



Release Document

XACTstep Version 5.2.1/6.0.1
Core Tools & Interfaces
July 1996

Read This Before Installation

Versions and Compatibility

The following master table indicates Xilinx core software with the current version numbers.

Software Versions

Program	Windows Version	DOS Version	Workstation Version
APR	5.2.0	5.2.0	5.2.0
APRLOOP	5.2.0	5.2.0	5.2.0
CstCvt	5.2.1	5.2.1	5.2.1
Design Manager	6.0.1	N/A	N/A
Floorplanner	6.0.1	N/A	5.2.1
Flow Engine	6.0.1	N/A	N/A
Hardware Debugger	6.0.0	N/A	N/A
HM2RPM	5.2.1	5.2.1	5.2.1
LCA2XNF	5.2.1	5.2.1	5.2.1
MakeBits	5.2.1	5.2.1	5.2.1
MakePROM	5.2.1	5.2.1	5.2.1
MAP2LCA	5.2.0	5.2.0	5.2.0
MemGen	5.2.1	5.2.1	5.2.1
PPR	5.2.1	5.2.1	5.2.1
PROM File Formatter	6.0.0	N/A	N/A
Report Browser	6.0.1	N/A	N/A
SymGen	5.2.1	5.2.1	5.2.1
Timing Analyzer	6.0.0	N/A	N/A

Program	Windows Version	DOS Version	Workstation Version
XACT	5.2.1	5.2.1	5.2.1
XBLOX	5.2.1	5.2.1	5.2.1
XChecker	5.2.0	5.2.0	5.2.0
XCK88	N/A	5.2.0	N/A
XDE	5.2.1	5.2.1	5.2.1
XDelay	5.2.1	5.2.1	5.2.1
XDM	N/A	N/A	5.2.1
xdm	5.2.1	5.2.1	5.2.1
XEMake	N/A	N/A	5.2.1
XEMake6	6.0.1	N/A	N/A
XKey	6.0.0	5.2.1	N/A
XMake	5.2.0	5.2.0	5.2.0
XNFBA	5.2.1	5.2.1	5.2.1
XNFCvt	5.2.0	5.2.0	5.2.0
XNFMAP	5.2.0	5.2.0	5.2.0
XNFMerge	5.2.0	5.2.0	5.2.0
XNFPrep	5.2.1	5.2.1	5.2.1
XPP	5.2.0	5.2.0	5.2.0
XPrint	5.2.1	5.2.1	5.2.1
XSimMake	5.2.1	5.2.1	5.2.1
Cadence-Related Beta Versions			
X2Vprep	NA	NA	9504-5.2c
XNF2Verilog	NA	NA	9504-1.30w
funcnetx	NA	NA	5.5.b
timenetx	NA	NA	5.2.1.b

Contents

	Software Versions	iii
Chapter 1	Introduction	
	Contents	1-1
	Software	1-1
	Hardware	1-2
	Documentation	1-2
	Online Documentation	1-2
	Xilinx PC Protection Key	1-3
	Maintenance and Support	1-3
Chapter 2	Installation	
	Disk Space	2-1
	Installing the 5.2.1/6.0.1 Software	2-2
	Installing the 5.2.1/6.0.1 Online Documentation	2-3
	Installing the ES-Verilog-XL Interface	2-3
	Installation Steps	2-3
	Environment Setup	2-5
	Using the Interface	2-5
	Important Information for XC4000E Simulation	2-5
Chapter 3	Features in This Release	
	XACT Software	3-1
	XDE	3-1
	Synchronous and Dual-Port RAMs	3-1
	Set Memory Command Initializes All XC4000E RAMs	3-2
	H Multiplexers	3-3
	IOB Clock Enable	3-3
	IOB O/EC Routing Model	3-3
	XC4000 to XC4000E Conversion	3-4
	CMOS Thresholds	3-4

Start-Up Sequence Options	3-5
MemGen	3-5
SYNC_RAM and DP_RAM	3-5
Default and Data Commands	3-6
Symbols	3-7
Pin Names of RAM Components	3-7
XBLOX	3-8
SYNC_RAM	3-8
DP_RAM	3-8
Conditions for Implementation in an IOB	3-9
PPR	3-9
Guided Design	3-9
Faster Routing for Difficult XC4000 and XC4000E Designs	3-10
XACT-Performance	3-10
Asynchronous RAMs	3-11
Single-Port Synchronous RAMs	3-11
Dual-Port Synchronous RAMs	3-12
CAE Tools	3-12
Viewlogic Interface	3-12
XC4000E Library	3-12
PC	3-12
Sun4, HP, and IBM RS6000	3-13
FXC4000E Library	3-13
PC	3-13
Sun4, HP, and IBM RS6000	3-14
Switching to XC4000E from XC4000	3-14
New Symbols For XC4000E	3-15
Location Constraints	3-15
Simulating XC4000E RAM	3-15
Xilinx-Synopsys Interface (XSI)	3-16
Configuring Synopsys to Synthesize and Simulate	
XC4000E Designs	3-16
New Components in XC4000E Synthesis and Simulation	
Libraries	3-17
Restrictions Imposed by Synopsys 3.3 and Later	3-20
Using Synchronous and/or Level-Sensitive RAM	3-20
Mentor Graphics Interface	3-21
Script Changes	3-25
MEN2XNF8, FNCSIM8, TIMSIM8	3-25
PLD_DVE	3-26
GEN_SCH8	3-26
GEN_SYM8	3-26

	Cadence Verilog-XL Interface	3-26
	New Features	3-27
	Script Changes	3-28
	OrCAD Interface	3-28
	XC4000E Library	3-28
	New Symbols for XC4000E	3-29
Chapter 4	Device and Package Support	
	Device and Package Support.....	4-1
Chapter 5	Xilinx Customer Support Information	
	Registration, Authorization, and Customer Service	5-1
	Technical Support	5-1
	Hotline Access	5-1
	Training	5-2

Introduction

Welcome to the XACTstep Core Tools from Xilinx!

Xilinx software products have prefixes to designate the type of products you receive.

- DS = Development System (new system)
- DX = Development System Upgrade (upgrade to current system)
- SC = Support Contract (update to current system)
- SR = Support Re-instatement (update for non-current system)

The labels on the box indicate the product that you have received.

Contents

The Development System (DS) product you received contains software, documentation, and/or hardware. New DS products contain hardware, software, and documentation. Update products have software and documentation only.

Software

The Xilinx software for all platforms is provided on CD-ROM. It consists of the following.

- Installation Program
- FPGA Core Implementation Tools (DS-502)
- Viewlogic Interface and Libraries (DS-391)
- OrCAD Interface and Libraries (DS-35) - PC Only
- Synopsys Interface and Libraries (DS-401) -Workstation Only

- Mentor 8 Interface and Libraries (DS-344) - Workstation Only
- ES-Verilog-XL Interface and Libraries - Workstation Only

Hardware

The hardware consists of the following items.

- JTAG Parallel Download Cable (included in DS-560 and DS-571 for the PC and all other PC platform packages)
- XChecker Download and Readback Cable set (Included in all packages and the DS-560 for the workstation (Not included in the DS-560 and DS-571 for the PC)
- Xilinx “C” Programmable Key (a beige key included in some Base and all Standard packages)

Documentation

The following documentation is provided in printed form for Xilinx Core Tools products.

- *Getting Started & Installation Guide*
- *Additional Products & Services Packet*
- *Release Documents*

Online Documentation

The following online documentation is included with your Xilinx Core Tools products. With this release, Sun Workstation customers are provided with the latest Adobe Acrobat Reader with Search, Version 2.1, and HP users no longer have to install a separate CD for online documentation.

- *Libraries Guide*
- *Libraries Supplement Guide*
- *Floorplanner Reference/User Guide*
- *Design Manager/Flow Engine Reference/User Guide*
- *Timing Analyzer Reference/User Guide*
- *Hardware Debugger Reference/User Guide*

- *PROM File Formatter Reference/User Guide*
- *Development System User Guide*
- *Development System Reference Guide, Vols 1-3*
- *Hardware & Peripherals User Guide*
- *XEPLD Reference Guide*
- *XEPLD Design Guide*
- *XEPLD Design Guide for Windows*
- *XEPLD Schematic Design Guide*
- *XEPLD Schematic Design Guide for Windows*

Note: Xilinx Core FPGA and EPLD documentation for workstation and PC platforms is available online via CD-ROM. Printed documents for CAE tool products are included depending on product configuration.

Selected Xilinx manuals are available in printed form. See the “Documentation Order Form” in the *Additional Products & Services Packet*.

Xilinx PC Protection Key

Xilinx and Viewlogic software are protected by a hardware key for the parallel port of your PC. This key is required to operate the software properly.

Maintenance and Support

This product comes with free technical and product information telephone support (toll-free in the U.S. and Canada). You can also fax and e-mail your questions. See the “Xilinx Customer Support Information” section of this release note for offices and phone numbers.

Most product come with one year of maintenance; you will receive all software and documentation updates automatically during that time. You will receive a notice at the end of the year giving instructions on how to renew your maintenance contract.

Chapter 2

Installation

This section includes information about installation of XACTstep core software from CD-ROM. This software package is basically an update to XACTstep Version 5.2/6.0 with XC4000E architecture support.

Disk Space

To install this XACTstep Release, you need the amount of disk space shown in the following table. All workstation installations require that you extract the tar files.

Tool/Interface	Platform	Disk Space Required
FPGA Core Tools	PC	164 MB
FPGA Core Tools	Sun 4	154 MB
FPGA Core Tools	HP 7	161 MB
FPGA Core Tools	IBM RS6000	160 MB
EPLD	PC	89 MB
EPLD	Sun 4	75 MB
EPLD	HP 7	26 MB
EPLD	IBM RS6000	34 MB
X-BLOX	PC	4 MB
X-BLOX	Sun 4	4 MB
X-BLOX	HP 7	4 MB
X-BLOX	IBM RS6000	6 MB
HW-112	PC	0.4 MB
Viewlogic	PC	65 MB
Viewlogic	Sun 4	34 MB

Tool/Interface	Platform	Disk Space Required
Viewlogic	HP 7	34 MB
Viewlogic	IBM RS6000	40 MB
Synopsys	Sun 4	81 MB
Synopsys	HP 7	80 MB
Synopsys	IBM RS6000	84 MB
Xilinx-ABEL	PC	9 MB
Xilinx-ABEL	Sun 4	15 MB
Xilinx-ABEL	HP 7	11 MB
Mentor Graphics	Sun 4	97 MB
Mentor Graphics	HP 7	99 MB
Cadence ES-Verilog-XL	Sun 4	4 MB
Cadence ES-Verilog-XL	HP 7	4 MB
OrCAD	PC	24 MB

Installing the 5.2.1/6.0.1 Software

Detailed instructions for installing this software on PCs are covered in Chapter 2 of the *Getting Started Installation Guide*, and detailed instructions for workstations are covered in Chapter 3. This manual is included with your package.

Note: HP700 users can now use the identical command sequence for installation outlined for Sun4 users, shown below.

```
# mkdir /cdrom_dir.↓
```

```
# mount -t hsfs -o ro /dev/sr0 /cdrom_dir.↓
```

Type the following to start the Install program:

```
% /cdrom_dir/install.↓
```

In addition, with this release, all the online documentation files are installed on the software CD, unlike the previous release where HP users received the online documentation files on a separate CD.

Installing the 5.2.1/6.0.1 Online Documentation

Instructions for installing the current online documentation are detailed in the Release Document *XACTstep Version 5.2/6.0 Core Tools* included with your Development System package. *All workstation users should follow the instructions for the HP platform, starting on page vi of that document.*

Installing the ES-Verilog-XL Interface

The ES-Verilog-XL Interface and Libraries is supplied as Engineering Software and is not installed by the XACTstep 5.2.1 install program. It is available on the XACTstep 5.2.1 CD in the /cdrom/xbbs/vlog_intf subdirectory.

Installation Steps

1. Check the system requirements defined in the *Getting Started and Installation Guide*.
2. Insert the CD into the drive.
3. Start a window manager (Open Windows or X-Windows).
4. Execute the platform-specific commands as indicated in the following:

Sun4

```
# mkdir /cdrom
# mount -t hsfs -o ro /dev/sro /cdrom
% cd xilinx_install_dir
```

(This changes your directory to the Xilinx install directory).

```
% uudecode /cdrom/xbbs/vlog_intf/xil_vlog_intf.sun4.tar.Z.uu
% uncompress xil_vlog_intf.sun4.tar.Z
% tar xvf xil_vlog_intf.sun4.tar
```

HP7

```
# mkdir /cdrom
# mount /dev/dsk/3s0 /cdrom
% cd xilinx_install_dir
```

(This changes your directory to the Xilinx install directory).

```
% uudecode /cdrom/xbbs/vlog_intfc/xil_vlog_intfc.hp7.tar.Z.uu
% uncompress xil_vlog_intfc.hp7.tar.Z
% tar xvf xil_vlog_intfc.hp7.tar
```

Note: Workstation users must have root privileges to use mount commands (“#” is the root prompt and “%” is the user prompt). Directories and device names can vary, therefore check these names with your system administrator.

5. Xilinx recommends that you use your XACT directory as the install directory. With XACT as the installation directory, the installation creates the following directory structure for you.

`$XACT/platform` (where *platform* is either *sparc* or *hppa*)

funcnetz (functional simulation netlister)

timenetx (timing simulation netlister)

x2vprep

xf2verilog

vlogintdoc.ps (Postscript format file of the 5.20 ES-Verilog documentation and Chapter 4 of the XACT 5.20 Core Tools Release Document.)

data/

xc2000.pin

xc3000.pin

xc4000.pin

xc4000e.pin

xc5200.pin

xc7000.pin

verilog2000/

verilog3000/

verilog4000/

verilog4000e

verilog5200/

```
verilog7000/
```

Environment Setup

In your `.cshrc` file or setup file:

1. Set your path to include the path to the `bin/platform` subdirectory:

For the Sun 4 platform:

```
set path = (install_dir/bin/sparc $path)
```

For the HP 7 platform:

```
set path = (install_dir/bin/hppa $path)
```

2. Define a `CDS_INTFC` environment variable pointing to the install directory.

```
setenv CDS_INTFC install_dir
```

Using the Interface

Examples

```
funcnetx design 4000e (XC4000E design, functional simulation  
netlist generation.)
```

```
timenetx design 5200 (XC5200 design, timing simulation netlist  
generation.)
```

Refer to Chapter 4 of the *XACTstep* version 5.2/6.0 Core Tools release document for complete instructions on using the ES-Verilog interface, or print the postscript format file, `vlogintdoc.ps`, which is included in the ES-Verilog archive.

Note: Post-synthesis functional simulation of Synopsys or other HDL synthesis-originated designs is not supported. Timing simulation netlists may be generated for these designs by running `timenetx` with the `-x` option to skip XNFBA.

Important Information for XC4000E Simulation

Users simulating XC4000E designs utilizing synchronous or dual-port RAM on the Sun4 platform need to install the Verilog-XL Version 2.2.8 patch. This fixes the core dump problem when Verilog-XL is

invoked with the `+neg_tchk` option. Users of the Verilog-XL Version 2.2.1 on the HP 7 platform do not see this problem.

You must specify the `+neg_tchk` and `+splitsuh` options when invoking the Verilog-XL simulator to make the simulator accept negative timing check values in the modified synchronous and dual port RAM models. If these options are not used, XNF2VERILOG issues error messages about delays being negative or too large.

Example: `verilog +neg_tchk +splitsuh mydesign.t.v mydesign.t.stim`

The Verilog-XL patch is available in both configurable and non-configurable versions from the Xilinx ftp site, <ftp.xilinx.com>. If only the non-configurable executable is required, download this file:

`verilog2.28sun4.t.Z.uu`

If you need to be able to configure your Verilog executable or link to PLI routines, download these two files:

`vconfig03.23-p007sun4.t.Z.uu`

`verilogx102.20-s018sun4.t.Z.uu`

The Cadence Verilog-XL archives outlined above are available in the customer download area of <ftp.xilinx.com>. Log in as “customer” with the password “xilinx” to access this area.

Please note that this is a blind area, so you will not be able to see a listing of the available files.

Features in This Release

The main purpose of this XACTstep Supplementary release is to provide full production support for the new XC4000E device family. Additional details covering XC4000E devices is found in the *XC4000E Field Programmable Gate Array Family Data Sheet*.

The XC4000E design flow is the same as the XC4000 design flow. If you received the XC4000E Version 1.0.0 Pre-release, the following information is the same as what was included in the XC4000E pre-release document except for some of the path names.

XACT Software

XDE

This section describes the new tag values that have been added to the XDE program to support the synchronous and dual-port RAMs, the H-MUXes, and the IOB clock enable. It also describes the new SetMemory command, which sets the initial values on RAMs; two options in MakeBits to set TTL/COMS thresholds; and two new options for the start-up sequence.

Synchronous and Dual-Port RAMs

This release offers a complete set of EDITBLK RAM tag values: RAM:DP:K:NOT:F:G:FG.

- RAM:DP indicates a dual-port RAM.
- RAM:K indicates a synchronous RAM.
- RAM:NOT indicates that the K pin is inverted.

However, the new RAM:DP tag value and the existing RAM:FG tag value are mutually exclusive.

Allowed values for XC4000E RAM tag are shown in the following table.

RAM:F:G RAM:F:G:K RAM:F:G:K:NOT	Two 16x1 non-synchronous RAMs Two 16x1 synchronous RAMs Two 16x1 synchronous RAMs, inverted clock
RAM:F RAM:G RAM:FG	One 16x1 non-synchronous RAM One 16x1 non-synchronous RAM One 32x1 non-synchronous RAM
RAM:F:K RAM:G:K RAM:FG:K	One 16x1 synchronous RAM One 16x1 synchronous RAM One 32x1 synchronous RAM
RAM:F:K:NOT RAM:G:K:NOT RAM:FG:K:NOT	One 16x1 synchronous RAM, inverted clock One 16x1 synchronous RAM, inverted clock One 32x1 synchronous RAM, inverted clock
RAM:F:G:DP:K RAM:F:G:DP:K:NOT	One 16x1 synchronous, dual-port RAM One 16x1 synchronous, dual-port RAM, inverted clock

Set Memory Command Initializes All XC4000E RAMs

RAM initialization is implemented by the SetMemory command from the Config menu. The SetMemory command is allowed only inside the block editor. The following table describes the values for the SetMemory options.

Setmemory F G FG value	Non-dual-port RAMs or ROMs. For F and G (16x1 case), a 16-bit value is expected. For FG (32x1), a 32-bit value is expected. G is in the left two bytes of the 32-bit number, and F is in the right two bytes.
Setmemory value	Dual-port RAMs. A 16-bit value is expected and loaded into F and G.

The *value* is a hexadecimal number with a leading “x” character.

In both cases, the MSB of the RAM initial value goes into the high RAM address, as in the following examples.

SETMEMORY F x8000 sets bit 15 of F RAM to 1

SETMEMORY G x0001 sets bit 0 of G RAM to 1.

H Multiplexers

This release offers a complete set of EDITBLK H function tag values: H:F:G:H1:DIN:SR.

- H:DIN indicates that the DIN pin sources the H function generator.
- H:SR indicates that the SR pin sources the H function generator.

However, the new H:DIN tag value and the existing H:F tag value are mutually exclusive. The new H:SR tag value and the existing H:G tag value are mutually exclusive.

Allowed values for XC4000E RAM tag are the following:

H:G	H:DIN:G	H:H1:F:G
H:SR	H:DIN:SR	H:H1:F:SR
H:F	H:H1	H:H1:DIN
H:F:G	H:H1:G	H:H1:DIN:G
H:F:SR	H:H1:SR	H:H1:DIN:SR
H:DIN	H:H1:F	

The H equation (Equate H) for XC4000E takes DIN and SR as arguments. Here are three examples:

```
Equate H = SR
Equate H = DIN
Equate H = (SR*DIN*H1)
```

IOB Clock Enable

The new EDITBLK IOB clock-enable tag values for XC4000E are INFF:EC and OUT:EC.

- INFF:EC indicates that the input register has a clock-enable pin.
- OUT:EC indicates that the output register has a clock-enable pin.

IOB O/EC Routing Model

The XC4000E parts now support a clock-enable function on each IOB that is common to both IOB flip-flops. This function is represented as the new EC pin on the IOB. The interconnect that connects to EC is shared between the EC pin and the existing O pin; this sharing results in some limitations in the use of the interconnect when both EC and O are used on the same IOB. In the Design Editor (XDE), the intercon-

nect attached to the O pin of an IOB is shown with two forks. One fork wraps around the IOB parallel to the interconnect attached to the EC pin and matches it PIP for PIP. The second fork extends into the XC4000E device.

When the IOB clock enable is not in use, any of the PIPs on the O-pin interconnect can be enabled to drive a signal to the O pin.

When both the O pin and the EC pin are in use, the PIPs on the O interconnect fork that match those on the EC interconnect cannot be used to drive a signal to the O pin. They can only connect a net to the EC pin. The only PIPs available to drive a signal onto the O pin are those on the fork of interconnect that extends into the device. A DRC check ensures the correct use of the interconnect in this condition.

XC4000 to XC4000E Conversion

To convert an XC4000 LCA file into an XC4000E LCA file, select the Convert command from the Programs menu, then follow the prompts.

An existing XC4000A/D/H LCA file cannot be converted into an XC4000E LCA file.

The Convert command does not support the conversion from a smaller XC4000E part to a larger part.

CMOS Thresholds

This release allows you to specify the input thresholds for all IOBs on the device by using the MakeBits command from the Programs menu in XDE, as follows:

- INPUT CMOS sets all input thresholds to CMOS.
- INPUT TTL sets all input thresholds to TTL. It is the default.

The XC4000E allows you to specify the output thresholds for all IOBs on the device by using the MakeBits command from the Programs menu in XDE, as follows:

- OUTPUT CMOS sets all output thresholds to CMOS.
- OUTPUT TTL sets all output thresholds to TTL. It is the default.

Note: For additional information see the “MakeBits Should Configure TTL Inputs by Default” item in the *Known Issues* release note.

Start-Up Sequence Options

These new options are available with the start-up sequences for the XC4000E.

- M0 Pin
Setting: **Pullup, Pulldown**
Default: **Neither**
- M2 Pin
Settings: **Pullup, Pulldown**
Default: **Neither**

Note: The Pullup and Pulldown options for the M0 and M2 pins act as toggles and are mutually exclusive. The default is inactive. Selecting one option enables it and disables the other. Selecting the same option a second time disables it.

MemGen

This section describes the new types that have been added to the MemGen program to support the synchronous and dual-port RAMs in the XC4000E architecture. It also provides the pin names of RAM components for instantiating cells in the HDL source code.

SYNC_RAM and DP_RAM

Two new TYPE values, SYNC_RAM and DP_RAM, have been added to the MemGen program. You can specify TYPE values interactively, through the command line, or in the MEM file.

- From the command line, enter the following:

```
type=[rom|ram|sync_ram|dp_ram]
```

Use SYNC_RAM for a single-port, edge-triggered RAM and DP_RAM for a dual-port, edge-triggered RAM.

Here is an example:

```
memgen ram64x1s type=sync_ram memory_depth=64  
word_width=1
```

- In interactive mode, you see the following:

What type of memory do you want to build?

Enter one of the following:

1 for RAMs

2 for ROMs

3 for SYNC_RAMs (XC4000E only)

4 for DP_RAM (XC4000E only)

- In the MEM file, type the following syntax:

type=[rom|ram|sync_ram|dp_ram]

A sample MEM file illustrates this syntax:

TYPE SYNC_RAM; The memory is a SYNC_RAM

DEPTH 64 ; The memory is 64-word deep

WIDTH 1 ; Each memory word is 1-bit wide

;

PART 4005EPG156

;

SYMBOL VIEWLOGIC PINS; Build a VIEWLOGIC symbol with
pin inputs

;

DEFAULT 0 ; Add a default value for unspecified
locations

DATA 0 ; Add your SYNC_RAM data here

;

Default and Data Commands

In the past, the Default and Data commands were only permitted on ROMs. Now they also apply to the XC4000E RAMs, SYNC_RAMs, and DP_RAMs. They are added to support the INIT attribute on all XC4000E RAMs.

Symbols

MemGen generates symbols for the new RAM types, SYNC_RAM and DP_RAM. To generate a symbol file for OrCAD, use the -o option. To generate a symbol file for Viewlogic, use the -v option.

Pin Names of RAM Components

This section describes how MemGen generates the pin names for RAM components. This information explains how pins are instantiated. The pin order is also important; that is, it should match the order that appears in the netlist.

1. MemGen generates pin names for XC4000/XC4000E RAMs from the following user inputs:

Address lines: An

Data lines: Dn

Output lines: On

Write-Enable line: WE

2. MemGen generates pin names for XC4000E SYNC_RAMs from the following user inputs:

Address lines: An

Data lines: Dn

Output lines: On

Write-Enable line: WE

Clock line: WCLK

3. MemGen generates pin names for XC4000E DP_RAMs from the following user inputs:

Address lines: An

Data lines: Dn

DP Read Address lines: DPRAn

SP Output lines: SPOn

DP Output lines: DPOn

Write-Enable line: WE

Clock line: WCLK

XBLOX

This section describes the new TYPE values that have been added to the XBLOX program to support the synchronous and dual-port RAMs in the XC4000E.

Two new modules have been added to X-BLOX, SYNC_RAM and DP_RAM.

SYNC_RAM

The new SYNC_RAM module synthesizes a synchronous read-write static random-access memory.

The inputs, outputs, and attributes are identical to the current SRAM module with the following exceptions.

- A positive edge on WR_CLK outputs data selected by ADDR on D_OUT, and when WR_EN is active, it loads data on D_IN into the RAM. The WR_CLK pin must be connected.
- ADDR_ERROR is not supported.
- The INIT and MEMFILE attributes have been added. See the “Specifying the Initial Contents of an SRAM, SYNC_RAM, or DP_RAM” section later in this document.
- The value of DEPTH must be a multiple of 16.
- CLB utilization is the same as that of the SRAM in XC4000.

DP_RAM

The DP_RAM module synthesizes a synchronous dual-port read-write static random-access memory.

The inputs, outputs, and attributes are identical to the current SRAM module with the following exceptions.

- The following inputs have been added.
 - WR_CLK

A positive edge on WR_CLK outputs data selected by ADDR on SP_OUT, and when WR_EN is active, it loads data on D_IN into the RAM. The WR_CLK pin must be connected.

- **DPRD_ADDR**

The DPRD_ADDR (dual-port read address) port selects the word that appears on DP_OUT. DPRD_ADDR cannot be connected to a bus with ENCODING=ONE-HOT.

- The following outputs have been added or changed.

- **SP_OUT**

The SP_OUT port reflects the addressed word of the RAM specified by ADDR when WR_CLK goes active.

- **DP_OUT**

The DP_OUT port reflects the addressed word of the RAM specified by DPRD_ADDR.

- ADDR_ERROR is not supported.
- The INIT and the MEMFILE attributes have been added. See the “Specifying the Initial Contents of an SRAM, SYNC_RAM, or DP_RAM” section, following.
- The value of DEPTH must be a multiple of 16.
- CLB utilization is twice that of the SRAM in XC4000.

Conditions for Implementation in an IOB

Item 2 on page 4-53 of the *X-BLOX Reference/User Guide* states that the flip-flop does not use its clock-enable pin. This statement is not valid for XC4000E, because clock enable on IOB registers is supported.

PPR

Guided Design

PPR supports the guiding feature for XC4000 or XC4000E designs guiding XC4000E designs in this release. See the *XACTstep Development System Reference Guide*, Volume 2, for information about guided design.

Faster Routing for Difficult XC4000 and XC4000E Designs

For the XC4000 families (including XC4000E), the “router_effort” variable is the only PPR command-line parameter that controls how hard the router works.

Increasing router_effort from 2 (the default) to 3 or 4 allows rip-up-and-retry work to continue longer when the router is trying to reduce unroutes. It also enables a special routing mode that can reduce resource utilization and sometimes improve chip performance.

With PPR 5.2.0 and 5.2.1, it is possible to enable this special routing mode *without* changing the default router effort. To do so, include the line:

```
/ppr/router/script = 3
```

in the file xactinit.dat, in your current working directory.

For a design that routes with difficulty, running with /ppr/router/script = 3 often improves the default result, without the cost in run time typically incurred by using router effort 3 or 4.

For designs that route easily with the default router_effort, /ppr/router/script = 3 is not needed; however, ongoing tests indicate that script = 3 tends to use less resources.

When router_effort is set to 3 or 4, script = 3 is selected automatically; therefore setting /ppr/router/script = 3 introduces no further effect on the result.

XACT-Performance

XACT-Performance support for the XC4000E family is mostly unchanged from its behavior for the XC4000 family, but there are new varieties of RAM to consider.

All grouping mechanisms that apply to RAMs in the XC4000 family also apply to RAMs in the XC4000E family. The RAMS qualifier matches any type of RAM, and the predefined RAMS group includes RAMs of all types.

See the “XACT-Performance Utility” chapter in the *XACTstep Development Reference Guide*, Volume 1, for more information on grouping mechanisms.

For all RAM modes, path-tracing behavior is best considered according to where the RAM in question appears in the paths associated with a particular TIMESPEC:

- Type 1: RAM at the end of paths, for example, “from:others:to:rams”
- Type 2: RAM at the start of paths; for example, “from:rams:to:others”
- Type 3: RAM in the middle of paths, for example, an enclosing “from:ffs:to:ffs”

Asynchronous RAMs

For RAMs configured in asynchronous mode, path-tracing behavior is unchanged from the XC4000.

- Type 1 paths (to RAM) are always traced. PPR determines the setup time appropriate to the destination pin.
- Type 2 paths (from RAM) are always traced. PPR determines the worst-case time from a change on D or WE to data valid.
- Type 3 paths (through RAM) are always traced if they arrive at address pins, and they are not traced if they arrive at D or WE pins. Changes on address inputs propagate just as they do for ordinary look-up tables. However, propagation of changes on D or WE is assumed to be of interest only when the RAM is being read during a write operation. If you want PPR to control the delay on paths through the D or WE inputs, you must split the delay requirement into two segments: one ending at the RAM input pin, and the other beginning at the RAM output.

Single-Port Synchronous RAMs

Path-tracing for synchronous RAMs is essentially the same as for asynchronous RAMs.

- Type 1 paths (to RAM) are always traced. PPR determines setup times with respect to the WCLK pin, for all other inputs.
- Type 2 paths (from RAM) are always traced. PPR determines worst-case time after WCLK transition to data valid.
- Type 3 paths (through RAM) are traced through address pins only.

Dual-Port Synchronous RAMs

Tracing behavior for dual-port synchronous RAMs is as follows.

- Type 1 paths (to RAM) are traced to every input pin destination except DPRA0 to DPRA3. Read address inputs cannot impact paths that end at a RAM (write function).
- Type 2 paths (from RAM) are always traced.
- Type 3 paths (through RAM) are traced through address inputs only. In particular, there are A?-to-SP0 paths and DPRA?-to-DPO paths.

CAE Tools

The following sections outline changes relevant to this release impacting CAE Tools.

Viewlogic Interface

This section describes the Viewlogic library and simulation support for the XC4000E architecture. See the *Libraries Guide Supplement* for details.

XC4000E Library

The XC4000E library is a superset of the XC4000 library, with the addition of XC4000E-specific elements. To access the XC4000E library, follow these platform-specific instructions.

PC

To access the XC4000E library on a personal computer, perform the following steps.

1. Include this line in your viewdraw.ini file:

```
DIR [m] d:\correct_path\unified\xc4000e (xc4000e)
```

where d: is the drive on which this 6.0.1 software is installed.

For example, if the software is installed in d:\xact, your viewdraw.ini file should contain the following lines:

```
DIR [m] d:\XACT\UNIFIED\XC4000E (xc4000e)
DIR [m] d:\XACT\UNIFIED\XBLOX (xblox)
```

```
DIR [m] d:\XACT\UNIFIED\XBUILTIN (xbuiltin)
DIR [m] d:\XACT\UNIFIED\BUILTIN (builtin)
```

2. Include the path to the \unified directory in your XACT environment variable:

```
set XACT=d:\XACT
```

Sun4, HP, and IBM RS6000

To access the XC4000E library on a Sun4, HP, or IBM RS6000 workstation, include this line in your viewdraw.ini file:

```
DIR [r] /correct_path/unified/xc4000e (xc4000e)
```

For example, if the software is installed in /customer/xilinx, your viewdraw.ini file should contain the following lines:

```
DIR [r] /customer/xilinx/unified/xc4000e (xc4000e)
DIR [r] /customer/xilinx/unified/xblox (xblox)
DIR [r] /customer/xilinx/unified/xbuiltin (xbuiltin)
DIR [r] /customer/xilinx/unified/builtin (builtin)
```

FXC4000E Library

The FXC4000E library is the same as the XC4000E library with one difference: the simulation models in the FXC4000E library do not perform timing violation checks like setup, hold, and minimum pulse width. The “f” prefix stands for fast simulation because the removal of timing violation checks can speed up the simulation. To access the FXC4000E library, follow these platform-specific instructions.

Refer to the FXC4000E library using the XC4000E alias. You should not have both the XC4000E and the FXC4000E libraries in one viewdraw.ini file.

PC

To access the FXC4000E library on a personal computer, perform the following steps.

1. Include this line in your viewdraw.ini file:

```
DIR [m] d:\correct_path\unified\fxc4000e (xc4000e)
```

where d: is the drive on which this 6.0.1 software is installed.

For example, if the software is installed in d:\xact, your viewdraw.ini file should contain the following lines:

```
dir [m] d:\xact\unified\fxc4000e (xc4000e)
dir [m] d:\xact\unified\xblox (xblox)
dir [m] d:\xact\unified\xbuiltin (xbuiltin)
dir [m] d:\xact\unified\builtin (builtin)
```

2. Include the path to the \unified directory in your XACT environment variable.

```
set XACT = d:\XACT
```

Sun4, HP, and IBM RS6000

To access the FXC4000E library on a Sun4, HP, or IBM RS6000 workstation, include this line in your viewdraw.ini file:

```
DIR [r] /correct_path/unified/fxc4000e(xc4000e)
```

For example, if the software is installed in /customer/xilinx, your viewdraw.ini file should contain the following lines:

```
DIR [r] /customer/xilinx/unified/fxc4000e (xc4000e)
DIR [r] /customer/xilinx/unified/xblox (xblox)
DIR [r] /customer/xilinx/unified/xbuiltin (xbuiltin)
DIR [r] /customer/xilinx/unified/builtin (builtin)
```

Switching to XC4000E from XC4000

If you have an existing Viewlogic schematic design implemented with the XC4000 library and you wish to switch to the XC4000E library, you can do so by changing your viewdraw.ini file, as illustrated in the following examples.

Example 1: On a workstation, if the XC4000E software is installed in /customer/xilinx and the XC4000 software is installed in /customer/old/xilinx, you can change the following line:

```
DIR [r] /customer/old/xilinx/unified/xc4000 (xc4000)
```

to:

```
DIR [r] /customer/xilinx/unified/xc4000e (xc4000e)
DIR [r] /customer/xilinx/unified/xc4000e (xc4000)
```

Example 2: On the PC, if the XC4000E software is installed in d:\xact and the XC4000 software is installed in d:\old\xact, you can change the line:


```
DIR [m] d:\OLD\XACT\UNIFIED\XC4000 (xc4000)
```

to:

```
DIR [m] d:\XACT\UNIFIED\XC4000E (xc4000e)
```

```
DIR [m] d:\XACT\UNIFIED\XC4000E (xc4000)
```

New Symbols For XC4000E

IFDX	IFDXI	IFDX4	IFDX8	IFDX16
IFDX_1	IFDXI_1			
ILDX	ILDXI	ILDX4	ILDX8	ILDX16
ILDX_1	ILDXI_1			
OFDX	OFDXI	OFDX4	OFDX8	OFDX16
OFDX_1	OFDXI_1			
OFDEX	OFDEXI	OFDEX4	OFDEX8	OFDEX16
OFDEX_1	OFDEXI_1			
OFDTX	OFDTXI	OFDTX4	OFDTX8	OFDTX16
OFDTX_1	OFDTXI_1			
RAM16X1D	RAM16X2D	RAM16X4D	RAM16X8D	
RAM16X1S	RAM16X2S	RAM16X4S	RAM16X8S	
RAM32X1S	RAM32X2S	RAM32X4S	RAM32X8S	

The X-BLOX XC4000E RAM symbols are DP_RAM and SYNC_RAM.

Location Constraints

In general, any parameter that is allowed on RAM16X1 or RAM32X1 for XC4000 is allowed on RAM16X1S, RAM32X1S, or RAM16X1D for XC4000E. As an exception, when an RLOC attribute is used on a RAM16X1D symbol, no extension is allowed, because a RAM16X1D occupies an entire CLB.

Simulating XC4000E RAM

An INIT value placed on a RAM16X1D, RAM16X1S, or RAM32X1S primitive must be processed by XSimMake to be shown in simulation. The XSimMake functional flow creates a file named *sdesign.xmm*, which contains the appropriate ViewSim commands to pre-load the RAM elements. This command file should be executed in ViewSim at the start of both functional and timing simulation. XSimMake also adds commands to the XMM file for asynchronous RAMs, ROMs, and memories created by X-BLOX and MemGen.

Xilinx-Synopsys Interface (XSI)

This release of XSI is a superset of the existing XSI 5.2 product. The principal changes include the following:

- Addition of several new cells to the FPGA Compiler/Design Compiler synthesis libraries and the VSS FTGS simulation libraries
- Additional timing information to support new XC4000E speed grades
- XSI executables that now recognize new XC4000E part types
- Equal application of all other XC4000-specific optimizations performed by FPGA-Compiler to the XC4000E architecture, including:
 - Direct synthesis to XC4000 IOB and CLB structures
 - Automatic inference of X-BLOX modules for optimized arithmetic
 - Clock-buffer insertion
 - Synthesis to flip-flops with clock enables

Configuring Synopsys to Synthesize and Simulate XC4000E Designs

The .synopsys_dc.setup file for XC4000E synthesis should be identical to that for XC4000 designs with the exception of the target- and link-library settings and the reference to the X-BLOX Design Ware library. An example.synopsys_dc.setup file follows. The target- and link-library settings were created using the Synlibs 4005e-3 command.

```
+++++++search_path = { . \
<XC4000E_DS401_install_path>/synopsys/libraries/syn
\ <Synopsys_install_path>/libraries/syn}

define_design_lib xblox_4000e -path \
<XC4000E_DS401_install_path>/synopsys/libraries/dw/
lib/fpga/xc4000e

compile_fix_multiple_port_nets = true
xlnx_hier_blknm = 1
xnfout_library_version = "2.0.0"
```

```
bus_naming_style = "%s<%d>"
bus_dimension_separator_style = "><"
bus_inference_style = "%s<%d>"

link_library = {xprim_4005e-3.db xprim_4000e-3.db \
xgen_4000e.db xfpga_4000e-3.db xio_4000e-3.db}
target_library = {xprim_4005e-3.db xprim_4000e-3.db \
xgen_4000e.db xfpga_4000e-3.db xio_4000e-3.db}
symbol_library = {xc4000e.sdb}
synthetic_library = {xblox_4000e.sldb standard.sldb}
```

Notice that the reference to the XC4000E X-BLOX Design Ware library is differentiated from the XC4000 X-BLOX library with a new name, `xblox_4000e`.

The `synopsys_vss.setup` file for XC4000E simulation should also be identical to that for XC4000 designs with the exception of the simulation library reference. The following is a sample `synopsys_vss.setup` file.

```
timebase=ns
time_res_factor=0.1
no_hazard_mesg=true
WORK > DEFAULT
DEFAULT: ./WORK
xc4000e:<XC4000E_DS401_install_path>/synopsys/
libraries/sim/lib/xc4000e
```

New Components in XC4000E Synthesis and Simulation Libraries

Several new components have been added to the XC4000E synthesis and simulation libraries, reflecting the XC4000E architecture's new features. The new components, listed in the following tables, include synchronous single-port RAMs, synchronous dual-port RAMs, and I/O flip-flops/latches with clock/latch-enables. Synopsys does not synthesize cells marked with an asterisk, so these components must be instantiated.

Synchronous RAM			
Name	Outputs	Inputs	Notes
RAM16X1S	O	D, A3, A2, A1, A0, WE, WCLK	*
RAM32X1S	O	D, A4, A3, A2, A1, A0, WE, WCLK	*
RAM16X1D	SPO,DPO	D, A3, A2, A1, A0, DPRA3, DPRA2, DPRA1, DPRA0, WE, WCLK	*

IOB Input Flip-Flops with Clock Enable			
Name	Output	Inputs	Notes
IFDX	Q	D, C, CE	
IFDX_F	Q	D, C, CE	NODELAY*
IFDX_U	Q	D, C, CE	UNBONDED*
IFDXI	Q	D, C, CE	INIT=S
IFDXI_F	Q	D, C, CE	INIT=S, NODELAY*
IFDXI_U	Q	D, C, CE	INIT=S, NODELAY, UNBONDED*

IOB Output Flip-Flops with Clock Enable			
Name	Outputs	Inputs	Notes
OFDX	Q	D, C, CE	
OFDX_F	Q	D, C, CE	Fast slew rate
OFDX_FU	Q	D, C, CE	Fast slew rate, unbonded*
OFDX_S	Q	D, C, CE	Slow slew rate
OFDX_U	Q	D, C, CE	Unbonded*
OFDXI	Q	D, C, CE	INIT=S
OFDXI_F	Q	D, C, CE	INIT=S, fast slew rate
OFDXI_S	Q	D, C, CE	INIT=S, slow slew rate

IOB Output Flip-Flops with Clock Enable			
Name	Outputs	Inputs	Notes
OFDXI_U	Q	D, C, CE	INIT=S, unbonded*

IOB Tristatable Output Flip-Flops with Clock Enable			
Name	Outputs	Inputs	Notes
OFDTX	O	D, C, CE, T	
OFDTX_F	O	D, C, CE, T	Fast slew rate
OFDTX_S	O	D, C, CE, T	Slow slew rate
OFDTX_U	O	D, C, CE, T	Unbonded*
OFDTXI	O	D, C, CE, T	INIT=S
OFDTXI_F	O	D, C, CE, T	INIT=S, fast slew rate
OFDTXI_S	O	D, C, CE, T	INIT=S, slow slew rate
OFDTXI_U	O	D, C, CE, T	INIT=S, unbonded*

IOB Input Latches with Gate Enable			
Name	Outputs	Inputs	Notes
ILDX_1	Q	D, G, GE	
ILDX_1F	Q	D, G, GE	NODELAY*
ILDX_1U	Q	D, G, GE	Unbonded*
ILDXI_1	Q	D, G, GE	INIT=S
ILDXI_1F	Q	D, G, GE	INIT=S, NODELAY
ILDXI_1U	Q	D, G, GE	INIT=S, unbonded*

Note: All IOB input gate-enabled latches have active-Low gate pins.

Restrictions Imposed by Synopsys 3.3 and Later

The Synopsys FPGA Compiler and Design Compiler products are currently unable to infer IOB registers with clock enables. The only way to access these features is to instantiate the appropriate cells in the HDL source code. This limitation will be addressed in a subsequent release of Synopsys' products; however, in the meantime, one of the optimizations performed by X-BLOX is to push clock-enabled CLB registers into IOBs where possible. Refer to the *XACTstep X-BLOX Reference/User Guide* for information about the rules governing this process.

Using Synchronous and/or Level-Sensitive RAM

The Synopsys FPGA Compiler and Design Compiler products are currently unable to infer RAM. (Although RAM can be described behaviorally, this methodology currently synthesizes to inefficient latch- or register-based implementations.) The only way to access these features is to instantiate the appropriate cells in the HDL source code. Alternatively, you can instantiate memory modules created by MemGen, which is explained in the "Pin Names of RAM Components" section earlier in this release note.

The XC4000E RAM modules allow you to specify their contents at power-on. When RAM modules are instantiated directly in the HDL source code, you can enter initialization values for the RAM using the following command:

```
set_attribute instance_name xnf_init init_value
-type string
```

For 16-location RAMs, specify a 4-digit hexadecimal value for *init_value*. For 32-location RAMs, specify an 8-digit hexadecimal value.

Although this mechanism permits RAM-initialization information to be carried into the FPGA implementation tools for incorporation into the configuration bitstream, it does not simulate behaviorally. For behavioral simulation, a RAM's contents remain unknown until they are defined by valid write access. However, back-annotated functional or timing simulation reflects this RAM initialization information.

Mentor Graphics Interface

This release of Mentor Interface (MN8) is a superset of the existing MN8 5.2 product. The major changes include the following:

- Addition of several new cells to the Xilinx library for Mentor Interface
- Changes in the scripts to support the new cells and features
- Single-Port Synchronous RAM. The two new RAM symbols (primitives) are the following:
 - RAM16X1S
 - RAM32X1S

There is a new WCLK pin, which occupies the same location as WE pin did in XC4000. The WE pin is now placed above the D data pin.

- Six new synchronous RAM macros:
 - RAM16X2S
 - RAM16X4S
 - RAM16X8S
 - RAM32X2S
 - RAM32X4S
 - RAM32X8S

The WE pin is now placed above the D pin.

- Dual-port RAM. RAM16X1D is the new primitive.
The WE pin is now placed above the D pin.
- Three new dual-port macros:
 - RAM16X2D
 - RAM16X4D
 - RAM16X8D

The WE pin is now placed above the D pin.

- CE on I/O registers. The IOBs of the XC4000E parts, which have a pin, the CE or GE pin, that is not available on other XC4000 parts. The Mentor primitives that are affected are the following:
 - INFF
 - INLAT
 - INREG
 - OUTFF
 - OUTFFT
- Eight new primitives:
 - IFDX
 - IFDXI
 - ILDX_1
 - ILDXI_1
 - OFDX
 - OFDXI
 - OFDTX
 - OFDTXI

The difference between these and the existing I/O register primitives is the addition of a CE pin between D and C pins for the flip-flops and the addition of a GE pin between the D and G pins for the latches.

- Twenty-seven new I/O macros:
 - IFDX4, IFDX8, IFDX16, IFDX_1, IFDXI_1
 - ILDX, ILDX4, ILDX8, ILDX16, ILDXI
 - OFDX4, OFDX8, OFDX16, OFDX_1, OFDXI_1
 - OFDEX, OFDEX4, OFDEX8, OFDEX16, OFDEXI, OFDEX_1, OFDEXI_1
 - OFDTX4, OFDTX8, OFDTX16, OFDTX_1, OFDTXI_1
- A new non-invertible pin required on the I/O register primitives. The pin name is CE for flip-flops and GE for latches. The XNF primitive names remain the same as they are currently.

- The INIT parameter on all RAM, RAMS, and RAMD primitives.
- Back-annotation. The -m option of the XNFBA program generates an MBA file for post-route timing back-annotation into Mentor. For RAM16X1S, RAM32X1S, and RAM16X1D, delays on all pins but the WCLK pin are routing delays and are put “on-parent.” The WCLK pin has double delay: routing and block. The delay on the WCLK pin is the block delay. The delay on the BUF driving the WCLK pin is the routing delay. The “on-parent” delay is written with the “RISEpin_name” format. The double delay is written with the “RISEpin_name” format for the routing delay and the “RISEpin_name” format for the block delay.

The following example shows how a RAM16X1S symbol is back-annotated.

Following is an example of a RAM16X1S before routing:

```
SYM, FIFO/RAMS/RAMSL/$1I92, RAMS, SCHNM=RAM16X1S,
INIT=0, LIBVER=2.0.0

PIN, O, O, DO7

PIN, WE, I, FIFO/CE_WR

PIN, D, I, FIFO/DIQ7

PIN, WCLK, I, CLK

PIN, A0, I, FIFO/A0

PIN, A1, I, FIFO/A1

PIN, A2, I, FIFO/A2

PIN, A3, I, FIFO/A3

END
```

The next example shows a RAM16X1S after routing (PPR, XDelay, LCA2XNF):

```
SYM, DO7, RAMS, LIBVER=2.0.0, SCHNM=RAM16X1S,
INIT=0000

PIN, O, O, DO7, 2.0

PIN, WE, I, FIFO/CE_WR, 5.0

PIN, D, I, FIFO/DIQ7, 1.1

PIN, WCLK, I, YSIG121, 6.0

PULSE, WCLK, +, 8.4
```

```
PIN, A3, I, FIFO/A3, 3.1
PIN, A2, I, FIFO/A0, 5.3
PIN, A1, I, FIFO/A2, 2.7
PIN, A0, I, FIFO/A1, 5.8
SETUP, D, WCLK, +, 3.3, 1.1
SETUP, WE, WCLK, +, 1.3, 0.4
SETUP, A3, WCLK, +, 2.4, 0.7
SETUP, A2, WCLK, +, 2.4, 0.7
SETUP, A1, WCLK, +, 2.4, 0.7
SETUP, A0, WCLK, +, 2.4, 0.7
END
SYM, XSYM220, BUF, LIBVER=2.0.0
PIN, O, O, YSIG121
PIN, I, I, CLK, 1.4
END
```

Here is an example of a RAM16X1S after back-annotation with XNFBA, which produces an MBA file:

```
/FIFO/RAMS/RAMSL/$1I92 N RISEO 2.0
/FIFO/RAMS/RAMSL/$1I92 N FALLO 2.0
/FIFO/RAMS/RAMSL/$1I92 N RISEA0 5.3
/FIFO/RAMS/RAMSL/$1I92 N FALLA0 5.3
/FIFO/RAMS/RAMSL/$1I92 N RISEA1 5.8
/FIFO/RAMS/RAMSL/$1I92 N FALLA1 5.8
/FIFO/RAMS/RAMSL/$1I92 N RISEA2 2.7
/FIFO/RAMS/RAMSL/$1I92 N FALLA2 2.7
/FIFO/RAMS/RAMSL/$1I92 N RISEA3 3.1
/FIFO/RAMS/RAMSL/$1I92 N FALLA3 3.1
/FIFO/RAMS/RAMSL/$1I92 N RISED 1.1
/FIFO/RAMS/RAMSL/$1I92 N FALLD 1.1
/FIFO/RAMS/RAMSL/$1I92 N RISEWE 5.0
/FIFO/RAMS/RAMSL/$1I92 N FALLWE 5.0
/FIFO/RAMS/RAMSL/$1I92 N RISEWCLK 1.4
```

```
/FIFO/RAMS/RAMSL/$1I92 N FALLWCLK 1.4
/FIFO/RAMS/RAMSL/$1I92 N RISE_WCLK 6.0
/FIFO/RAMS/RAMSL/$1I92 N FALL_WCLK 6.0
/FIFO/RAMS/RAMSL/$1I92 N SETUPA1 2.4
/FIFO/RAMS/RAMSL/$1I92 N HOLDA1 0.7
/FIFO/RAMS/RAMSL/$1I92 N SETUPA2 2.4
/FIFO/RAMS/RAMSL/$1I92 N HOLDA2 0.7
/FIFO/RAMS/RAMSL/$1I92 N SETUPA0 2.4
/FIFO/RAMS/RAMSL/$1I92 N HOLDA0 0.7
/FIFO/RAMS/RAMSL/$1I92 N SETUPA3 2.4
/FIFO/RAMS/RAMSL/$1I92 N HOLDA3 0.7
/FIFO/RAMS/RAMSL/$1I92 N SETUPWE 1.3
/FIFO/RAMS/RAMSL/$1I92 N HOLDWE 0.4
/FIFO/RAMS/RAMSL/$1I92 N SETUPD 3.3
/FIFO/RAMS/RAMSL/$1I92 N HOLDD 1.1
/FIFO/RAMS/RAMSL/$1I92 N PULSEWCLK 8.4
/FIFO/RAMS/RAMSL/$1I92 S PULSE_POLARITY_WCLK +
/FIFO/RAMS/RAMSL/$1I92 S INIT 0
```

Script Changes

The Mentor scripts have been changed to support the new cells in this release. In addition to these changes, XBLXGS and EDIF2XNF have been modified to support the new XC4000E library.

MEN2XNF8, FNCSIM8, TIMSIM8

From the Design Manager (pld_dmgr), you can invoke commands to execute specific scripts, as follows.

- **PLD_Men2XNF8** executes the script to translate your design to an XNF file. You must run this command before you can use either **PLD_FNCSIM8** or **PLD_TIMSIM8**.
- **PLD_FNCSIM8** executes the script to prepare your design for functional simulation.

- PLD_TIMSIM8 executes the script to prepare your design for timing simulation.

Refer to the *Mentor Graphics Interface/Tutorial Guide* for more details these commands.

PLD_DVE

This command executes the script that invokes the Mentor Graphics Design Viewpoint Editor (DVE) configured for Xilinx designs. The following example shows the `Pld_dve` command as you would use it in Men2XNF8 to check for the valid technology:

```
pld_dve design_name xc4000e
```

GEN_SCH8

This program creates a new schematic composed of only schematic elements that can be used for functional simulation. An example follows:

```
gen_sch8 xnf_file
```

where *xnf_file* is any XC4000E file; that is, it specifies an XC4000E part type (e.g. 4000epg223).

The output should be a valid design. To verify its validity, bring the design into the Design Architect and run Check Sheet on its sheets.

GEN_SYM8

This program automatically creates a symbol based on information in the XSF file. An example follows:

```
gen_sym8 xnf_file
```

where *xnf_file* is any XC 4000E file.

The output should be a valid symbol. To verify its validity, bring the design into the Design Architect symbol editor and examine the symbol.

Cadence Verilog-XL Interface

The 5.2.1 Beta release of ES-Verilog-XL is a superset of the existing ES-Verilog-XL product essentially adding support for XC4000E and XC5200 simulation. Principal changes include the following.

New Features

- Addition of new cells to the Xilinx library for the ES-Verilog interface.
- ES-Verilog executables now recognize XNF Version 6.0
- Changes in the scripts to support the new XC4000E and XC5200 cells and features
- XC4000E Single-port synchronous RAM. The two new synchronous RAM primitives are the following:
 - RAM16x1S
 - RAM32x1S
- XC4000E Dual-port RAM. The new primitive is
 - RAM16x1D
- XC4000E I/O registers. The IOBs of the XC4000E parts have a CE or GE pin that is not available on other XC4000 parts. There are eight new XC4000E primitives:
 - IFDX
 - IFDXI
 - ILDX1 (equivalent to ILDX_1)
 - ILDXI1 (equivalent to ILDXI_1)
 - OFDX
 - OFDXI
 - OFDTX
 - OFDTXI

The difference between these and the existing I/O register primitives is the addition of a CE pin to the I/O flip-flops and a GE pin to the latches. The CE and GE pins are non-invertible.

- XC4000E Back annotation. For RAM16x1S, RAM32x1S and RAM16x1D, delays on all pins except the WCLK pin are routing delays. The WCLK pin has two types of delays, routing and block. The delay on the BUF driving the WCLK pin is the routing delay. The delay on the WCLK pin is the clock-to-out block delay.

New XC5200 primitives:

- CY_MUX
- F5_MUX
- F5_MAP
- LDCE
- OSC52
- CK_DIV

New XC5200 versions of primitives:

- BSCAN
- Startup

Script Changes

Both the funcnetx and timenetx scripts have been modified to recognize and support XC4000E and XC5200 architectures.

OrCAD Interface

This section describes the OrCAD library and simulation support for the XC4000E architecture. See the *Libraries Guide Supplement* for details.

XC4000E Library

The XC4000E library is a superset of the XC4000 library, with the addition of XC4000E-specific elements. To access the XC4000E library for both schematic and simulation, perform the following steps.

1. Set the XACT environment variable to point to the installed directory of the Xilinx DS35 OrCAD interface.
2. Create the design directory in either of the following two ways:
 - Run the Create Design command from the Design Management Tools in the OrCAD ESP.
 - Create a new directory, then copy into it the OrCAD SDT/VST configuration files (sdt.cfg and vst.cfg, respectively) from the OrCAD template directory.

3. In the newly created directory, run the Xdraft 4e command from the MS-DOS command line. This command configures both sdt.cfg and vst.cfg to use the XC4000E libraries for schematic and simulation.

New Symbols for XC4000E

IFDX	IFDXI	IFDX4	IFDX8	IFDX16
IFDX_1	IFDXI_1			
ILDX	ILDXI	ILDX4	ILDX8	ILDX16
ILDX_1	ILDXI_1			
OFDX	OFDXI	OFDX4	OFDX8	OFDX16
OFDX_1	OFDXI_1			
OFDEX	OFDEXI	OFDEX4	OFDEX8	OFDEX16
OFDEX_1	OFDEXI_1			
OFDTX	OFDTXI	OFDTX4	OFDTX8	OFDTX16
OFDTX_1	OFDTXI_1			
RAM16X1D	RAM16X2D	RAM16X4D	RAM16X8D	
RAM16X1S	RAM16X2S	RAM16X4S	RAM16X8S	
RAM32X1S	RAM32X2S	RAM32X4S	RAM32X8S	

The X-BLOX XC4000E RAM symbols are DP_RAM and SYNC_RAM.

Note: The OrCAD library does not support the RAM, RAMS, or RAMD simulation initialization in this release

Chapter 4

Device and Package Support

The following is a master table of Xilinx devices for this release. For more information on architectural families and specific device parameters, see *The Programmable Logic Data Book*.

Device	Packages	Speed Grades
XC2018 ^a	PC44, VQ64, PC68, PC84, PG84, TQ100	-33, -50, -70, -100, -130
XC2064 ^a	PC44, PC68, PD48, PG68	-33, -50, -70, -100, -130
XC2018L ^a	PC84, VQ64, VQ100	-10
XC2064L ^a	VQ64, PC68	-10
XC3020 ^a	PC68, PC84, PG84, CB100, CQ100, PQ100	-50, -70, -100, -125
XC3030 ^a	PC44, PC68, PC84, PG84, PQ100, TQ100	-50, -70, -100, -125
XC3042 ^a	PC84, PG84, CB100, CQ100, PQ100, TQ100, PG132, PP132	-50, -70, -100, -125
XC3064 ^{a b}	PC84, PG132, PP132, PQ160	-50, -70, -100, -125
XC3090 ^{a b}	PC84, CB164, CQ164, PQ160, PG175, PP175, PQ208	-50, -70, -100, -125
XC3020A	PC68, PC84, PG84, CB100, PQ100	-6, -7
XC3030A	PC44, VQ64, PC68, PC84, PG84, PQ100, VQ100	-6, -7
XC3042A	PC84, PG84, CB100, PQ100, VQ100, PG132, PP132, TQ144	-6, -7
XC3064A ^b	PC84, PG132, PP132, TQ144, PQ160	-6, -7
XC3090A ^b	PC84, CB164, PC84, PG175, PP175, TQ176, PQ160, PQ208	-6, -7
XC3020L	PC84	-8
XC3030L	PC84, VQ64, VQ100	-8
XC3042L	PC84, VQ100, TQ144	-8

Device	Packages	Speed Grades
XC3064L ^b	PC84, TQ144	-8
XC3090L ^b	PC84, TQ176	-8
XC3120 ^a	PC68, PC84, PG84, CB100, PQ100	-3, -4, -5
XC3130 ^a	PC44, PC68, PC84, PG84, PQ100, TQ100	-3, -4, -5
XC3142 ^a	PC84, PG84, CB100, PQ100, TQ100, PG132, PP132, TQ144	-3, -4, -5
XC3164 ^{a b}	PC84, PG132, PP132, PQ160	-3, -4, -5
XC3190 ^{a b}	PC84, PQ160, CB164, PG175, PP175, PQ208	-3, -4, -5
XC3195 ^{a b}	PC84, PQ160, CB164, PG175, PP175, PQ208, PG223	-3, -4, -5
XC3120A	PC68, PC84, PG84, CB100, PQ100	-1, -2, -3, -4, -5
XC3130A	PC44, VQ64, PC68, PC84, PG84, PQ100, VQ100	-1, -2, -3, -4, -5
XC3142A	PC84, PG84, CB100, PQ100, VQ100, PG132, PP132, TQ144	-1, -2, -3, -4, -5
XC3164A ^b	PC84, PG132, PP132, TQ144, PQ160	-1, -2, -3, -4, -5
XC3190A ^b	PC84, PQ160, CB164, PG175, PP175, TQ176, PQ208	-1, -2, -3, -4, -5
XC3195A ^b	PC84, PQ160, CB164, PG175, PP175, PQ208, PG223	-1, -2, -3, -4, -5
XC4003	PC84, P100, PG120	-4, -5, -6
XC4005 ^b	PC84, PQ100, PG156, PQ160, CB164, PQ208	-4, -5, -6, -6B, -10
XC4006 ^b	PC84, PG156, PQ160, PQ208	-4, -5, -6
XC4008 ^b	PC84, PQ160, PG191, MQ208, PQ208	-4, -5, -6
XC4010 ^b	PC84, PQ160, PG191, CB196, MQ208, PQ208, BG225	-4, -5, -6, -10
XC4013 ^b	PQ160, MQ208, PQ208, PG223, BG225, CB228, PQ240, MQ240	-4, -5, -6, -10
XC4002A	PC84, PQ100, VQ100, PG120	-5, -6
XC4003A	PC84, CB100, PQ100, VQ100, PG120	-4, -5, -6, -10
XC4004A ^b	PC84, PG120, TQ144, PQ160,	-5, -6
XC4005A ^b	PC84, TQ144, PG156, PQ160, PQ208, BG225	-4, -5, -6
XC4010D ^b	PC84, PQ160, PQ208	-5, -6
XC4013D ^b	PQ160, PQ208, BG225, PQ240	-5, -6
XC4003E	PC84, PQ100, VQ100	-3, -4
XC4005E ^b	PC84, PQ100, TQ144, PQ160, CB164, PQ208	-3, -4

Device	Packages	Speed Grades
XC4006E ^b	PC84, PG156, PQ160, PQ208, TQ144	-3, -4
XC4008E ^b	PC84, PQ160, PG191, PQ208	-3, -4
XC4010E ^b	PC84, PQ160, PG191, CB196, PQ208, BG225	-3, -4
XC4013E ^b	PQ160, PQ208, HQ208, PG223, CB228, BG225, PQ240, HQ240	-3, -4
XC4020E ^b	HQ208, PG223, HQ240	-3, -4
XC4003H	PG191, PQ208	-5, -6
XC4005H ^b	PG223, MQ240, PQ240	-5, -6
XC5202	PC84, PQ100, VQ100, TQ144, PG156,	-5, -6
XC5204	PC84, PQ100, VQ100, TQ144, PG156, PQ160,	-5, -6
XC5206 ^b	PC84, PQ100, VQ100, TQ144, PQ160, PG191, PQ208	-5, -6
XC5210 ^b	PC84, TQ144, PQ160, PG223, PQ208, BG225, PQ240	-5, -6
XC5215 ^b	HQ208, HQ240, PG299, HQ304	-5, -6
XC7236A ^a	PC44	-16, -20, -25
XC7318 ^a	PC44, PQ44	-5, -7
XC7336 ^a	PC44, PQ44	-5, -7, -10, -12, -15
XC7354 ^a	PC44, PC68	-7, -10, -12, -15
XC7372 ^a	PC68, PC84, PQ100	-7, -10, -12, -15
XC7336Q ^a	PC44, PQ44, VQ44	-10, -12, -15
XC73108 ^a	PC84, PQ100, PG144, PQ100, PQ160, BG225	-7, -10, -12, -15, -20
XC73144 ^a	PQ160, BG225	-7, -10, -12, -15

a. Not supported in X-BLOX.

b. Not supported in Base packages.

Xilinx Customer Support Information

For registration, authorization codes, update information, warranty status, shipping, product issues, and technical support, call Monday through Friday, 8 a.m. to 5 p.m. Pacific time.

Registration, Authorization, and Customer Service

- United States and Canada.....1-800-624-4782
- Europe..... Contact your local Distributor
- Japan.....81-33-297-9191
- Southeast Asia/All Other Countries.....852-2424-5200
- Facsimile Transmission.....1-408-559-0115

Technical Support

Hotline Access

Location	Telephone	Electronic Mail
U.S. and Canada	1-800-255-7778	hotline@xilinx.com
Japan	81-3 -3297-9163	jhotline@xilinx.com
France	33-1-3463-0100	frhelp@xilinx.com
Germany	49-89-9915-4930	dlhelp@xilinx.com
United Kingdom	44-1-932-820821	ukhelp@xilinx.com
To Contact Factory	1-408-879-5199	hotline@xilinx.com

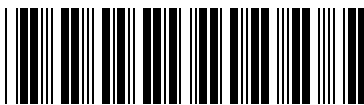
- Technical Support FAX(24 hours/7 days)1-408-879-4442
- Technical Support BBS (24 hours/7 days)1-408-559-9327
- Internet E-mail Address (24 hours/7 days).....hotline@xilinx.com
- Xilinx Worldwide Web Site <http://www.xilinx.com>

Training

- Xilinx Training Administrator1-408-879-5090
- International customers, contact your local sales representative or distributor.



The Programmable Logic CompanySM



0401533