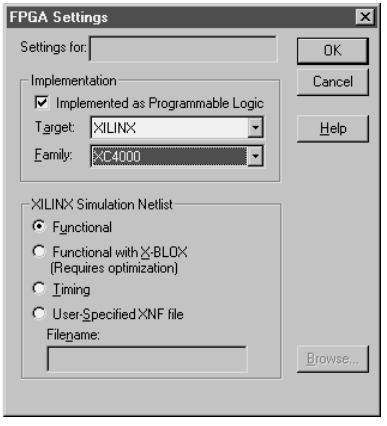
This procedure describes how to change the device family and libraries for the current top-level schematic and all subschematics contained within blocks.

The Xilinx Design Manager does not allow you to change the device family for an existing project. You can get around this limitation either of the following ways:

- Use the Design Manager to delete the old project.
- Rename the top-level Xilinx schematic. This creates a new subdirectory for the Xilinx files.

  If the schematic has a parent FPGA block in a mixed design, also change the file name in the view named XILINX on the parent block.

To change the device family



1. From the Tools menu, select FPGA Settings.
2. In the FPGA Settings dialog box, select a different device family for the Family list.
3. Click OK. Any open macro schematics are closed.

Error messages are generated if the schematic contains symbols which are not part of the new family's library. These symbols must be replaced with symbols from the new family. If error messages are displayed in the Message Viewer, double-click the error messages to locate the missing symbols.

Note *Use Create All XNF from the Tools menu to make sure that all XNF files come from the correct family.*

Using X-BLOX

X-BLOX allows you to use a library of high-level modules to capture your design. This section describes the use of X-BLOX within MicroSim FPGA.

Refer to your Xilinx *X-BLOX Reference/User Guide* for more information about using X-BLOX.

The X-BLOX Library

X-BLOX is supported on XC3000A, XC3000L, XC3100A, all XC4000, and XC5200 devices. When you select the XC3000, XC4000, or XC5200 family in the FPGA Settings dialog box, the X-BLOX symbol library (`xblox.slb`) is configured automatically as a local library.

Note *The X-BLOX symbol library is configured for the XC3000 family even though X-BLOX does not support all XC3000 devices.*

Using Buses

X-BLOX requires buses to be specified without a size. When connecting bus pins between X-BLOX symbols, you can use either buses or wires. To improve traceability, all X-BLOX buses should be labeled.

When connecting X-BLOX buses to non-X-BLOX buses or interface ports, use the bus interface symbols, such as BUS_IF02 or BUS_IF08.

The X-BLOX symbols INPUTS and OUTPUTS are not available in the MicroSim FPGA library. All input and output signals must pass through interface ports, which take the place of pads in MicroSim FPGA.

Example: To connect an X-BLOX bus to output pads, you would do the following:

1. Connect the X-BLOX bus to a regular bus using a BUS_IFnn symbol.
2. Connect the regular bus to the appropriate OBUF symbol(s).
3. Connect the OBUF symbols to an IF_OUT port.

Using X-BLOX Attributes

The majority of the X-BLOX symbols included with MicroSim FPGA do not have pre-defined attributes (such as ASYNC_VAL and STYLE) as shown in the Xilinx *X-BLOX User/Reference Guide*.

You can add attributes to an X-BLOX symbol using the normal attribute mechanism in Schematics.

X-BLOX Example

The EXAMPLES\XILINX\FIBXBLOX subdirectory contains the schematic file `fibonacc.sch`, which is an example of a Fibonacci sequence generator circuit using X-BLOX. This example includes functional and timing XNF netlist files so that you can run functional or timing simulations without running the XACTstep program.

If you make changes to the design, you need to run XACTstep through the Optimize phase before doing a functional simulation. To do this, run the Flow Engine from the XACTstep Design Manager and click Step. This runs the X-BLOX program and synthesizes the X-BLOX modules into primitives which can be simulated by PSpice A/D.

Using MEMGEN

The XACTstep utility MEMGEN is used to create RAMs and ROMs. The MEMGEN program produces an XNF netlist file which you can attach to a block in a schematic.

Where to Put MEMGEN XNF Files

The XACTstep program merges the MEMGEN-produced XNF file with the XNF files generated for the schematic-based portion of the design. It is important that XACTstep be able to locate the XNF file you create using MEMGEN.

When Schematics creates the XNF netlist, all XNF files are put in a separate subdirectory below the directory containing the top-level Xilinx schematic. This subdirectory has the name of the top-level Xilinx schematic. For example, you could have a top-level Xilinx schematic named as follows:

```
d:\projects\myfpga.sch
```

For this schematic, the XNF files would be put in the following subdirectory:

```
d:\projects\myfpga\
```

When you add the FILE attribute to a block, the location of the XNF file name you specify is relative to this subdirectory. This makes the XNF file subdirectory the easiest place to put the MEMGEN-produced XNF file because you can then specify the XNF file name in the FILE attribute without a path.

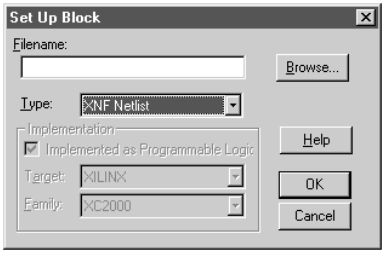
Note *If you have not created the XNF netlist, the XNF file subdirectory will not exist. If so, create the subdirectory before running MEMGEN and then save the MEMGEN XNF in the new directory.*

To Attach a MEMGEN XNF file to a Block

It is not necessary for MEMGEN to create a symbol.

1. Run MEMGEN to produce an XNF file.
   - Make sure that the XNF file name is not the same as any schematic files in your design.
2. From the Draw menu, select Block and place a block on your schematic.
3. Double-click the block to display the Set Up Block dialog box.
4. From the Type list, select XNF Netlist.
5. In the File Name text box, type the name of the XNF netlist file.
6. Click OK.
7. Add a pin to the block for each address line on the RAM or ROM. Name the pins A0 through A*n*-1, where *n* is the number of address pins.
8. Add a pin to the block for each output of the RAM or ROM. Name the pins O0 through O*m*-1, where *m* is the number of output pins.

Note *You can also connect a bus to the block. Name the address pin A[n-1:0] and the output pin O[m-1:0].*



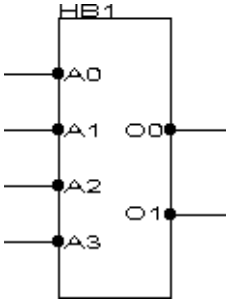


Figure 2-2 *Block Set Up for MEMGEN XNF File*

Using the TIMESPEC and TIMEGRP Symbols

The TIMESPEC and TIMEGRP symbols allow you to specify timing constraints for use by the XACT-Performance utility.

You can also place TNM attributes on primitive symbols, macro symbols, or wires using Attributes from the Edit menu.

For more information about TIMESPEC, TIMEGRP, and the XACT-Performance utility, refer to your XACTstep *Development System Reference Guide, Volume 1*.

To use the TIMESPEC symbol

1. Place a TIMESPEC symbol on your schematic.
2. Click the symbol to select it, then, from the Edit menu, select Attributes.
3. In the Name text box, type a valid TS identifier, such as TS01.
4. In the Value text box, type a valid TS from-to specification, such as FROM:PADS:TO:FFS=20NS.
5. Click Save Attr, then click OK.
6. New attributes appear at the upper left corner of the symbol. Select the attribute value and drag it into an empty row in the TIMESPEC.

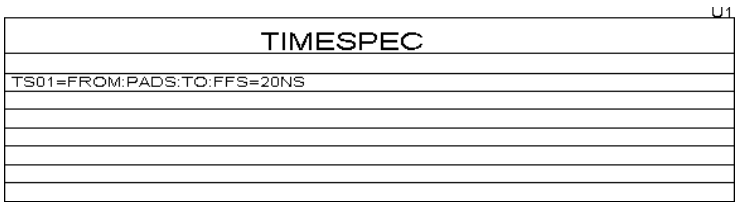


Figure 2-3 *TIMESPEC Symbol with TS01 Attribute*

To use the TIMEGRP symbol

1. Place a TIMEGRP symbol on your schematic.
2.  Click the symbol to select it, then, from the Edit menu, select Attributes.
3. In the Name text box, type a valid group name, such as FLOPS.
4. In the Value text box, type a valid group definition, such as FFS1:FFS2.
5. Click Save Attr, then click OK.
6. New attributes appear at the upper left corner of the symbol. Select the attribute value and drag it into an empty row in the TIMEGRP.

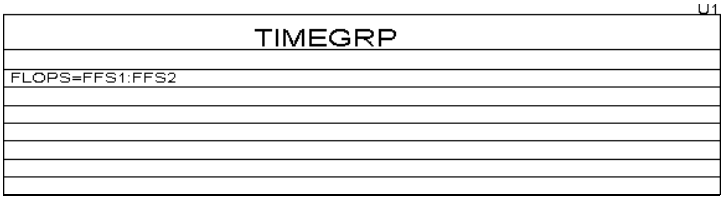


Figure 2-4 *TIMEGROUP Symbol with FLOPS Attribute*

Creating Your Own Symbol Library

The process of creating your own symbol library consists of the following:

1. Add symbols to the library either of the following ways:
   - Copy a symbol from the Xilinx Universal Library.
   - Create a macro symbol.
2. Configure the library

Any library you create must be compatible with a specific Xilinx device family, and you cannot mix symbols from multiple device families in a library. For example, you could create a library of XC4000 symbols.

Adding Symbols to Your Library

To copy a symbol from the Xilinx Universal Library

1. In the Symbol Editor, from the Edit menu, select Copy to copy the symbol.
2. Create the schematic to be represented by the symbol. The schematic must contain only symbols from the same family. For example, if you are creating a schematic for a custom XC4000 macro, you must use XC4000 symbols in the schematic.
3. In the Symbol Editor, associate the symbol with a subschematic using Set Schematic from the Edit menu.
4. In the Symbol Editor, from the Part menu, select Attributes and do the following:
   - If present, remove the LIBVER attribute for the symbol.
   - If not present, add an IMPL=XILINX attribute for the symbol.

To create a custom macro symbol

1. Create the symbol.
2. Create the schematic to be represented by the symbol. The schematic must only contain symbols from the same family. For example, if you are creating a schematic for a custom XC4000 macro, you must use XC4000 symbols in the schematic.
3. In the Symbol Editor, from the Part menu, select Attributes and add an IMPL=XILINX attribute for the symbol.

Directory Structure for Custom Libraries

All symbol and schematic files for a library must be in a separate directory. Figure 1-1 on page 1-2 shows the directory structure for Xilinx device families with each device family library in its own directory. If you have a custom library, you need to put all files for the library in its own directory; Figure 2-5 shows two directories for custom libraries (MY3KLIB and MY4KLIB) along with directories for Xilinx device family libraries.

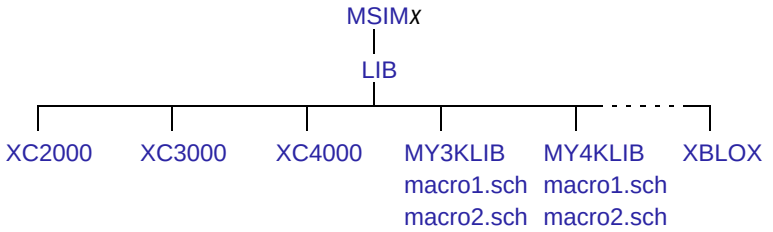


Figure 2-5 *Directory Structure for Custom Libraries*

The macro schematics and other files in custom directories do not need to be uniquely named across all directories. As shown in Figure 2-5, there are files named macro1.sch and macro2.sch in both the MY3KLIB and MY4KLIB directories.

Configuring a Custom Library

You can configure a custom library for use by any design (global) or use it for one design only (local).

To configure a custom library for use by any design (global):

1. Exit Schematics.
2. Use a text editor to modify `msim.ini` in the MSIM*x* directory.

   In the [XILINX] section of `msim.ini`, add the directory name for the custom library on the appropriate line. For example, if your custom library has macro symbols for the XC4000 family and the library files are in a directory named MY4KLIB, you would modify the XC4000 line in the [XILINX] section like this:

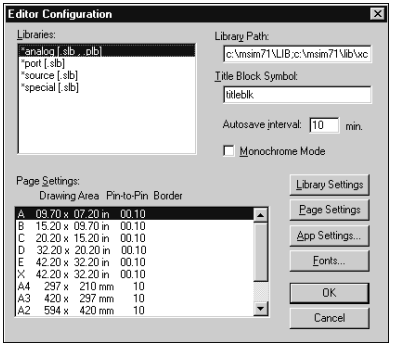
   ```
   XC4000=XC4000;XBLOX;MY4KLIB
   ```

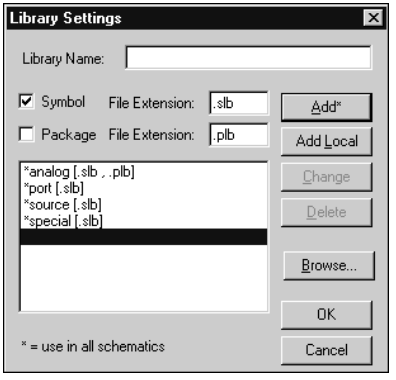
3. Restart Schematics.
4. From the Options menu, select Editor Configuration.
5. In the Library Path text box, add the path for the directory containing the custom library.
6. Click OK.

To configure a custom library for use for one design (local):

1. With the design displayed in Schematics, from the Options menu, select Editor Configuration, then click Library Settings.
2. In the Library Name text box, type the name of your custom library.
3. Click Add Local.

Figure 2-5 describes the directory structure to use for a custom library.





Running XACTstep

3

Overview

This chapter describes how to start XACTstep from within Schematics. This chapter includes the following:

Setting Up an FPGA in XACTstep on page 3-2

Regenerating All Netlists without Running XACTstep on page 3-4

Setting Up an FPGA in XACTstep

Starting XACTstep

Once you have entered the circuit design for an FPGA, you can start XACTstep from Schematics by selecting Run XACTstep from the Tools menu.

XACTstep uses XNF netlist files for a schematic. When you select Run XACTstep for the first time, the XNF files are created for the schematic.

Each time you select Run XACTstep after that, the schematic and the XNF files are checked to see if the XNF files require updating. If changes have been made to the schematic which affect the XNF files, the XNF files are updated and XACTstep is started. If no changes have been made, XACTstep is started without updating the XNF files.

You can also use Create All XNF from the Tools menu to update XNF files without starting XACTstep. See Regenerating All Netlists without Running XACTstep on page 3-4 for more information.

Setting Up and Translating an FPGA in XACTstep

1. With the schematic displayed, select Run XACTstep from the Tools menu.

   If you have a mixed design with two or more FPGA blocks, a dialog box appears that requires you to select the subdesign. XNF files are created or updated as needed, and XACTstep is started.

2. Select New Design from the File menu.
3. In the New Project dialog box, type a project name.
4. For the input design, do the following:

   a. Click Browse, and set the Files of Type list to XNF.

b. Select the subdirectory which has the Xilinx files for the schematic.

c. Select *file*.xnf, where *file* is the file name for the top-level Xilinx schematic. For example, if the schematic is `fpga.sch`, you would select `fpga.xnf`.

5. For Target Family, select the device family used for the schematic.
6. Leave the Design Uses Unified Library check box selected. If it is disabled, click it to enable it.
7. Click Translate.
8. In the Translate Options dialog box, clear the Read Part From Design check box.
9. Click Select Part, and complete the Part Selector dialog box as needed. Be sure to set the device family to the same family used for the schematic.
10. Change Preserve Floor Plan as needed.
11. Click OK.

Note *The XACTstep Design Manager does not automatically load the design you are currently working on. It reloads the last project worked on in XACTstep.*

Xilinx files for a schematic are in a subdirectory under the `.sch` file for the schematic, and the Xilinx file subdirectory has the same name as the schematic file name (without the `.sch` extension). For example, if this is your schematic file:

```
D:\PROJECTS\fpga.sch
```

then Xilinx files for the schematic would be in the following directory:

```
D:\PROJECTS\FPGA\
```

Regenerating All Netlists without Running XACTstep

When you select Run XACTstep from the Tools menu in XACTstep, the system normally detects whether the schematic has been changed and the XNF files for the schematic require updating.

If you want, you can also use Create All XNF from the Tools menu to regenerate the XNF files. Create All XNF does the following:

- It forces regeneration of the XNF files, whether they appear to need updating or not.
- It does not start XACTstep.

Simulation

4

Overview

This chapter describes how to set up and run simulation of FPGA designs.

Specifying Stimulus on page 4-2 provides information about specifying stimulus for an FPGA design.

Special Signals on page 4-3 describes how to specify global signals and access oscillators.

Running Simulations on page 4-7 describes how to set up and run functional and timing simulations for an FPGA design.

After simulating an FPGA design, you can display the results of the simulation in MicroSim Probe. Xilinx Net Names in Probe on page 4-9 describes the FPGA net names as they appear in Probe.

Specifying Stimulus

For information about specifying stimulus for an FPGA design, refer to sections on specifying stimulus for digital devices in your PSpice user's guide.

Note *Do not use stimulus devices such as DIGSTIM in an FPGA-only design.*

Special Signals

Specifying Global Signals

Specifying Global Reset on XC2000 and XC3000

The XC2000 and XC3000 devices have a dedicated RESET (overbar) pin which is represented in the simulation netlist by the signal named GR. To clear the flip-flops at the beginning of the simulation, the GR signal must have a stimulus defined. If a stimulus definition does not exist for the GR signal, PSpice A/D produces the following error message:

```
Can't find .STIMULUS "GR_designname" definition
```

where *designname* is the name of the FPGA design.

To add the GR stimulus definition

1. In Schematics, from the Analysis menu, select Edit Stimuli to start the Stimulus Editor.
2. From the Stimulus menu, select New to display the New Stimulus dialog box.
3. In the Name text box, type GR_*designname*.
4. Change the Type to Signal. Because the GR signal is active low, leave the Initial Value defined as 0.
5. Click OK.
6. Click the Add Transition button on the toolbar.
7. Add a transition to 1 at an appropriate time.

Specifying Global/Set Reset on XC4000 and XC4000E Devices

The XC4000/XC4000E devices have a global set/reset signal (GSR) which you can access by placing a STARTUP symbol on the schematic and driving its GSR pin. Pulsing this pin high returns all flip-flops to their initial (set or reset) state.

You do not have to use the GSR pin on the STARTUP symbol to simulate XC4000/XC4000E designs. The simulation netlist contains a default signal (named GSR_I) that automatically pulses the GSR pins on all flip-flops at the beginning of the simulation.

Specifying Global Reset on XC5200 Devices

The XC5200 devices have a global reset signal (GR) which you can access by placing a STARTUP symbol on the schematic and driving its GR pin. Pulsing this pin high returns all flip-flops to their initial reset state.

You do not have to use the GR pin on the STARTUP symbol to simulate XC5200 designs. The simulation netlist contains a default signal (named GR_I) that automatically pulses the GR pins on all flip-flops at the beginning of the simulation.

Specifying Global Tristate on XC4000, XC4000E, and XC5200 Devices

The XC4000, XC4000E, and XC5200 devices have a global tristate control signal (GTS), which you can access by placing a STARTUP symbol on the schematic and driving its GTS pin. Setting this pin low disables all IOB outputs.

You do not have to use the GTS pin on the STARTUP symbol to simulate XC4000, XC4000E, or XC5200 designs. The simulation netlist contains a default signal (named GTS_I) that is set high at the beginning of the simulation, and thus enables all of the outputs for the duration simulation.

Accessing Oscillators

Using the OSC (XC2000/XC3000)

The OSC symbol is used to configure the oscillator amplifier in the XC2000 and XC3000 family devices. The output of the OSC symbol must be connected to the ACLK symbol input.

The Universal Library GXTL symbol is not supported by MicroSim FPGA. Use the OSC symbol driving an ACLK symbol instead.

To define stimulus values for the OSC

1. Place an OSC symbol on the schematic and double-click it.
2. Type a name for the stimulus. Click OK to start the Stimulus Editor.
3. Select Clock as the stimulus type and click OK.
4. Type values as needed for Frequency, Duty Cycle, and the other parameters, and click OK.

The default name is the reference designator on the symbol. You may want to enter a more descriptive name.

Using the OSC4 (XC4000/XC4000E)

The OSC4 symbol is used to access the internal clock generator in the XC4000 and XC4000E families. The symbol has five outputs: 8 MHz, 500 KHz, 16 KHz, 490 Hz, and 15 Hz. When you use one of these symbols, the simulation netlist automatically includes the appropriate stimulus to drive the connected outputs at the correct frequencies.

Using the OSC5 (XC5200)

OSC5 is implemented as a primitive in MicroSim FPGA.

The OSC5 symbol is used to access the internal clock generator in the XC5200 device family. The symbol has two outputs: OSC1 and OSC2. The frequency of OSC1 and OSC2 can be controlled by specifying values in the DIVIDE1_BY and DIVIDE2_BY attributes, respectively. These values divide the internal 16 Mhz clock to produce the final output frequency.

The similar CK_DIV symbol is used to divide a clock signal input, instead of the internal 16 Mhz clock.

Legal values for DIVIDE1_BY and DIVIDE2_BY are shown in Table 4-1 and Table 4-2.

Table 4-1 *OSC1 Frequencies for DIVIDE1_BY Values*

DIVIDE1_BY Value	OSC1 Frequency
4	4 MHz
16	1 MHz
64	250 kHz
256	63 kHz

Table 4-2 *OSC2 Frequencies for DIVIDE2_BY Values*

DIVIDE2_BY Value	OSC2 Frequency
2	8 MHz
8	2 MHz
32	500 kHz
128	125 kHz
1024	16 kHz
4096	4 kHz
16384	1 kHz
65536	244 Hz

Running Simulations

FPGA-specific simulation settings are specified using the FPGA Settings dialog box. To access the FPGA Settings dialog box, select FPGA Settings from the Tools menu.

Note *For a design using X-BLOX symbols, the design must be optimized in XACTstep before running simulation.*

In the XILINX Simulation Netlist frame, select one of the following options.

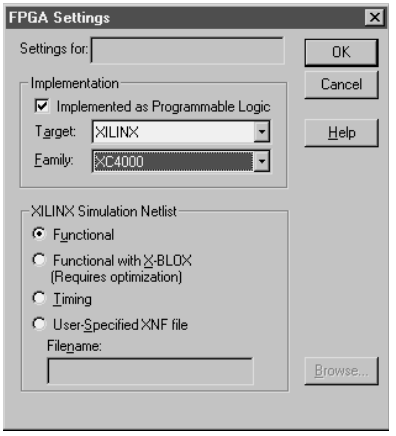
Select this...	To do this...
Functional	Run a functional simulation using the most recent functional information.
Functional with X-BLOX*	Run a functional simulation for a design using X-BLOX symbols.
Timing	Run a timing simulation using the latest timing file from XACTstep.
User-specified XNF File **	Enter an XACTstep generated file name and run a simulation based on the information in the specified file.

*. You must run XACTstep and optimize the design before using the Functional with X-BLOX option.

**. Whether a timing or functional simulation will be run is based on whether the XNF file you specified has timing information in it.

When you are finished with the FPGA Settings dialog box, you are ready to setup and run a transient analysis:

1. From the Analysis menu, select Setup to set up the transient analysis and any simulation options.
2. From the Analysis menu, select Simulate to run the simulation.



The FPGA Settings dialog box applies only to the current (active) schematic. If you are working on a mixed design and need to change the simulation settings for an FPGA block, complete the following:

1. Double-click the FPGA block.
2. From the Tools menu, select FPGA Settings.
3. Change the settings for the FPGA.

Using Probe Markers

After place-and-route, some signals can be optimized away. Therefore, if you simulate after place-and-route, you may not have waveform data for some signals.

By default, PSpice A/D saves simulation results for all FPGA designs, with the exception of signals within macro schematics. For this reason, you can place schematic markers on any wire or pin in your schematic, except if it is a macro schematic. You can then see the waveform results displayed in Probe. For more information about schematic markers, see your PSpice user's guide.

Xilinx Net Names in Probe

After simulating an FPGA design, you can display the simulation results in MicroSim Probe. This section describes the net names for FPGA designs in Probe.

Xilinx Net Names for FPGA-Only Designs

When the top-level schematic is a Xilinx schematic, nets are named as follows:

- Net names are unchanged at the top level.

Figure 4-1 shows net name examples for an FPGA-only design.

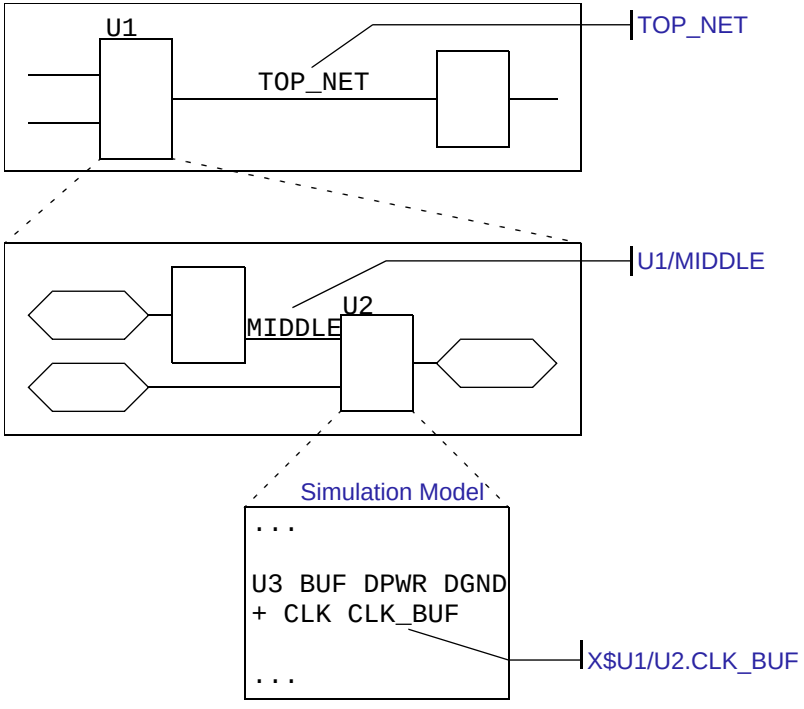


Figure 4-1 *Xilinx Net Names in an FPGA-Only Design*

- Nets on a child schematic are prefixed with the hierarchical path of REFDES attributes of the parent symbols, separated by slashes (/). For example:

  U1/MIDDLE

- PSpice A/D simulation models are used to model XNF primitives. Within those models, there may internal nets used only within the model. These names begin with X$, followed by the REFDES path, followed by a period then the net name.
  For example:

  X$U1/U2.CLK_BUF

Net Names for Mixed Designs

When the top-level schematic is a board-level (non-FPGA) schematic, nets are named as follows:

- Net names are unchanged at the top level.
- A PSpice A/D subcircuit is used to contain the simulation model for the FPGA. All nets within the FPGA symbol or block begin with X_*refdes*. For example:

  X_HB1.

- Nets within the topmost Xilinx schematic are named as they appear on the schematic. For example:

  X_HB1.MIDDLE

- At lower levels of the hierarchy, slashes are used in the hierarchy path. For example:

  X_HB1.U2/BOTTOM

- PSpice A/D simulation models are used to model XNF primitives. Within those models, internal nets can be used only within the model. These names begin with X$, followed by the REFDES path, followed by a period. For example:

  X_HB1.X$U2/U3.CLK_BUF

Figure 4-2 shows net name examples for an FPGA-only design.

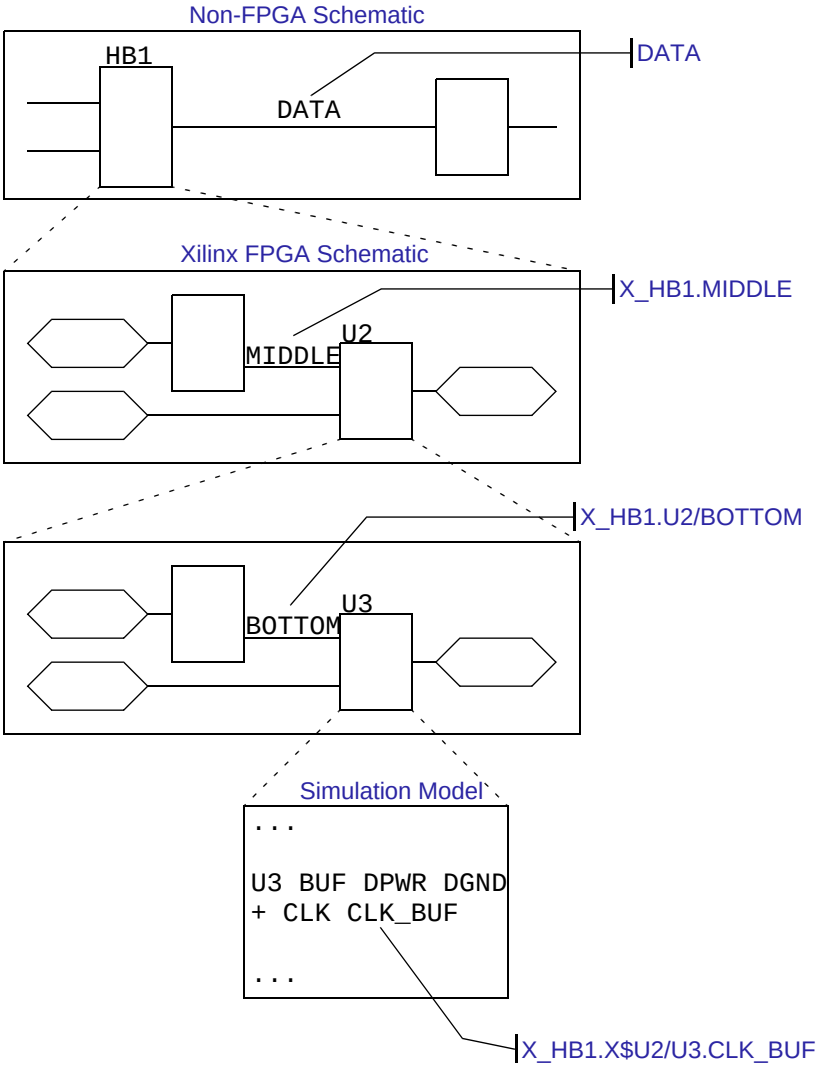


Figure 4-2 *Xilinx Net Names in a Mixed Design*

Glossary

attribute A characteristic or property of a symbol which can be used to identify a device or adjust the operation of a device.

child schematic A schematic represented by a block in a higher level schematic. If a block is placed in a top-level design to represent an FPGA, for example, only the block is visible in the top-level schematic. To display the child schematic, double-click the block.

design entry The process of placing and connecting symbols to represent a circuit.

device family Symbols in the Xilinx Universal Library are grouped into device families, and each FPGA design is targeted to a specific device family. MicroSim FPGA includes the following device families:

- XC2000
- XC3000
- XC4000
- XC4000E
- XC5200

FPGA Field programmable gate array.

FPGA-only design Schematic which includes Xilinx devices only. *See also* **mixed design**.

functional simulation Circuit simulation which identifies potential problems with circuit operation other than timing problems. Functional simulation is typically run prior to performing place-and-route of an FPGA in XACTstep.

MEMGEN XACTstep utility used to create RAMs and ROMs. The MEMGEN program produces an XNF netlist file which you can attach to a block in a schematic.

MicroSim FPGA DesignLab feature that allows you to use MicroSim Schematics and MicroSim PSpice A/D to design Xilinx FPGAs. The FPGA devices are implemented with Xilinx XACTstep. MicroSim FPGA provides an interface between Schematics and XACTstep.

mixed design Schematic which includes a block representing an FPGA within a design that can also include analog and discrete digital devices. The block's child schematic defines the contents of the FPGA. *See also* **FPGA-only design**.

parameters Used in Xilinx programs; the equivalent of symbol attributes defined in MicroSim programs.

place-and-route Conversion of symbols for a circuit design into specific parts. Place-and-route for FPGAs is performed with XACTstep.

Probe MicroSim program used to display simulation results.

PSpice A/D MicroSim program used to simulate circuit designs, including FPGAs.

symbol library Collection of symbols used to enter circuit designs. For MicroSim FPGA, each symbol library corresponds to a particular Xilinx device family.

timing simulation Circuit simulation which identifies propagation delays that can affect circuit operation. For FPGAs, timing simulation is run after performing place-and-route in Xilinx XACTstep.

XACTstep Program developed by Xilinx which is used in combination with MicroSim programs to design and implement FPGAs. Schematics with MicroSim FPGA and PSpice A/D are used to enter and simulate FPGA designs, and the FPGA devices are implemented with XACTstep. MicroSim FPGA supports XACTstep 5.2/6.0.

X-BLOX A library of high-level modules which can be used for FPGA design entry. X-BLOX is supported on XC3000A, XC3000L, XC3100A, all XC4000, and XC5200 devices. *See also* **device family**.

Xilinx Universal Library Library of primitive symbols and macro symbols developed by Xilinx for design of FPGAs and included with MicroSim FPGA. *See also* **device family**.

XNF Xilinx netlist file, a file which contains a textual representation of the symbols, signals, and attributes in a schematic.

Index

A

ACLK symbol, 2-3
ASYNC_VAL attribute, 2-12
attributes
  ASYNC_VAL, 2-12
  DIVIDE*n*_BY, 4-6
  IMPL=XILINX, 2-5, 2-6, 2-17, 2-18
  LIBVER, 2-17
  LOC, 2-8, 2-9
  REFDES, 4-10
  specifying, 2-9
  STYLE, 2-12
  TMN, 2-15
  TS (for TIMESPEC), 2-15
  X-BLOX, 2-12

B

blocks
  attaching a MEMGEN XNF file to, 2-14
  creating for a Xilinx schematic, 2-6
  creating for an XNF file, 2-7
  setting up for an FPGA, 2-5
  specifying attributes for, 2-9
BUS_IF*nn* symbols, 2-11
buses, 2-14

C

CK_DIV symbol, 2-3, 4-6

D

design flow, 1-4
device configuration, 1-4
device families
  changing the family for an FPGA, 2-10
  directory structure, 1-2
  selecting for FPGA-only design, 2-4
  supported by MicroSim FPGA, 2-3
DIVIDE*n*_BY attribute, 4-6

E

error messages, 2-10, 4-3

F

FPGA-only designs
entering, 2-4
specifying stimulus for, 4-2
Xilinx node names for, 4-9
FPGAs
See Also FPGA-only designs, mixed designs
design flow, 1-4
device configuration, 1-4
functional simulation, 1-4, 4-7

G

Global Reset (GR signal), specifying for simulation, 4-3, 4-4
Global Set/Reset (GSR signal), specifying for simulation, 4-4
Global Tristate (GTS signal), specifying for simulation, 4-4
GXTL symbol, 2-3

I

IF_IN symbol, 2-8
IF_OUT symbol, 2-8, 2-11
IMPL=XILINX attribute, 2-5, 2-6, 2-17, 2-18
inputs, 2-8
INPUTS symbol, 2-3, 2-11
installing MicroSim FPGA, 1-2
interface ports, 2-3, 2-8, 2-9, 2-11
INTERFACE symbol, 2-8
IOB symbol, 4-4
IOPAD symbol, 2-3, 2-8
IPAD symbol, 2-3, 2-8

L

LIBVER attribute, 2-17
LOC attribute, 2-8, 2-9

M

macro symbol, creating, 2-18
MEMGEN, 2-13
mixed designs
considerations when changing the device family, 2-10
creating an FPGA block for a Xilinx schematic, 2-6
creating an FPGA block for an XNF file, 2-7
defining inputs and outputs, 2-8
entering, 2-5
Xilinx node names for, 4-10
msim.ini, 2-19

N

naming conventions, 2-2
netlist files, *See* XNF files, 1-4
node names in Probe, 4-9

O

OBUF symbols, 2-11
OPAD symbol, 2-3, 2-8
optimizing an X-BLOX design, 2-12, 4-7
OSC and OSC*n* symbols, 2-3, 4-5
outputs, 2-8
OUTPUTS symbol, 2-3, 2-11

P

parameters, *See* attributes
place and route, 1-4
Probe, Xilinx node names in, 4-9

R

RAM, adding with MEMGEN, 2-13
REFDES attribute, 4-10
reference designators, 2-2
ROM, adding with MEMGEN, 2-13

S

signal names, 2-2
simulation, 1-4, 4-7
STARTUP, 4-4
STARTUP symbol, 4-4
stimulus, specifying for an FPGA, 4-2
STYLE attribute, 2-12
symbol libraries
configuring as global, 2-19
configuring as local, 2-4, 2-19
creating your own, 2-17

library directories, 1-2
X-BLOX, 2-11
Xilinx Universal Library, 1-4
symbolizing a Xilinx schematic, 2-6
symbols
ACLK, 2-3
BUS_IF*nn*, 2-11
CK_DIV, 2-3, 4-6
copying from Xilinx Universal Library, 2-17
creating a macro symbol, 2-18
GXTL, 2-3
IF_IN, 2-8
IF_OUT, 2-8, 2-11
INPUTS, 2-3, 2-11
INTERFACE, 2-8
IOB, 4-4
IOPAD, 2-3, 2-8
IPAD, 2-3, 2-8
OBUF, 2-11
OPAD, 2-3, 2-8
OSC and OSC*n*, 2-3, 4-5
OUTPUTS, 2-3, 2-11
specifying attributes for, 2-9
STARTUP, 4-4
TIMEGRP, 2-15
TIMESPEC, 2-15
UPAD, 2-8

T

TIMEGRP symbol, 2-15, 2-16
TIMESPEC symbol, 2-15
timing simulation, 1-4, 4-7
TNM attribute, 2-15
translating an FPGA, 3-2
TS attribute for TIMESPEC, 2-15

U

UPAD symbol, 2-8

X

XACT Performance utility, 2-15
xactinit.dat, 1-3
XACT-Performance utility, 2-15
XACTstep
device configuration, 1-4
installing, 1-2, 1-3
modifying xactinit.dat, 1-3
optimizing for X-BLOX, 2-12, 4-7
place and route, 1-4
starting, 3-2
translating an FPGA, 3-2
X-BLOX
attributes, 2-12
example, 2-12
optimizing in XACTstep, 2-12, 4-7
symbol library, 2-11
using buses, 2-11
XBLO-X
See Also device families
XC2000 device family
See Also device families
specifying Global Reset (GR signal), 4-3
XC3000 device family
See Also device families
specifying Global Reset (GR signal), 4-3
using X-BLOX with, 2-11
XC4000 device family
See Also device families
specifying Global Set/Reset (GSR signal), 4-4
XC4000E device family
See Also device families
specifying Global Set/Reset (GSR signal), 4-4
XC5200 device family
See Also device families
specifying Global Reset (GR signal), 4-4
Xilinx node names in Probe, 4-9
Xilinx parameters, *See* attributes
Xilinx Universal Library, 1-4, 2-3, 2-17
XNF files
creating/updating, 1-4, 3-2, 3-4
for MEMGEN designs, 2-13
for X-BLOX example, 2-12
specifying for simulation, 4-7